MC68705R3

Technical Summary
8-Bit EPROM Microcontroller Unit

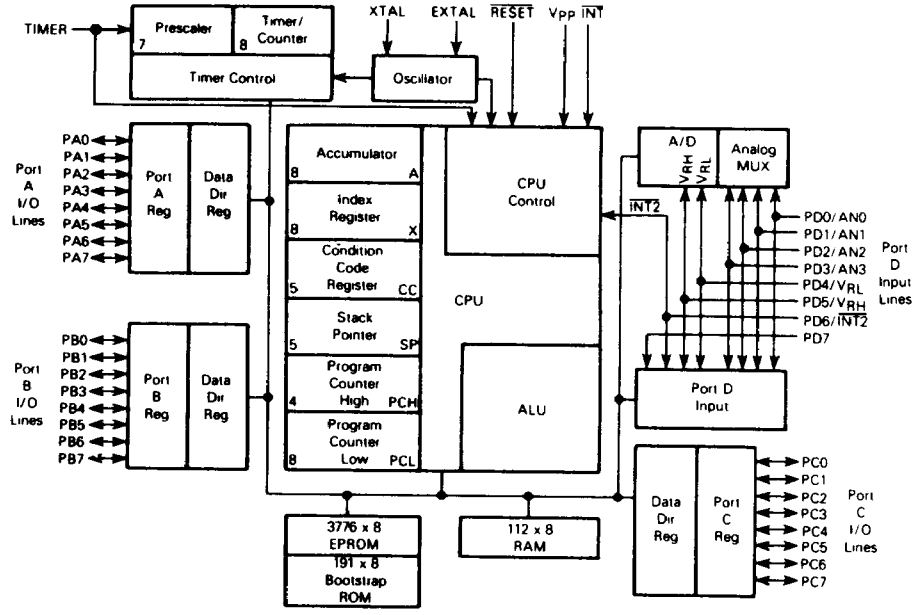
The MC68705R3 (HMOS) Microcontroller Unit (MCU) is an EPROM member of the MC6805 Family of microcontrollers. The user programmable EPROM allows program changes and lower volume applications. This low cost MCU has parallel I/O capability with pins programmable as input or output. This publication contains condensed information on the MCU; for detailed information, refer to *M6805 HMOS, M146805 CMOS Family User's Manual* (M6805UM(AD2)) or contact your local Motorola sales office.

Refer to the block diagram for the hardware features and to the list below for additional features available on the MCU.

- Internal 8-Bit Timer with 7-Bit Programmable Prescaler
- On-chip Oscillator
- Memory Mapped I/O
- Versatile Interrupt Handling
- Bit Manipulation
- Bit Test and Branch Instruction
- Vectored Interrupts
- Bootstrap Program in ROM
- 112 Bytes of RAM
- 3776 Bytes of Eprom
- 24 I/O Pins
- 4-Channel Analog-to-Digital Converter

3

BLOCK DIAGRAM



This document contains information on a new product. Specifications and information herein are subject to change without notice.

SIGNAL DESCRIPTION

V_{CC} AND V_{SS}

Power is supplied to the microcontroller using these two pins. V_{CC} is +5.25 volts (±0.5Δ) power, and V_{SS} is ground.

V_{PP}

This pin is used when programming the EPROM. In normal operation, this pin is connected to V_{CC}.

$\overline{INT}$

This pin provides the capability for asynchronously applying an external interrupt to the MCU. Refer to **INTERRUPTS** for more detailed information.

EXTAL, XTAL

These pins provide control input for the on-chip clock oscillator circuit. A crystal, a resistor/capacitor combination, or an external signal (depending on mask option register setting) is connected to these pins to provide a system clock.

RC Oscillator

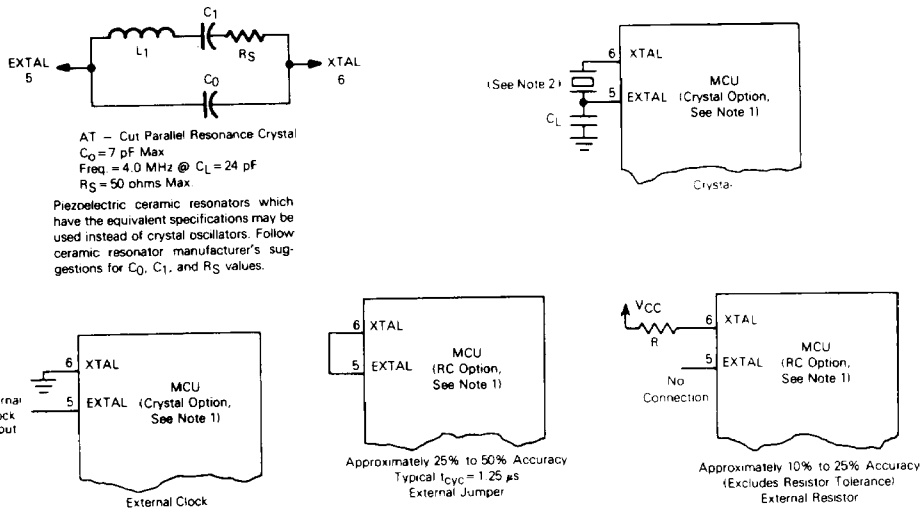
With this option, a resistor is connected to the oscillator pins as shown in Figure 1. The relationship between R and f_{osc} is shown in Figure 2.

Crystal

The circuit shown in Figure 1 is recommended when using a crystal. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and startup stabilization time. Refer to **ELECTRICAL SPECIFICATIONS** for V_{CC} specifications.

External Clock

An external clock should be applied to the EXTAL input with the XTAL input connected to V_{SS}, as shown in Figure 1. This option may only be used with the crystal oscillator option selected in the mask option register.



