HC05C5GRS/D
REV 1.2

68HC05C5

SPECIFICATION
(General Release)

CSIC System Design Group
Austin, Texas

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SECTION 1 INTRODUCTION

1.1 GENERAL

The MC68HC05C5 is a 40-pin device based on the MC68HC05P7. The memory map includes 5184 bytes of user ROM, 176 bytes of RAM, and 128 bytes of EEPROM. The MCU has four 8-bit I/O ports: A, B, C and D. Port C has high sink current capability. The MC68HC05C5 includes a simple Serial I/O Peripheral (SIOP), 16-bit Timer, and an on-chip Computer Operating Properly (COP) watchdog circuit.

1.2 FEATURES

- HC05 Core
- 40-pin DIP or 44-pin plastic-leaded chip carrier (PLCC) package
- On-Chip Oscillator with resistor capacitor (RC) or Crystal/Ceramic Resonator Mask Options
- 5184 Bytes of User ROM
- 176 Bytes of On-Chip RAM
- 128 Bytes of EEPROM
- EEPROM Low Voltage Program Inhibit (LVPI)
- Hardware EEPROM Program Enable
- 16-Bit Timer
- COP Watchdog Timer Mask Option
- 32 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
- Simple Serial Input/Output Port
- 10 mA Sink Capability on One 8-Bit Port

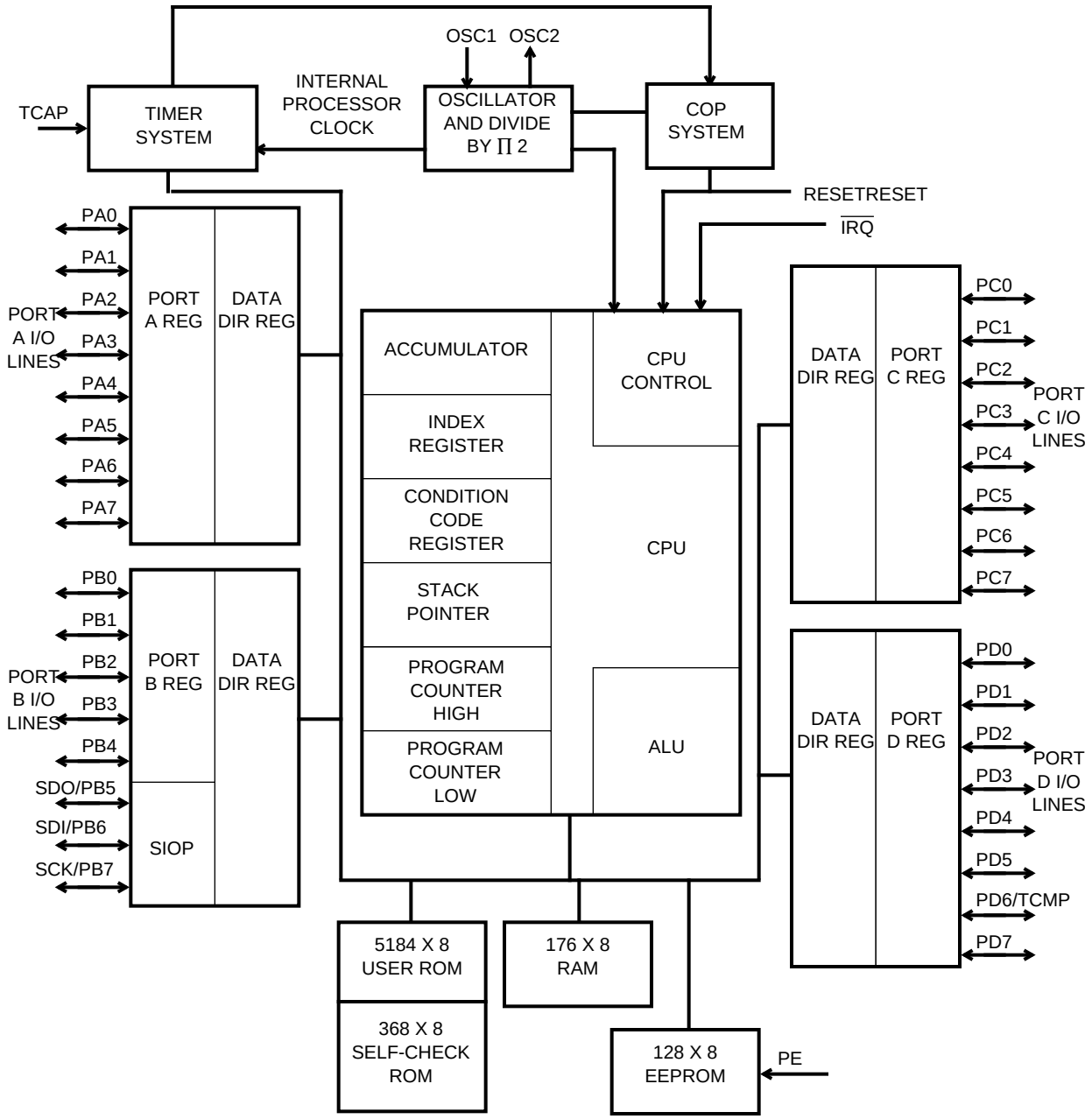


Figure 1-1: Self-Check Mode Schematic for the MC68HC05C5

1.3 MASK OPTIONS

There are five mask options on the MC68HC05C5: CLOCK (RC or Crystal), $\overline{IRQ}$ (edge-sensitive only or edge and level-sensitive), SIOP (MSB or LSB first), COP Watchdog Timer (enable/disable) and LVPI (enable/disable).

NOTE: Negative true signals like $\overline{\text{RESET}}$ and $\overline{\text{IRQ}}$ will be denoted with either an asterisk or an overline.

1.4 SIGNAL DESCRIPTION

1.4.1 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.

1.4.2 PE

PE is the Program Enable for the EEPROM. This pin has a very weak internal pullup. If this pin is held at a logic "1" level or left not connected, the EEPROM can be programmed and erased. If this pin is held at a logic "0" level, EEPROM programming and erasing is disabled. The pullup will not be able to pull the pin high if any circuit is connected to PE.

1.4.3 $\overline{\text{IRQ}}$

This pin has a mask option that provides two different choices of interrupt triggering sensitivity. The $\overline{\text{IRQ}}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity. Refer to **4.5 INTERRUPTS** for more detail.

1.4.4 OSC1 AND OSC2

These pins provide control input for an on-chip clock oscillator circuit. A crystal, a ceramic resonator, a resistor/capacitor combination, or an external signal connects to these pins providing a system clock. A mask option selects either a crystal/ceramic resonator or a resistor/capacitor as the frequency determining element. The oscillator frequency is two times the internal bus rate.

1.4.5 $\overline{\text{RESET}}$

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.

1.4.6 TCAP

This pin controls the input capture feature for the on-chip programmable timer. The TCAP pin contains an internal Schmitt trigger as part of its input to improve noise immunity.

1.4.7 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. Refer to **5.5 INPUT/ OUTPUT PROGRAMMING** for a detailed description of I/O programming.

1.4.8 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. Refer to **5.5 INPUT/ OUTPUT PROGRAMMING** for a detailed description of I/O programming. Three of the port B pins (PB5-PB7) are shared with the SIOP subsystem. Refer to **SECTION 7 SIMPLE SERIAL INPUT/OUTPUT PORT** for a detailed description of the SIOP.

1.4.9 PC0-PC7

These eight 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. Refer to **5.5 INPUT/ OUTPUT PROGRAMMING** for a detailed description of I/O programming.

1.4.10 PD0-PD7

These eight I/O lines comprise port D. The state of any pin is software programmable and all port D lines are configured as input during power-on or reset. Refer to **5.5 INPUT/ OUTPUT PROGRAMMING** for a detailed description of I/O programming. PD6 is shared with TCMP. Refer to **SECTION 6 TIMER** for more information on the TCMP pin.

SECTION 2 OPERATING MODES

The MCU has two modes of operation: Single-Chip Mode and Self-Check Mode. Table 2-1 shows the conditions required to go into each mode, where $V_{TST} = 2 \times V_{DD}$.

Table 2-1: Operating Mode Conditions

RESET	$\overline{IRQ}$	TCAP	MODE
⌐↑	V_{SS} to V_{DD}	V_{SS} to V_{DD}	Single-chip
	V_{TST}	V_{DD}	Self-Check

2.1 SINGLE-CHIP MODE

In Single-Chip Mode, the address and data buses are not available externally, but there are four 8-bit I/O ports. 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 $\overline{RESET}$ if the $\overline{IRQ}$ pin is within normal operating range.

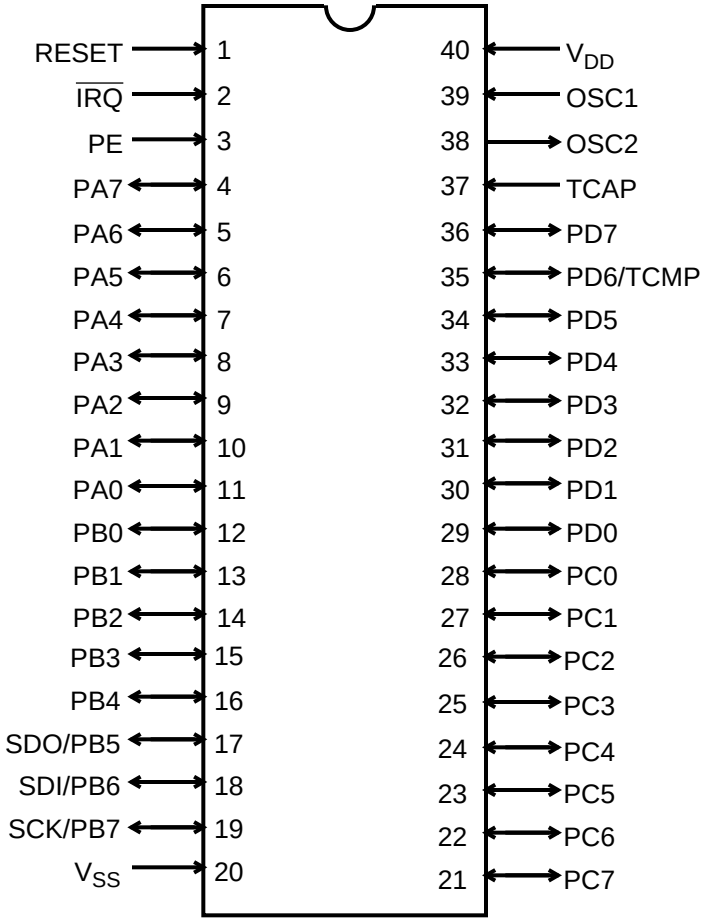


Figure 2-1: Single-Chip Mode Pinout of the MC68HC05C5

2.2 SELF-CHECK MODE

The Self-Check program resides at mask ROM location $1E80 to $1FEF. This program is designed to check the part's functionality with a minimum of support hardware.

