MOTOROLA

SEMICONDUCTORS

EAST KILBRIDE, SCOTLAND

Advance Information

MC68HC04P3

8-BIT MICROCOMPUTER

JUNE, 1985

EADI 0007

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION 1
INTRODUCTION

1.1 GENERAL

The MC68HC04P3 HCMOS microcomputer unit (MCU) is a member of the 68HC04 family of very low-cost single-chip microcomputers. This 8-bit microcomputer contains a CPU, on-chip clock, ROM, RAM, I/O and timer. It is designed for the user who needs an economical microcomputer with the proven capabilities of an M6805-based instruction set.

1.2 FEATURES

The following are some of the hardware and software features of the MC68HC04P3 MCU.

HARDWARE FEATURES

- Low Power HCMOS Technology
- Power Saving STOP and WAIT Modes
- Pin Compatability with the MC6804P2
- Data RAM: 124 Bytes
- Program ROM: 1696 Bytes (Including 8 Bytes Reserved for Vectors)
- Data ROM: 72 Bytes
- Memory Mapped I/O
- 20 CMOS Compatible Bidirectional I/O Lines
- Versatile Indirect X and Y Registers
- On-chip Clock Generator
- Master Reset
- External and timer interrupts
- Single 2-6V Supply
- Self Check and ROM Verify Modes
- Software Programmable 8-bit Timer with 7-bit Prescaler
- Complete Development System Support

SOFTWARE FEATURES

- Similar to M6805 Family
- Byte Efficient Instruction Set
- Easy to Program
- True Bit Manipulation
- Bit Test and Branch Instruction
- Versatile Indirect Registers
- Conditional Branches

SOFTWARE FEATURES (Continued)

- Single Instruction Memory Examine/Change
- Timer Pin is Software Programmable as Clock Input or Timer Output
- 9 Powerful Addressing Modes
- STOP and WAIT Instructions

USER SELECTABLE OPTIONS

- Crystal/Ceramic Resonator, or Low-Cost Resistor Capacitor Oscillator
- Mask Selectable Edge- or Level-Sensitive Interrupt Pin
- Oscillator Divide Ratio (by 1, 2 or 4)
- Temperature Range (−40 to + 85°C or −55 to +125°C)
- Supply Voltage Range (2.7 to 5.5V or 2 to 5.5V Operation)
- Pull Down Devices on I/O Ports.

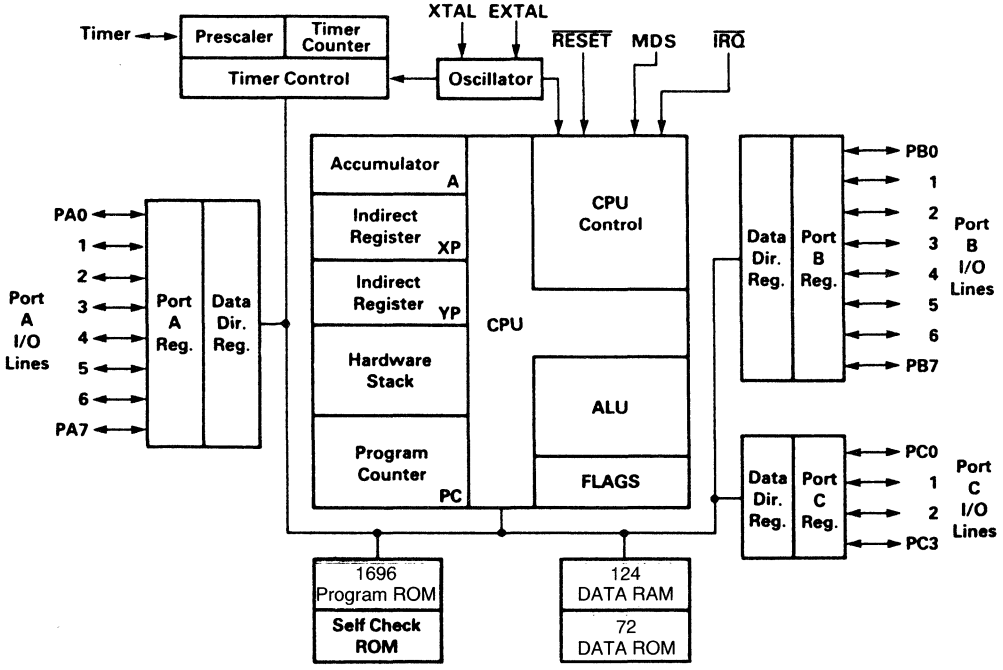


Figure 1-1. 68HC04P3 MCU Block Diagram

SECTION 2
FUNCTIONAL PIN DESCRIPTION, MEMORY, CPU AND REGISTERS

This section provides a description of the functional pins, memory spaces, central processing unit (CPU) and various registers and flags.

2.1 FUNCTIONAL PIN DESCRIPTION

2.1.1 V_{DD} and V_{SS}

Power is supplied to the MCU using these two pins. V_{DD} is power and V_{SS} is the ground connection.

2.1.2 IRQ

This pin provides the capability for asynchronously applying an external interrupt to the MCU. Refer to **4.1 INTERRUPT** for additional information.

2.1.3 XTAL and EXTAL

These pins provide connections to the on-chip oscillator circuit. A crystal, ceramic resonator resistor and capacitor, or an external signal, depending on the user selectable manufacturing mask option, can be connected to these pins to provide a system clock source with various stability/cost trade-offs. Lead lengths and stray capacitance on these two pins should be minimised. Refer to **4.6 INTERNAL CLOCK GENERATOR OPTIONS** for recommendations concerning these inputs.

2.1.4 TIMER

Two input modes are available, the timer pin being configured either as a timer enable or as the timer clock. In the output mode, the timer pin signals that a time out of the timer has occurred. Refer to **SECTION 3 TIMER** for additional information.

2.1.5 RESET

The RESET pin is used to restart the processor of the MC68HC04P3 at the beginning of the program. This pin, together with the MDS pin, is also used to select the operating mode of the MC68HC04P3. See **2.1.6**. Refer to **4.5 RESET** for additional information.

2.1.6 MDS

The MDS (mode select) pin is used to place the MCU into special operating modes. If MDS is held logic high at the exit of the reset state, the decoded state of PA6 and PA7 is latched to determine the operating mode (single-chip, self-check or ROM verify). However, if MDS is held at zero volts at the exit of the reset state, the single chip operating mode is automatically selected (regardless of PA6 and PA7 state).

2.1.7 Input/Output Lines (PA0-PA7, PB0-PB7, PC0-PC3)

These 20 lines are arranged into two 8-bit ports (A and B) and one 4-bit port (C). All lines are programmable as either inputs or outputs under software control of the data direction registers. Refer to **SECTION 5 INPUT/OUTPUT PORTS** for additional information.

2.2 MEMORY

The MCU operates in three different memory spaces: program space, data space and stack space. A representation of these memory spaces is shown in Figure 2.1. The program space (Figure 2-1a) contains all of the instructions that are to be executed, including the data required for the immediate adressing mode instructions and the self-check and user vectors. The data space (Figure 2-1b) contains all of the RAM locations, plus I/O locations and some ROM for storage of tables and constants. The stack space (Figure 2-1c) contains RAM which is used for stacking subroutine and interrupt return addresses. It is not directly available to the user.

The MCU is capable of addressing 4096 bytes of program space memory with its program counter and 256 bytes of data space memory with its instructions. The data space memory contains:

- 3 bytes for port data registers
- 3 bytes for port data direction registers
- 1 byte for timer status/control
- 72 bytes ROM
- 124 bytes RAM (including two bytes for XP and YP indirect registers)
- 2 bytes for timer prescaler and count registers
- 1 byte for the accumulator

The program space includes:

- 352 bytes of self-check ROM
- 1688 bytes program ROM
- 8 bytes of vectors for self-check and user programs.

2.2.1 Program ROM Protect

A maskable option is available to the user to enable Program ROM Protection. If enabled, this option prevents the ROM contents being output during self check/ROM verify.

2.3 CENTRAL PROCESSING UNIT

The CPU of the MC68HC04 family is implemented independently from the I/O or memory configuration. Consequently, it can be treated as a independent central processor communicating with I/O and memory via internal address, data, and control buses.

Figure. 2-1b

DATA SPACE

bytes decimal			
	DATA I/O REGISTER	A	$00
	DATA I/O REGISTER	B	$01
	DATA I/O REGISTER	C	$02
	DIRECTION REG	A	$04
	DIRECTION REG	B	$05
	DIRECTION REG	C	$06
	TIMER CONTROL REG.		$09
	CRC LOW BYTE		$0A
	CRC HIGH BYTE		$0B
8	Reserved ROM		$10 → $17
72	DATA SPACE ROM		$18 → $5F
32	future expansion all 1's		$60 → $7F
124	POINTER REG	X	$80 ← short-direct
	POINTER REG	Y	$81 ← short-direct
		V	$82 ← short-direct
		W	$83 ← short-direct
	DATA SPACE RAM		$84 → $FB
1	unimplemented all 1's		$FC
	TIMER PRESCALER		$FD
	TIMER COUNT REG		$FE
	ACCUMULATOR		$FF

Figure. 2-1a

PROGRAM SPACE

all 0's			$000
			$7FF
SELF-CHECK ROM			$800
			$960
PROGRAM ROM			
			$FF7
SELF-CHECK	IRQ	VECTOR	$FF8, $FF9
SELF-CHECK	RESTART	VECTOR	$FFA, $FFB
USER	IRQ	VECTOR	$FFC, $FFD
USER	RESTART	VECTOR	$FFE, $FFF

Figure. 2-1c

STACK SPACE

Level 1
Level 2
Level 3
Level 4

Figure 2-1. 68HC04P3 Memory Map

2.4 REGISTERS

The M6804 family CPU has four registers and two flags available to the programmer. They are shown in Figure 2-2 and explained in the following paragraphs.

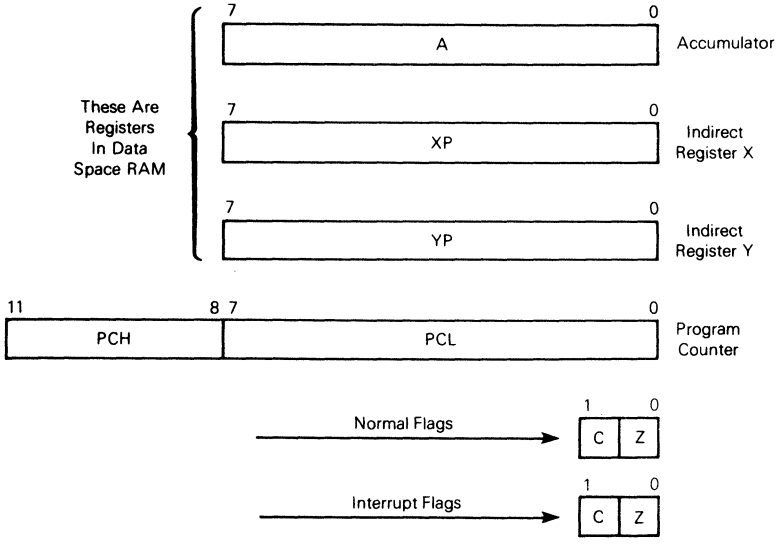


Figure 2-2. Programming Model

2.4.1 Accumulator (A)

The accumulator is an 8-bit general purpose register used in all arithmetic calculations, logical operations and data manipulations. The accumulator is implemented as the highest RAM location ($FF) in data space. Since the accumulator is implemented in RAM, it may be manipulated in the same way as any RAM memory location. This allows the implementation of "implied" instructions, some of which are recognised by the assembler e.g. ASLA. Refer to **6.3 IMPLIED INSTRUCTIONS** for additional information.

2.4.2 Indirect Registers (XP, YP)

These two indirect registers are used to maintain pointers to other memory locations in data space. They are used in the register-indirect addressing mode, and can be accessed with the direct, indirect, short direct or bit set/clear addressing modes. These registers are implemented as two of the 124 RAM locations ($80, $81) and as such may generate implied instructions. Refer to **6.3 IMPLIED INSTRUCTIONS** for additional information.

