68HC05P4A

SPECIFICATION

(General Release)

CSIC MCU Design Group
Oak Hill, Texas

TABLE OF CONTENTS

Section **Title** **Page**

SECTION 1
GENERAL DESCRIPTION

SECTION 2
MEMORY

SECTION 3
CENTRAL PROCESSING UNIT

TABLE OF CONTENTS

Section **Title** **Page**

TABLE OF CONTENTS

TABLE OF CONTENTS

Section **Title** **Page**

LIST OF FIGURES

LIST OF TABLES

SECTION 1
GENERAL DESCRIPTION

The MC68HC05P4A is a 28-pin device based on the MC68HC05P4. The memory map includes 4160 bytes of user ROM and 176 bytes of RAM. The MCU has two 8-bit input/output (I/O) ports, A and C. Port B has three I/O pins and port D has two pins, one that is I/O and the other input only. The MC68HC05P4A includes a simple serial I/O peripheral (SIOP) and an on-chip mask programmable computer operating properly (COP) watchdog circuit.

1.1 Features

- Low Cost
- HC05 Core
- 28-Pin Package
- On-Chip Oscillator with RC or Crystal/Ceramic Resonator Mask Options
- 4160 Bytes of User ROM Including 16 User Vector Locations
- ROM Security Feature
- 176 Bytes of On-Chip RAM
- 16-Bit Timer

- Edge-Sensitive or Edge- and Level-Sensitive Interrupt Trigger Mask Option
- Simple Serial Input/Output Port
- Mask Option Selectable Watchdog Timer (COP)

1.2 Mask Options

There are 13 mask options on the MC68HC05P4A:

- CLOCK (RC or Crystal)
- $\overline{\text{IRQ}}$ (Edge-Sensitive Only or Edge- and Level-Sensitive)
- SIOP (MSB or LSB First)
- COP Watchdog Timer (Enable/Disable)
- Keyscan Pullups and Interrupts on Port A (Enable/Disable by Pin).
- Stop Instruction (Option to Convert to Halt)

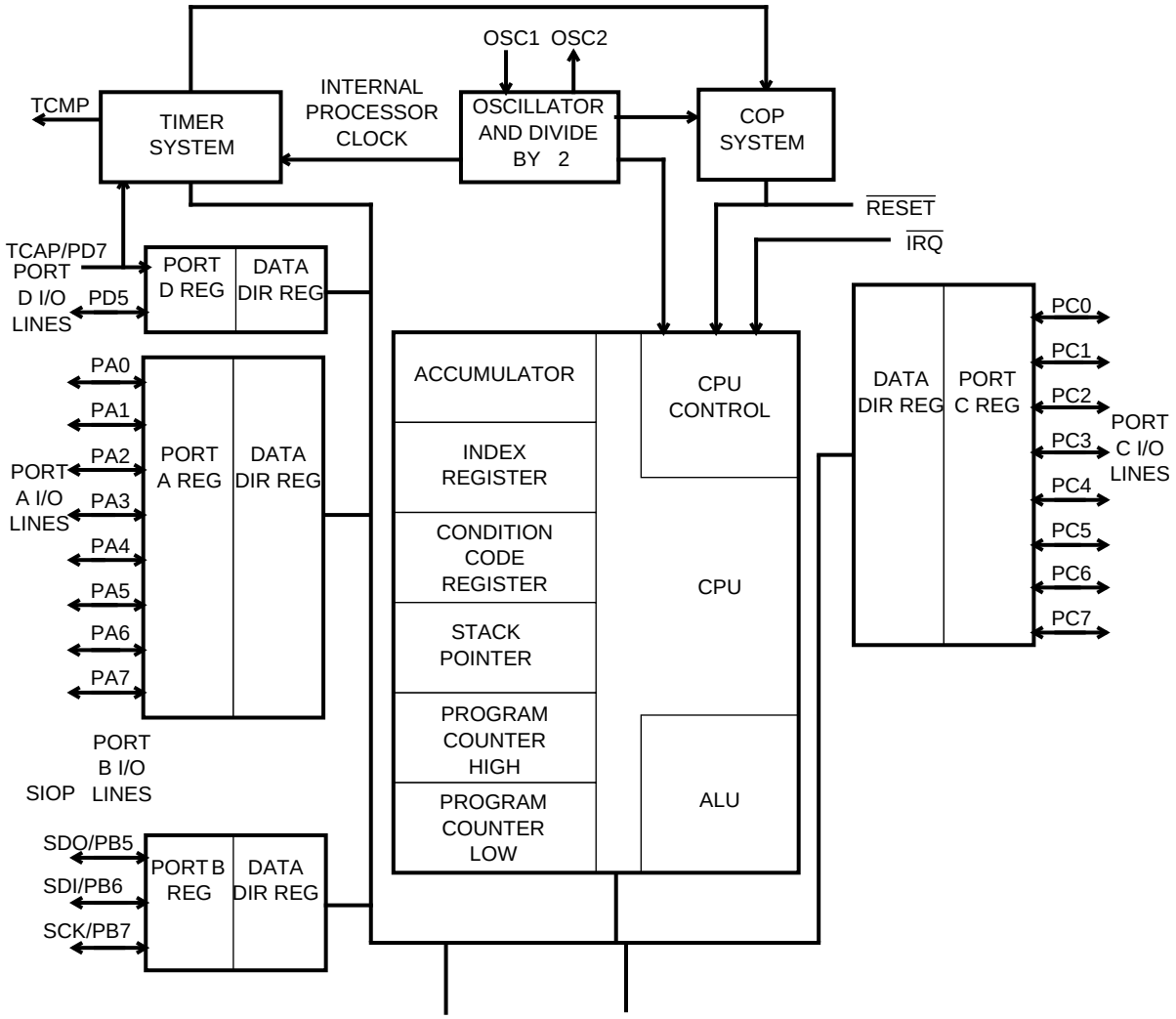
All mask options and the user ROM are programmed on the 01 layer in fabrication.

NOTE

Negative true signals like $\overline{\text{RESET}}$ and $\overline{\text{IRQ}}$ will be denoted with an overline.

1.3 MCU Structure

Figure 1-1 shows the structure of the MC68HC05P4A.



1.4 Pin Assignments

The MC68HC05P4A pin assignments are shown in Figure 1-2..

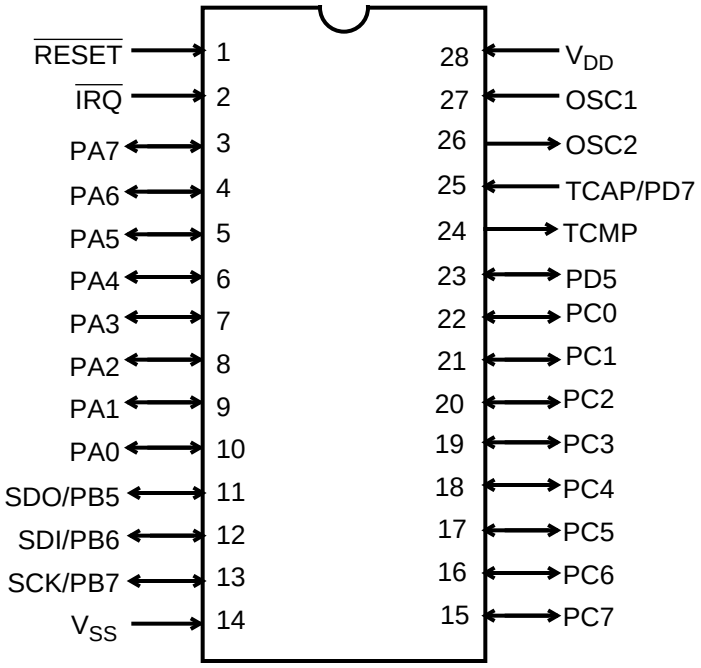


Figure 1-2. Pin Assignments

1.5 Signal Description

The following paragraphs provide a description of the signals.

1.5.3 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 and provides 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.

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

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.

1.5.5 TCMP

This pin provides an output for the output compare feature of the on-chip timer system.

1.5.6 PA0 through PA7

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 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. Port A has mask option enabled pullup devices and interrupt capability by pin. For a detailed description of I/O programming, refer to **1.6 Input/Output Programming**.

1.5.7 SDO/PB5, SDI/PB6, and SCK/PB7

Port B is a 3-bit bidirectional port. These pins are shared with the SIOP subsystem. Refer to **SECTION 7 SIMPLE SERIAL INPUT/OUTPUT PORT** for a detailed description of the SIOP. The address of the port B data register is $0001 and the data direction register 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.

1.5.8 PC0 through PC7

Port C is an 8-bit bidirectional port which does not share any of its pins with other subsystems. The address of the port C data register is $0002 and the data direction register is at address $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. Two of the port C pins, PC0 and PC1, have a higher current drive capability. See **SECTION 12 ELECTRICAL SPECIFICATIONS**.

1.5.9 PD5 and TCAP/PD7

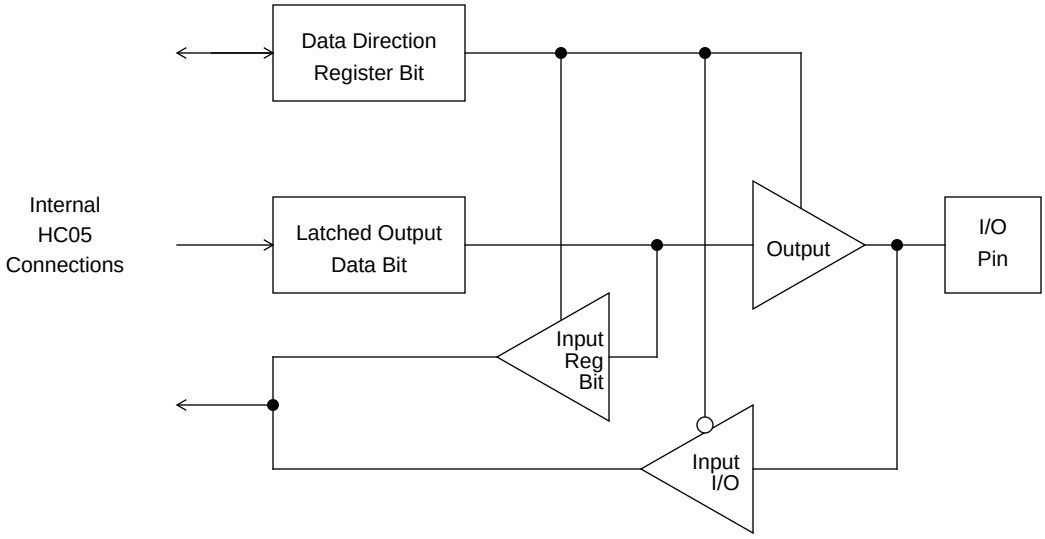
Port D is a 2-bit port. PD5 is I/O and TCAP/PD7 is input-only shared with the timer input capture. The address of the port D data register is $0003 and the data direction register is at address $0007. Reset does not affect the data registers, but clears the data direction registers, thereby returning the ports to inputs. Writing a one to a DDR bit sets the corresponding port bit to output mode. The TCAP/PD7 pin controls the input capture feature for the on-chip programmable timer. This pin can be read at any time even if the TCAP function is enabled.