The Self-Check Mode is entered on the rising edge of $\overline{\text{RESET}}$ if the $\overline{\text{IRQ}}$ pin is at V_{TST} volts and the TCAP pin is at logic one. $\overline{\text{RESET}}$ must be held low for 4064 cycles after Power-on Reset (POR), or for a time t_{RL} for any other reset. After reset, the I/O, RAM, ROM, Timer, SIOP and Interrupts are tested.

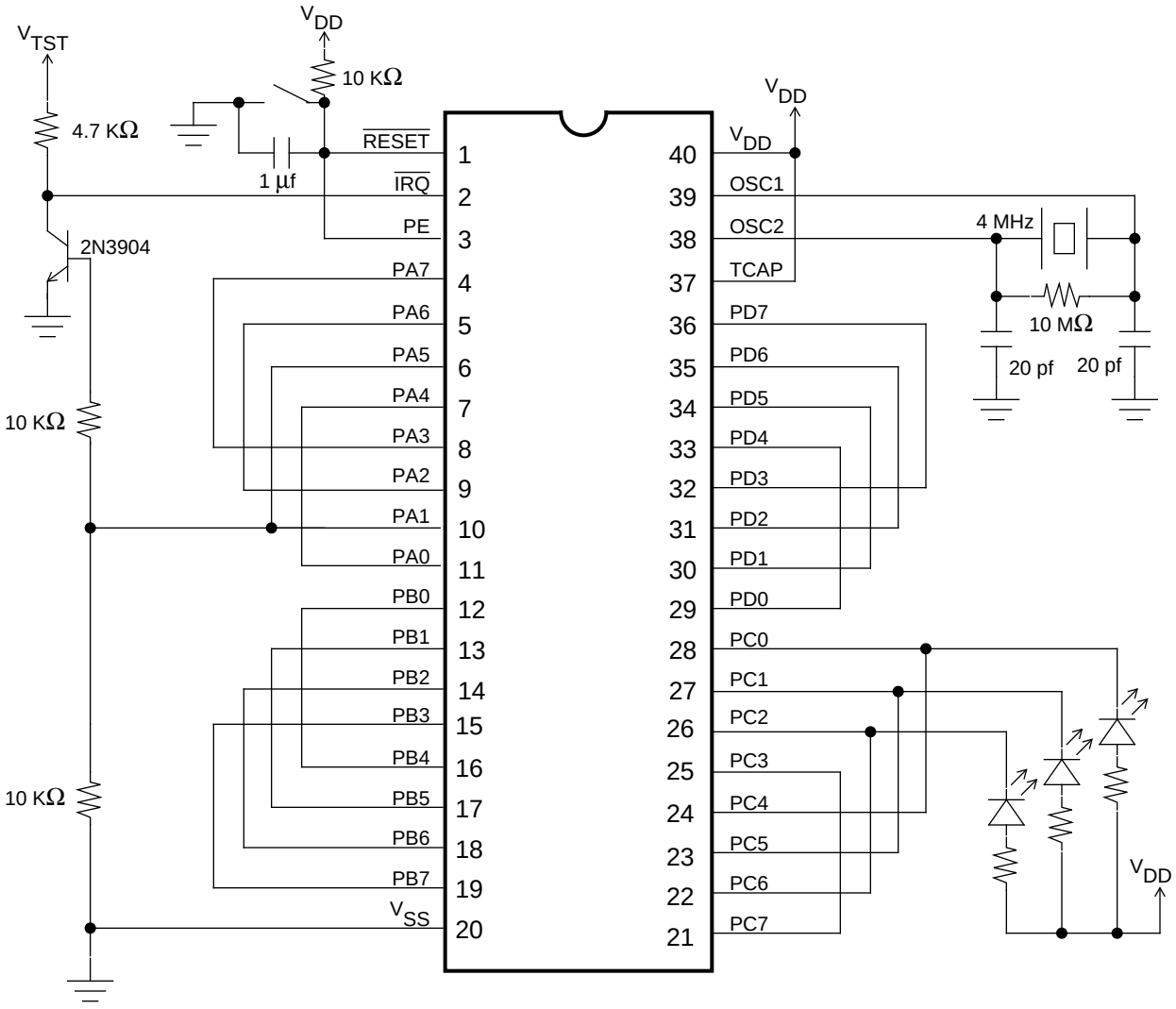


Figure 2-2: Self-Check Mode Schematic for the MC68HC05C5

SECTION 3 MEMORY

The MC68HC05C5 has an 8 K-byte memory map, consisting of user ROM, user RAM, Self-Check ROM, EEPROM, and I/O. See **Figure 3-1** and **Figure 3-2**.

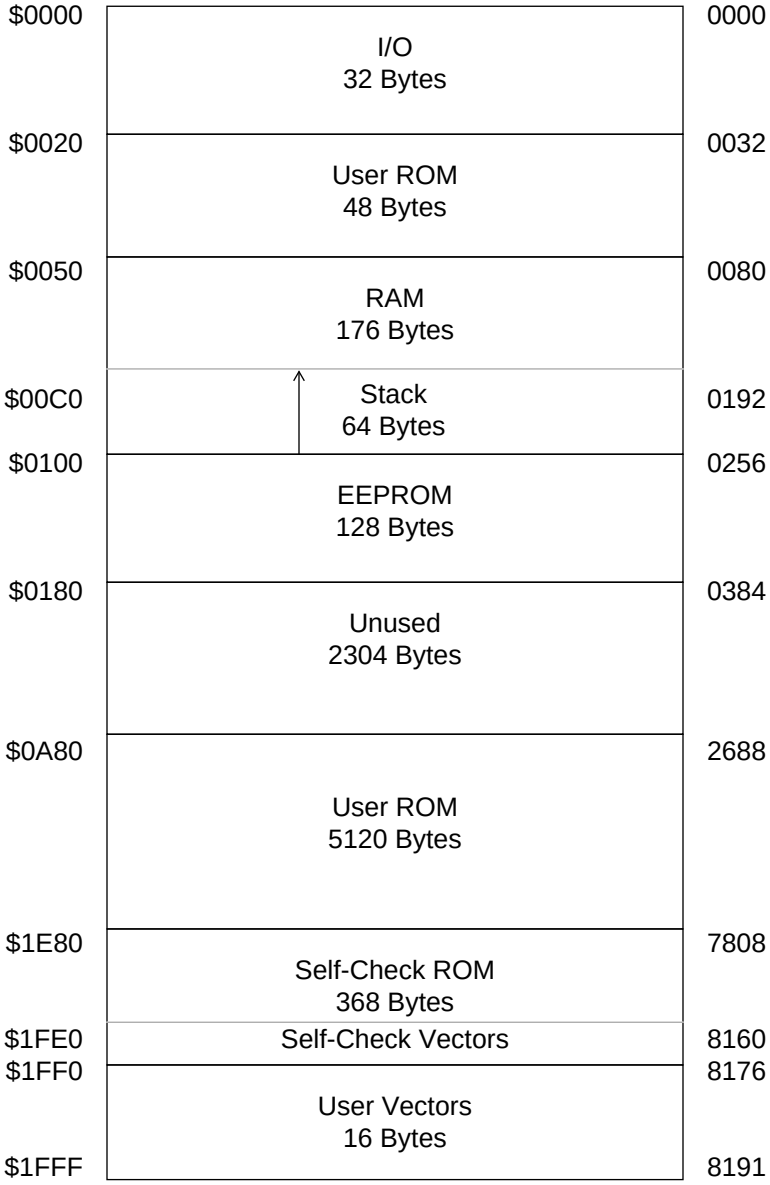


Figure 3-1: The 8K Memory Map of the MC68HC05C5

ADDRESS 0000 TO 001F	DATA 7	6	5	4	3	2	1	0
$00 PORT A DATA								
$01 PORT B DATA								
$02 PORT C DATA								
$03 PORT D DATA								
$04 PORT A DDR								
$05 PORT B DDR								
$06 PORT C DDR								
$07 PORT D DDR								
$08 UNUSED	--	--	--	--	--	--	--	--
$09 UNUSED	--	--	--	--	--	--	--	--
$0A SERIAL CTRL	0	SPE	0	MSTR	CPOL	0	0	0
$0B SERIAL STAT	SPIF	DCOL	0	0	0	0	0	0
$0C SERIAL DATA								
$0D UNUSED	--	--	--	--	--	--	--	--
$0E UNUSED	--	--	--	--	--	--	--	--
$0F UNUSED	--	--	--	--	--	--	--	--
$10 UNUSED	--	--	--	--	--	--	--	--
$11 UNUSED	--	--	--	--	--	--	--	--
$12 TIMER CONTROL	ICIE	OCIE	TOIE	0	0	COE	IEDG	OLVL
$13 TIMER STATUS	ICF	OCF	TOF	0	0	0	0	0
$14 CAPTURE HIGH								
$15 CAPTURE LOW								
$16 COMPARE HIGH								
$17 COMPARE LOW								
$18 COUNTER HIGH								
$19 COUNTER LOW								
$1A DUAL TM HIGH								
$1B DUAL TM LOW								
$1C PROGRAM REG	LVPI	CPEN	0	ER1	ER0	LATCH	EERC	EEPGM
$1D UNUSED	--	--	--	--	--	--	--	--
$1E UNUSED	--	--	--	--	--	--	--	--
$1F TEST REGISTER	--	MSCAN	ROMON	IOOFF	--	COPON	TCNT	RAMON

Figure 3-2: I/O Registers for the MC68HC05C5

3.1 ROM

The user ROM consists of 48 bytes of page zero ROM from $0020 to $004F, 5120 bytes of ROM from $0A80 to $1E7F and 16 bytes of user vectors from $1FF0 to $1FFF. The Self-Check ROM and vectors are located from $1E80 to $1FEF.

Eight of the user vectors, $1FF8 through $1FFF, are dedicated to reset and interrupt vectors. The remaining eight locations, $1FF0 through $1FF7, are general purpose user ROM locations.

3.2 RAM

The user RAM consists of 176 bytes of a shared stack area. The stack begins at address $00FF. The stack pointer can access 64 bytes of RAM in the range $00FF to $00C0. See **4.1.4 STACK POINTER (SP)**.

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.

3.3 EEPROM

The EEPROM on this device is 128 bytes long and is located at address $0100. Programming the EEPROM can be done by the user on a single byte basis by manipulating the Programming Register, located at address $001C.

3.3.1 PROGRAMMING REGISTER $1C

The contents and use of the programming register are discussed below. This device includes low-voltage programming inhibit (LVPI) circuitry which inhibits the use of the programming register when the supply voltage (V_{DD}) falls below V_{LVPI}.