2.4.3 Program Counter (PC)

The program counter is a 12-bit register which contains the address of the next ROM word to be used (may be opcode, operand or address of operand). The 12-bit program counter is contained in PCL (low byte) and PCH (high nibble).

2.4.4 Flags (C,Z)

The carry (C) bit is set on a carry or a borrow out of the ALU. It is cleared if the result of an arithmetic operation does not result in a carry or a borrow. The (C) bit is also set to the value of the bit tested in a bit test instruction and participates in the rotate left (ROLA) instruction.

The zero (Z) bit is set if the result of the last arithmetic or logical operation was equal to zero, otherwise it is cleared.

There are two sets of these flags, one set for interrupt processing, the other for all other routines. When an interrupt occurs, a context switch is made from the program flags to the interrupt flags (interrupt mode). An RTI forces the context switch back to the program flags (program mode). While in either mode, only the flags for that mode are available. Further, the interrupt flags will not be cleared upon entering the interrupt mode. Instead, the flags will be in the same state as at the exit of the last interrupt mode. Both sets of flags are cleared by reset.

2.4.5 Stack

There is a true LIFO stack in the MC68HC04P3 which eliminates the need for a stack pointer. Stack space is implemented in separate RAM (12-bits wide) shown in Figure 2-1c. Whenever a subroutine call (or interrupt) occurs, the contents of the PC are shifted into the top register of the stack. At the same time (same cycle), the top register is shifted to the next level deeper. This happens to all registers with the bottom register falling out the bottom of the stack.

Whenever a subroutine or interrupt return occurs, the top register is shifted into the PC and all lower registers are shifted up one level higher. The stack RAM is four levels deep. If the stack is pulled more than four times without any pushes, the fifth pull will return the same PC address as the third pull. The sixth pull will return the same address as the fourth pull and so on. This is because stack level three is copied back into level four during a stack pull.

2.4.6 CRC Registers

Two eight bit registers are implemented in RAM primarily for use in self-check and ROM verify modes. These two registers are memory mapped in data space at addresses $0A (CRC low) and $0B (CRC high).

Provided no write or read/modify/write operation is performed to change the contents of these two locations, the registers are configured to perform CRC calculations. Hence by simply reading a register, a pseudo random number may be generated. If a write or read/modify/write is performed on addresses $0A or $0B then the CRC circuitry is disabled and both registers can be used as a RAM location until such time as a $\overline{\text{RESET}}$ enables the CRC circuitry again.

SECTION 3
TIMER

3.1 INTRODUCTION

A block diagram of the MC68HC04P3 timer circuitry is shown in Figure 3-1. the timer logic is comprised of an 8-bit counter (timer count register, TCR) with a 7-bit prescaler and a timer status control register (TSCR).

The timer count register, which may be loaded under program control, is decremented towards zero by a clock input (prescaler output). The prescaler is used to extend the maximum interval of the overall timer. The prescaler tap is selected by bits 0-2 (PS0-PS2) of the TSCR. Bits PS0-PS2 control the actual division of the prescaler within the range of divide-by-1 (2^0) to divide-by-128 (2^7). The timer count register (TCR) and prescaler are decremented on rising clock edges. The coding of the TSCR PS0-PS2 bits produce a division in the prescaler as shown in Table 3-1.

Table 3-1. Prescaler Coding Table

PS2	PS1	PS0	Divide By
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8

PS2	PS1	PS0	Divide By
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

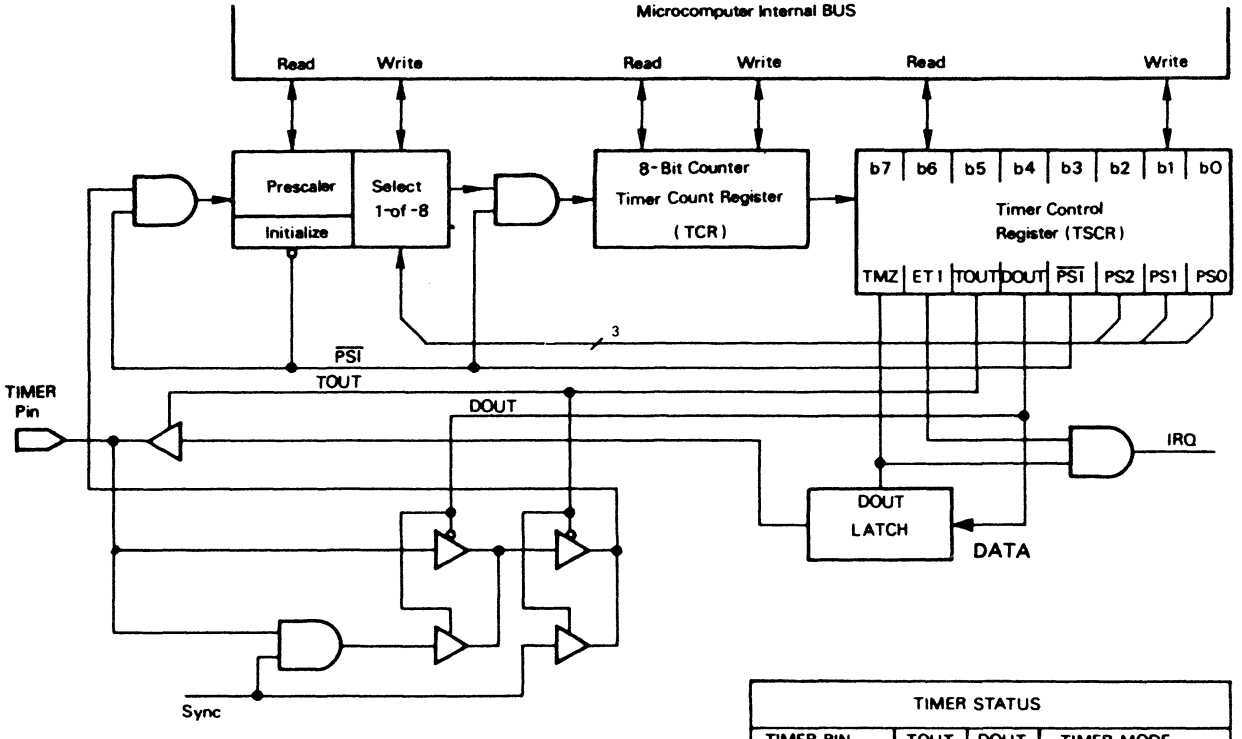
The TIMER pin may be programmed as either an input or an output depending on the status of TOUT (TSCR bit 5). Two distinct input modes exist, namely Input Gated and Input Event Counter. Refer to figure 3-1.

3.1.1 INPUT EVENT COUNTER MODE

In the Input Event Counter Mode, both TOUT and DOUT are at logic zero. TIM is effectively connected directly to the prescaler input. Therefore, the TIMER/PRESCALER is clocked by the signal applied from TIM. The PRESCALER divides its clock input by a value determined by the coding of TSCR bits PS0-PS2 as shown in Table 3-1. The divided prescaler output then clocks the 8-bit TCR. When TCR is decremented to zero it sets the TMZ bit in TSCR. The TMZ bit can be tested under program control to perform a timer function whenever it goes high. The frequency of the external clock applied to TIM must be less than $1/T_{byte}$ i.e. f_{osc}/(internal divide value $\times$ 12).

3.1.2 INPUT GATED MODE

In the Input Gated Mode. TOUT is logic zero and DOUT is logic one. The TIMER (TIM) pin is an input which enables the TIMER/PRESCALER to decrement every machine cycle as long as it is held high. Counting is inhibited when TIM is low. This mode allows the counting of time during which TIM is held high, based on the system clock and prescaler values.



TIMER STATUS			
TIMER PIN	TOUT	DOUT	TIMER MODE
INPUT	0	0	EVENT COUNTER
	0	1	INPUT GATED MODE
OUTPUT	1	0	OUTPUT
	1	1	

Figure 3-1. Timer Block Diagram

3.1.3 OUTPUT MODE

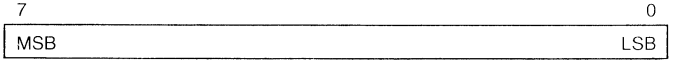
In the Output mode, TOUT is logic one and TIM is connected to the DOUT latch. Therefore, the timer prescaler is clocked by the internal sync pulse (divide-by-12, 24 or 48 of the internal oscillator depending on the mask option). Operation is similar to that described for the Input Event Counter mode. However, in the output mode, the low-to-high TMZ bit transition is used to latch the DOUT bit of TSCR and provide it as output for the TIM pin.

NOTE

TMZ is normally set to logic one when the timer times out (TCR decrements to $00); however, it may be set by a write of $00 to TCR or by a write to bit 7 of TSCR. During reset, TCR and PRESCALER are set to $FF while TSCR is cleared to $00 and the DOUT Latch (TIM pin is in the high-impedance input mode) is forced to a logic high. The PRESCALER and TCR are implemented in data space RAM locations ($FD, $FE); therefore they are both readable and writable. A write to either will predominate over the TCR decrement-to-$00 function, i.e., if a write and a TCR decrement-to-$00 occur simultaneously, the write will take precedence, and the TMZ bit is not set until the next TIMER time out.

3.2 TIMER REGISTERS

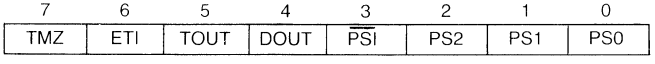
3.2.1 Timer Count Register (TCR)



TCR Address=$FE

The timer count register indicates the state of the internal 8-bit counter.

3.2.2 Timer Status/Control Register (TSCR)



TSCR Address=$09

b7, TMZ — Low-to-high transition indicates that TCR has decremented to zero since TSCR was last read. Cleared by a read of TSCR register if TMZ was read as logic one.

b6, ETI — Timer interrupt is enabled when high and disabled when low.

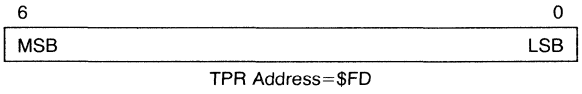
b5, TOUT — When low, this bit selects the input mode for the timer. When high, the output mode is selected.

b4, DOUT — In output mode, data is sent to the timer ouput pin when TMZ is set high. In input mode, event counter mode is selected if DOUT is low, gated mode if high. See Figure 3-1.

b3, $\overline{PSI}$ — Used to initialise the prescaler and inhibit its counting while $\overline{PSI} = 0$. The initialised value is set to $FF. TCR will also be inhibited (contents unchanged). When $\overline{PSI}=1$ the prescaler begins to count downward.

b0, b1, b2 PS0-PS2	These bits are used to select the prescaler divide-by ratio, therefore affecting the clock input frequency to TCR. Refer to Table 3-1.

3.2.3 Timer Prescaler Register



TPR Address=$FD

The timer prescaler register indicates the state of the internal 7-bit prescaler. This 7-bit prescaler divide ratio is normally determined by bits PS0-PS2 of TSCR. (See table 3-1).

SECTION 4
INTERRUPT, POWER SAVING MODES, SELF-CHECK, RESET, AND INTERNAL CLOCK GENERATOR

4.1 INTERRUPT

There are two ways in which the MC68HC04P3 can be interrupted. Firstly by applying a logic low signal to the $\overline{IRQ}$ Pin and secondly by a positive transition of the TMZ bit of TSCR with ETI bit set. Note that both types of interrupt share the same vector ($FFC). The only way to differentiate a timer interrupt from an external interrupt is to test the TMZ bit.

4.1.1 External Interrupt Edge-Sensitive Option

When the $\overline{IRQ}$ pin is pulled low, the internal interrupt request latch is set. Prior to each instruction fetch, the interrupt request latch is tested and, if its output is low, an interrupt sequence is initiated at the end of the current instruction (provided the interrupt mask is cleared). Figure 4-1a illustrates the instruction processing sequence.

The external and timer interrupt sequences are detailed in figure 4-1b. The interrupt sequence consists of one cycle during which:

— interrupt request latch is cleared
— interrupt mode flags are selected
— PC is saved on the stack
— interrupt mask is set
— $\overline{IRQ}$ vector is loaded indirectly into the PC.