1.6 Input/Output Programming

Table 1-1. I/O Pin Functions

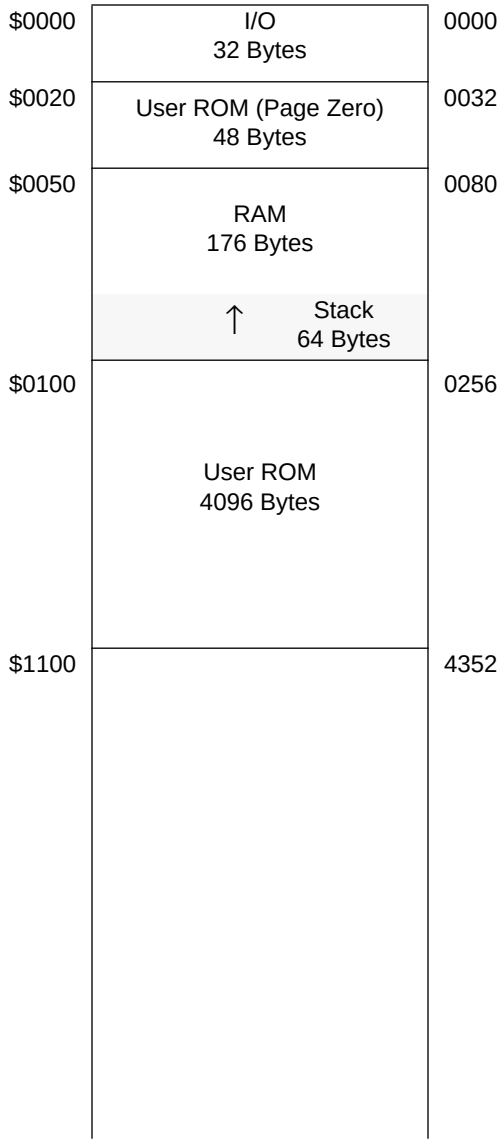
R/$\overline{W}$	DDR	I/O Pin Function
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode. The output data latch is read.

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



SECTION 2
MEMORY

The MC68HC05P4A has a 8 Kbyte memory map, consisting of user ROM, user RAM, self-check ROM, and I/O. See Figure 2-1 and Figure 2-2.



ADDRESS $0000 to $001F	DATA							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
$00 PORT A DATA								
$01 PORT B DATA				0	0	0	0	0
$02 PORT C DATA								
$03 PORT D DATA		0		1	0	0	0	0
$04 PORT A DDR								
$05 PORT B DDR				1	1	1	1	1
$06 PORT C DDR								
$07 PORT D DDR	0	0		0	0	0	0	0
$08 UNUSED								
$09 UNUSED								
$0A SERIAL CTRL		SPE		MSTR				
$0B SERIAL STAT	SPF	DCOL						
$0C SERIAL DATA								
$0D UNUSED								
$0E UNUSED								
$0F UNUSED								
$10 UNUSED								
$11 UNUSED								
$12 TIMER CONTROL	ICIE	OCIE	TOIE				IEDG	OLVL
$13 TIMER STATUS	ICF	OCF	TOF					
$14 CAPTURE HIGH								
$15 CAPTURE LOW								

2.1 ROM

The user ROM consists of 48 bytes of page zero ROM from $0020 to $004F, 4096 bytes of ROM from $0100 to $10FF, and 16 bytes of user vectors from $1FF0 to $1FFF. The self-check ROM and vectors are located from $1F00 to $1FEF.

2.2 ROM Security Feature

A security feature has been incorporated into the MC68HC05P4A to help prevent external reading of code in the ROM. Placing unique customer code at ROM locations $0028-$002F aids in keeping customer developed software proprietary.

2.3 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.

NOTE

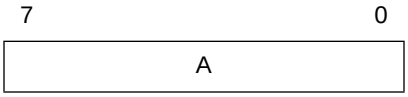
Using the stack area for data storage or temporary work locations requires care to prevent it from being overwritten due to stacking from an interrupt or subroutine call.

SECTION 3
CENTRAL PROCESSING UNIT

This section describes the five CPU registers. CPU registers are not part of the memory map.

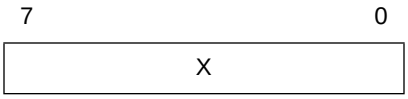
3.1 Accumulator (A)

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



3.2 Index Register (X)

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



3.3 Condition Code Register (CCR)

3.3.2 I — Interrupt

When this bit is set, timer and external interrupts are 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.

3.3.3 N — Negative

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

3.3.4 Z — Zero

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

3.3.5 C — Carry/Borrow

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

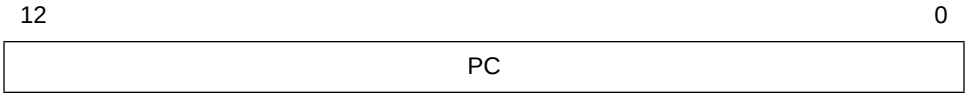
3.4 Stack Pointer (SP)

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

When accessing memory, the seven most significant bits are permanently set to

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 a 64 Kbyte memory map. For this implementation, however, the addressing registers are limited to an 8 Kbyte memory map.

SECTION 4
INTERRUPTS

The MCU can be interrupted four different ways: the two maskable hardware interrupts ($\overline{IRQ}$ and timer), the non-maskable software interrupt instruction (SWI), and by the optional external asynchronous interrupt on each port A pin (enabled by pullup mask option).

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 $\overline{RESET}$, 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

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 Capture	TIMER	$1FF8-$1FF9
TSR	TOF	Timer Overflow	TIMER	$1FF8-$1FF9

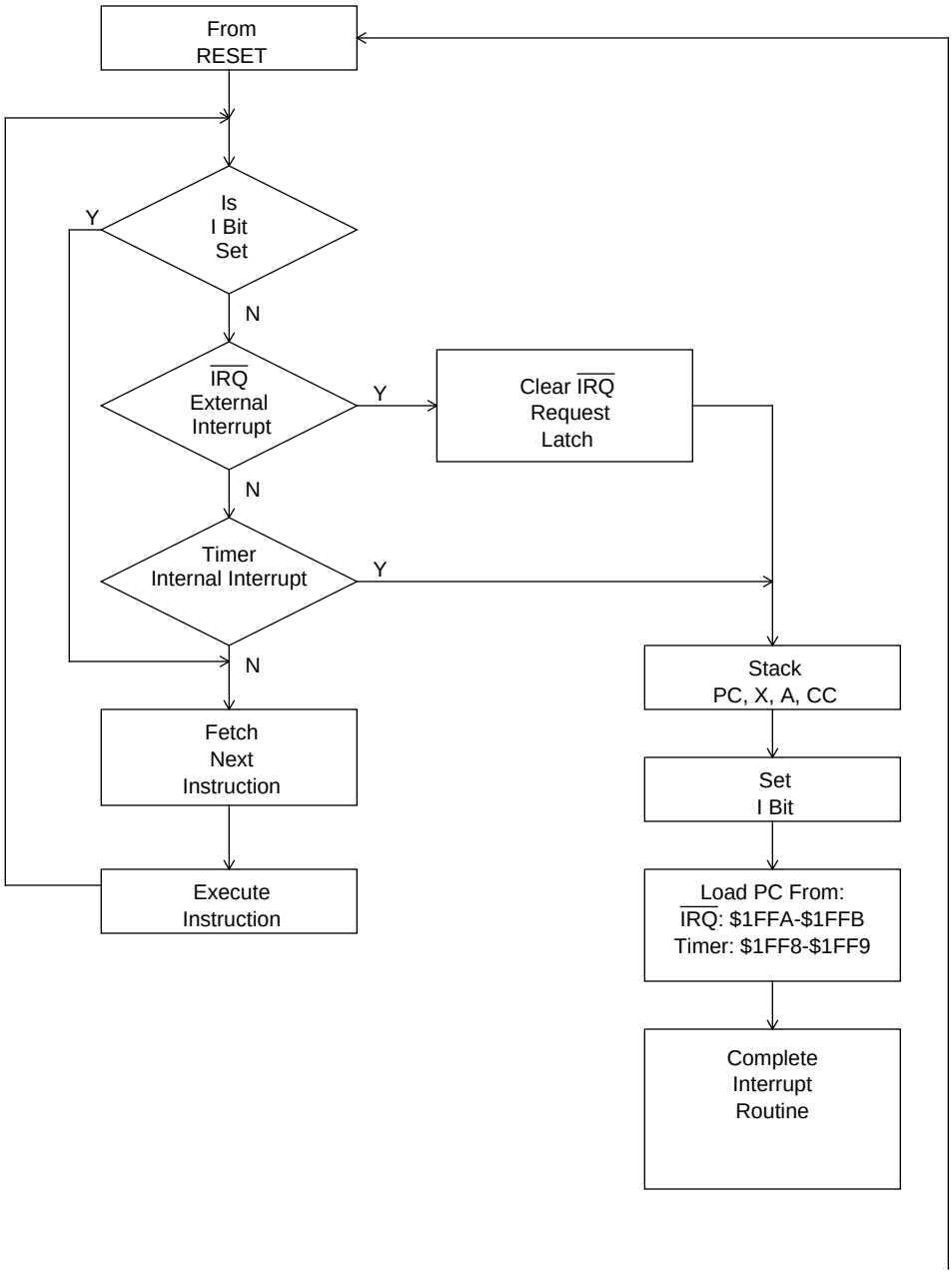
4.1 Hardware Controlled Interrupt Sequence

The following three functions ($\overline{\text{RESET}}$, STOP, and WAIT) are not interrupts in the strictest sense. However, they are acted upon in a similar manner. Flowcharts for hardware interrupts are shown in Figure 4-1, and for STOP and WAIT in Figure 6-1. 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 also is set. Much of the MCU is configured to a known state during this type of reset as described in **SECTION 5 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.See **6.1 Stop Mode**.