$1C	LVPI	CPEN	0	ER1	ER0	LATCH	EERC	EEPGM
RESET:	0	0	0	0	0	0	0	0

Figure 3-3: : Programming Register

3.3.1.1 LVPI - Low-Voltage Programming Inhibit

LVPI is automatically set and cleared by the LVPI circuit and is not writable. The bit is set when V_{DD} falls below V_{LVPI} and is cleared when V_{DD} is above V_{LVPR}. Note that the V_{DD} rise and fall slew rates (t_{VDDR} and t_{VDDF}) must be within the specification for proper LVPI operation. If the specification is not met, the circuit will operate properly

following a delay of V_{DD}/Slew rate. When set, LVPI clears bits 0 through 6 in the programming register to disable the charge pump and prevent programming.

CPEN cannot be set when LVPI is set. During reset, LVPI is set until V_{DD} reaches V_{LVPI}, at which time it is cleared. The LVPI circuitry continues to function while the processor is in STOP mode.

The LVPI function is a mask option. If this function is disabled, bit 7 will be set to a value of "0".

3.3.1.2 CPEN - Charge Pump Enable

When set, CPEN enables the charge pump which produces the internal EEPROM programming voltage. This bit should be set concurrently with the LATCH bit. The programming voltage will not be available until EEPGM is set. The charge pump should be disabled when not in use. This bit is automatically cleared by the LVPI circuit when LVPI is set, and cannot be set until LVPI is cleared. CPEN is readable and writable and is cleared by reset.

3.3.1.3 ER1:ER0 - Erase Select Bits

ER1 and ER0 form a 2-bit field which is used to select one of three erase modes: byte, block, or bulk. **Table 3-1** shows the modes selected for each bit configuration. These bits are automatically cleared when LVPI is set. These bits are readable and writable and are cleared by reset.

In byte erase mode, only the selected byte is erased. In block mode, a 32-byte block of EEPROM is erased. The EEPROM memory space is divided into four 32-byte blocks ($100-$11F, $120-$13F, $140-$15F, $160-$17F), and doing a block erase to any address within a block will erase the entire block. In bulk erase mode, the entire 128-byte EEPROM section is erased.

Table 3-1: Erase Mode Select

ER1	ER0	MODE
0	0	Program (no Erase)
0	1	Byte Erase
1	0	Block Erase
1	1	Bulk Erase

3.3.1.4 LATCH

When set, LATCH configures the EEPROM address and data bus for programming. When LATCH is set, writes to the EEPROM array cause the data bus and the address bus to be latched. This bit is readable and writable, but reads from the array are inhibited if the LATCH bit is set and a write to the EEPROM space has taken place. When clear,

address and data buses are configured for normal operation. LATCH is automatically cleared when LVPI is set. Reset clears this bit.

3.3.1.5 EERC - EEPROM RC Oscillator Control

When this bit is set, the EEPROM section uses the internal RC oscillator instead of the CPU clock. After setting the EERC bit, delay a time t_{RCON} to allow the RC oscillator to stabilize. This bit is readable and writable and should be set by the user when the internal bus frequency falls below 1.5 MHz. EERC is automatically cleared when LVPI is set. Reset clears this bit.

3.3.1.6 EEPGM - EEPROM Programming Power Enable

EEPGM must be written to enable (or disable) the EEPGM function. When set, EEPGM turns on the charge pump and enables the programming (or erasing) power to the EEPROM array. When clear, this power is switched off. This will enable pulsing of the programming voltage to be controlled internally. This bit can be read at any time, but can only be written to if LATCH = 1. If LATCH is not set, then EEPGM cannot be set. LATCH and EEPGM can not both be set with one write if LATCH is cleared. EEPGM is automatically cleared when LVPI is set. EEPGM is automatically cleared when LATCH is cleared. Reset clears this bit.

3.3.2 PROGRAMMING/ERASING PROCEDURES

To program a byte of EEPROM, set LATCH = CPEN = 1, set ER1 = ER0 = 0, write data to the desired address and then set EEPGM for a time t_{EPGM}.

NOTE: Any bit should be erased before it is programmed. However, if write/erase cycling is a concern, a procedure can be followed to minimize the cycling of each bit in each EEPROM byte. Here is the procedure:

- If PB•EB* = 0, then program the new data over the existing data without erasing it first
- If PB•EB* ≠ 0, then erase byte before programming
- Where PB = Byte data to be programmed and EB = Existing EEPROM byte data.

To erase a **byte** of EEPROM, set LATCH = 1, CPEN = 1, ER1 = 0 and ER0 = 1, write to the address to be erased, and set EEPGM for a time t_{EBYT}.

To erase a **block** of EEPROM, set LATCH = 1, CPEN = 1, ER1 = 1 and ER0 = 0, write to any address in the block, and set EEPGM for a time t_{EBLOCK}.

For a **bulk** erase, set LATCH = 1, CPEN = 1, ER1 = 1, and ER0 = 1, write to any address in the array, and set EEPGM for a time t_{EBULK}.

To terminate the programming or erase sequence, clear EEPGM, delay for a time t_{FPV} to allow the program voltage to fall, and then clear LATCH and CPEN to free up the buses. Following each erase or programming sequence, clear all programming control bits.

SECTION 4 CPU CORE

4.1 REGISTERS

The MCU contains five registers as shown in the programming model of **Figure 4-1**. The interrupt stacking order is shown in **Figure 4-2**.

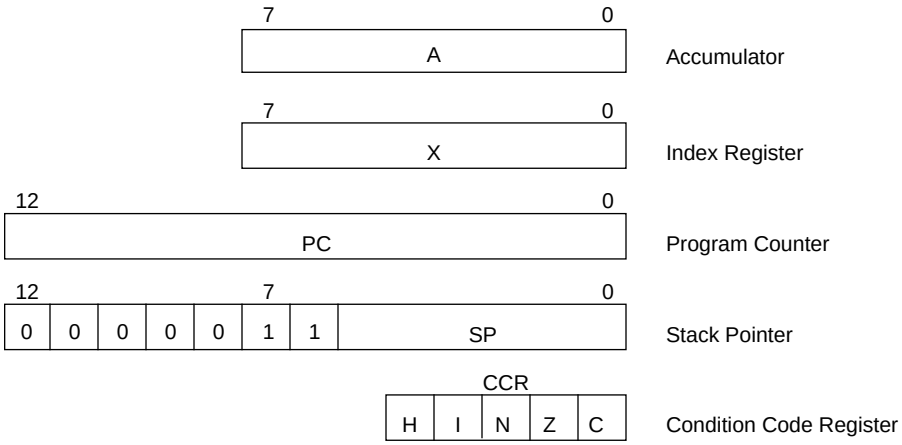
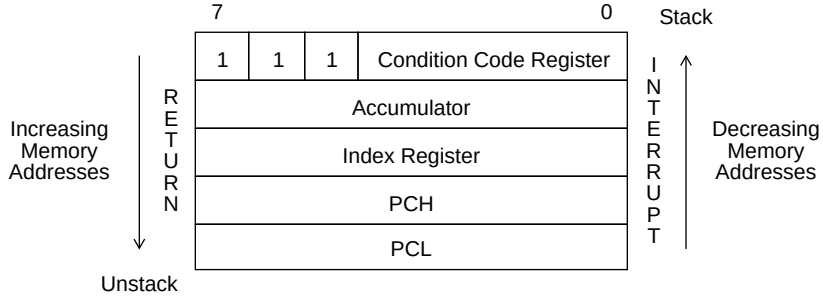


Figure 4-1: Programming Model



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 4-2: Stacking Order

4.1.1 ACCUMULATOR (A)

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

4.1.2 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.

4.1.3 PROGRAM COUNTER (PC)

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

4.1.4 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 7 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.

4.1.5 CONDITION CODE REGISTER (CCR)

The CCR is a 5-bit register in which 4 bits are used to indicate the results of the instruction just executed, and the fifth bit indicates whether interrupts are masked. 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.

4.1.5.1 Half Carry (H)

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

4.1.5.2 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 interrupt bit is cleared.

4.1.5.3 Negative (N)

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

4.1.5.4 Zero (Z)

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

4.1.5.5 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.

4.2 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 Technical Data* (MC68HC05C4/D).

4.2.1 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

4.2.2 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 Through Carry	ROL
Rotate Right Through Carry	ROR
Logical Shift Left	LSL
Logical Shift Right	LSR
Arithmetic Shift Right	ASR
Test for Negative or Zero	TST

4.2.3 BRANCH INSTRUCTIONS

This set of instructions branches if a particular condition is met; otherwise, no operation is performed. Branch instructions are 2-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

4.2.4 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)

4.2.5 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

4.3 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 (3 bytes) permit accessing tables throughout memory. Short and long absolute addressing is also included. One or 2-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 to describe the various addressing modes. Effective address is defined as the address from which the argument for an instruction is fetched or stored.

4.3.1 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 (for example, a constant used to initialize a loop counter).

4.3.2 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 2-byte instruction.

4.3.3 EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the 2 bytes following the opcode byte. Instructions with extended addressing mode are capable of referencing arguments anywhere in memory with a single 3-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.

4.3.4 RELATIVE

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.

4.3.5 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 1 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.

4.3.6 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. This addressing mode is useful for selecting the K^{th} element in an n element table. With this 2-byte instruction, K would typically be in X and the address of the beginning of the table would be in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510 ($01FE). This is the last location which can be accessed in this way.

4.3.7 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 2 unsigned bytes following the opcode. This address mode can be used in a manner similar to indexed, 8-bit offset except that this 3-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.

4.3.8 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 addressing 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 by selectively set or cleared with a single 2-byte instruction.

4.3.9 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 3-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.