The last action is achieved by the user loading the vector locations with the appropriate two-byte JMP instruction. For single-chip mode, the locations are $FFC/$FFD and for self-test mode $FF8/$FF9.

Internal processing of the interrupt continues until an RTI (return from interrupt) instruction is processed. During the RTI instruction, the interrupt mask is cleared and the program mode flags are selected. The next instruction of the program is then fetched and executed.

When STOP is processed, the interrupt mask is cleared and the oscillator stopped. Checks are made for either $\overline{RESET}$ or $\overline{IRQ}$. If $\overline{RESET}$ is detected, the $\overline{RESET}$ sequence is initiated. If $\overline{IRQ}$ is detected, the system oscillator is enabled along with the clock. In both cases, a delay is executed by the timer to allow oscillator stabilisation before the CPU is enabled and the interrupt serviced.

When WAIT is processed, the interrupt mask is cleared and the CPU clock disabled. The interrupt latch is tested. Detection of $\overline{RESET}$ initiates the $\overline{RESET}$ sequence. Detection of IRQ or TIRQ enables the CPU clock and initiates the servicing of the interrupts.

When RTI is processed, the program counter is pulled from the stack, the program flags are selected and the interrupt mask cleared. The interrupt request latch is then tested *before* the next instruction.

When the interrupt is initially detected and the interrupt sequence started, the interrupt request latch is cleared so that the next (second) interrupt may be detected even while the previous (first) one is being serviced. However, even though the second interrupt sets the interrupt request latch during processing of the first interrupt, the second interrupt sequence will not be initiated until completion of the interrupt service routine for the first interrupt. Completion of an interrupt service routine is always accomplished using an RTI instruction to return to the main program. The interrupt mask (which is not directly available to the programmer) is cleared during the last cycle of the RTI instruction.

4.1.2 External Interrrupt Level-Sensitive Option

The operation of the level-sensitive and edge-sensitive options are similar except that in the case of the former a check is made on the pin level. The differences are illustrated in the flowcharts of Figure 4-1.

4.1.3 Power Up and Timing

During the power-up sequence the interrupt mask is set to preclude any false or "ghost" interrupts from occurring. To clear the interrupt mask, the programmer should write a JSR (instead of a JMP) instruction to an initialisation routine as the first instruction in a program. The initialisation routine should end with an RTI (instead of RTS).

Maximum interrupt response time is eight machine (t_{byte}) cycles (see **4.6 INTERNAL CLOCK GENERATOR OPTIONS).** This includes five machine cycles for the longest instruction, plus one machine cycle for stacking PC and switching flags (see **2.4.4 Flags (C,Z))** and a maximum of two cycles to synchronise $\overline{IRQ}$ input with the internal machine cycle frequency.

4.1.4 Timer Interrupt

A timer interrupt is requested by a positive transition of the TMZ bit of the timer status/control register (TSCR). A TMZ positive transition can be caused either by the timer count register (TCR) reaching the all-zero state or by any program instruction that writes a one to the TMZ bit. Timer interrupt request is maskable by clearing ETI, bit 6 of TSCR.

ETI is cleared by RESET. Note that only one interrupt request can be stored in the interrupt request latch. Therefore if a subsequent interrupt is sensed before processing of an initial interrupt has commenced, then the initial interrupt will be lost. This has to be carefully considered when more than one interrupt source is available.

4.2 STOP MODE

The STOP instruction places the MC68HC04P3 in its lowest power consumption mode. In STOP mode, the internal oscillator is turned off causing all internal processing to be halted and the current consumption to drop to leakage levels. (See section **7 ELECTRICAL SPECIFICATION**).

The contents of the timer status/control register, the accumulator and all data space RAM are unchanged by STOP providing that the supply voltage, V_{DD}, remains within data sheet limits. The processor can only be brought out of STOP mode by pulling low IRQ or RESET input pins. The timer is used to provide a delay for the oscillator to stabilise during exit from STOP. Hence, the contents of timer count register (TCR) and the prescaler must be considered to be corrupted.

4.3 WAIT MODE

The WAIT instruction places MC68HC04P3 in a low power consumption mode. In WAIT mode, the clock is disabled from all internal circuitry except the timer circuit, halting all internal processing. The timer may continue to count down if PSI bit of TSCR is set. External interrupts are enabled. All other registers, memory and I/O lines remain in their last state. ETI bit of TSCR may be enabled by software prior to entering WAIT. This allows an exit from WAIT via a timer interrupt in addition to an external interrupt (IRQ) or RESET.

4.4 SELF CHECK

Two forms of internal check are implemented on the 68HC04P3. Self check performs an extensive functional check of the MCU using a signature analysis technique. ROM verify checks the contents of the program ROM by a similar method.

Self check mode is selected by holding MDS and PA7 high and PA6 low during RESET. The external circuitry required is illustrated in Figure 4-2a. Figure 4.3 illustrates the self check program flow. Successful completion of each stage is indicated on the LED display as shown on this flowchart.

ROM verify mode is entered by holding MDS, PA7 and PA6 high during RESET. The result of the check is indicated on the LED display as shown in Figure 4.2b.

Note that unimplemented program space ROM locations are also tested and should read as $00.

4.5 RESET

The MCU can be reset in two ways: by initial power up (see Figure 4-1) and by the external reset input (RESET). During a power-on reset, the timer is used to count 1920 external clock cycles to allow the oscillator to stabilise before releasing the internal reset irrespective of the RESET pin state. If the RESET pin is low at the end of the delay. the processor remains in the reset condition.

A reset can also be achieved by pulling the RESET pin low for a minimum of two clock cycles. The delay is not implemented in this case.

The reset states are as follows:

- All DDRs ($04, $05, $06) set to $00.
- TCR ($FE) and prescaler register ($FD) set to $FF.
- TSCR ($09) set to $00.
- STOP and WAIT latches cleared.
- All flags cleared.
- CRC ($0A, $OB) registers set to $FF.
- All other registers are not affected.
- Interrupt request latch cleared.
- Restart vector loaded into P.C.

4.6 INTERNAL CLOCK GENERATOR OPTIONS

The internal clock generator circuit is designed to require a minimum of external components. A crystal, a resistor-capacitor, a ceramic resonator or an external signal may be used to generate a system clock with various stability/cost tradoffs. A manufacturing mask option is requred to select either the crystal oscillator or the RC oscillator circuit. The different clock generator option connection methods are shown in Figure 4-4, crystal specifications and suggested PC board layouts are given in Figure 4-5, and a timing diagram is illustrated in Figure 4-6. The crystal oscillator startup time is a function of many variables: crystal parameters (especially R_S), oscillator load capacitance (C_L), IC parameters, ambient temperature, and supply voltage. To ensure rapid oscillator startup, neither the crystal characteristics nor the load capacitance should exceed recommendations. Any stray capacitance should be minimised.

The oscillator output frequency is internally divided by 1, 2 or 4 to produce the internal $\phi 1$ clock. The divisor is determined by a manufacturing mask option. The $\phi 1$ clock is divided by twelve to produce one machine cycle. An instruction may need two, four, or five cycles to execute.