3. WAIT or HALT — The WAIT or HALT instruction causes all processor clocks to stop, but leaves the timer clock running. This rest state of the



4.3 External Interrupt

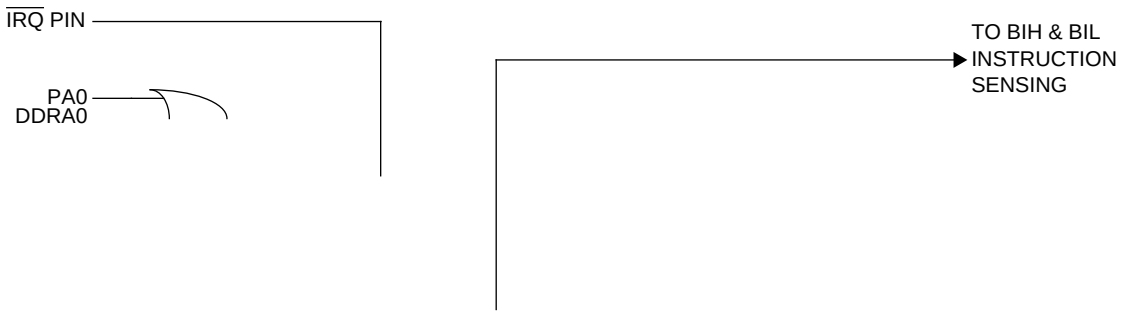
The $\overline{\text{IRQ}}$ pin drives an asynchronous interrupt to the CPU. An edge detector flip-flop is latched on the falling edge of $\overline{\text{IRQ}}$. If either the output from the internal edge detector flip-flops or the level on the $\overline{\text{IRQ}}$ pin is low, a request is synchronized to the CPU to generate the IRQ interrupt. If the edge-sensitive only mask 0ption is selected, the output of the internal edge detector flip-flop is sampled and the input level on the $\overline{\text{IRQ}}$ pin is ignored. The interrupt service routine address is specified by the contents of memory locations $1FFA and $1FFB. A block diagram of the IRQ function is shown in Figure 4-2.

NOTE

The internal interrupt latch is cleared nine PH2 clock cycles after the interrupt is recognized (after location $1FFA is read). Therefore, another external interrupt pulse can be latched during the IRQ service routine.

NOTE

When the edge- and level-sensitive mask option is selected, the voltage applied to the $\overline{\text{IRQ}}$ pin must return to the high state before the RTI instruction in the interrupt service routine is executed to avoid the processor re-entering the IRQ service routine.



The $\overline{IRQ}$ pin is one source of an IRQ interrupt and a mask option can also enable the port A pins (PA0 thru PA7) to act as other IRQ interrupt sources. These sources are all combined into a single ORing function to be latched by the IRQ latch.

Any enabled IRQ interrupt source sets the IRQ latch on the falling edge of the $\overline{IRQ}$ pin or a port A pin if port A interrupts have been enabled. If edge-only sensitivity is chosen by a mask option, only the IRQ latch output can activate a request to the CPU to generate the IRQ interrupt sequence. This makes the IRQ interrupt sensitive to the following cases:

1. Falling edge on the $\overline{IRQ}$ pin with all enabled port A interrupt pins at a high level.

2. Falling edge on any enabled port A interrupt pin with all other enabled port A interrupt pins and the $\overline{IRQ}$ pin at a high level.

If level sensitivity is chosen, the active high state of the IRQ input can also activate an IRQ request to the CPU to generate the IRQ interrupt sequence. This makes the IRQ interrupt sensitive to the following cases:

1. Low level on the $\overline{IRQ}$ pin.

2. Falling edge on the $\overline{IRQ}$ pin with all enabled port A interrupt pins at a high level.

3. Low level on any enabled port A interrupt pin.

4. Falling edge on any enabled port A interrupt pin with all enabled port A interrupt pins on the $\overline{IRQ}$ pin at a high level.

This interrupt is serviced by the interrupt service routine located at the address specified by the contents of $1FFA and $1FFB. The IRQ latch is automatically

4.4 Optional External Interrupts (PA0-PA7)

The IRQ interrupt can be triggered by the inputs on the PA0 thru PA7 port pins if enabled by individual mask options. With pullup enabled, each port A pin can activate the IRQ interrupt function and the interrupt operation will be the same as for inputs to the $\overline{\text{IRQ}}$ pin. Once enabled by mask option, each individual port A pin can be disabled as an interrupt source if its corresponding DDR bit is configured for output mode.

NOTE

The BIH and BIL instructions apply to the output of the logic OR function of the enabled PA0 thru PA7 interrupt pins and the $\overline{\text{IRQ}}$ pin. The BIH and BIL instructions to do not exclusively test the state of the $\overline{\text{IRQ}}$ pin.

NOTE

If enabled, the PA0 thru PA7 pins will cause an IRQ interrupt only if these individual pins are configured as inputs.

4.5 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.

SECTION 5
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 computer operating properly (COP) watchdog-timer timeout.

5.1 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.

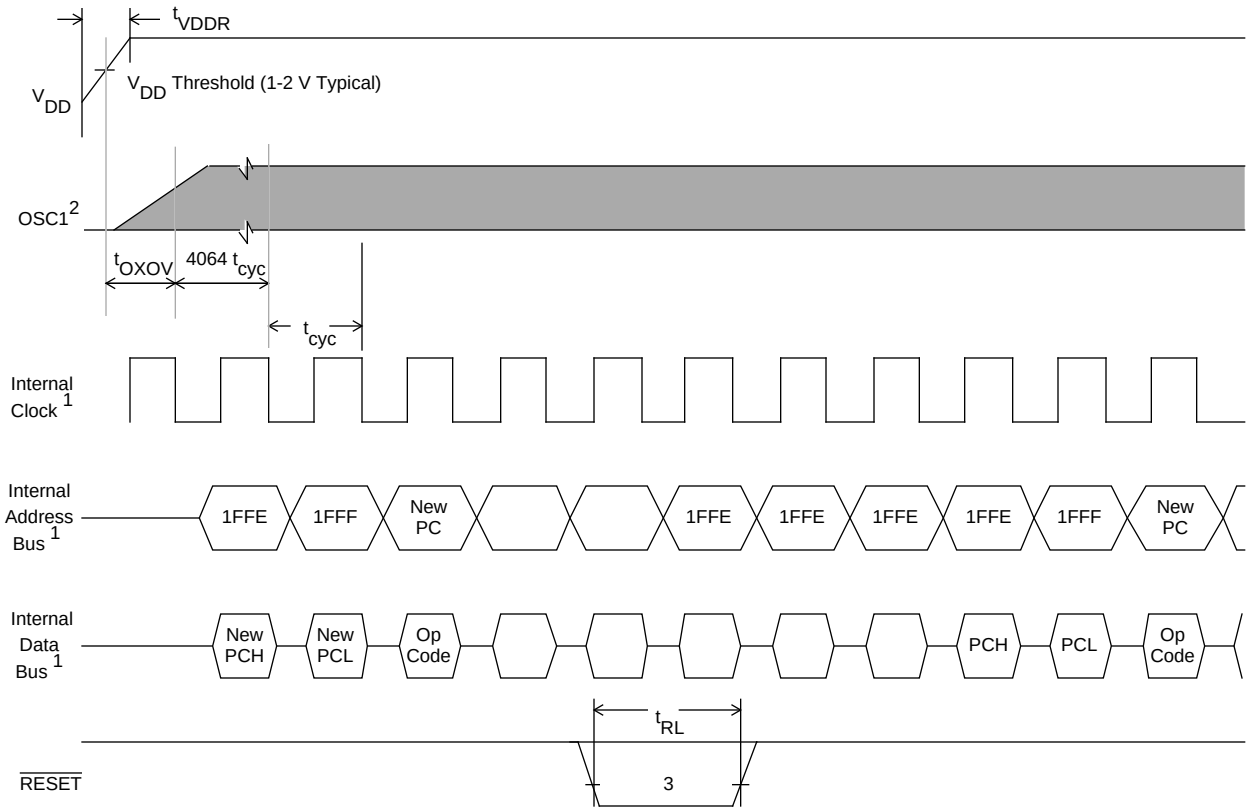
There is a 4064 internal processor clock cycle (t_{cyc}) oscillator stabilization delay after the oscillator becomes active. If the $\overline{\text{RESET}}$ pin is low at the end of this 4064-cycle delay, the MCU will remain in the reset condition until $\overline{\text{RESET}}$ goes high.

5.2 $\overline{\text{RESET}}$ 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}).

5.3 Computer Operating Properly (COP) Reset

The MCU contains a watchdog timer that automatically times out if not reset



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 RESET initiates the reset sequence.

Figure 5-1. Power-On Reset and RESET

SECTION 6
LOW POWER MODES

The MC68HC05P4A is capable of running in a low-power mode in each of its configurations. The WAIT and STOP instructions provide two modes that reduce the power required for the MCU by stopping various internal clocks and/or the on-chip oscillator. The STOP and WAIT instructions are not normally used if the COP watchdog timer is enabled. The stop conversion mask option is used to modify the behavior of the STOP instruction from stop mode to halt mode. The flow of the stop, halt, and wait modes is shown in Figure 6-1.

6.1 STOP Instruction

The STOP instruction can result in one of two modes of operation, depending on the stop conversion mask option. If the stop conversion is not chosen, the STOP instruction will behave like a normal STOP instruction in the MC68HC05 Family and place the MCU in the stop mode. If the stop conversion is chosen, the STOP instruction will behave like a WAIT instruction (with the exception of a variable delay at startup) and place the MCU in the halt mode.

6.1.1 Stop Mode

Execution of the STOP instruction without conversion to halt places the MCU in its lowest power consumption mode. In the stop mode the internal oscillator is turned off, halting *all* internal processing, including the COP watchdog timer. Execution of the STOP instruction automatically clears the I bit in the condition code register

6.1.2 Halt Mode

Execution of the STOP instruction with the conversion to halt places the MCU in this low-power mode. Halt mode consumes the same amount of power as wait mode. (Both halt and wait modes consume more power than stop mode.)