4.3.10 INHERENT

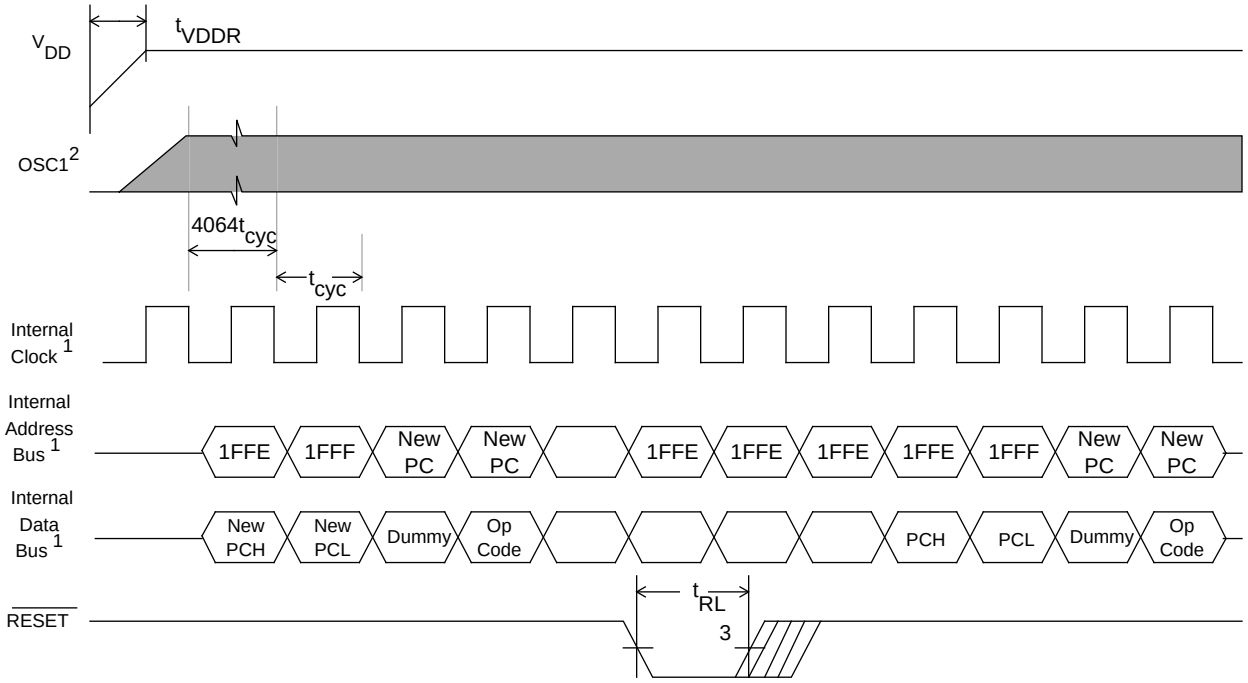
In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register and/or accumulator as well as the control instructions with no other arguments are included in this mode. These instructions are 1 byte long.

4.4 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, and by a COP watchdog-timer reset.

4.4.1 POWER-ON RESET (POR)

An internal reset is generated upon 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.



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 4-3: Power-On Reset and $\overline{\text{RESET}}$

4.4.2 $\overline{\text{RESET}}$ 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}).

4.4.3 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 time-out, 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 **SECTION 8 COMPUTER OPERATING PROPERLY** for more information on the COP Watchdog timer.

4.5 INTERRUPTS

The MCU can be interrupted three different ways: by the two maskable hardware interrupts ($\overline{\text{IRQ}}$ and timer) 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 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 if 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 4-1 lists vector addresses for all interrupts including reset.

Table 4-1: Vector Address for Interrupts and Reset

Register	Flag Name	Interrupts	CPU Interrupt	Vector Address
N/A	N/A	Reset	$\overline{\text{RESET}}$	$1FFE-$1FFF
N/A	N/A	Software	SWI	$1FFC-$1FFD
N/A	N/A	External Interrupt	$\overline{\text{IRQ}}$	$1FFA-$1FFB
TSR	ICF	Timer Input Capture	TIMER	$1FF8-$1FF9
TSR	OCF	Timer Output Compare	TIMER	$1FF8-$1FF9
TSR	TOF	Timer Overflow	TIMER	$1FF8-$1FF9

4.5.1 HARDWARE CONTROLLED INTERRUPT SEQUENCE

The following three functions ($\overline{\text{RESET}}$, STOP, and WAIT) are not in the strictest sense interrupts; however, they are acted upon in a similar manner. Flowcharts for hardware interrupts are shown in **Figure 4-4**, and for STOP and WAIT in **Figure 4-6**. 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 $1FFF. The I bit in the condition code register is also set. Much of the MCU is configured to a known state during this type of reset as previously described in **4.4 RESETS**.
2. STOP - The STOP instruction causes the oscillator to be turned off and the processor to "sleep" until and external interrupt ($\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.

4.5.2 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.

4.5.3 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 as specified by the contents of $1FFA and $1FFB.

Either a level-sensitive and edge-sensitive trigger or an edge-sensitive-only trigger is available 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.

4.5.4 TIMER INTERRUPT

There are three different timer interrupt flags that cause a timer interrupt whenever they are set and enabled. The interrupt flags are in the Timer Status Register (TSR), and the enable bits are in the Timer Control Register (TCR). Any of these interrupts will vector to the same interrupt service routine, located at the address specified by the contents of memory locations $1FF8 and $1FF9.

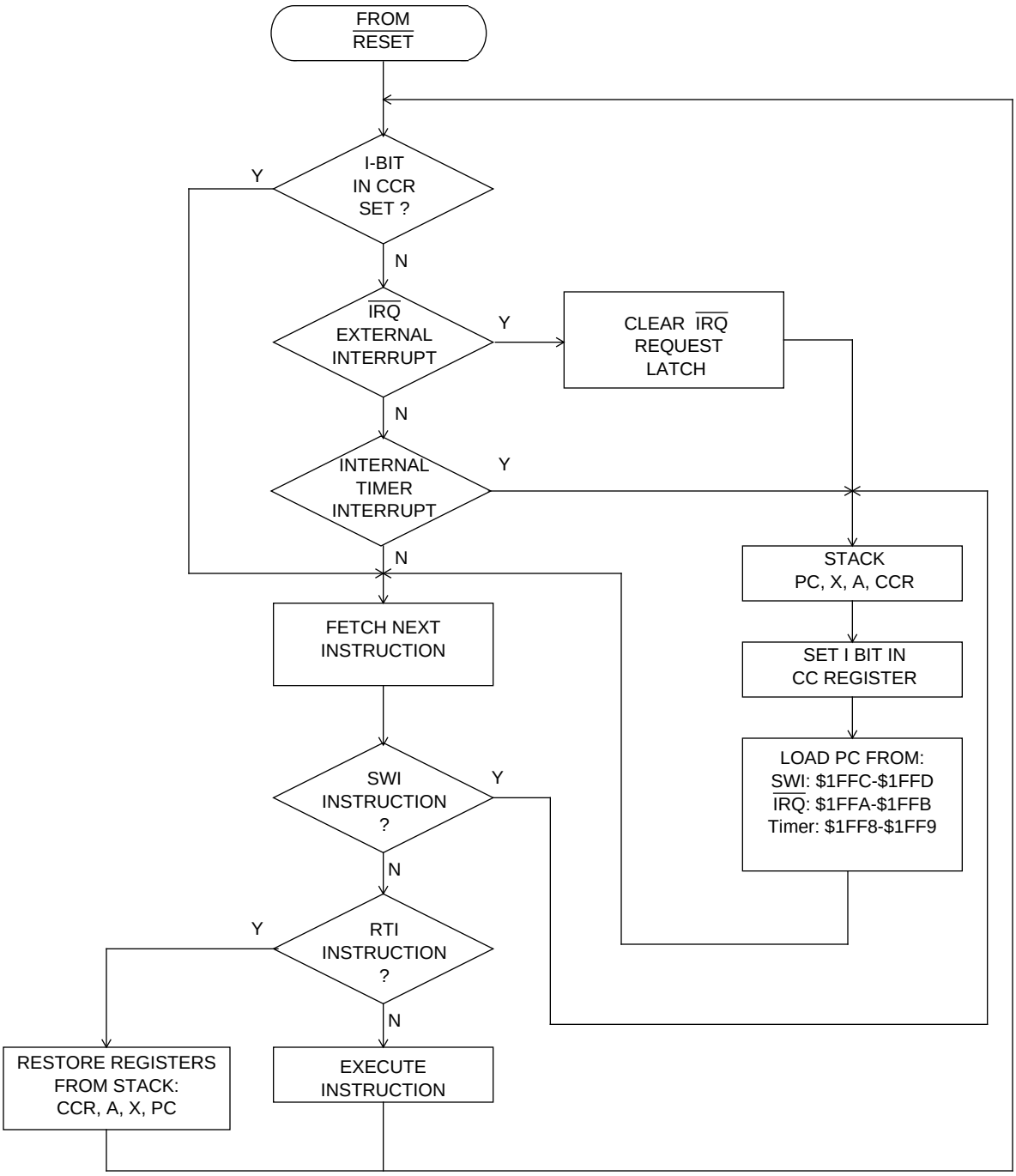


Figure 4-4: Interrupt Flowchart

4.6 LOW-POWER MODES

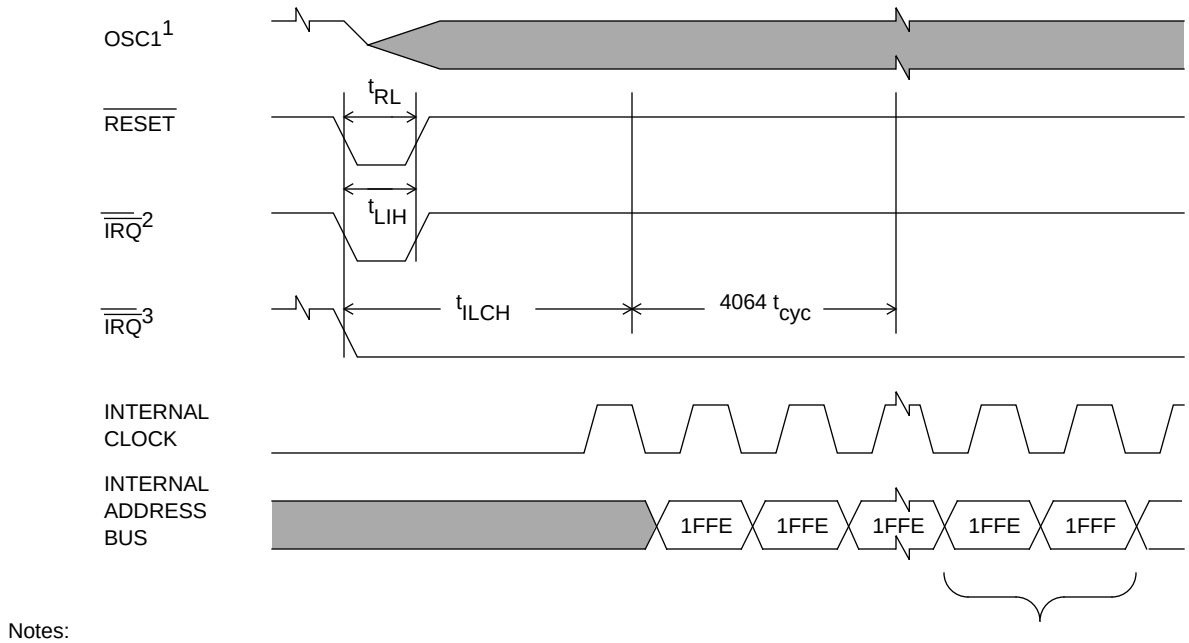
4.6.1 STOP

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 operation. The RC oscillator is also turned off during STOP mode, and is not available for use by the EEPROM system.

During the STOP mode, the TCR 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 the STOP mode only by an external interrupt or reset.

4.6.1.1 STOP RECOVERY

The processor can be brought out of the STOP mode only by an external interrupt or RESET. See **Figure 4-5**.



Notes:
1. Represents the internal gating of the OSC1 pin.
2. IRQ pin edge-sensitive mask option.
3. IRQ pin level- and edge-sensitive mask option.