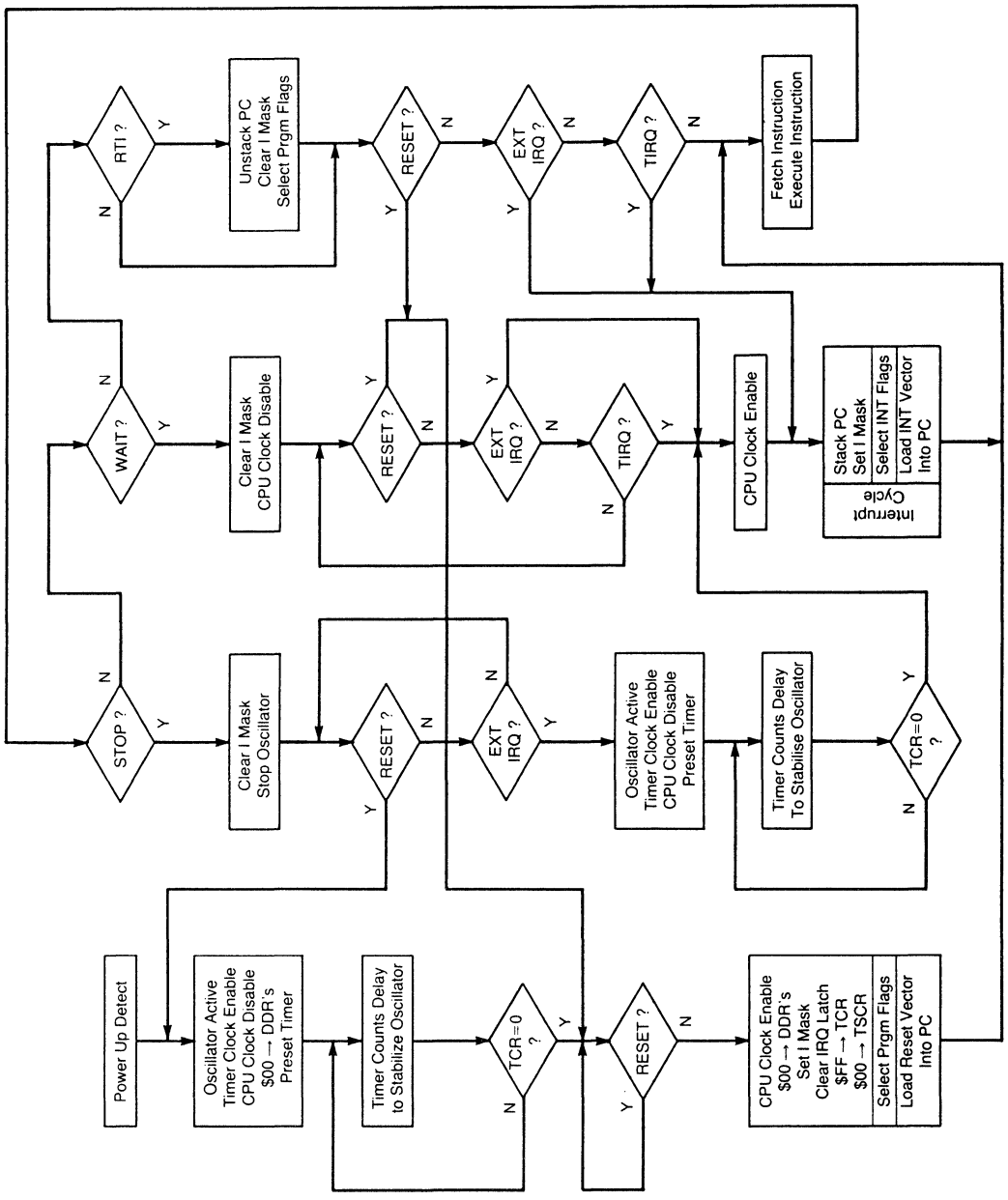


Figure 4-1a Instruction Processing Sequence

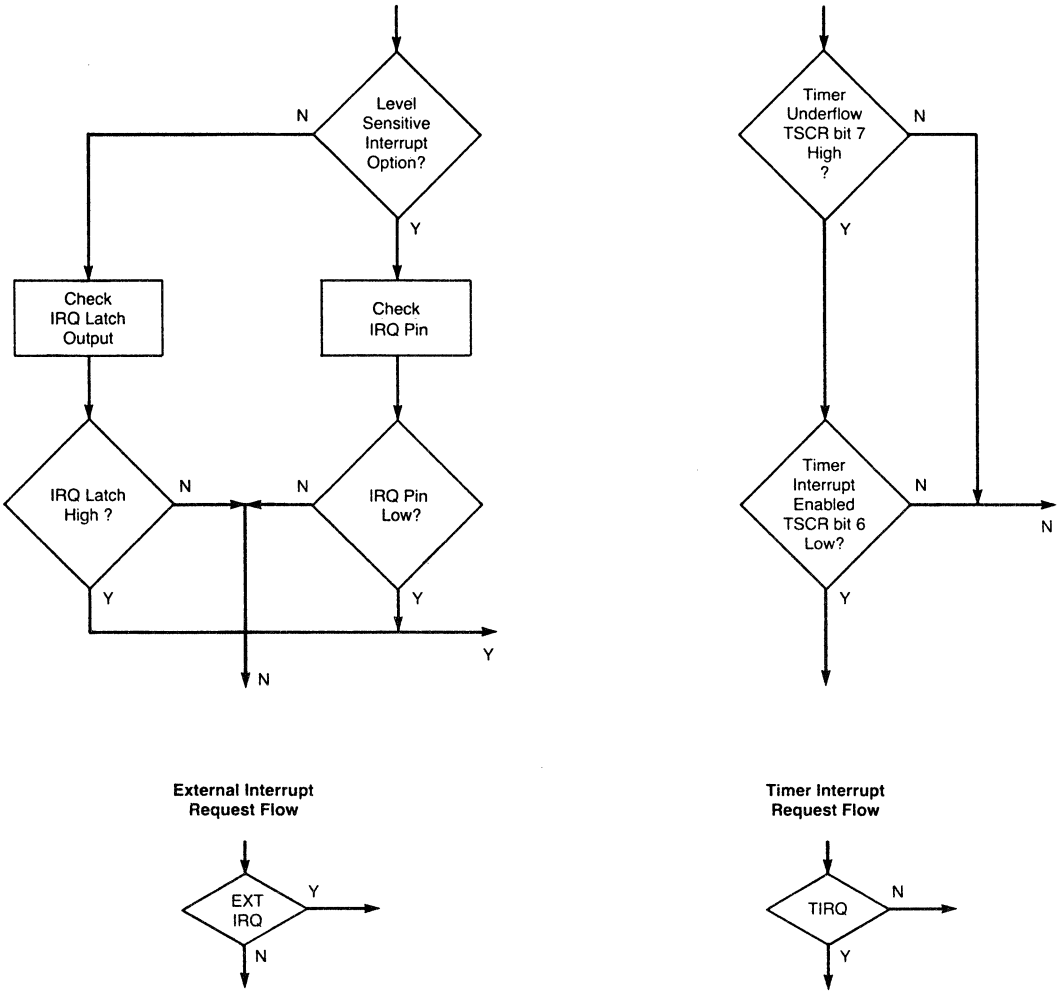
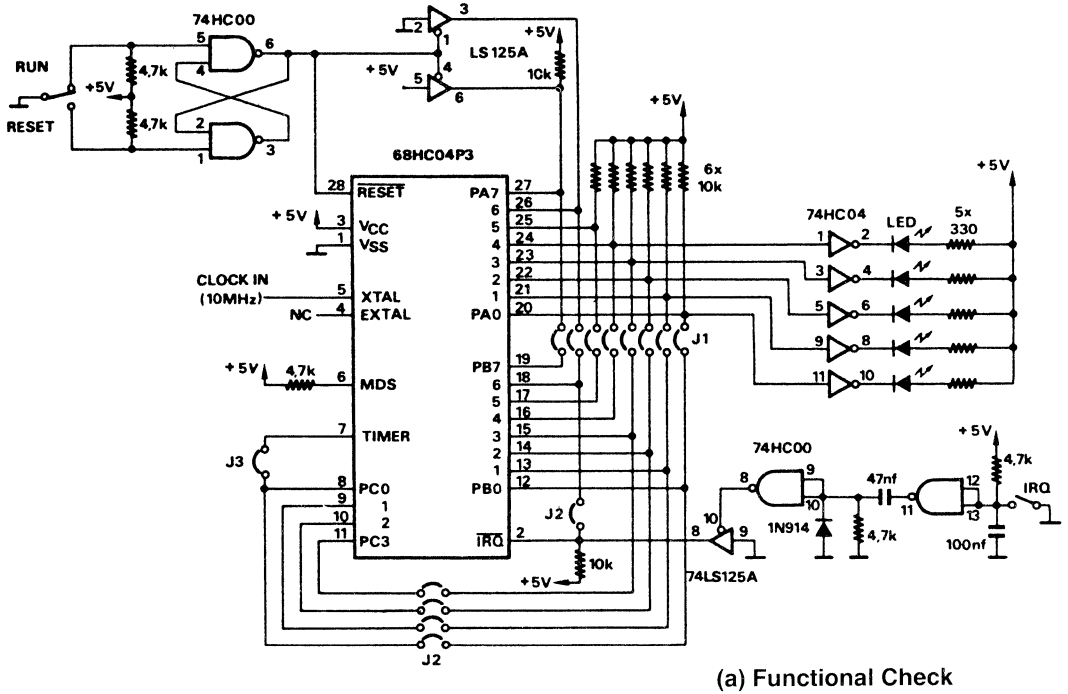


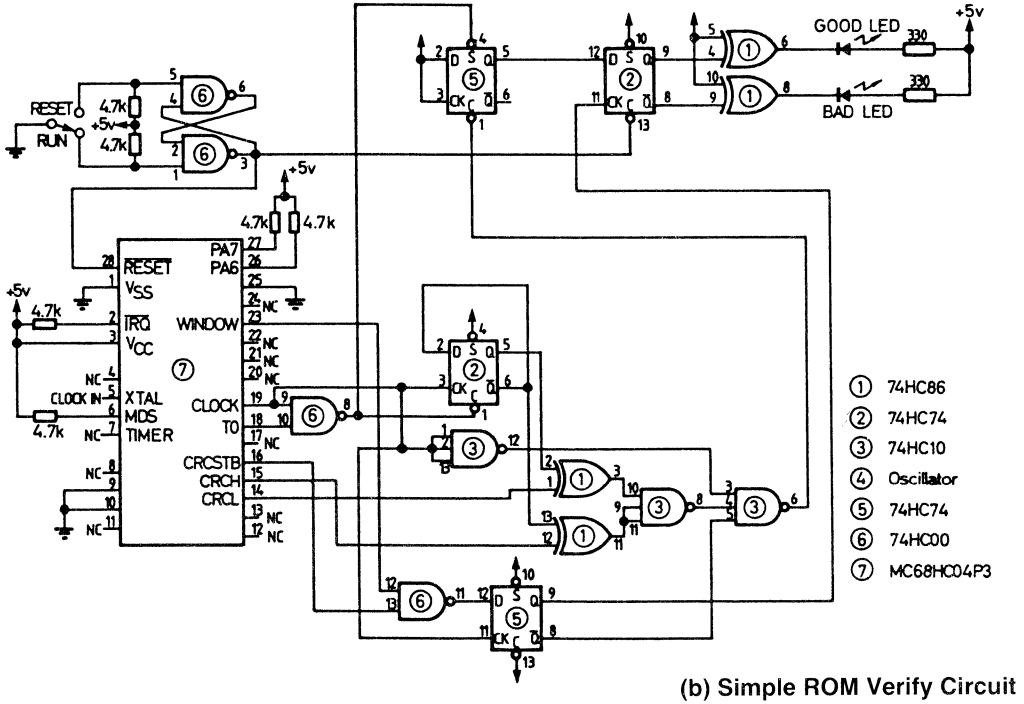
Figure 4-1b Interrupt Sequences

DDR = I/O Port Data Direction Register
TCR = Timer Count Register
TSCR = Timer Status/Control Register
I Mask = Interrupt Mask Flip Flop

IRQ Latch = Interrupt Request Latch Used to Memorize Interrupt Request for Edge Sensitive (External) Interrupt
EXT IRQ = External Interrupt Request
TIRQ = Timer Interrupt Request