In halt mode the PH2 clock is halted, suspending all processor and internal bus activity. Internal timer clocks remain active, permitting interrupts to be generated from the 16-bit timer or a reset to be generated from the COP watchdog timer. Execution of the STOP instruction automatically clears the I bit in the condition code register enabling the $\overline{\text{IRQ}}$ external interrupt. All other registers, memory, and input/output lines remain in their previous states.

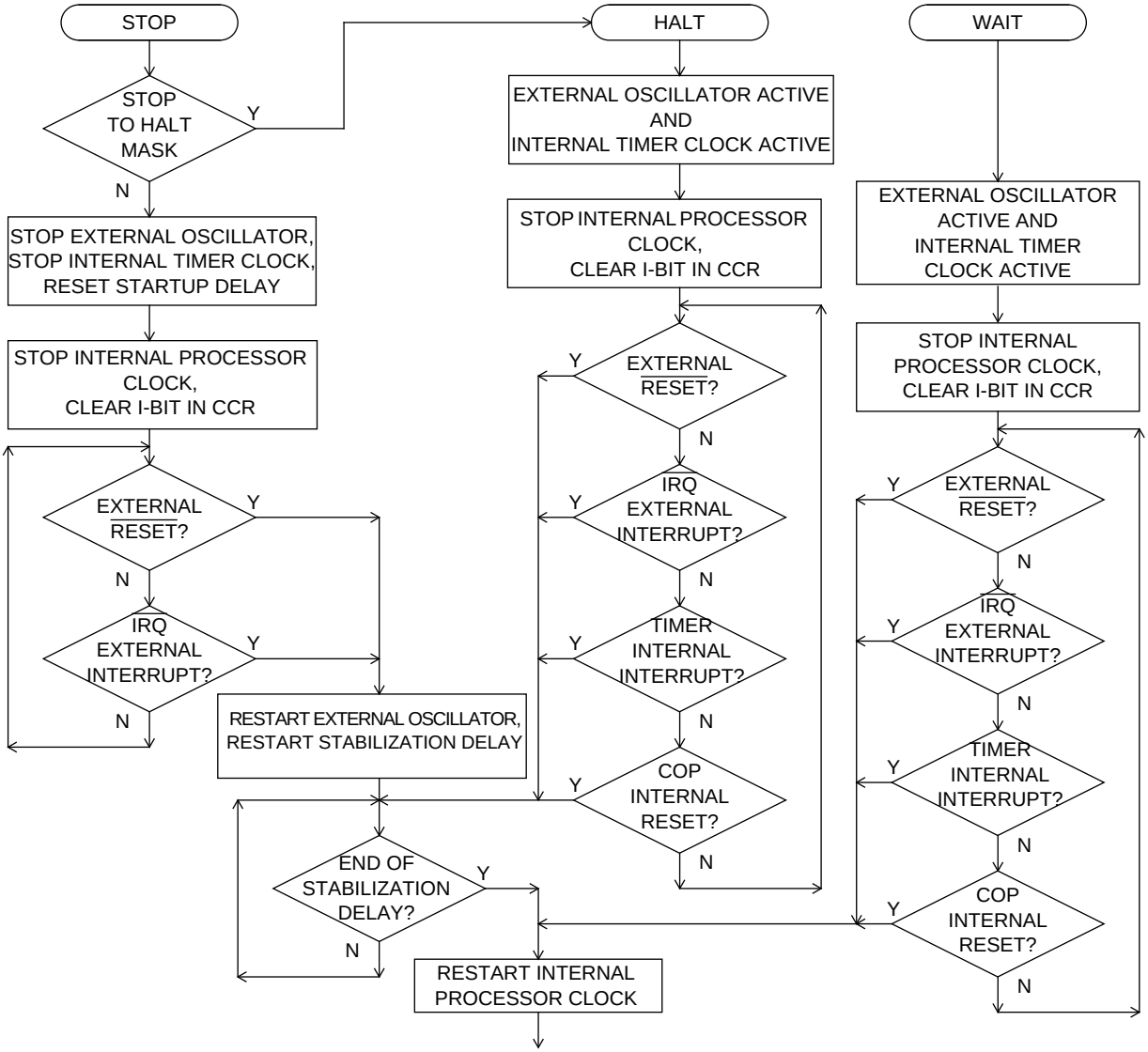
If the 16-bit timer interrupt is enabled, the processor will exit the halt mode and resume normal operation. The halt mode can also be exited when an $\overline{\text{IRQ}}$ external interrupt or external $\overline{\text{RESET}}$ occurs. When exiting the halt mode, the PH2 clock will resume after a delay of one to 4064 PH2 clock cycles. This varied delay time is the result of the halt mode exit circuitry testing the oscillator stabilization delay timer (a feature of the stop mode), which has been free-running (a feature of the wait mode).

NOTE

The halt mode is not intended for normal use. This feature is provided to keep the COP watchdog timer active in the event a STOP instruction is inadvertently executed.

6.2 WAIT Instruction

The WAIT instruction places the MCU in a low-power mode, which consumes more power than the stop mode. In wait mode, the PH2 clock is halted,



SECTION 7
SIMPLE SERIAL INPUT/OUTPUT PORT

This device includes a simple synchronous serial I/O port. 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.

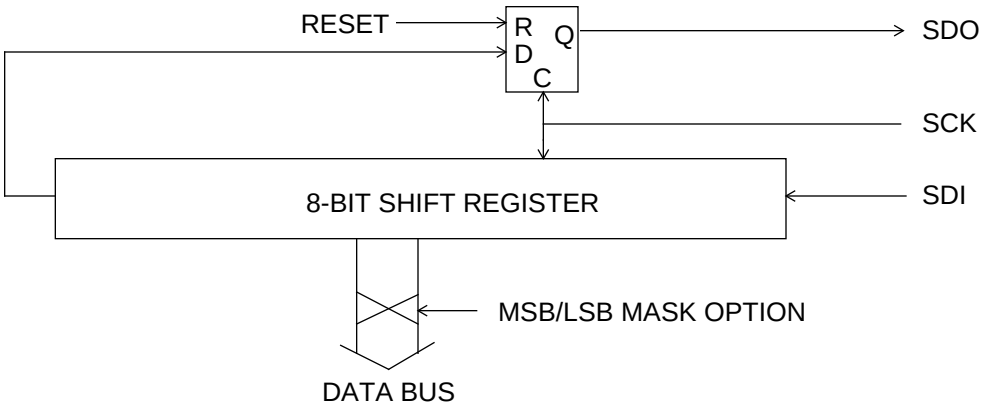


Figure 7-1. SIOP Block Diagram

7.1 Signal Format

The following paragraphs describe the SIOP signal format.

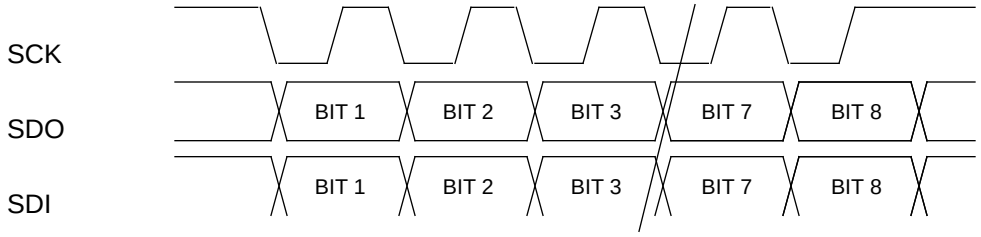
In master mode, the format is identical except that the SCK pin is an output and the shift clock now originates internally. The master mode transmission frequency is fixed at E/4.

7.1.2 Serial Data Out (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 always will reflect the value of the first bit received on the previous transmission if there was one. Prior to enabling the SIOP, PB5 can be initialized to determine the beginning state if necessary. 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. On the first falling edge of SCK, the first data bit to be shifted out is presented to the output pin.

7.1.3 Serial Data In (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 100 ns before the rising edge of the clock and remain valid for 100 ns after the edge.



7.2 SIOP Registers

The following paragraphs describe the SIOP registers.

7.2.1 SIOP Control Register (SCR)

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

$0A	0	SPE	0	MSTR	0	0	0	0
RESET:	0	0	0	0	0	0	0	0

Figure 7-3. SIOP Control Register

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 altered subsequently 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.

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 four. While the device is in master mode, the SDO and SDI

7.2.2 SIOP Status Register (SSR)

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

$0B	SPIF	DCOL	0	0	0	0	0	0
RESET:	0	0	0	0	0	0	0	0

Figure 7-4. SIOP Status Register

SPIF — Serial Peripheral Interface Flag

This bit is set upon occurrence of the last rising clock edge and 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 it is cleared before the last edge of the next byte, it will be set again. Reset clears this bit.

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 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. If the DCOL bit is set and the SPIF is not set, clearing the DCOL requires turning the SIOP off then turning it back on. 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).

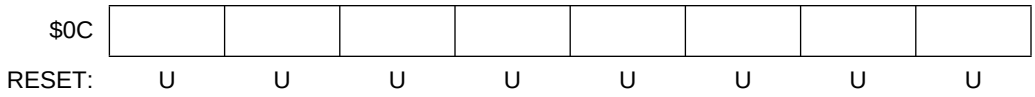


Figure 7-5. SIOP Data Register

SECTION 8
TIMER

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 8-1 for a timer block diagram.

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 also is 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.