Figure 4-5: Stop Recovery Timing Diagram

4.6.2 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 and the oscillator remain active. Any interrupt or reset (including a COP reset) will 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.

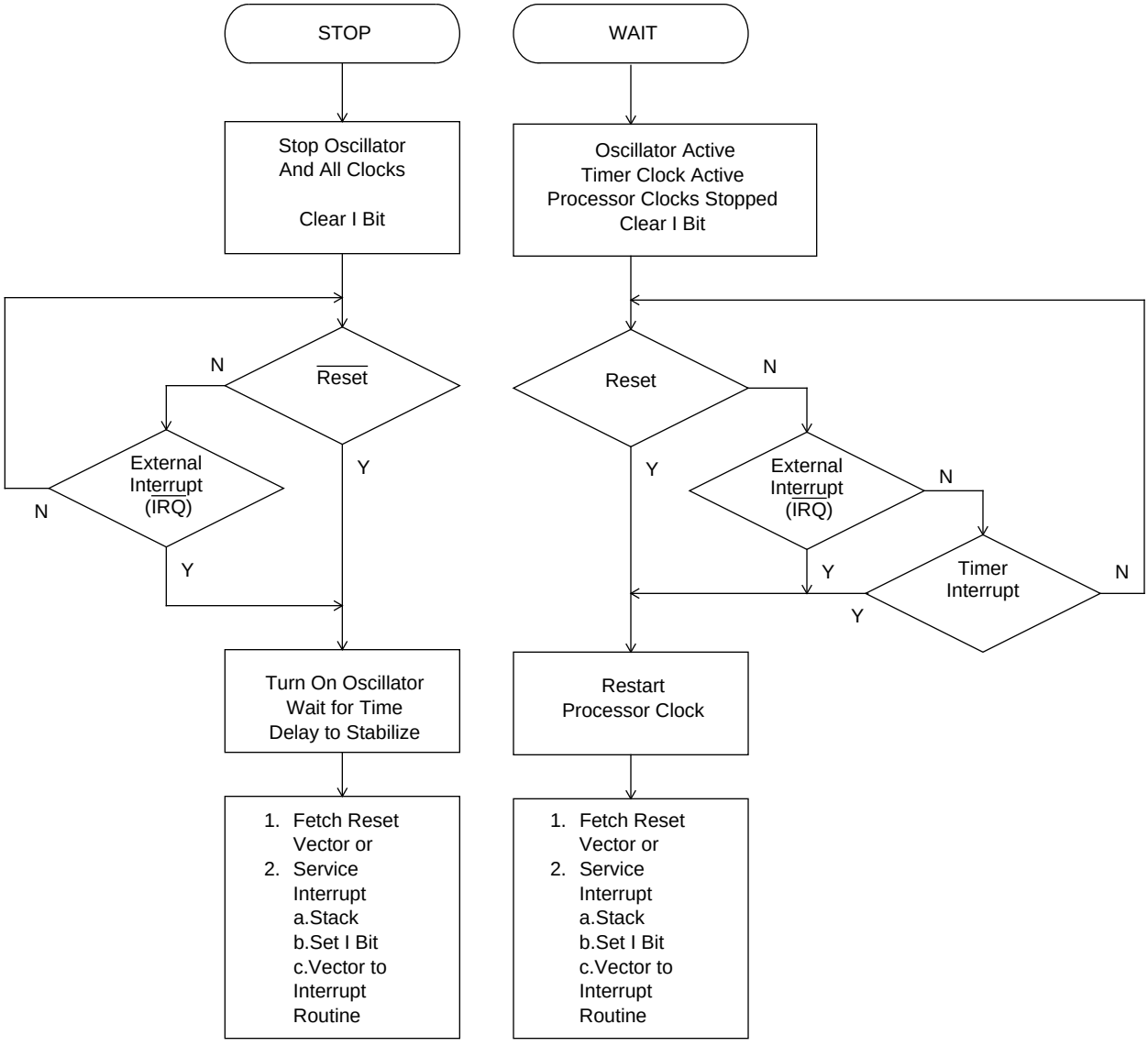


Figure 4-6: STOP/WAIT Flowcharts

SECTION 5 INPUT/OUTPUT PORTS

In single-chip mode there are 32 lines arranged as four 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.

5.1 PORT A

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

5.2 PORT B

Port B is an 8-bit bidirectional port. Three of the port B pins (PB5 thourgh PB7) are shared with the SIOP subsystem. Refer to **SECTION 7 SIMPLE SERIAL INPUT/OUTPUT PORT** for a detailed description of the SIOP. The port B data register is at $0001 and the data direction register (DDR) is at $0005. Reset does not affect the data registers, but clears the data direction registers, thereby returning the ports to inputs. Writing a 1 to a DDR bit sets the corresponding port bit to output mode.

5.3 PORT C

Port C is an 8-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 1 to a DDR bit sets the corresponding port bit to output mode. Port C has a high current sink capability. To minimize current spikes, these pins should be switched one at a time.

5.4 PORT D

Port D is an 8-bit bidirectional port. PD6 is shared with TCMP. If the PD6 pin is configured as TCMP by setting the COE bit in the Timer Control Register, this pin will become an output controlled by the Timer subsection. Refer to **SECTION 6 TIMER** for more information. The port D data register is at $0003 and the data direction register (DDR) is at $0007. Reset does not affect the data registers, but clears the data direction

registers, thereby returning the ports to inputs. Writing a 1 to a DDR bit sets the corresponding port bit to output mode.

5.5 INPUT/OUTPUT PROGRAMMING

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

At power-on or reset, all DDRs are cleared, which configures all pins as inputs. The data direction registers are capable of being written to or read by the processor. During the programmed output state, a read of the data register actually reads the value of the output data latch and not the I/O pin. Refer to **Table 5-1** and **Figure 5-1**.

Table 5-1: 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 mode. The output data latch is read.

R/W is an internal signal.

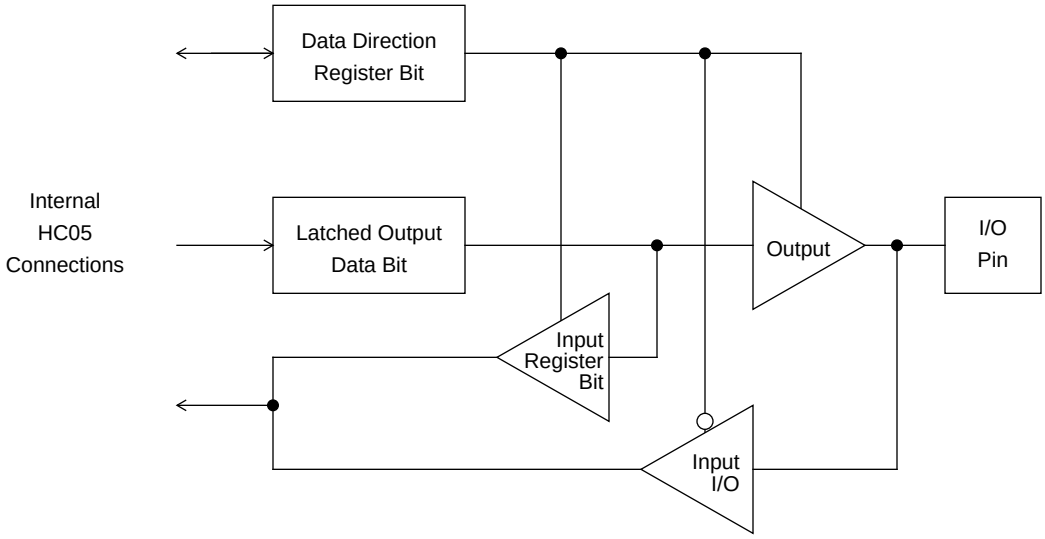


Figure 5-1: Port I/O Circuitry

SECTION 6 TIMER

6.1 INTRODUCTION

The timer consists of a 16-bit, software-programmable counter driven by a fixed divide-by-four prescaler. This timer can be used for many purposes, including input waveform measurements while simultaneously generating an output waveform. Pulse widths can vary from several microseconds to many seconds. Refer to **Figure 6-1** for a timer block diagram.

Because the timer has a 16-bit architecture, each specific functional segment (capability) is represented by two registers. These registers contain the high and low byte of that functional segment. Generally, accessing the low byte of a specific timer function allows full control of that function; however, an access of the high byte inhibits that specific timer function until the low byte is also accessed.

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

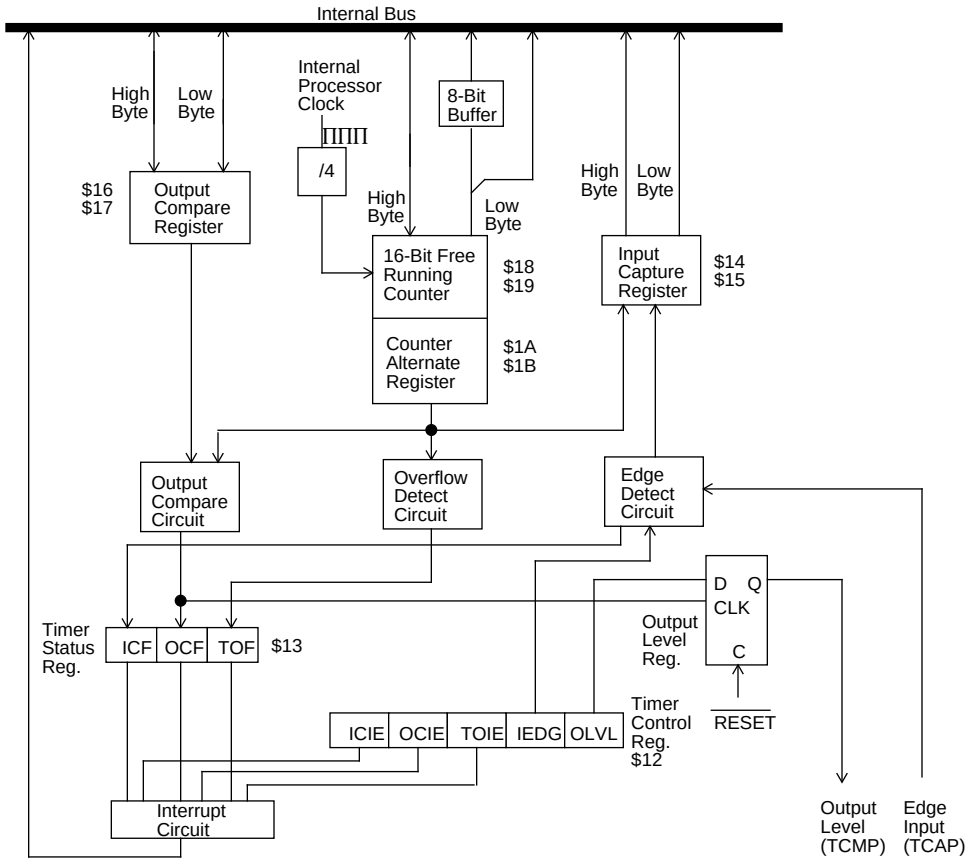


Figure 6-1: Timer Block Diagram

6.2 COUNTER

The key element in the programmable timer is a 16-bit, free-running counter or counter register, preceded by a prescaler that divides the internal processor clock by four. The prescaler gives the timer a resolution of 2.0 microseconds if the internal bus clock is 2.0 MHz. The counter is incremented during the low portion of the internal bus clock. Software can read the counter at any time without affecting its value.