(a) Functional Check



(b) Simple ROM Verify Circuit

Figure 4-2. Self Check Circuits

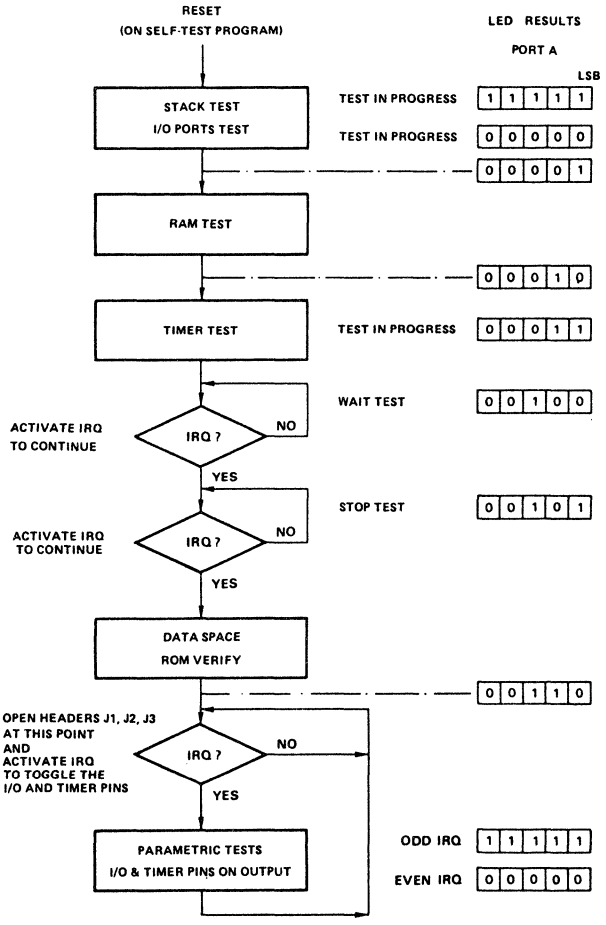


Figure 4-3. Self-Test Flowchart

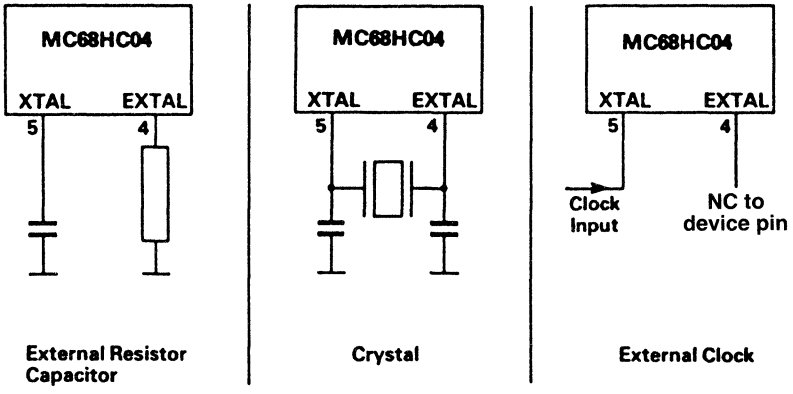


Figure 4-4. Clock Generator Options

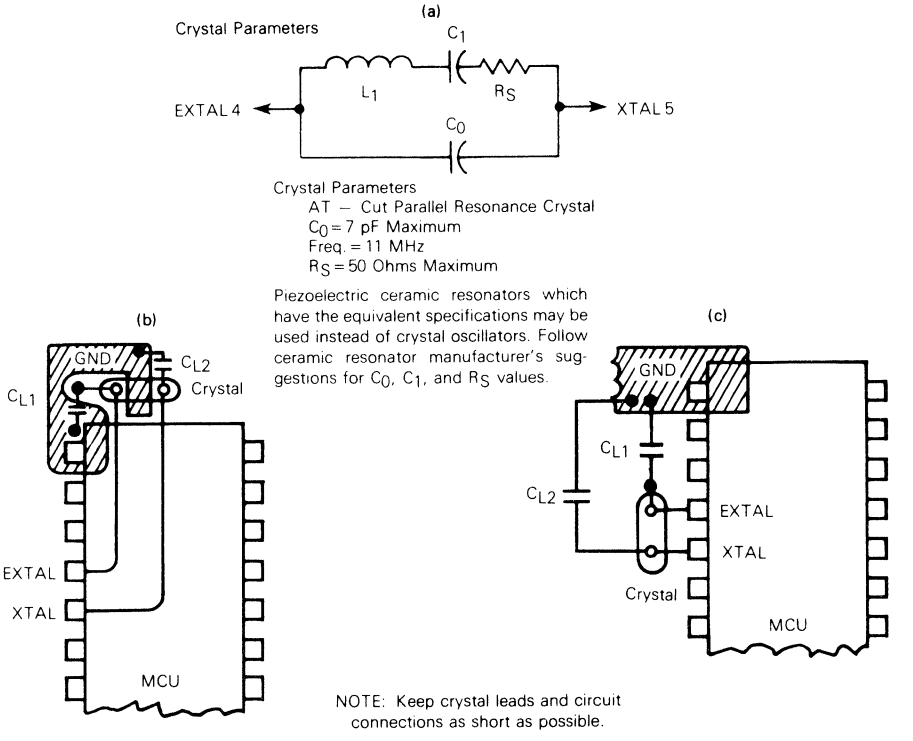


Figure 4-5. Crystal Motional Arm Parameters and Suggested PC Board Layout

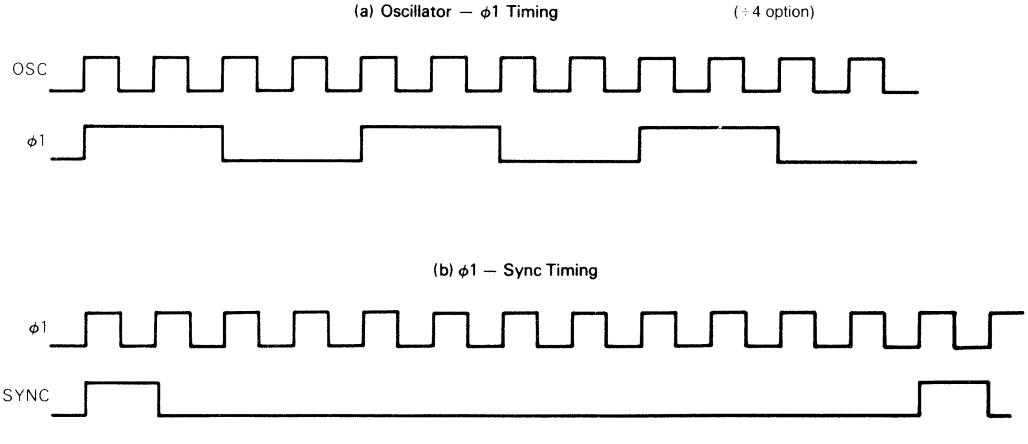


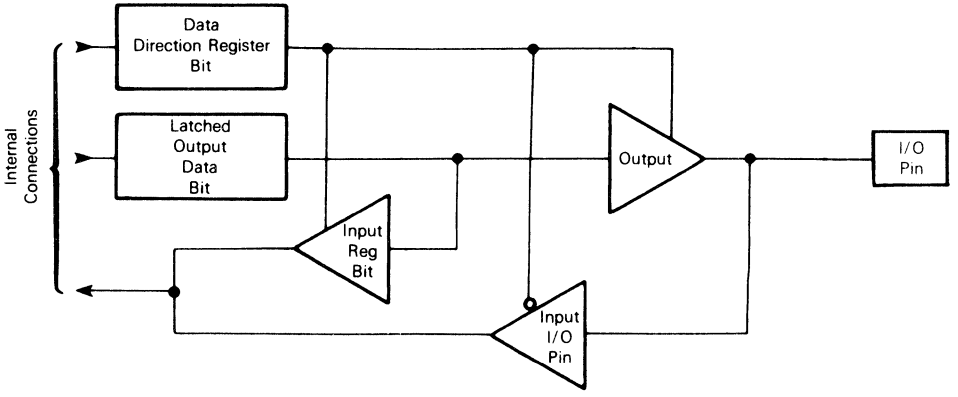
Figure 4-6. Clock Generator Timing Diagrams

SECTION 5
INPUT/OUTPUT

5.1 INPUT/OUTPUT

There are 20 input/output pins. All pins (port A, B and C) are programmable as either inputs or outputs under software control of the corresponding data direction register (DDR). The port I/O programming is accomplished by writing the corresponding bit in the port DDR to a logic one for output or a logic zero for input. On reset, all the DDRs are initialised to configure the ports in the input mode. The port output registers are not initialised on reset but should be initialised before changing the DDR bits to avoid undefined levels. When programmed as outputs, the latched output data is readable as input data, regardless of the logic levels at the output pin due to output loading; see figure 5-1. The address map in figure 2-1 gives the address of data registers and DDRs. The register configuration is discussed under the registers paragraph below.

The latched output data bit (see figure 5-1) may always be written. Therefore any write to a port writes to all of its data bits even though the port DDR is set to input. This may be used to initialise the data registers and avoid undefined outputs; however, care must be exercised when using read-modify-write instructions since the data read corresponds to the pin level if the DDR is an input (0) and corresponds to the latched output data when the DDR is an output (1).



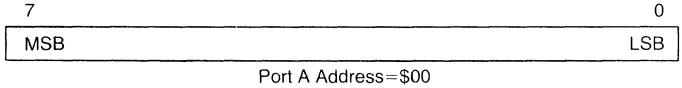
Data Direction Register Bit	Output Data Bit	Output State	Input To MCU
1	0	0	0
1	1	1	1
0	X	Hi-Z	Pin

Figure 5-1. Typical I/O Port Circuitry

5.2 REGISTERS

The registers described below are implemented as RAM locations and thus may be read or written.

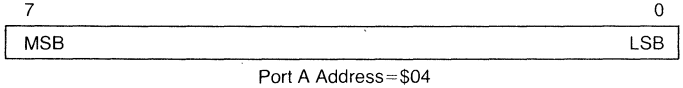
5.2.1 Port Data Registers



Port A Address=$00
Port B Address=$01
Port C Address=$02 (Bits 0-3)

The source of data read from the port data register will be the port I/O pin or previously latched output data depending upon the contents of the corresponding data direction register (DDR). The destination of data written to the port data register will be an output data latch. If the corresponding data direction register (DDR) for the port I/O pin is programmed as an output, the data will then appear on the port pin.

5.2.2 Port Data Direction Register



Port A Address=$04
Port B Address=$05
Port C Address=$06 (Bits 0-3)

The port DDRs configure the port pins as either inputs or outputs. Each port pin can be programmed individually to act as an input or an output. A zero in the pin's corresponding bit position will program that pin as an input while a one in the pin's corresponding bit position will program that pin as an output.

5.3 PULL DOWN DEVICE OPTION

The implementation of pull down devices on particular groupings of I/O ports is a maskable option available to the user. This is typically of benefit in applications where keyboards are interfaced directly to the MCU and similar situations.

The permitted groupings are as follows:

— Port B
— Port C and pin PA0
— Pins PA4, PA3, PA2, PA1
— Pins PA5, PA6
— Pin PA7.

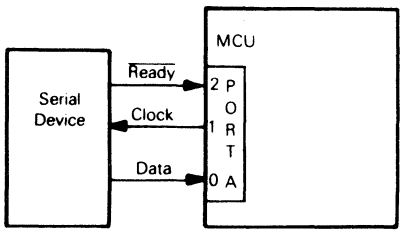
SECTION 6
SOFTWARE AND INSTRUCTION SET

6.1 SOFTWARE

6.1.1 Bit Manipulation

The MC68HC04P3 MCU has the ability to set or clear any register or single random access memory (RAM) writable bit with a single instruction (BSET, BCLR). Any bit in data space, including ROM, can be tested, using the BRSET and BRCLR instructions, and the program may branch as a result of its state. The carry bit equals the value of the bit referenced by BRSET or BRCLR. A rotate instruction may then be used to accumulate serial input data in a RAM location or register. The capability to work with any bit in RAM, ROM or I/O allows the user to have individual flags in RAM or to handle I/O bits as control lines.

The coding example in Figure 6-1 illustrates the usefulness of the bit manipulation and test instructions. Assume that the MCU is to communicate with an external serial device. The external device has a data ready signal, a data output line, and a clock line (to clock data one bit at a time, MSB first, out of the device). The MCU waits until the data is ready, clocks the external device, picks up the data in the carry flag (C bit), clears the clock line, and finally accumulates the data bit in the accumulator.




```
                .
                .
                .
SELF    BRSET   2,PORTA,SELF
                .
                .
                .
        BSET    1,PORTA
        BRCLR   0,PORTA,CONT
CONT    BCLR    1,PORTA
        ROLA    .
                .
                .
                .
```

Figure 6-1. Bit Manipulation Example

6.1.2 Addressing Modes

The MC68HC04P3 MCU has nine addressing modes which are explained briefly in the following paragraphs. The MC68HC04P3 deals with objects in three different address spaces: program space, data space, and stack space. Program space contains the instructions which are to be executed, plus the data for immediate mode instructions. Data space contains all of the RAM locations, XP and YP registers, accumulator, timer, I/O locations, and some ROM (for storage of tables and constants). Stack space contains RAM for use in stacking the return addresses for subroutines and interrupts.

The term "Effective Address" (EA) is used in describing the address modes. EA is defined as the address from which the argument for an instruction is fetched or stored.

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

6.1.2.2 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 256 bytes in data space memory with a single two-byte instruction.

6.1.2.3 SHORT DIRECT. The MCU also has four locations in data space RAM ($80, $81, $82, $83) which may be used in a short-direct addressing mode. In this mode the opcode determines the data space RAM location, and the instruction is only one byte. Short direct addressing is a subset of the direct addressing mode. (The XP and YP registers are at locations $80 and $81 respectively.)

6.1.2.4 EXTENDED. In the extended addressing mode, the effective address is obtained by concatenating the four least significant bits of the opcode with the byte following the opcode (12-bit address). Instructions using the extended addressing mode (JMP, JSR) are capable of branching anywhere in program space. An extended addressing mode instruction is two bytes long.

6.1.2.5 RELATIVE. The relative addressing mode is only used in conditional branch instructions. In relative addressing, that address if formed by adding the sign extended lower five bits of the opcode (the offset) to the program counter if and only if the condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from −15 to +16 from the opcode address. The programmer need not worry about calculating the correct offset when using the Motorola assembler since it calculates the proper offset and checks to see if it is within the span of the branch.

6.1.2.6 BIT SET/CLEAR. In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode specifies the direct address of the byte in which the specified bit is to be set or cleared. Thus, any bit in the 256 locations of data space memory, which can be written to, can be set or cleared.

6.1.2.7 BIT TEST AND BRANCH. The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit and condition (set or clear) which is to be tested is included in the opcode, and the data space address of the byte to be tested is in the single byte immediately following the opcode byte. The third byte is sign extended to twelve bits and becomes the offset added to the program counter if the condition is true. The single three-byte instruction allows the program to branch based on the condition of any bit in data space memory. The span of branching is from −125 to +130 from the opcode address. The state of the tested bit is also transferred to the carry flag.

6.1.2.8 REGISTER-INDIRECT. In the register-indirect addressing mode, the operand is at the address (in data space) pointed to by the contents of one of the indirect registers (X or Y). The particular X or Y register is selected by bit 4 of the opcode. Bit 4 of the opcode is then decoded into an address which selects the desired X or Y register ($80 or $81). A register-indirect instruction is one byte long.

6.1.2.9 INHERENT. In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. These instructions are one byte long.

6.2 INSTRUCTION SET

The MC68HC04P3 MCU has a set of 42 basic instructions, which when combined with nine addressing modes produce 242 usable opcodes. 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. All the instructions within a given type are presented in individual tables.

6.2.1 Register/Memory Instructions

Most of these instructions use two operands. One operand is the accumulator and 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 operands. Refer to Table 6-1.

6.2.2 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. There are ten instructions which utilize read-modify-write cycles. All INC and DEC forms along with all bit manipulation instructions use this method. Refer to Table 6-2.

6.2.3 Branch Instructions

The branch instructions cause a branch from the program when a certain condition is met. Refer to Table 6-3.

6.2.4 Bit Manipulation Instructions

These instructions are used on any bit in data space memory. One group either sets or clears. The other group performs the bit test branch operations. Refer to Table 6-4.