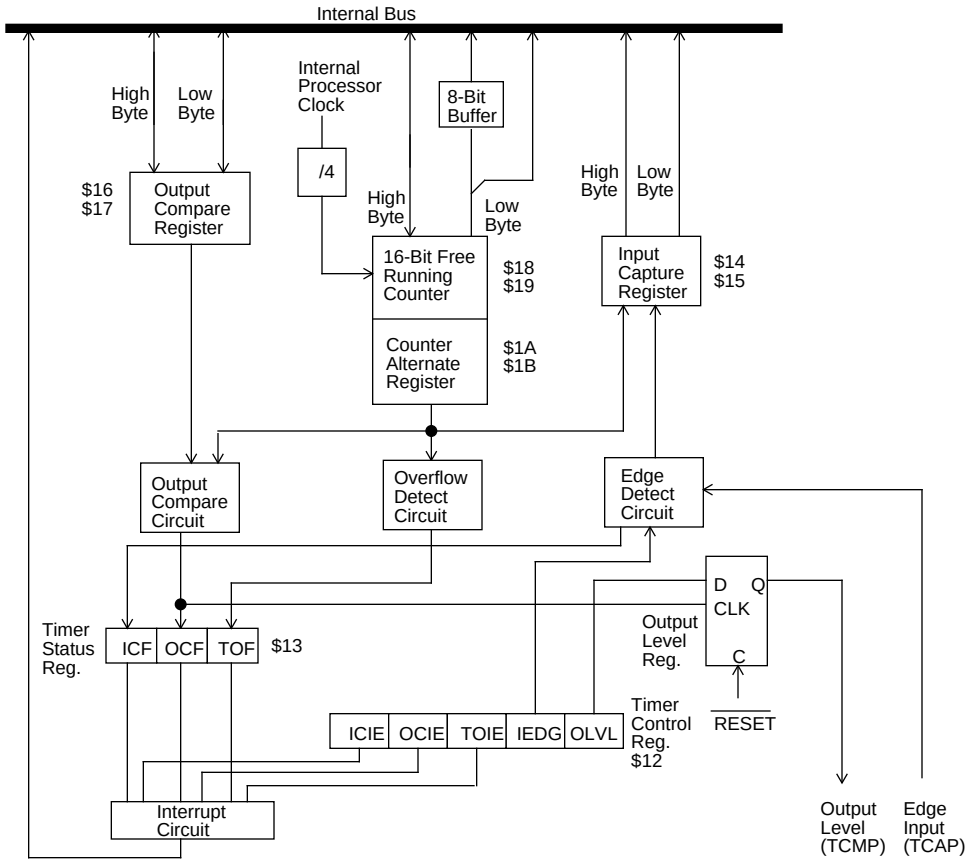


Figure 8-1. Timer Block Diagram

8.1 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 also must be read to complete the sequence.

The counter alternate register differs from the counter register in one respect: Aread 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).

8.2 Output Compare Register

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

The output compare register contents are 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.

8.3 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.

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.

8.4 Timer Control Register (TCR) $12

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

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

Figure 8-2. 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

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 2, 3, and 4 — Not used

Always read zero

8.5 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 8-3. 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.

8.6 Timer During Wait or Halt Mode

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

8.7 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 $\overline{\text{RESET}}$ 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 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 $\overline{\text{RESET}}$ is used to exit stop mode, then no input capture flag or data remains, even if a valid input capture edge occurred.

SECTION 9
COMPUTER OPERATING PROPERLY (COP)

This device includes a watchdog 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 power-on reset (POR) or external reset.

9.1 Resetting The COP

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

9.2 COP During Wait or Halt Mode

The COP will continue to operate normally during wait or halt mode. The software should pull the device out of wait or halt mode periodically and reset the COP by writing a logic zero to the COPR bit to prevent a COP reset.

9.3 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 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.

NOTE

The halt mode is not intended for normal use. This feature is provided to keep the COP watchdog timer active in the event a STOP instruction is inadvertently executed.

SECTION 10
SELF-CHECK MODE

The self-check program resides at mask ROM locations $1F00 to $1FEF. This program is designed to check the part's functionality with a minimum of support hardware. The COP subsystem is disabled in the self-check mode so that routines that feed the COP do not exist in the self-check program.

The self-check mode is entered on the rising edge of $\overline{\text{RESET}}$ if the $\overline{\text{IRQ}}$ pin is driven to double the supply voltage and the TCAP/PD7 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 I/O, RAM, ROM, timer, SIOP and Interrupts are tested. Self-check results (using LED's as monitors) are shown in Table 10-1**.** It is not recommended that the user code use any of the self-check code. The self-check code is subject to change at any time to improve testability or manufacturability.

Table 10-1. Self-Check Results

PC2	PC1	PC0	REMARKS
0	0	1	Bad I/O
0	1	0	Bad RAM
0	1	1	Bad Timer
1	0	0	Bad ROM
1	0	1	Bad Serial
1	1	0	Bad Interrupt
Flashing			Good Device
All Others			Bad Device

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

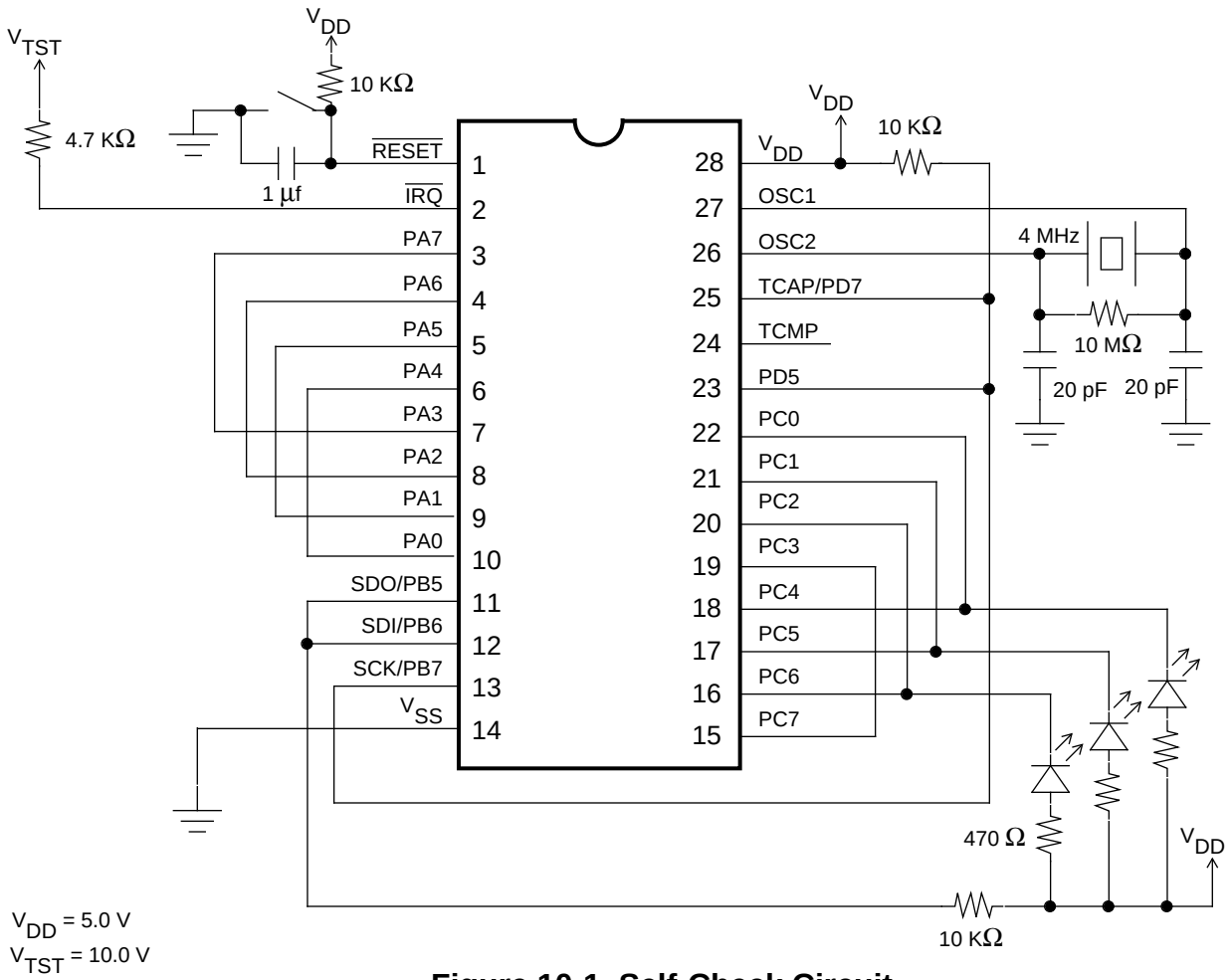


Figure 10-1. Self-Check Circuit

SECTION 11
INSTRUCTION SET

This section describes the M68HC05P4A addressing modes and instruction types.

11.1 Addressing Modes

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

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

11.1.1 Inherent

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

11.1.2 Immediate

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

11.1.3 Direct

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

11.1.4 Extended

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

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

11.1.5 Indexed, No Offset

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

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

11.1.6 Indexed, 8-Bit Offset

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

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

11.1.7 Indexed, 16-Bit Offset

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

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

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

11.1.8 Relative

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

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

11.2 Instruction Types

The MCU instructions fall into the following five categories:

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

11.2.1 Register/Memory Instructions

Most of these instructions use two operands. One operand is in either the accumulator or the index register. The CPU finds the other operand in memory. Table 11-1 lists the register/memory instructions.

Table 11-1. Register/Memory Instructions

Instruction	Mnemonic
Add Memory Byte and Carry Bit to Accumulator	ADC
Add Memory Byte to Accumulator	ADD
AND Memory Byte with Accumulator	AND
Bit Test Accumulator	BIT
Compare Accumulator	CMP
Compare Index Register with Memory Byte	CPX
EXCLUSIVE OR Accumulator with Memory Byte	EOR
Load Accumulator with Memory Byte	LDA
Load Index Register with Memory Byte	LDX
Multiply	MUL
OR Accumulator with Memory Byte	ORA
Subtract Memory Byte and Carry Bit from Accumulator	SBC
Store Accumulator in Memory	STA
Store Index Register in Memory	STX
Subtract Memory Byte from Accumulator	SUB

11.2.2 Read-Modify-Write Instructions

These instructions read a memory location or a register, modify its contents, and write the modified value back to the memory location or to the register. The test for negative or zero instruction (TST) is an exception to the read-modify-write sequence because it does not write a replacement value. Table 11-2 lists the read-modify-write instructions.

Table 11-2. Read-Modify-Write Instructions

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

11.2.3 Jump/Branch Instructions

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

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

the third byte to the program counter if the specified bit tests true. The bit to be tested and its condition (set or clear) is part of the opcode. The span of branching is from –128 to +127 from the address of the next location after the branch instruction. The CPU also transfers the tested bit to the carry/borrow bit of the condition code register. Table 11-3 lists the jump and branch instructions.