The double-byte, free-running counter can be read from either of two locations, $18-$19 (counter register) or $1A-$1B (counter alternate register). A read from only the least significant byte (LSB) of the free-running counter ($19, $1B) receives the count value at the time of the read. If a read of the free-running counter or counter alternate register first addresses the most significant byte (MSB) ($18, $1A), the LSB ($19, $1B) is transferred to a buffer. This buffer value remains fixed after the first MSB read, even if the user reads the MSB several times. This buffer is accessed when reading the free-running counter or counter alternate register LSB ($19 or $1B) and, thus, completes a read sequence of the total counter value. In reading either the free-running counter or counter alternate register, if the MSB is read, the LSB must also be read to complete the sequence.

The counter alternate register differs from the counter register in one respect: A read of the counter register MSB can clear the timer overflow flag (TOF). Therefore, the counter alternate register can be read at any time without the possibility of missing timer overflow interrupts due to clearing of the TOF.

The free-running counter is configured to $FFFC during reset and is always a read-only register. During a power-on reset, the counter is also preset to $FFFC and begins running after the oscillator start-up delay. Because the free-running counter is 16 bits preceded by a fixed divided-by-four prescaler, the value in the free-running counter repeats every 262,144 internal bus clock cycles. When the counter rolls over from $FFFF to $0000, the TOF bit is set. An interrupt can also be enabled when counter rollover occurs by setting its interrupt enable bit (TOIE).

6.3 OUTPUT COMPARE REGISTER

The 16-bit output compare register is made up of two 8-bit registers at locations $16 (MSB) and $17 (LSB). The output compare register is used for several purposes, such as indicating when a period of time has elapsed. All bits are readable and writable and are not altered by the timer hardware or reset. If the compare function is not needed, the 2 bytes of the output compare register can be used as storage locations.

The output compare register contents are compared with the contents of the free-running counter continually, and if a match is found, the corresponding output compare flag (OCF) bit is set and the corresponding output level (OLVL) bit is clocked to an output level register. The output compare register values and the output level bit should be changed after each successful comparison to establish a new elapsed timeout. An interrupt can also accompany a successful output compare provided the corresponding interrupt enable bit (OCIE) is set.

After a processor write cycle to the output compare register containing the MSB ($16), the output compare function is inhibited until the LSB ($17) is also written. The user must write both bytes (locations) if the MSB is written first. A write made only to the LSB ($17) will not inhibit the compare function. The free-running counter is updated every four internal bus clock cycles. The minimum time required to update the output compare register is a function of the program rather than the internal hardware.

The processor can write to either byte of the output compare register without affecting the other byte. The output level (OLVL) bit is clocked to the output level register regardless of whether the output compare flag (OCF) is set or clear.

6.4 INPUT CAPTURE REGISTER

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

The result obtained by an input capture will be one more than the value of the free-running counter on the rising edge of the internal bus clock preceding the external transition. This delay is required for internal synchronization. Resolution is one count of the free-running counter, which is four internal bus clock cycles.

The free-running counter contents are transferred to the input capture register on each proper signal transition regardless of whether the input capture flag (ICF) is set or clear. The input capture register always contains the free-running counter value that corresponds to the most recent input capture.

After a read of the input capture register ($14) MSB, the counter transfer is inhibited until the LSB ($15) is also read. This characteristic causes the time used in the input capture software routine and its interaction with the main program to determine the minimum pulse period.

A read of the input capture register LSB ($15) does not inhibit the free-running counter transfer since they occur on opposite edges of the internal bus clock.

6.5 TIMER CONTROL REGISTER (TCR) $12

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

$12	ICIE	OCIE	TOIE	0	0	COE	IEDG	OLVL
RESET:	0	0	0	0	0	0	U	0

Figure 6-2: Timer Control Register

ICIE - Input Capture Interrupt Enable

1 = Interrupt enabled

0 = Interrupt disabled

OCIE - Output Compare Interrupt Enable

1 = Interrupt enabled

0 = Interrupt disabled

TOIE - Timer Overflow Interrupt Enable

1 = Interrupt enabled

0 = Interrupt disabled

COE - TCMP Pin Enable

1 = TCMP pin enabled

0 = TCMP pin disabled (pin 35 is PD6)

IEDG - Input Edge

Value of input edge determines which level transition on TCAP pin will trigger free-running counter transfer to the input capture register

1 = Positive edge

0 = Negative edge

Reset does not affect the IEDG bit (U=unaffected).

OLVL - Output Level

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

1 = High output

0 = Low output

Bits 3 and 4 - Not used

Always read zero

6.6 TIMER STATUS REGISTER (TSR) $13

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

$13	ICF	OCF	TOF	0	0	0	0	0
RESET:	U	U	U	0	0	0	0	0

Figure 6-3: Timer Status Register

ICF - Input Capture Flag

1 = Flag set when selected polarity edge is sensed by input capture edge detector

0 = Flag cleared when TSR and input capture low register ($15) are accessed

OCF - Output Compare Flag

1 = Flag set when output compare register contents match the free-running counter contents

0 = Flag cleared when TSR and output compare low register ($17) are accessed

TOF - Timer Overflow Flag

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

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

Bits 0-4 - Not used

Always read zero

Accessing the timer status register satisfies the first condition required to clear status bits. The remaining step is to access the register corresponding to the status bit.

A problem can occur when using the timer overflow function and reading the free-running counter at random times to measure an elapsed time. Without incorporating the proper precautions into software, the timer overflow flag could unintentionally be cleared if:

1) The timer status register is read or written when TOF is set, and

2) The LSB of the free-running counter is read but not for the purpose of servicing the flag.

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

6.7 TIMER DURING WAIT MODE

The CPU clock halts during the WAIT mode, but the timer remains active. If interrupts are enabled, a timer interrupt will cause the processor to exit the WAIT mode.

6.8 TIMER DURING STOP MODE

In the STOP mode, the timer stops counting and holds the last count value if STOP is exited by an interrupt. If RESET is used, the counter is forced to $FFFC. During STOP, if at least one valid input capture edge occurs at the TCAP pin, the input capture detect circuit is armed. This does not set any timer flags nor wake up the MCU, but when the MCU does wake up, there is an active input capture flag and data from the first valid edge that occurred during the STOP mode. If RESET is used to exit STOP mode, then no input capture flag or data remains, even if a valid input capture edge occurred.

SECTION 7 SIMPLE SERIAL INPUT/OUTPUT PORT

This device includes a simple synchronous Serial I/O Port (SIOP). The SIOP is a three-wire master/slave system including Serial Clock (SCK), Serial Data Input (SDI), and Serial Data Output (SDO). A mask programmable option determines whether the SIOP is MSB or LSB first.

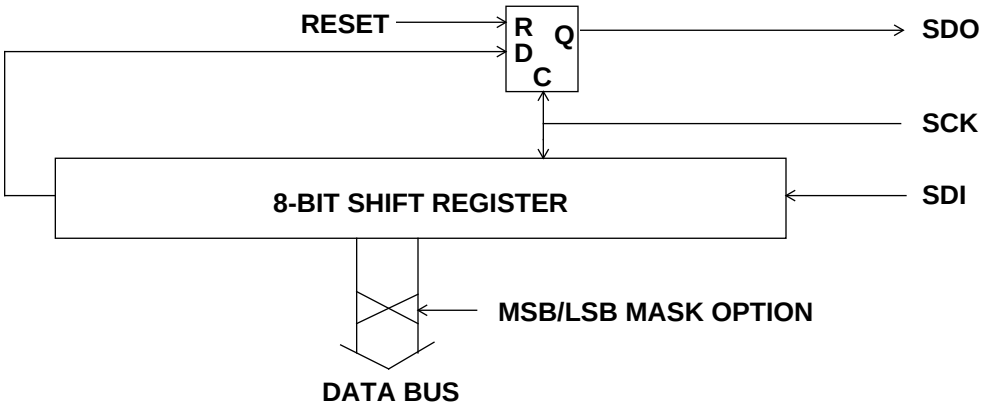


Figure 7-1: SIOP Block Diagram

7.1 SIGNAL FORMAT

7.1.1 SCK

The state of SCK between transmissions must be logic ’1’ for CPOL set and logic ’0’ for CPOL clear. The first transition of SCK signals the beginning of a transmission. At this time, the first bit of received data is accepted at the SDI pin and the first bit of transmitted data is presented at the SDO pin. Data is captured at the SDI pin on the rising edge of SCK. Subsequent falling edges shift the data and accept or present the next bit. The transmission is ended upon the eighth rising edge of SCK. The maximum frequency of SCK in slave mode is equal to E (bus clock) divided by 4. That is for a 4 MHz oscillator input E becomes 2 MHz and the maximum SCK frequency is 500 KHz. There is no minimum SCK frequency.

In master mode, the format is identical except that the SCK pin is an output and the shift clock now originates internally. The master mode transmission frequency is fixed at E/4.

7.1.2 SDO

A mask programmable option will be included to allow data to be transmitted in either MSB first format or LSB first format. In either case, the state of the SDO pin will always reflect the value of the first bit received on the previous transmission if there was one. Upon enabling the SIOP, SDO will always be driven to a logic one by the SIOP subsystem.

While the SIOP is enabled, PB5 can not be used as a standard output since that pin is coupled to the last stage of the serial shift register. If CPOL is set, the first falling edge of SCK will shift the first data bit out to the output pin. If CPOL is clear, the first data bit will be on the SDO pin waiting for the transmission.

7.1.3 SDI

The SDI pin becomes an input as soon as the SIOP is enabled. New data may be presented to the SDI pin on the falling edge of SCK. Valid data must be present at least t_S before the rising edge of the clock and remain valid for t_H after the edge.