Table 6-1. Register/Memory Instructions

		Addressing Modes																			
		Indirect				Immediate			Direct			Inherent			Extended			Short-Direct			
Function	Mnem	Opcode XP	Opcode YP	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Special Notes
Load A from Memory	LDA	E0	F0	1	4	E8	2	4	F8	2	4	—	—	—	—	—	—	AC-AF	1	4	1
Load XP from Memory	LDXI	—	—	—	—	B0	3	4	—	—	—	—	—	—	—	—	—	—	—	—	4
Load YP from Memory	LDYI	—	—	—	—	B0	3	4	—	—	—	—	—	—	—	—	—	—	—	—	4
Store A in Memory	STA	E1	F1	1	4	—	—	—	F9	2	4	—	—	—	—	—	—	BC-BF	1	4	2
Add to A	ADD	E2	F2	1	4	EA	2	4	FA	2	4	—	—	—	—	—	—	—	—	—	—
Subtract from A	SUB	E3	F3	1	4	EB	2	4	FB	2	4	—	—	—	—	—	—	—	—	—	—
Arithmetic Compare with Memory	CMP	E4	F4	1	4	EC	2	4	FC	2	4	—	—	—	—	—	—	—	—	—	—
AND Memory to A	AND	E5	F5	1	4	ED	2	4	FD	2	4	—	—	—	—	—	—	—	—	—	—
Jump to Subroutine	JSR	—	—	—	—	—	—	—	—	—	—	—	—	—	8 (TAR)	2	4	—	—	—	3
Jump Unconditional	JMP	—	—	—	—	—	—	—	—	—	—	—	—	—	9 (TAR)	2	4	—	—	—	3
Clear A	CLRA	—	—	—	—	—	—	—	FB	2	4	—	—	—	—	—	—	—	—	—	—
Clear XP	CLRX	—	—	—	—	—	—	—	FB	2	4	—	—	—	—	—	—	—	—	—	—
Clear YP	CLRY	—	—	—	—	—	—	—	FB	2	4	—			—	—	—	—	—	—	—
Complement A	COMA	—	—	—	—	—	—	—	—	—	—	B4	1	4	—	—	—	—	—	—	—
Move Immediate Value to Memory	MVI	—	—	—	—	B0	3	4	B0	3	4	—	—	—	—	—	—	—	—	—	5
Rotate A Left and Carry	ROLA	—	—	—	—	—	—	—	—	—	—	B5	1	4	—	—	—	—	—	—	—
Arithmetic Left Shift of A	ASLA	—	—	—	—	—	—	—	FA	2	4	—	—	—	—	—	—	—	—	—	—

SPECIAL NOTES

1. In Short-Direct addressing, the LDA mnemonic represents opcode AC, AD, AE, and AF. This is equivalent to RAM locations $80 (AC), $81 (AD), $82 (AE), and $83 (AF)
2. In Short-Direct addressing, the STA mnemonic represents opcode BC, BD, BE, and BF. This is equivalent to RAM locations $80 (BC), $81 (BD), $82 (BE), and $83 (BF).
3. In Extended addressing, the four LSBs of the opcode (Mnemonic JSR and JMP) are formed by the four MSBs of the target address.
4. In Immediate addressing, the LDXI and LDYI are mnemonics which are recognized as follows:
   LDXI = MVI $80,data
   LDYI = MVI $81,data Where data is a one-byte hexadecimal number.
5. In both Immediate and Direct addressing, the MVI instruction has the same opcode (80).

Table 6-2. Read-Modify-Write Instructions

Function	Mnem	Addressing Modes										Special Notes
		Indirect				Direct			Short-Direct			
		Opcode		#	#		#	#		#	#	
		XP	YP	Bytes	Cycles	Opcode	Bytes	Cycles	Opcode	Bytes	Cycles	
Increment Memory Location	INC	E6	F6	1	4	FE	2	4	A8-AB	1	4	1, 3
Increment A	INCA	—	—	—	—	FE	2	4	—	—	—	—
Increment XP	INCX	—	—	—	—	—	—	—	A8	1	4	—
Increment YP	INCY	—	—	—	—	—	—	—	A9	1	4	—
Decrement Memory Location	DEC	E7	F7	1	4	FF	2	4	B8-BB	1	4	2, 4
Decrement A	DECA	—	—	—	—	FF	2	4	—	—	—	—
Decrement XP	DECX	—	—	—	—	—	—	—	B8	1	4	—
Decrement YP	DECY	—	—	—	—	—	—	—	B9	1	4	—

SPECIAL NOTES

1. In Short-Direct addressing, the INC mnemonic represents opcode A8, A9, AA, and AB. These are equivalent to RAM locations $80 (A8), $81 (A9), $82 (AA), and $83 (AB).
2. In Short-Direct addressing, the DEC mnemonic represents opcode B8, B9, BA, and BB. These are equivalent to RAM locations $80 (B8), $81 (B9), $82 (BA), and $83 (BB).
3. In Indirect addressing, the INC mnemonic represents opcode E6 or F6, and causes the location pointed to by XP (E6 opcode) or YP (F6 opcode) to be incremented.
4. In Indirect addressing, the INC mnemonic represents opcode E7 or F7, and causes the location pointed to by XP (E7 opcode) or YP (F7 opcode) to be incremented.

Table 6-3. Branch Instructions

		Relative Addressing Mode			
Function	Mnem	Opcode	# Bytes	# Cycles	Special Notes
Branch if Carry Clear	BCC	40-5F	1	2	1
Branch if Higher or Same	(BHS)	40-5F	1	2	1, 2
Branch if Carry Set	BCS	60-7F	1	2	1
Branch if Lower	(BLO)	60-7F	1	2	1, 3
Branch if Not Equal	BNE	00-1F	1	2	1
Branch if Equal	BEQ	20-3F	1	2	1

SPECIAL NOTES

1. Each mnemonic of the Branch Instructions covers a range of 32 opcodes; e.g., BCC ranges from 40 through 5F. The actual memory location (target address) to which the branch is made is formed by adding the sign extended lower five bits of the opcode to the contents of the program counter.
2. The BHS instruction (shown in parentheses) is identical to the BCC instruction. The C bit is clear if the register was higher or the same as the location in the memory to which it was compared.
3. The BLO instruction (shown in parentheses) is identical to the BCS instruction. The C bit is set if the register was lower than the location in memory to which it was compared.

Table 6-4. Bit Manipulation Instructions

		Addressing Modes						
		Bit Set/Clear			Bit Test and Branch			
Function	Mnem	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Special Note
Branch IFF Bit n is set	BRSET n (n=0 . . . . 7)	–	–	–	C8+n	3	5	1
Branch IFF Bit n is clear	BRCLR n (n=0 . . . . 7)	–	–	–	C0+n	3	5	1
Set Bit n	BSET n (n=0 . . . .7)	D8+n	2	4	–	–	–	1
Clear Bit n	BCLR n (n=0 . . . . 7)	D0+n	2	4	–	–	–	1

SPECIAL NOTE

1. The opcode is formed by adding the bit number (0-7) to the basic opcode. For example: to clear bit six using the BSET6 instruction the opcode becomes DE (D8+6); BCLR5 becomes (C0+5); etc.

6.2.5 Control Insructions

The control instructions control the MCU operations during program execution. Refer to Table 6-5.

6.2.6 Alphabetical Listing

The complete instruction set is given in alphabetical order in Table 6-6. There are certain mnemonics recognized by the Motorola assembler and converted to other instructions. The fact that all registers and accumulator are in RAM allows many implied instructions to exist. The implied instructions recognized by the Motorola assembler are identified in Table 6-6.

6.2.7 Opcode Map Summary

Table 6-7 contains an opcode map for the instructions used on the MCU.

6.3 IMPLIED INSTRUCTIONS

Since the accumulator and all other registers are located in RAM many implied instructions exist. The assembler-recognized implied instructions are given in Table 6-6. Some examples not recognized by the assembler are shown below.

BCLR 7,$FF	Ensures accumulator is plus
BSET 7,$FF	Ensures accumulator is minus
BRCLR 7,$FF	Branch iff accumulator is plus
BRSET 7,$FF	Branch iff accumulator is minus
BRCLR 7,$80	Branch iff X is plus (BXPL)
BRSET 7,$80	Branch iff X is minus (BXMI)
BRCLR 7,$81	Branch iff Y is plus (BYPL)
BRSET 7,$81	Branch iff Y is minus (BYMI)

		Addressing Modes									
		Short-Direct			Inherent			Relative			
Function	Mnem	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Special Notes
Transfer A to XP	TAX	BC	1	4	—	—	—	—	—	—	—
Transfer A to YP	TAY	BD	1	4	—	—	—	—	—	—	—
Transfer XP to A	TXA	AC	1	4	—	—	—	—	—	—	—
Transfer YP to A	TYA	AD	1	4	—	—	—	—	—	—	—
Return from Subroutine	RTS	—	—	—	B3	1	2	—	—	—	—
Return from Interrupt	RTI	—	—	—	B2	1	2	—	—	—	—
No-Operation	NOP	—	—	—	—	—	—	—	—	—	1
Stop	STOP	—	—	—	B6	1	2	—	—	—	—
Wait	WAIT	—	—	—	B7	1	2	—	—	—	—

SPECIAL NOTE

1 . The NOP instruction is equivalent to a branch if equal (BEQ) to the location designated by PC + 1.

Table 6-5. Control Instructions

	Addressing Modes									Flags	
Mnemonic	Inherent	Immediate	Direct	Short Direct	Bit Set Clear	Bit Test Branch	Register Indirect	Extended	Relative	Z	C
ADD		X	X				X			Λ	Λ
AND		X	X				X			Λ	•
ASLA	Assembler converts this to "ADD $FF"									•	•
BCC									X	•	•
BCLR					X					•	•
BCS									X	•	•
BEQ									X	•	•
BHS	Assembler converts this to "BCC"									•	•
BLO	Assembler converts this to "BCS"									•	•
BNE									X	•	•
BRCLR						X				•	Λ
BRSET						X				•	Λ
BSET					X					•	•
CLRA	Assembler converts this to "SUB $FF"									Λ	Λ
CLRX	Assembler converts this to "MVI $80,#0"									•	•
CLRY	Assembler converts this to "MVI $81,#0"									•	•
CMP		X	X				X			Λ	Λ
COMA	X									Λ	Λ
DEC			X	X			X			Λ	•
DECA	Assembler converts this to "DEC $FF"									Λ	•
DECX	Assembler converts this to "DEC $80"									Λ	•
DECY	Assembler converts this to "DEC $81"									Λ	•
INC			X	X			X			Λ	•
INCA	Assembler converts this to "INC $FF"									Λ	•
INCX	Assembler converts this to "INC $80"									Λ	•
INCY	Assembler converts this to "INC $81"									Λ	•
JMP								X		•	•
JSR								X		•	•
LDA		X	X	X			X			Λ	•
LDXI	Assembler converts this to "MVI $80,DATA"									•	•
LDYI	Assembler converts this to "MVI $81,DATA"									•	•
MVI		X	X							•	•
NOP	Assembler converts this to "BEQ (PC) + 1"									•	•
ROLA	X									Λ	Λ
RTI	X									Λ	Λ
RTS	X									•	•
STA			X	X			X			Λ	•
STOP	X									•	•
SUB		X	X				X			Λ	Λ
TAX	Assembler converts this to "STA $80"									•	•
TAY	Assembler converts this to "STA $81"									•	•
TXA	Assembler converts this to "LDA $80"									•	•
TYA	Assembler converts this to "LDA $81"									•	•
WAIT	X									•	•

Flag Symbols: Z = Zero, C = Carry/Borrow, Λ = Test and Set if True, Cleared Otherwise, • = Not Affected

Table 6-6. Instruction Set

Table 6-7. MC68HC04P3

Branch Instructions

Hi / Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111
0 0000	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
1 0001	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
2 0010	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
3 0011	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
4 0100	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
5 0101	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
6 0110	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
7 0111	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
8 1000	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
9 1001	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
A 1010	2 BNE 1 REL	2 BNE 2 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
B 1011	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
C 1100	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
D 1101	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
E 1110	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
F 1111	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL

Abbreviations for Address Modes

iNH	Inherent
S-D	Short Direct
B-T-B	Bit Test and Branch
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
R-IND	Register Indirect

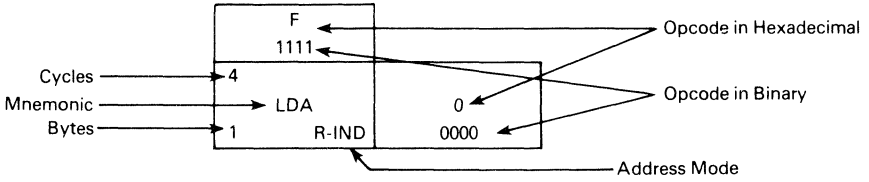
* Indicates Instruction Reserved for Future Use