Table 11-3. Jump and Branch Instructions

Instruction	Mnemonic
Branch if Carry Bit Clear	BCC
Branch if Carry Bit Set	BCS
Branch if Equal	BEQ
Branch if Half-Carry Bit Clear	BHCC
Branch if Half-Carry Bit Set	BHCS
Branch if Higher	BHI
Branch if Higher or Same	BHS
Branch if $\overline{\text{IRQ}}$ Pin High	BIH
Branch if $\overline{\text{IRQ}}$ Pin Low	BIL
Branch if Lower	BLO
Branch if Lower or Same	BLS
Branch if Interrupt Mask Clear	BMC
Branch if Minus	BMI
Branch if Interrupt Mask Set	BMS
Branch if Not Equal	BNE
Branch if Plus	BPL
Branch Always	BRA
Branch if Bit Clear	BRCLR
Branch Never	BRN
Branch if Bit Set	BRSET
Branch to Subroutine	BSR
Unconditional Jump	JMP
Jump to Subroutine	JSR

11.2.4 Bit Manipulation Instructions

The CPU can set or clear any writable bit in the first 256 bytes of memory. Port registers, port data direction registers, timer registers, and on-chip RAM locations are in the first 256 bytes of memory. The CPU can also test and branch based on the state of any bit in any of the first 256 memory locations. Bit manipulation instructions use direct addressing. Table 11-4 lists these instructions.

Table 11-4. Bit Manipulation Instructions

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

11.2.5 Control Instructions

These register reference instructions control CPU operation during program execution. Control instructions, listed in Table 11-5, use inherent addressing.

Table 11-5. Control Instructions

Instruction	Mnemonic
Clear Carry Bit	CLC
Clear Interrupt Mask	CLI
No Operation	NOP
Reset Stack Pointer	RSP
Return from Interrupt	RTI
Return from Subroutine	RTS
Set Carry Bit	SEC
Set Interrupt Mask	SEI
Stop Oscillator and Enable $\overline{\text{IRQ}}$ Pin	STOP
Software Interrupt	SWI
Transfer Accumulator to Index Register	TAX
Transfer Index Register to Accumulator	TXA
Stop CPU Clock and Enable Interrupts	WAIT

11.3 Instruction Set Summary

Table 11-6 is an alphabetical list of all M68HC05 instructions and shows the effect of each instruction on the condition code register.

Table 11-6. Instruction Set Summary

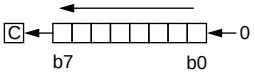
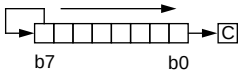
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
ADC #*opr* ADC *opr* ADC *opr* ADC *opr*,X ADC *opr*,X ADC ,X	Add with Carry	A ← (A) + (M) + (C)	↕	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A9 B9 C9 D9 E9 F9	ii dd hh ll ee ff ff	2 3 4 5 4 3
ADD #*opr* ADD *opr*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

Table 11-6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
BHCC *rel*	Branch if Half-Carry Bit Clear	PC ← (PC) + 2 + *rel* ? H = 0	—	—	—	—	—	REL	28	rr	3
BHCS *rel*	Branch if Half-Carry Bit Set	PC ← (PC) + 2 + *rel* ? H = 1	—	—	—	—	—	REL	29	rr	3
BHI *rel*	Branch if Higher	PC ← (PC) + 2 + *rel* ? C ∨ Z = 0	—	—	—	—	—	REL	22	rr	3
BHS *rel*	Branch if Higher or Same	PC ← (PC) + 2 + *rel* ? C = 0	—	—	—	—	—	REL	24	rr	3
BIH *rel*	Branch if $\overline{\text{IRQ}}$ Pin High	PC ← (PC) + 2 + *rel* ? $\overline{\text{IRQ}}$ = 1	—	—	—	—	—	REL	2F	rr	3
BIL *rel*	Branch if $\overline{\text{IRQ}}$ Pin Low	PC ← (PC) + 2 + *rel* ? $\overline{\text{IRQ}}$ = 0	—	—	—	—	—	REL	2E	rr	3
BIT #*opr* BIT *opr* BIT *opr* BIT *opr*,X BIT *opr*,X BIT ,X	Bit Test Accumulator with Memory Byte	(A) ∧ (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	A5 B5 C5 D5 E5 F5	ii dd hh ll ee ff ff p	2 3 4 5 4 3
BLO *rel*	Branch if Lower (Same as BCS)	PC ← (PC) + 2 + *rel* ? C = 1	—	—	—	—	—	REL	25	rr	3
BLS *rel*	Branch if Lower or Same	PC ← (PC) + 2 + *rel* ? C ∨ Z = 1	—	—	—	—	—	REL	23	rr	3
BMC *rel*	Branch if Interrupt Mask Clear	PC ← (PC) + 2 + *rel* ? I = 0	—	—	—	—	—	REL	2C	rr	3
BMI *rel*	Branch if Minus	PC ← (PC) + 2 + *rel* ? N = 1	—	—	—	—	—	REL	2B	rr	3
BMS *rel*	Branch if Interrupt Mask Set	PC ← (PC) + 2 + *rel* ? I = 1	—	—	—	—	—	REL	2D	rr	3
BNE *rel*	Branch if Not Equal	PC ← (PC) + 2 + *rel* ? Z = 0	—	—	—	—	—	REL	26	rr	3
BPL *rel*	Branch if Plus	PC ← (PC) + 2 + *rel* ? N = 0	—	—	—	—	—	REL	2A	rr	3
BRA *rel*	Branch Always	PC ← (PC) + 2 + *rel* ? 1 = 1	—	—	—	—	—	REL	20	rr	3
BRCLR *n opr rel*	Branch if bit n clear	PC ← (PC) + 2 + *rel* ? Mn = 0	—	—	—	—	↕	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	01 03 05 07 09 0B 0D 0F	dd rr	5
BRSET *n opr rel*	Branch if Bit n Set	PC ← (PC) + 2 + *rel* ? Mn = 1	—	—	—	—	↕	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	00 02 04 06 08 0A 0C 0E	dd rr	5
BRN *rel*	Branch Never	PC ← (PC) + 2 + *rel* ? 1 = 0	—	—	—	—	—	REL	21	rr	3

Table 11-6. Instruction Set Summary (Continued)

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

Table 11-6. Instruction Set Summary (Continued)

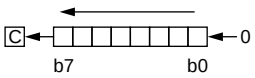
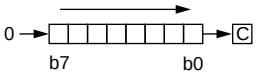
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
INC *opr* INCA INCX INC *opr*,X INC ,X	Increment Byte	M ← (M) + 1 A ← (A) + 1 X ← (X) + 1 M ← (M) + 1 M ← (M) + 1	—	—	↕	↕	—	DIR INH INH IX1 IX	3C 4C 5C 6C 7C	dd ff 	5 3 3 6 5
JMP *opr* JMP *opr* JMP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

Table 11-6. Instruction Set Summary (Continued)

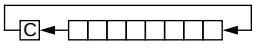
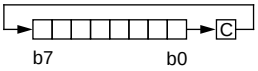
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
ORA #*opr* ORA *opr* ORA *opr* ORA *opr*,X ORA *opr*,X ORA ,X	Logical OR Accumulator with Memory	A ← (A) ∨ (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	AA BA CA DA EA FA	ii dd hh ll ee ff ff	2 3 4 5 4 3
ROL *opr* ROLA ROLX ROL *opr*,X ROL ,X	Rotate Byte Left through Carry Bit		—	—	↕	↕	×	DIR INH INH IX1 IX	39 49 59 69 79	dd ff	5 3 3 6 5
ROR *opr* RORA RORX ROR *opr*,X ROR ,X	Rotate Byte Right through Carry Bit		—	—	↕	↕	×	DIR INH INH IX1 IX	36 46 56 66 76	dd ff	5 3 3 6 5
RSP	Reset Stack Pointer	SP ← $00FF	—	—	—	—	—	INH	9C		2
RTI	Return from Interrupt	SP ← (SP) + 1; Pull (CCR) SP ← (SP) + 1; Pull (A) SP ← (SP) + 1; Pull (X) SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)	↕	↕	↕	↕	↕	INH	80		6
RTS	Return from Subroutine	SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)						INH			
SBC #*opr* SBC *opr* SBC *opr* SBC *opr*,X SBC *opr*,X SBC ,X	Subtract Memory Byte and Carry Bit from Accumulator	A ← (A) – (M) – (C)	—	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A2 B2 C2 D2 E2 F2	ii dd hh ll ee ff ff	2 3 4 5 4 3
SEC	Set Carry Bit	C ← 1	—	—	—	—	1	INH	99		2
SEI	Set Interrupt Mask	I ← 1	—	1	—	—	—	INH	9B		2
STA *opr* STA *opr* STA *opr*,X STA *opr*,X STA ,X	Store Accumulator in Memory	M ← (A)	—	—	↕	↕	—	DIR EXT IX2 IX1 IX	B7 C7 D7 E7 F7	dd hh ll ee ff ff	4 5 6 5 4
STOP	Stop Oscillator and Enable $\overline{\text{IRQ}}$ Pin		—	0	—	—	—	INH	8E		2
STX *opr* STX *opr* STX *opr*,X STX *opr*,X STX ,X	Store Index Register In Memory	M ← (X)	—	—	↕	↕	—	DIR EXT IX2 IX1 IX	BF CF DF EF FF	dd hh ll ee ff ff	4 5 6 5 4

Table 11-6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
SUB #*opr* SUB *opr* SUB *opr* SUB *opr*,X SUB *opr*,X SUB ,X	Subtract Memory Byte from Accumulator	A ← (A) – (M)	—	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A0 B0 C0 D0 E0 F0	ii dd hh ll ee ff ff	2 3 4 5 4 3
SWI	Software Interrupt	PC ← (PC) + 1; Push (PCL) SP ← (SP) – 1; Push (PCH) SP ← (SP) – 1; Push (X) SP ← (SP) – 1; Push (A) SP ← (SP) – 1; Push (CCR) SP ← (SP) – 1; I ← 1 PCH ← Interrupt Vector High Byte PCL ← Interrupt Vector Low Byte	—	1	—	—	—	INH	83		10
TAX	Transfer Accumulator to Index Register	X ← (A)	—	—	—	—	—	INH	97		2
TST *opr* TSTA TSTX TST *opr*,X TST ,X	Test Memory Byte for Negative or Zero	(M) – $00	—	—	—	—	—	DIR INH INH IX1 IX	3D 4D 5D 6D 7D	dd ff	4 3 3 5 4
TXA	Transfer Index Register to Accumulator	A ← (X)	—	—	—	—	—	INH	9F		2
WAIT	Stop CPU Clock and Enable Interrupts		—	↕	—	—	—	INH	8F		2

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

Table 11-7. Opcode Map

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

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

LSB of Opcode in Hexadecimal

MSB / LSB	0
0	5 BRSET0 3 DIR

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

SECTION 12
ELECTRICAL SPECIFICATIONS

12.1 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 Modes ($\overline{IRQ}$ 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
Operating Temperature Range 68HC05P4AP (Standard) 68HC05P4ACP (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 (e.g., either V_{SS} or V_{DD}).