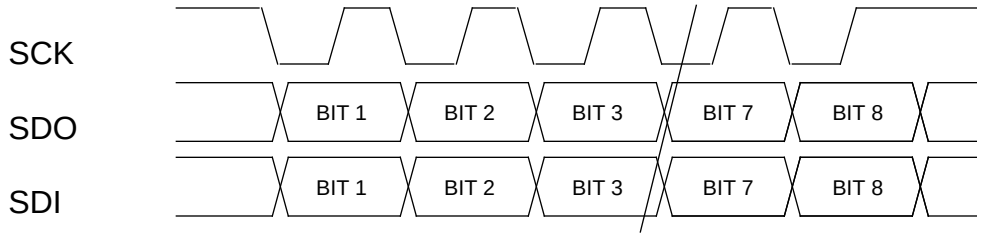


Figure 7-2: Serial I/O Port Timing (CPOL=1)

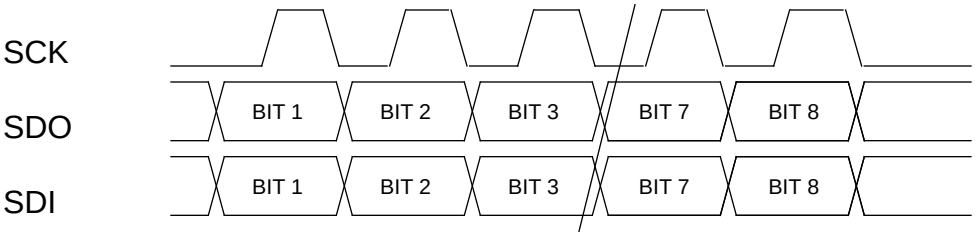


Figure 7-3: Serial I/O Port Timing (CPOL=0)

7.2 SIOP REGISTERS

7.2.1 SIOP CONTROL REGISTER (SCR)

This register is located at address $000A and contains 3 bits.

$0A	0	SPE	0	MSTR	CPOL	0	0	0
RESET:	0	0	0	0	1	0	0	0

Figure 7-4: SIOP Control Register

7.2.1.1 SPE - SERIAL PERIPHERAL ENABLE

When set, this bit enables the Serial I/O Port and initializes the Port B DDR such that PB5 (SDO) is output, PB6 (SDI) is input and PB7 (SCK) is input (slave mode only). The Port B DDR can be subsequently altered as the application requires and the Port B data register (except for PB5) can be manipulated as usual. However, these actions could affect the transmitted or received data. When SPE is cleared, Port B reverts to standard parallel I/O without affecting the Port B data register or DDR. SPE is readable and writable any time but clearing SPE while a transmission is in progress will abort the transmission, reset the bit counter, and return Port B to its normal I/O function. Reset clears this bit.

7.2.1.2 MSTR - MASTER MODE

When set, this bit configures the SIOP for master mode. This means that the transmission is initiated by a write to the data register and the SCK pin becomes an output providing a synchronous data clock at a fixed rate of E (bus clock) divided by 4. While the device is in master mode, the SDO and SDI pins do not change function. These pins behave exactly as they would in slave mode. Reset clears this bit and configures the SIOP for slave operation. MSTR may be set at any time regardless of the state of SPE. Clearing MSTR will abort any transmission in progress.

7.2.1.3 CPOL - CLOCK POLARITY

The Clock Polarity bit controls the SCK polarity between transmissions. When this bit is cleared, SCK will be low between transmissions. When this bit is set, SCK will be high between transmissions. In both cases, the data is latched on the rising edge of SCK for serial input and is valid on the rising edge of SCK for serial output. Reset sets this bit.

When using the Clock Polarity low mode (CPOL=0), the proper mode should be entered before enabling the serial system. The CPOL bit should be cleared first. Then the SPE bit should be set during a second write to the SCR. The following example shows a proper sequence.

```
* For Master Mode CPOL=0

LDA#$00

STASCRclear CPOL

LDA#$50

STASCRset Mstr, set SPE


* For Slave Mode CPOL=0

LDA#$00

STASCRclear CPOL

LDA#$40
```

STASCRset SPE

7.2.2 SIOP STATUS REGISTER (SSR)

This register is located at address $000B and contains only 2 bits.

$0B	SPIF	DCOL	0	0	0	0	0	0
$\overline{\text{RESET}}$:	0	0	0	0	0	0	0	0

Figure 7-5: SIOP Status Register

7.2.2.1 SPIF - SERIAL PERIPHERAL INTERFACE FLAG

This bit is set upon occurrence of the last rising clock edge if CPOL is set and the last falling clock edge of CPOL is clear to indicates that a data transfer has taken place. It has no effect on any further transmissions and can be ignored without problem. SPIF is cleared by reading the SSR with SPIF set followed by a read or write of the serial data register. If SPIF is cleared before the last edge of the next byte, it will be set again. Reset clears this bit.

7.2.2.2 DCOL - DATA COLLISION

This is a read-only status bit which indicates that an invalid access to the data register has been made. This can occur any time after the first falling edge of SCK if CPOL is set and after the first rising edge of SCK if CPOL is clear and before SPIF is set. A read or write of the data register during this time will result in invalid data being transmitted or received.

DCOL is cleared by reading the status register with SPIF set followed by a read or write of the data register. If the last part of the clearing sequence is done after another transmission has been started, DCOL will be set again. Reset also clears this bit.

7.2.3 SIOP DATA REGISTER (SDR)

This register is located at address $000C and is both the transmit and receive data register. This system is not double buffered and any write to this register will destroy the previous contents. The SDR can be read at any time, but if a transmission is in progress the results may be ambiguous. Writes to the SDR while a transmission is in progress can cause invalid data to be transmitted and/or received. This register can be read and written only when the SIOP is enabled (SPE=1).

$0C								
RESET:	U	U	U	U	U	U	U	U

Figure 7-6: SIOP Data Register

SECTION 8 COMPUTER OPERATING PROPERLY

8.1 INTRODUCTION

This device includes a "Watchdog" Computer Operating Properly (COP) feature as a mask option. The COP is implemented with an 18-bit ripple counter. This provides a timeout period of 64 milliseconds at a bus rate of 2 MHz. If the COP should timeout, a system reset will occur and the device will be re-initialized in the same fashion as a POR or external reset.

8.2 RESETTING THE COP

Preventing a COP reset is done by writing a "0" to the COPF bit. This action will reset the counter and begin the timeout period again. The COPF bit is bit 0 of address $1FF0. A read of address $1FF0 will result in the user defined ROM data at that location.

8.3 COP TEST FEATURES

For speeding up the COP test, a feature was added in the Self-Check mode to split the 18-bit counter into 6-bit and 12-bit counters clocked in parallel where the output of the 6-bit counter drives the COP logic. Splitting the counter is accomplished by writing a "1" to bit 7 of $1FF0. Writing a "0" to bit 7 of $1FF0 will reconnect the two halves of the COP counter.

8.4 COP DURING WAIT MODE

The COP will continue 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.

8.5 COP DURING STOP MODE

STOP mode disables the oscillator circuit and thereby turns the clock off for the entire device. The COP counter will be reset when STOP mode is entered. If a reset is used to exit STOP mode, the COP counter will be reset after the 4064 cycles of delay after STOP mode. If an $\overline{\text{IRQ}}$ is used to exit STOP mode, the COP counter will not be reset after the 4064-cycle delay and will have that many cycles already counted when control is returned to the program.

SECTION 9 ELECTRICAL SPECIFICATIONS

9.1 MAXIMUM RATINGS

(Voltages referenced to V_{SS})

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 $2 \times V_{DD}$ + 0.3	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	25	mA
Operating Temperature Range MC68HC05C5P (Standard) MC68HC05C5CP (Extended)	T_A	T_L to T_H 0 to +70 -40 to +85	°C
Storage Temperature Range	T_{STG}	-65 to +150	°C

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

9.2 THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic DIP Plastic Leaded Chip Carrier	 θ_{JA} θ_{JA}	 60 70	 °C/W °C/W

9.3 DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc, T_A = -40°C to +85°C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output voltage I_{Load} = 10.0 µA I_{Load} = -10.0 µA	 V_{OL} V_{OH}	 — V_{DD}-0.1	 — —	 0.1 —	 V
Output High Voltage (I_{Load} = -0.8 mA) PA0-7, PB0-7,PC0-7,PD0-7	V_{OH}	V_{DD}-0.8	—	—	V
Output Low Voltage (I_{Load} = 1.6 mA) PA0-7, PB0-7,PD0-7 (I_{Load} = 10 mA) PC0-7	V_{OL}	—	—	0.4	V
Input High Voltage PA0-7, PB0-7,PC0-7, PD0-7, $\overline{IRQ}$,RESET,OSC1, TCAP	V_{IH}	0.7 × V_{DD}	—	V_{DD}	V
Input Low Voltage PA0-7, PB0-7,PC0-7, PD0-7, $\overline{IRQ}$,RESET,OSC1, TCAP	V_{IL}	V_{SS}	—	0.2 × V_{DD}	V
Low Voltage Programming Inhibit	V_{LVPI}	3.5	—	—	V
Low Voltage Programming Recover	V_{LVPR}	—	—	4.5	V
Low Voltage Programming Inhibit/Recover Hysteresis	H_{LVPI}	0.1	—	1.0	V
Supply Current (see Notes) Run Wait Stop with LVPI Enabled 25°C -40°C to +85°C Stop with LVPI Disabled 25°C -40°C to +85°C	 I_{DD} I_{DD} I_{DD} I_{DD} I_{DD} I_{DD}	 — — — — — —	 3.0 1.0 200 200 2 5	 7.0 4.0 300 300 50 140	 mA mA µA µA µA µA
I/O Ports Hi-Z Leakage Current PA0-7, PB0-7,PC0-7,PD0-7	I_{OZ}	—	—	10	µA
Input Current $\overline{RESET}$, $\overline{IRQ}$, OSC1,TCAP, PE PE	 I_{IN} I_{IN}	 — —	 — —	 1 1	 µA µA
Capacitance Ports (as Input or Output) $\overline{RESET}$, $\overline{IRQ}$, TCAP,PE PE	 C_{OUT} C_{IN} C_{IN}	 — — —	 — — —	 12 8 8	 pF pF pF

NOTES:

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

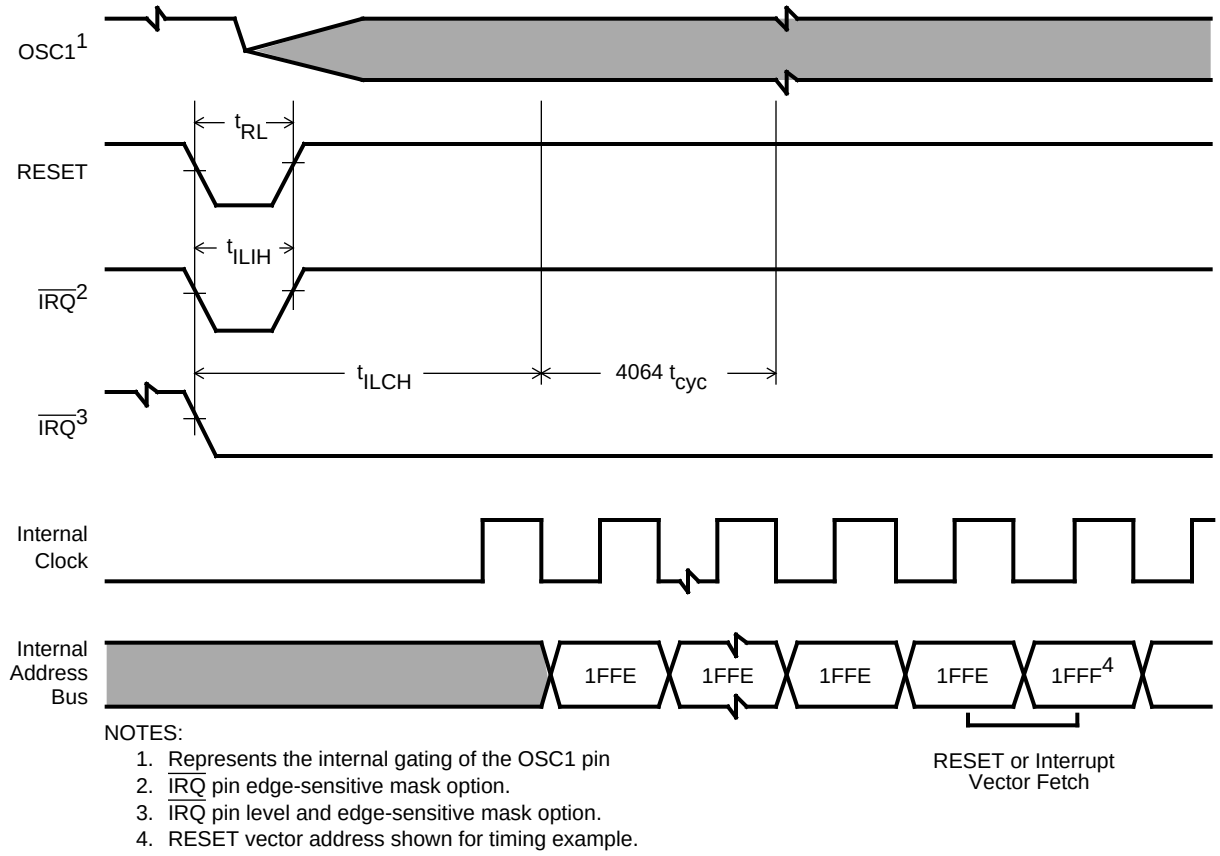
9.4 CONTROL TIMING

(V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc, T_A = -40°C to +85 °C, unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Crystal Option External Clock Option	 f_{osc} f_{osc}	 — dc	 4.2 4.2	 MHz MHz
Internal Operating Frequency Crystal (f_{osc} ÷ 2) External Clock (f_{osc} ÷ 2)	 f_{OP} f_{OP}	 — dc	 2.1 2.1	 MHz MHz
Cycle Time	t_{CYC}	480	—	ns
Crystal Oscillator Start-up Time	t_{OXOV}	—	100	ms
Stop Recovery Start-up Time (Crystal Oscillator)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ Pulse Width	t_{RL}	1.5	—	t_{CYC}
Interrupt Pulse Width Low (Edge-Triggered)	t_{ILIH}	125	—	ns
Interrupt Pulse Period	t_{ILIL}	*	—	t_{CYC}
OSC1 Pulse Width	t_{OH},t_{OL}	90	—	ns
EEPROM Byte Programming Time	t_{EPGM}	—	15.0	ms
EEPROM Byte Erase Time	t_{EBYT}	—	15.0	ms
EEPROM Block Erase Time	t_{EBLOCK}	—	30.0	ms
EEPROM Bulk Erase Time	t_{EBULK}	—	100.0	ms
EEPROM Programming Voltage Fall Time Normal Operation After LVPI set	 t_{FPV} t_{FPVL}	 — —	 10.0 10.0	 µs µs
V_{DD} Slew Rate Rising Falling	 t_{VDDR} t_{VDDF}	 — —	 0.05 0.1	 V/µs V/µs
RC Oscillator Stabilization Time (EEPROM)	t_{RCON}	—	5.0	µs

* The minimum period T_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.

CONTROL TIMING



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

Figure 9-1: Stop Recovery Timing Diagram

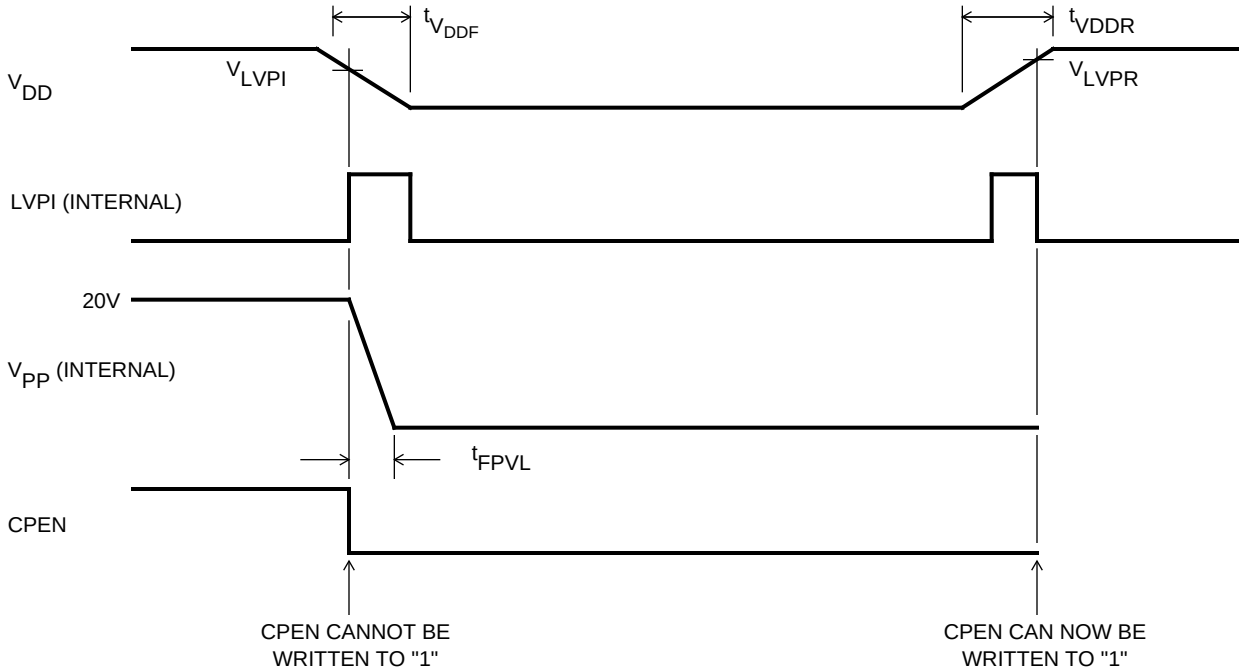


Figure 9-2: LVPI Timing Diagram

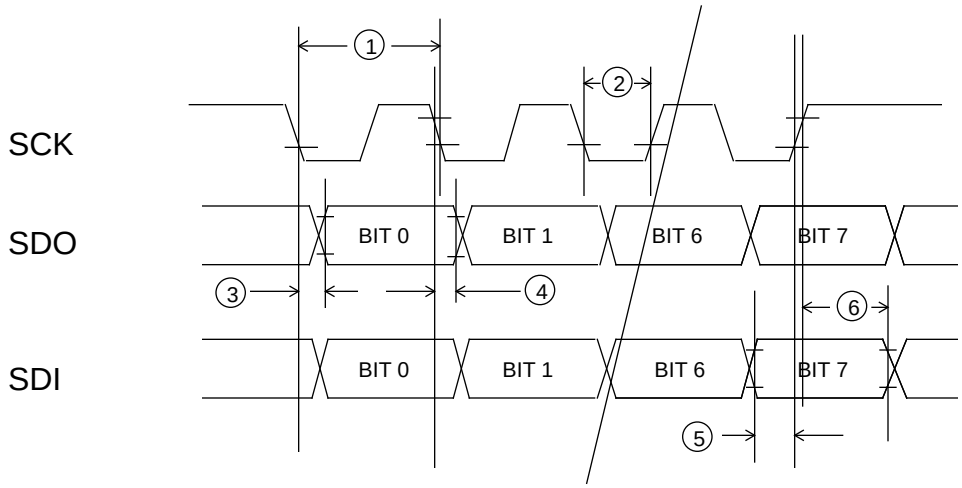


Figure 9-3: SIOP Timing Diagram

NOTE: Clock Polarity (CPOL) = 1 and Data LSB first shown for example only.

9.5 SIOP TIMING

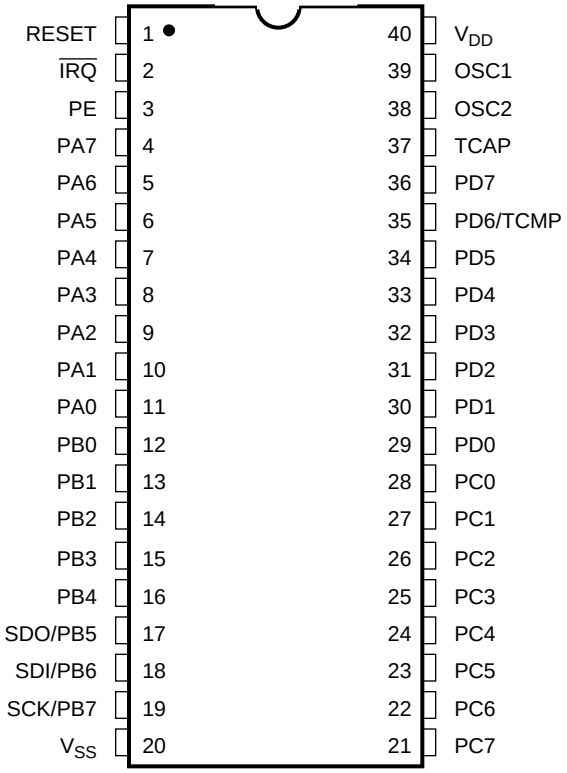
(V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc, T_A = -40°C to +85°C, unless otherwise noted)

Num.	Characteristic	Symbol	Min	Max	Unit
	Operating Frequency Master Slave	 $f_{OP(M)}$ $f_{OP(S)}$	 dc dc	 0.25 0.25	 f_{OP} f_{OP}
1	Cycle Time Master Slave	 $t_{CYC(M)}$ $t_{CYC(S)}$	 4.0 —	 4.0 4.0	 t_{CYC} t_{CYC}
2	Clock (SCK) Low Time	t_{CYC}	932	—	ns
3	SDO Data Valid Time	t_V	—	200	ns
4	SDO Hold Time	t_{HO}	0	—	ns
5	SDI Setup Time	t_S	100	—	ns
6	SDI Hold Time	t_H	100	—	ns

NOTES:
1. f_{OP} = 2.1 MHz max.

SECTION 10 MECHANICAL SPECIFICATIONS

10.1 40-PIN DUAL INLINE PACKAGE



10.2 44-PIN PLCC PACKAGE