\# Indicates Illegal Instruction

Instruction Set Opcode Map

	Register/Memory, Control, and Read/Modify/Write Instructions				Bit Manipulation Instructions		Register/Memory and Read/Modify/Write		
	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi / Low
	4 JSRn 2 EXT	4 JMPn 2 EXT	*	4 MVI 3 IMM	5 BRCLR0 3 B-T-B	4 BCLR0 2 BSC	4 LDA 1 R-IND	4 LDA 1 R-IND	0 0000
	4 JSRn 2 EXT	4 JMPn 2 EXT	*	*	5 BRCLR1 3 B-T-B	4 BCLR1 2 BSC	4 STA 1 R-IND	4 STA 1 R-IND	1 0001
	4 JSRn 2 EXT	4 JMPn 2 EXT	*	2 RTI 1 INH	5 BRCLR2 3 BT B	4 BCLR2 2 BSC	4 ADD 1 R-IND	4 ADD 1 R-IND	2 0010
	4 JSRn 2 EXT	4 JMPn 2 EXT	*	2 RTS 1 INH	5 BRCLR3 3 B-T-B	4 BCLR3 2 BSC	4 SUB 1 R-IND	4 SUB 1 R-IND	3 0011
	4 JSRn 2 EXT	4 JMPn 2 EXT	*	4 COMA 1 INH	5 BRCLR4 3 B-T-B	4 BCLR4 2 BSC	4 CMP 1 R-IND	4 CMP 1 R-IND	4 0100
	4 JSRn 2 EXT	4 JMPn 2 EXT	*	4 ROLA 1 INH	5 BRCLR5 3 B-T-B	4 BCLR5 2 BSC	4 AND 1 R-IND	4 AND 1 R-IND	5 0101
	4 JSRn 2 EXT	4 JMPn 2 EXT	*	**2 STOP 1 INH**	5 BRCLR6 3 B-T-B	4 BCLR6 2 BSC	4 INC 1 R-IND	4 INC 1 R-IND	6 0110
	4 JSRn 2 EXT	4 JMPn 2 EXT	*	**2 WAIT 1 INH**	5 BRCLR7 3 B-T-B	4 BCLR7 2 BSC	4 DEC 1 R-IND	4 DEC 1 R-IND	7 0111
	4 JSRn 2 EXT	4 JMPn 2 EXT	4 INC 1 S-D	4 DEC 1 S-D	5 BRSET0 3 B-T-B	4 BSET0 2 BSC	4 LDA 2 IMM	4 LDA 2 DIR	8 1000
	4 JSRn 2 EXT	4 JMPn 2 EXT	4 INC 1 S-D	4 DEC 1 S-D	5 BRSET1 3 B-T-B	4 BSET1 2 BSC	#	4 STA 2 DIR	9 1001
	4 JSRn 2 EXT	4 JMPn 2 EXT	4 INC 1 S-D	4 DEC 1 S-D	5 BRSET2 3 B-T-B	4 BSET2 2 BSC	4 ADD 2 IMM	4 ADD 2 DIR	A 1010
	4 JSRn 2 EXT	4 JMPn 2 EXT	4 INC 1 S-D	4 DEC 1 S-D	5 BRSET3 3 B-T-B	4 BSET3 2 BSC	SUB IMM	4 SUB 2 DIR	B 1011
	4 JSRn 2 EXT	4 JMPn 2 EXT	4 LDA 1 S-D	4 STA 1 S-D	5 BRSET4 3 B-T-B	4 BSET4 2 BSC	4 CMP 2 IMM	CMP DIR	C 1100
	4 JSRn 2 EXT	4 JMPn 2 EXT	4 LDA 1 S-D	4 STA 1 S-D	5 BRSET5 3 B-T-B	4 BSET5 2 BSC	4 AND 2 IMM	4 AND 2 DIR	D 1101
	4 JRSn 2 EXT	4 JMPn 2 EXT	4 LDA 1 S-D	4 STA 1 S-D	5 BRSET6 3 B-T-B	4 BSET6 2 BSC	#	4 INC 2 DIR	E 1110
	4 JSRn 2 EXT	4 JMPn 2 EXT	4 LDA 1 S-D	4 STA 1 S-D	5 BRSET7 3 B-T-B	4 BSET7 2 BSC	#	4 DEC 2 DIR	F 1111

LEGEND



SECTION 7
ELECTRICAL CHARACTERISTICS

7.1 MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.5 to +7	V
Input Voltage	V_{in}	V_{SS}−0.5 to V_{DD}+0.5	V
Current drain per pin Excluding V_{DD} and V_{SS}	I	10	mA
Total current for sink Ports A, B, C EXTAL, TIM source	I I	30 15	mA mA
Operating temperature range	T_A	−40 to +85	°C
Storage temperature range	T_{stg}	−55 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 applications 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 $V_{SS} \leq (V_{in}) \leq V_{DD}$. Reliability of operation is enhanced if unused inputs except EXTAL are tied to an appropriate logic voltage level (e.g. either V_{SS} or V_{DD}).

7.2 THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal resistance			
Plastic	θ_{JA}	100	°C/W
Ceramic	θ_{JA}	50	°C/W

7.3 POWER CONSIDERATIONS

The average chip-junction temperature T_J in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \quad (1)$$

Where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D = P_{INT} + P_{I/O}$

$P_{INT} = I_{DD} \times V_{CC}$, Watts — Chip Internal Power

$P_{I/O}$ = Power Dissipation on Input and Output Pins — User Determined

An approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is:

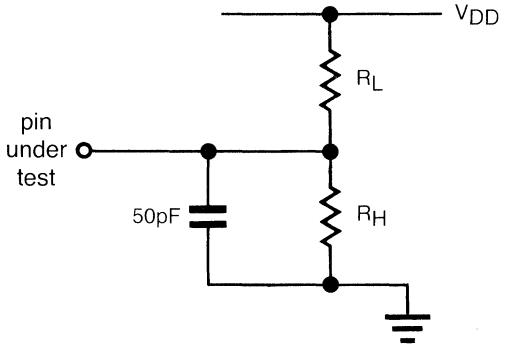
$$P_D = K - (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives:

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

7.4 EQUIVALENT TEST LOAD



V_{DD} = +4.5V

I_{OL}/I_{OH} = 800µA

$R_L = R_H$ = 4.6kΩ

V_{DD} = +2.7V

I_{OL}/I_{OH} = 200µA

$R_L = R_H$ = 10.5kΩ

V_{DD} = +2.0V

I_{OL}/I_{OH} = 100µA

$R_L = R_H$ = 16kΩ

7.5 DC ELECTRICAL CHARACTERISTICS V_{DD}=+5Vdc±10%, V_{SS}=0Vdc, T_A=−40 to +85°C

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage, I_{Load}<10.0μA	V_{OL} V_{OH}	— V_{DD}−0.1	— —	0.1 —	V V
Output High Voltage, I_{Load}=−800μA) — Ports, TIM Output Low Voltage, I_{Load}=+800μA) — Ports, TIM	V_{OH} V_{OL}	V_{DD}−0.4 —	— —	— 0.4	V V
Input High Voltage — Ports, TIM, XTAL, MDS — IRQ, RESET	V_{IH} V_{IH}	0.7×V_{DD} 0.8×V_{DD}	— —	V_{DD} V_{DD}	V V
Input Low Voltage — Ports, TIM, MDS, XTAL — IRQ, RESET	V_{IL} V_{IL}	V_{SS} V_{SS}	— —	0.3×V_{DD} 0.2×V_{DD}	V V
Total Supply Current C_L=50pF, Ports, TIM No dc load, t_{CYC}=1/f_{CL}(max) V_{IL}=0.2v, V_{IH}=V_{DD}−0.2V — RUN — WAIT* — STOP*	 I_{DD} I_{DD} I_{DD}	 — — —	 3 1 <1	 TBD TBD TBD	 mA mA μA
I/O Ports Input Leakage V_{SS}<V_I<V_{DD}	I_{IL}	—	—	±1	μA
Input Current RESET, IRQ, TIM	I_{in}	—	—	±1	μA
Capacitance per pin — PORTS (as input or output) — RESET, IRQ, TIM, XTAL, MDS	 C_{out} C_{in}	 — —	 — —	 12 8	 pF pF

*Measured under the following conditions:
— all ports and timer pin are configured as input
— XTAL is driven by a square wave input
— EXTAL is open circuit
— port pull downs not enabled

Note: typical pull down sink current for V_{out}=V_{DD} is 50μA.

7.6 CONTROL TIMING CHARACTERISTICS V_{DD}=+5Vdc±10%, V_{SS}=0Vdc, T_A=−40 to +85°C

Characteristic	Symbol	Min	Typ	Max	Unit
Oscillator Frequency	f_{OSC}	0	—	11.0	MHz
PHI1 Clock Frequency	f_{CL}	0	—	5.5	MHz
Cycle time (min)	t_{CYC}	2.2	—	—	μs
IRQ Pulse Width	t_{IWL}	2×t_{CYC}	—	—	μs
RESET Pulse Width	t_{RWL}	2×t_{CYC}	—	—	μs
Oscillator Clock Pulse Width	t_{OL},t_{OH}	45	—	—	ns

7.7 DC ELECTRICAL CHARACTERISTICS V_{DD}=+3Vdc±10%, V_{SS}=0Vdc, T_A=−40 to +85°C

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage, I_{Load}<10.0μA	V_{OL} V_{OH}	— V_{DD}−0.1	— —	0.1 —	V V
Output High Voltage, I_{Load}=−200μA) — Ports, TIM Output Low Voltage, I_{Load}=+200μA) — Ports, TIM	V_{OH} V_{OL}	V_{DD}−0.3 —	— —	— 0.3	V V
Input High Voltage, — Ports, TIM, XTAL, MDS IRQ, RESET	V_{IH} V_{IH}	0.7×V_{DD} 0.8×V_{DD}	— —	V_{DD} V_{DD}	V V
Input Low Voltage — Ports, TIM, MDS, XTAL IRQ, RESET	V_{IL} V_{IL}	V_{SS} V_{SS}	— —	0.3×V_{DD} 0.2×V_{DD}	V V
Total Supply Current C_L=50pF, Ports, TIM No dc load, t_{CYC}=1/f_{CL}(max) V_{IL}=0.2v, V_{IH}=V_{DD}−0.2V — RUN — WAIT* — STOP*	 I_{DD} I_{DD} I_{DD}	 — — —	 1.5 0.5 <1	 TBD TBD TBD	 mA mA μA
I/O Ports Input Leakage V_{SS}<V_I<V_{DD}	I_{LL}	—	—	±1	μA
Input Current RESET, IRQ, TIM	I_{in}	—	—	±1	μA
Capacitance per pin — PORTS (as input or output) — RESET, IRQ, TIM, XTAL, MDS	 C_{out} C_{in}	 — —	 — —	 12 8	 pF pF

*Measured under the following conditions:
— all ports and timer pin are configured as input
— XTAL is driven by a square wave input
— EXTAL is open circuit
— port pull downs not enabled

7.8 CONTROL TIMING CHARACTERISTICS V_{DD}=+3Vdc±10%, V_{SS}=0Vdc, T_A=−40 to +85°C

Characteristic	Symbol	Min	Typ	Max	Unit
Oscillator Frequency	f_{OSC}	—	—	11	MHz
PHI1 Clock Frequency	f_{CL}	—	—	4.2	MHz
Cycle time (min)	t_{CYC}	2.9	—	—	μs
IRQ Pulse Width	t_{IWL}	2×t_{CYC}	—	—	μs
RESET Pulse Width	t_{RWL}	2×t_{CYC}	—	—	μs
Oscillator Clock Pulse Width	t_{OL},t_{OH}	45	—	—	ns