12.2 Thermal Characteristics

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic DIP Plastic SOIC	θ_{JA}	 60 71	 °C/W °C/W

12.3 DC Electrical Characteristics

Table 12-1. DC Electrical Characteristics (V_{DD} = 5 V)

(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 V
Output High Voltage (I_{LOAD} = -0.8 mA) PA0-PA7, PB5-PB7, PC2-PC7, PD5 (I_{LOAD} = -5.0 mA) PC0-PC1	 V_{OH} V_{OH}	 V_{DD}-0.8 V_{DD}-0.8	 — —	 — —	 V V
Output Low Voltage (I_{LOAD} = 1.6 mA) PA0-PA7, PB5-PB7, PC2-PC7, PD5 (I_{LOAD} = 15 mA) PC0-PC1	 V_{OL} V_{OL}	 — —	 — —	 0.4 0.4	 V V
Input High Voltage PA0-PA7, PB5-PB7, PC0-PC7, PD5, TCAP/PD7, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	0.7 × V_{DD}	—	V_{DD}	V
Input Low Voltage PA0-PA7, PB5-PB7, PC0-PC7, PD5, TCAP/PD7, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	0.2 × V_{DD}	V
Supply Current Run Wait/Halt Stop 25°C 0°C to +70°C (Standard)	 I_{DD} I_{DD} I_{DD} I_{DD}	 — — — —	 TBD TBD TBD —	 TBD TBD TBD TBD	 mA mA µA µA
I/O Ports Hi-Z Leakage Current PA0-PA7, PB5-PB7, PC0-PC7, PD5	I_{OZ}	—	—	±10	µA
Input Current $\overline{RESET}$, $\overline{IRQ}$, OSC1, TCAP/PD7	I_{IN}	—	—	±1	µA
Capacitance Ports (as Input or Output) $\overline{RESET}$, $\overline{IRQ}$	 C_{OUT} C_{IN}	 — —	 — —	 12 8	 pF pF

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

Table 12-2. DC Electrical Characteristics (V_{DD} = 3.3 V)

(V_{DD} = 3.3 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	V_{OL}	—	—	0.1	V
I_{LOAD} = -10.0 µA	V_{OH}	V_{DD}-0.1	—	—	V
Output High Voltage					
(I_{LOAD} = -0.2 mA) PA0-PA7, PB5-PB7, PC2-PC7, PD5, TCMP	V_{OH}	V_{DD}-0.3	—	—	V
(I_{LOAD} = 6.0 mA) PC0-PC1	V_{OH}	V_{DD}-0.3	—	—	V
Output Low Voltage					
(I_{LOAD} = 0.4 mA) PA0-PA7, PB5-PB7, PC2-PC7, PD5, TCMP	V_{OL}	—	—	0.3	V
(I_{LOAD} = 6.0 mA) PC0-PC1	V_{OL}	—	—	0.3	V
Input High Voltage					
PA0-PA7, PB5-PB7, PC0-PC7, PD5, TCAP/PD7, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	0.7 × V_{DD}	—	V_{DD}	V
Input Low Voltage					
PA0-PA7, PB5-PB7, PC0-PC7, PD5, TCAP/PD7, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	0.2 × V_{DD}	V
Supply Current					
Run	I_{DD}	—	TBD	TBD	mA
Wait/Halt	I_{DD}	—	TBD	TBD	mA
Stop					
25°C	I_{DD}	—	TBD	TBD	µA
0°C to +70°C (Standard)	I_{DD}	—	—	TBD	µA
I/O Ports Hi-Z Leakage Current					
PA0-PA7, PB5-PB7, PC0-PC7, PD5	I_{OZ}	—	—	±10	µA
Input Current					
$\overline{RESET}$, $\overline{IRQ}$, OSC1, TCAP/PD7	I_{IN}	—	—	±1	µA
Capacitance					
Ports (as Input or Output)	C_{OUT}	—	—	12	pF
$\overline{RESET}$, $\overline{IRQ}$	C_{IN}	—	—	8	pF

NOTES:

1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C.
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}= 2.0 MHz), all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2.
5. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 V, V_{IH} = V_{DD} -0.2 V.
6. Wait I_{DD} is affected linearly by the OSC2 capacitance.

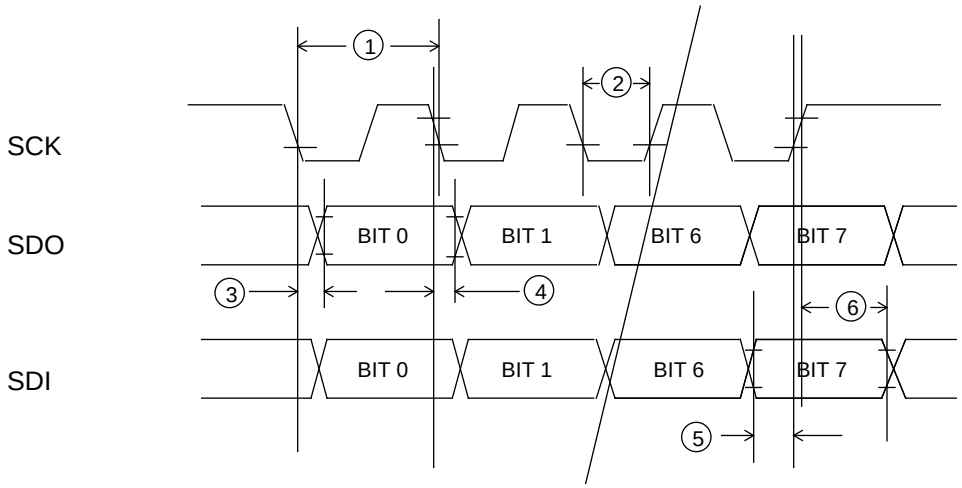


Figure 12-1. SIOP Timing Diagram

12.4 SIOP Timing

Table 12-3. SIOP Timing (V_{DD} = 5 V)

(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	 $f_{op(m)}$ $f_{op(s)}$	 0.25 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

NOTE: f_{op} = 2.1 MHz maximum

Table 12-4. SIOP Timing (V_{DD} = 3.3 V)

(V_{DD} = 3.3 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	 $f_{op(m)}$ $f_{op(s)}$	 0.25 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}	1980	—	ns
3	SDO Data Valid Time	t_v	—	400	ns
4	SDO Hold Time	t_{ho}	0	—	ns
5	SDI Setup Time	t_s	200	—	ns
6	SDI Hold Time	t_h	200	—	ns

NOTE: f_{op} = 1.0 MHz maximum

12.5 Control Timing

Table 12-5. Control Timing (V_{DD} = 5 V)

(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 Startup Time	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (Crystal Oscillator)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ Pulse Width	t_{RL}	1.5	—	t_{cyc}
Interrupt Puolse Width Low (Edge-Triggered)	t_{ILIH}	125	—	ns
Interrupt Pulse Period	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	90	—	ns

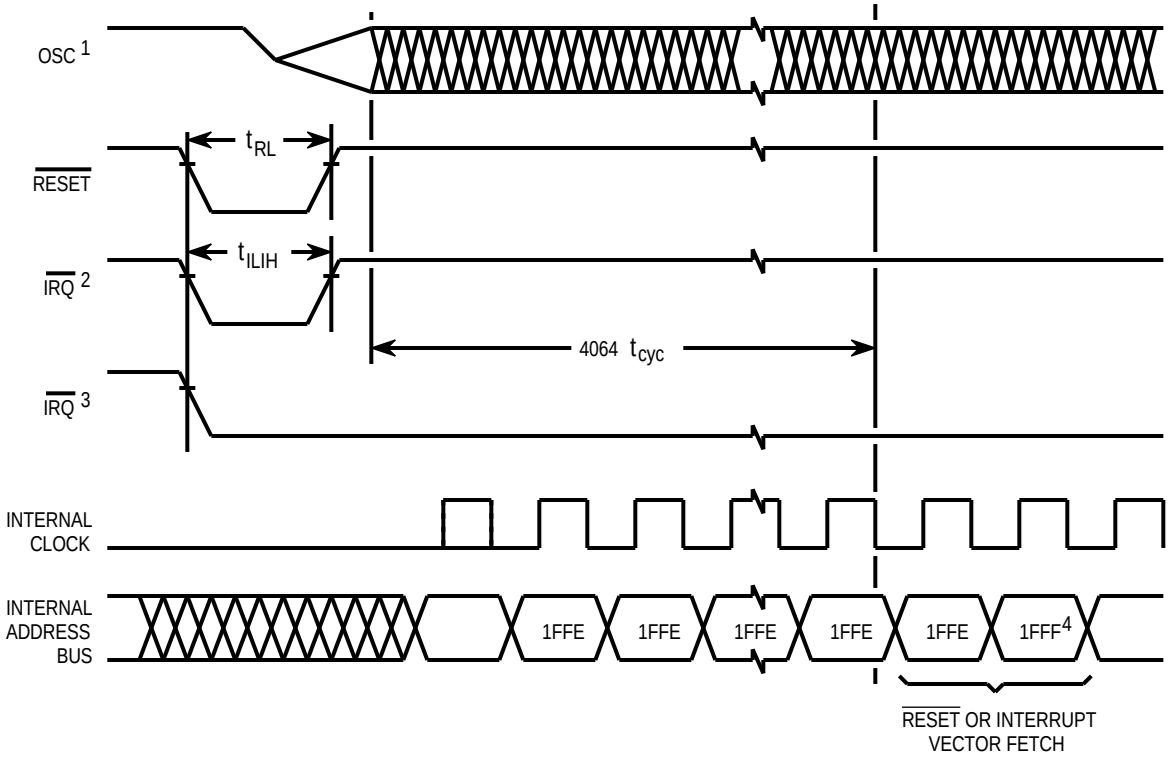
*The minimum period T_{ILIL} should not be less than the number of cycles it takes to execute the interrupt service routine plus 19 t_{cyc}.

Table 12-6. Control Timing (V_{DD} = 3.3 V)

(V_{DD} = 3.3 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	 2.0 2.0	 MHz MHz
Internal Operating Frequency Crystal (f_{osc} ÷ 2) External Clock (f_{osc} ÷ 2)	 f_{op} f_{op}	 — dc	 1.0 1.0	 MHz MHz
Cycle Time	t_{cyc}	1000	—	ns
Crystal Oscillator Startup Time	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (Crystal Oscillator)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ Pulse Width, Excluding Powerup	t_{RL}	1.5	—	t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered)	t_{ILIH}	250	—	ns
Interrupt Pulse Period	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	200	—	ns

*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}.



NOTES:
1. Represents the internal clocking 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 12-2. STOP Recovery Timing

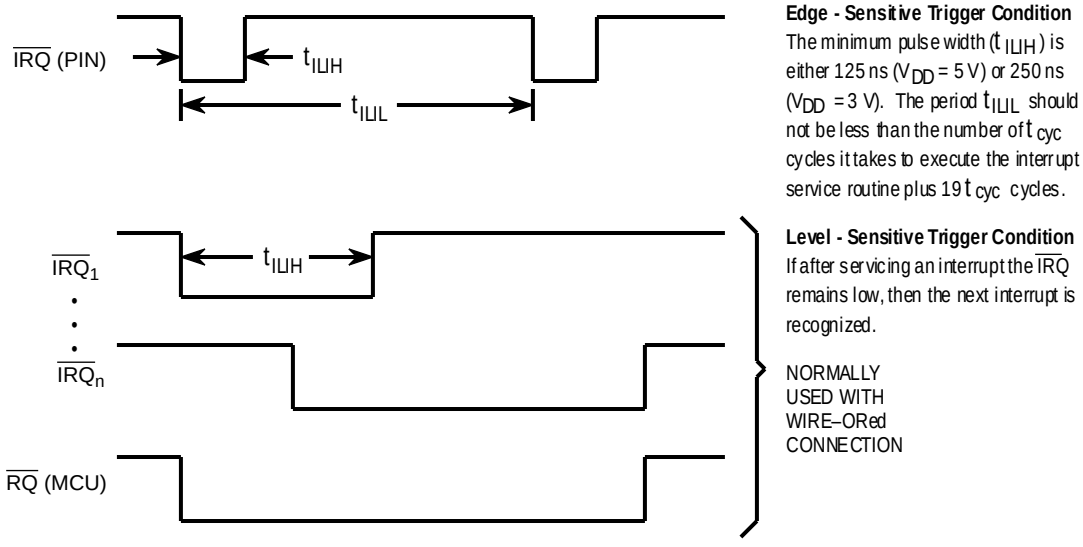
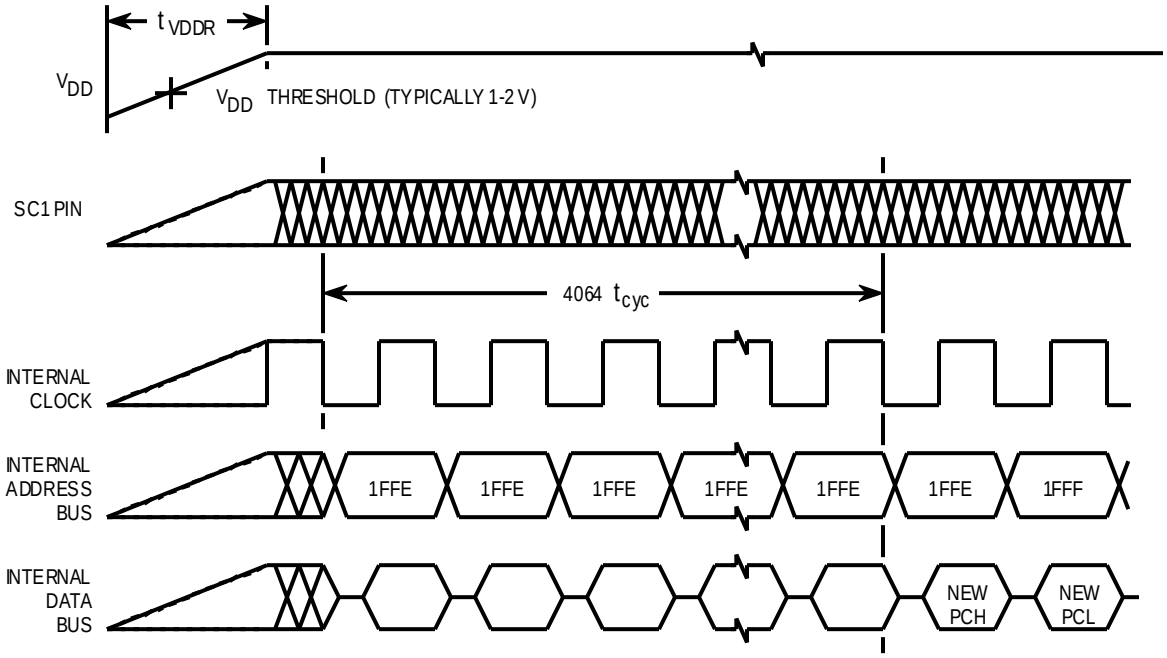
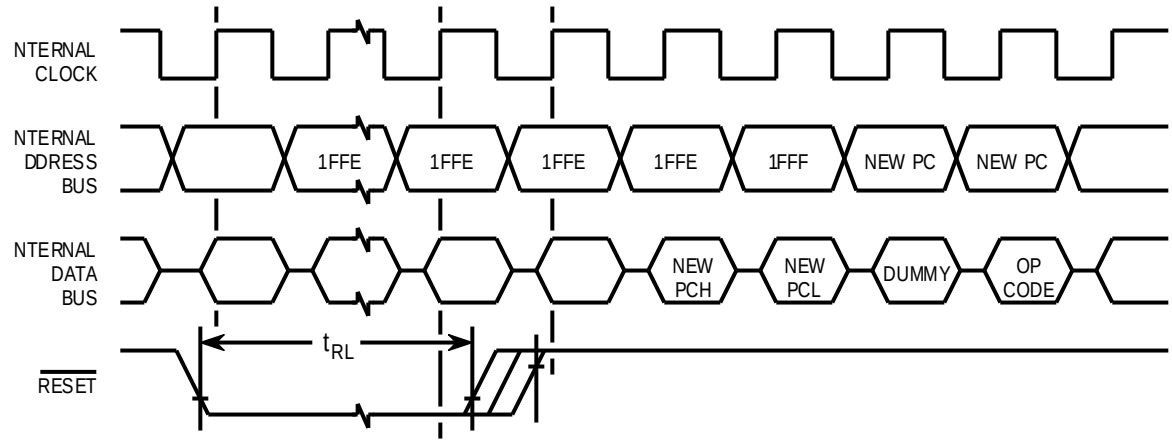


Figure 12-3. External Interrupt Timing



OTES:
1. Internal clock, internal address bus, and internal data bus signals are not available externally.
2. An internal POR reset is triggered as V_{DD} rises through a threshold (typically 1-2 V).

Figure 12-4. Power-On Reset Timing



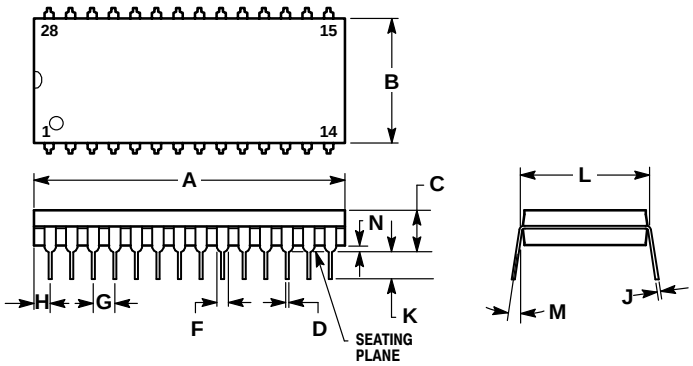
OTES:
1. Internal clock, internal address bus, and internal data bus signals are not available externally.
2. The next rising edge of the internal processor clock after the rising edge of $\overline{\text{RESET}}$ initiates the reset sequence.

Figure 12-5. External Reset Timing

SECTION 13
MECHANICAL SPECIFICATIONS

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

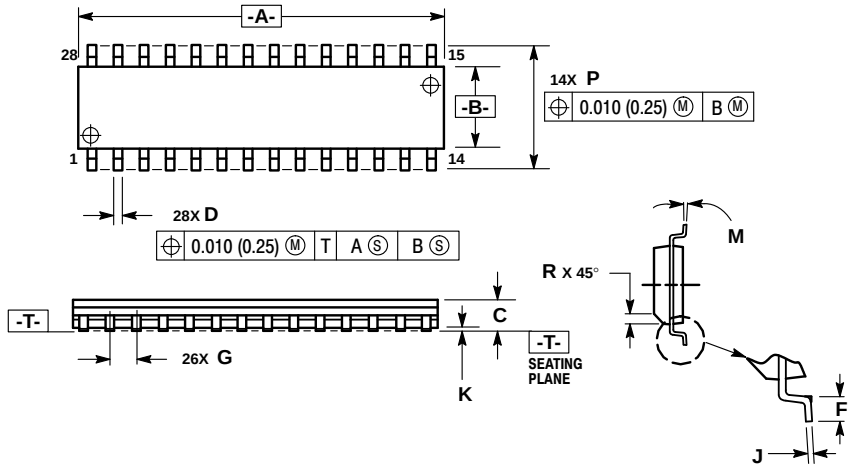
13.1 28-Pin Plastic Dual In-Line Package (Case 710-02)



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

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

13.2 28-Pin Small Outline Integrated Circuit Package (Case 751F-04)



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

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

SECTION 14
ORDERING INFORMATION

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

14.1 MCU Ordering Forms

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

- A current MCU ordering form that is **completely filled out** (Contact your Motorola sales office for assistance.)
- A copy of the customer specification if the customer specification deviates from the Motorola specification for the MCU
- Customer's application program on one of the media listed in **14.2 Application Program Media**

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

14.2 Application Program Media

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

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

Use positive logic for data and addresses.

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

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

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

NOTE

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

---

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

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

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

14.3 ROM Program Verification

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

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

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

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

14.4 ROM Verification Units (RVUs)

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

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.

Freescale Semiconductor, Inc.