7.9 DC ELECTRICAL CHARACTERISTICS V_{DD}=+2.2Vdc±10%, V_{SS}=0Vdc, T_A=−40 to +85°C

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage, I_{Load}<10.0μA	V_{OL} V_{OH}	— V_{DD}−0.1	— —	0.1 —	V V
Output High Voltage, I_{Load}=−100μA) — Ports, TIM Output Low Voltage, I_{Load}=+100μA) — Ports, TIM	V_{OH} V_{OL}	V_{DD}−0.3 —	— —	— 0.3	V V
Input High Voltage, — Ports, TIM, XTAL, MDS IRQ, RESET	V_{IH} V_{IH}	0.7×V_{DD} 0.8×V_{DD}	— —	V_{DD} V_{DD}	V V
Input Low Voltage — Ports, TIM, MDS, XTAL IRQ, RESET	V_{IL} V_{IL}	V_{SS} V_{SS}	— —	0.3×V_{DD} 0.2×V_{DD}	V V
Total Supply Current C_L=50pF, Ports, TIM No dc load, t_{CYC}=1/f_{CL}(max) V_{IL}=0.2v, V_{IH}=V_{DD}−0.2V — RUN — WAIT* — STOP*	 I_{DD} I_{DD} I_{DD}	 — — —	 1 0.3 <1	 TBD TBD TBD	 mA mA μA
I/O Ports Input Leakage V_{SS}<V_I<V_{DD}	I_{LL}	—	—	±1	μA
Input Current RESET, IRQ, TIM	I_{in}	—	—	±1	μA
Capacitance per pin — PORTS (as input or output) — RESET, IRQ, TIM, XTAL, MDS	 C_{out} C_{in}	 — —	 — —	 12 8	 pF pF

*Measured under the following conditions:
— all ports and timer pin are configured as input
— XTAL is driven by a square wave input
— EXTAL is open circuit
— port pull downs not enabled

7.10 CONTROL TIMING CHARACTERISTICS V_{DD}=+2.2Vdc±10%, V_{SS}=0Vdc, T_A=−40 to +85°C

Characteristic	Symbol	Min	Typ	Max	Unit
Oscillator Frequency	f_{OSC}	0	—	8.4	MHz
PHI1 Clock Frequency	f_{CL}	0	—	2.1	MHz
Cycle time (min)	t_{CYC}	5.7	—	—	μs
IRQ Pulse Width	t_{IWL}	2×t_{CYC}	—	—	μs
RESET Pulse Width	t_{RWL}	2×t_{CYC}	—	—	μs
Oscillator Clock Pulse Width	t_{OL},t_{OH}	45	—	—	ns

Note: 2V operation is a user-selectable option only. Prior consultation with the factory is required.

SECTION 8
ORDERING INFORMATION

8.1 INTRODUCTION

The following information is required when ordering a custom MCU. The information may be transmitted to Motorola in the following media:

EPROM(s), MCM2716 or MCM2532
MDOS, disk file

To initiate a ROM pattern for the MCU, it is necessary to first contact your local field service office, local sales person, or your local Motorola representative.

8.1.1 EPROMs

An MCM2716 or MCM2532 type EPROM, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. Since all program and data space information will fit in one MCM2617 or MCM2532 EPROM, the EPROM must be programmed as follows in order to emulate the MC68HC04P3 MCU. Start the data space ROM at EPROM address $000 and start program space ROM at EPROM address $3E8 and continue to memory space $FFF. All unused bytes, including the user's space, must be set to zero. For shipment to Motorola, the EPROMs should be placed in a conductive IC carrier and packed securely. Do no use styrofoam.

8.1.2 MDOS Disk File

An MDOS disk, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. When using the MDOS disk, include the entire memory image of both data and program space. All unused bytes, including the user's space, must be set to zero.

8.2 VERIFICATION MEDIA

All original pattern media (EPROMs or floppy disk) are filed for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked and the verification form completed, signed, and returned to Motorola. The signed verification form constitutes the contractural agreement for creation of the customer mask.

8.3 ROM VERIFICATION (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask but are for the purposes of ROM verification only. For expediency they are usually unmarked, packaged in ceramic, and tested only at room temperature and five volts. These RVUs are included in the mask charge and are not production parts. These RVUs are not backed nor guaranteed by Motorola Quality Assurance.

8.4 FLEXIBLE DISKS

The disk media submitted must be single-sided, single density, 8-inch, MDOS compatible floppies. The customer must clearly label the disk with the ROM pattern file name. The minimum MDOS system files as well as the absolute binary object file (filename.LO type of file) from the M6805 cross assembler must be on the disk. An object file made from a memory dump; using the ROLLOUT command is also admissable. Consider submitting a source listing as well as: filename, .LX (EXORciser loadable format). This file will of course be kept confidential and is used 1) to speed up the process in house if any problems arise, and 2) to speed up our customer to factory interface if a user finds any software errors and needs assistance quickly from the factory representative.

MDOS is Motorola's Disk Operating System available on development systems such as EXORciser, EXORset, etc.

Figure 8-1 ORDERING FORM

MOTOROLA MASK PROGRAMMING INFORMATION

Device '*MC68HC04P3* ' SC Number ( )

Customer ' ' Salesman ' '
Location ' ' Sales Office ' '
Phone ' ' Phone ' '

Technical Options

Internal OSC Input — CRYSTAL ☐ Resistor ☐

CTM OSC Frequency — : :MHz (e.g. 1.0)

External Interrupt — Edge Sensitive ☐ Level Sensitive ☐

Clock Predivide Ratio — 1 ☐ 2 ☐ 4 ☐

PROM — Protect ☐ No Protect ☐

Voltage Supply — 2.7V to 5.5V ☐ 2.0V to 5.5V ☐

Temperature Range — −40° – +85°C ☐ −55° – +125°C ☐

Additional Marking — ' ' (e.g. CTM Pat No. 10 Chars Max)

Complete Next Section in Reference to Additional Marking —

Customer Spec/Drawing/Print — No ☐ Yes ☐

Package — Ceramic ☐ Plastic ☐ Other ☐ . . . . . .

RVU Approval Waived — No ☐ Yes ☐ Qty ( )

Give indication of initial production requirements below —

(Example– Qty (2000) Date '*27-Mar-1985*')

1st Mth Prod Required Qty ( ) Date ' '
2nd Mth Prod Required Qty ( ) Date ' '
3rd Mth Prod Required Qty ( ) Date ' '

Comments —

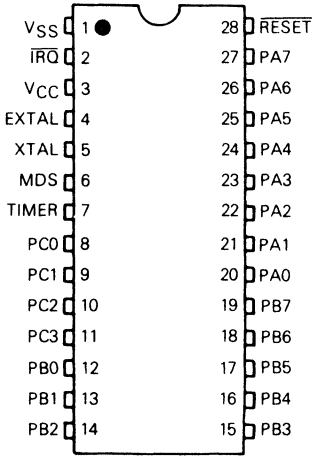
For Office Use Only Date ' ' (e.g. '*27-Mar-1985*))
Mask + Layer ' ' Mask Shop ' ' *Wafer Fab* ' '
Revision ( ) Revision Date '*27-Mar-1985*' PEA ☐

NB. All other options require prior factory approval

SECTION 9
MECHANICAL DATA

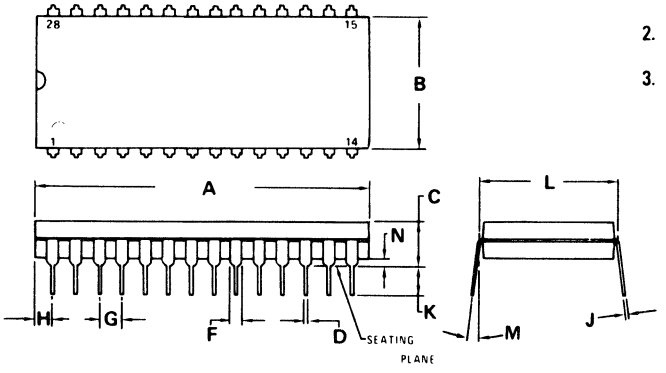
This section contains the pin assignment and package dimension diagrams for the MC68HC04P3 microcomputer.

9.1 PIN ASSIGNMENTS



9.2 PACKAGING DATA

P SUFFIX
PLASTIC 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	0^0	15^0	0^0	15^0
N	0.51	1.02	0.020	0.040

MOTOROLA SEMICONDUCTOR SALES OFFICES
EUROPE

AUSTRIA
Motorola Ges.m.b.H.
Alserbachstrasse 30 — A-1090 Wien
Tel. (0222) 31 65 45

DENMARK
Motorola Semiconductors A/S
Tornerosevej 127
2730 Herlev
Tel. 452 92 00 99

FRANCE
Motorola Semiconducteurs Commerciale S.A.
Main Sales Office
2, rue Auguste-Comte — Bp. 39
92173 Vanves-Cedex
Tel. 4736 01 99

Sales Office
Chemin de Malacher-Zirst
38240 Meylan (Grenoble)
Tel. (76) 90 22 81

Sales Office
Zone artisanale — 35740 Pace
Tel. (99) 60 65 48

Sales Office
Avenue Général-Eisenhower
31023 Toulouse Cedex
Tel. (61) 41 90 00

WEST GERMANY
Motorola GmbH.
Geschäftsbereich Halbleiter
Main Sales Office
Arabellastrasse 17
8000 München 81
Tel. (089) 92 720

Sales Office
Hans-Böckler-Strasse 30
3012 Langenhagen
Hanover
Tel. (0511) 78 99 11

Sales Office
Donaustrasse 43
8500 Nürnberg
Tel. (0911) 643044

Sales Office
Stralsunder-Strasse 1
7032 Sindelfingen
Tel. (0703) 18 30 74/75

Sales Office
Abraham-Lincoln-Strasse 28
6200 Wiesbaden
Tel. (06121) 76 19 21

HOLLAND
Motorola B.V.
Semiconductor Division
Maarssenbroeksedijk 37
3606 AG Maarssen
Tel. (030) 43 96 53

ITALY
Motorola S.p.A.
Divisione Semiconduttori
Main Sales Office
Centro Milanofiori-Strada 2-C2
20090 Assago (Milano)
Tel. (02) 822 01

Sales Office
Via Costantino Maes 68
00162 Roma
Tel. (06) 8322 186

NORWAY
Motorola A/S
Plogveien 1A, N-9679 Oslo 6
Tel. (02) 198070

SPAIN
Motorola España S.A.
Alberto Alcocer, 46 Dpdo
28016 Madrid
Tel. 457 82 04

SWEDEN
Motorola AB.
Dalvägen 2
S-17136 Solna
Tel. (08) 83 02 00

SWITZERLAND
Motorola (Schweiz) AG
Main Sales Office
Uitikonerstr.9
8952 Schlieren
Tel. (01) 730 40 74

Sales Office
16, chemin de la Voie-Creuse
P.O. Box 8
1211 Genève 20
Tel. (022) 99 11 11

UNITED KINGDOM
Motorola Ltd.
Main Sales Office
Motorola House
69 Buckingham Street
Aylesbury, Bucks, HP20 2NF
Tel. 0296 35252

Sales Office
Colvilles Road, Kelvin Estate
East Kilbride, Scotland G75 0TG
Tel. (03552) 39101

European Literature Centre
88 Tanners Drive, Blakelands
Milton Keynes MK14 5BP
Tel. (0908) 614614

HEADQUARTERS EUROPEAN OPERATIONS
Motorola Inc.
European Semiconductor Division
16, chemin de la Voie-Creuse
P.O. Box 8
12111 Genève 20
Switzerland
Tel. (022) 99 11 11

EUROPEAN FACTORIES

FRANCE
Motorola Semiconducteurs S.A.
Avenue Général-Eisenhower
31023 Toulouse CEDEX
Tel. (61) 41 90 00

GERMANY
Motorola GmbH
Münchner Strasse 18
8043 Unterföhring
Tel. (089) 92 481

UNITED KINGDOM
Motorola Limited
Colvilles Road, Kelvin Estate
East Kilbride, Scotland G75 0TG
Tel. (03552) 39101

YOUR LOCAL DISTRIBUTOR IS:

MOTOROLA LTD. *Semiconductor Products Group*
COLVILLES ROAD, KELVIN ESTATE, EAST KILBRIDE, GLASGOW, G75 0TG, SCOTLAND

Printed in Great Britain by Galliard (Printers) Ltd 6/86 2000