NOTES:

1. For the MC68705R3 MOR b7 = 0 for the crystal option and MOR b7 = 1 for the RC option. When the TIMER input pin is in the V_{IHTP} range (in the bootstrap EPROM programming mode), the crystal option is forced. When the TIMER input is at or below V_{CC}, the clock generator option is determined by bit 7 of the mask option register (CLK).
2. The recommended C_L value with a 4.0 MHz crystal is 27 pF maximum, including system distributed capacitance. There is an internal capacitance of approximately 25 pF on the XTAL pin. For crystal frequencies other than 4 MHz, the total capacitance on each pin should be scaled as the inverse of the frequency ratio. For example, with a 2 MHz crystal, use approximately 50 pF on EXTAL and approximately 25 pF on XTAL. The exact value depends on the motional-arm parameters of the crystal used.

Figure 1. Oscillator Connections

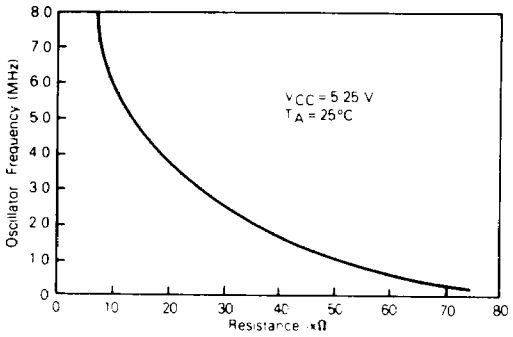


Figure 2. Typical Frequency vs Resistance for RC Oscillator Option Only

3

TIMER

This pin is used as an external input to control the internal timer-counter circuitry. This pin also detects a higher voltage level used to initiate the bootstrap program.

$\overline{\text{RESET}}$

This pin has a Schmitt trigger input and an on-chip pullup. The MCU can be reset by pulling $\overline{\text{RESET}}$ low.

INPUT/OUTPUT LINES (PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7)

These 32 lines are arranged into four 8-bit ports (A, B, C, and D). Ports A, B, and C are programmable as either inputs or outputs under software control of the data direction registers. Port D is a fixed input port. It has up to four analog inputs, plus two voltage reference inputs when the analog-to-digital converter is used (PD5/V_{RH}, PD4/V_{RL}), and an $\overline{\text{INT2}}$ input. Port D lines can be read directly and used as binary inputs. If an analog input is used, then the voltage reference pins must be used in the analog mode. Refer to **PROGRAMMING** for additional information.

PROGRAMMING

INPUT/OUTPUT PROGRAMMING

Ports A, B, and C are programmable as either input or output under software control of the corresponding data direction register (DDR). Port D lines are input only. The port I/O programming is accomplished by writing the corresponding bit in the port DDR to a logic one for output and a logic zero for input. On reset, all the DDRs are initialized to a logic zero state to put the ports in the input mode. The port output registers are not initialized on reset and should be written to before setting the DDR bits.

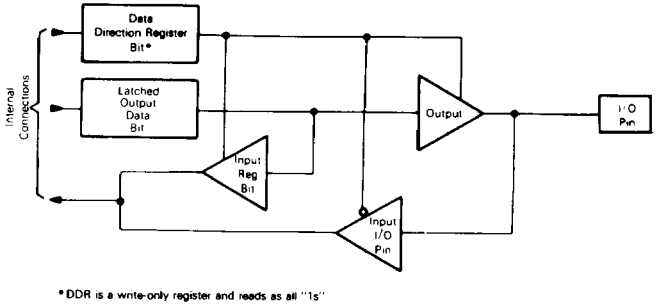
When programmed as outputs, the latched output data is readable as input data regardless of the logic levels at the output pin due to output loading. The latched output data bit may always be written. Therefore, any write to a port writes all of its data bits, even though the port DDR is set to input. This port write may be used to initialize the data registers and avoid undefined outputs. 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 (zero) and, also, to the latched output when the DDR is an output (one). Refer to Table 1 for I/O functions and to Figure 3 for typical port circuitry.

Table 1. I/O Pin Functions

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

**Port B and C are three-state ports. Port A has an internal pullup device to provide CMOS data drive capability.

Port D provides reference voltage and multiplexed analog inputs. The V_{RL} and V_{RH} lines are internally connected to the A/D resistor. Port D can always be used as



*DDR is a write-only register and reads as all "1s"

Figure 3. Typical Port I/O Circuitry and Register Configuration

digital inputs, but for analog inputs, V_{RH} and V_{RL} must be connected to the appropriate reference voltage.

NOTE

Read-modify-write instructions should be not used when writing to DDRs always read as 'one'.

MEMORY

The MCU is capable of addressing 4096 bytes of memory and I/O registers. The memory map is shown in Figure 4. The locations consists of user EPROM, bootstrap ROM, user RAM, a mask option register (MOR), a program control register, miscellaneous register, A/D control registers, and I/O. The interrupt vectors are located from $FF8 to $FFF. The bootstrap is a mask-programmed ROM that allows the MCU to program its own EPROM.

The stack area is used during processing of an interrupt or subroutine call to save the CPU state. The stack pointer decrements during pushes and increments during pulls. Refer to **INTERRUPTS** for additional information.

NOTE

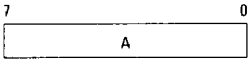
Using the stack area for data storage or temporary work locations requires care to prevent it from being overwritten due to stacking from an interrupt or subroutine call.

REGISTERS

The MCU contains the registers described in the following paragraphs.

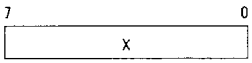
ACCUMULATOR (A)

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



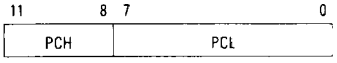
INDEX REGISTER (X)

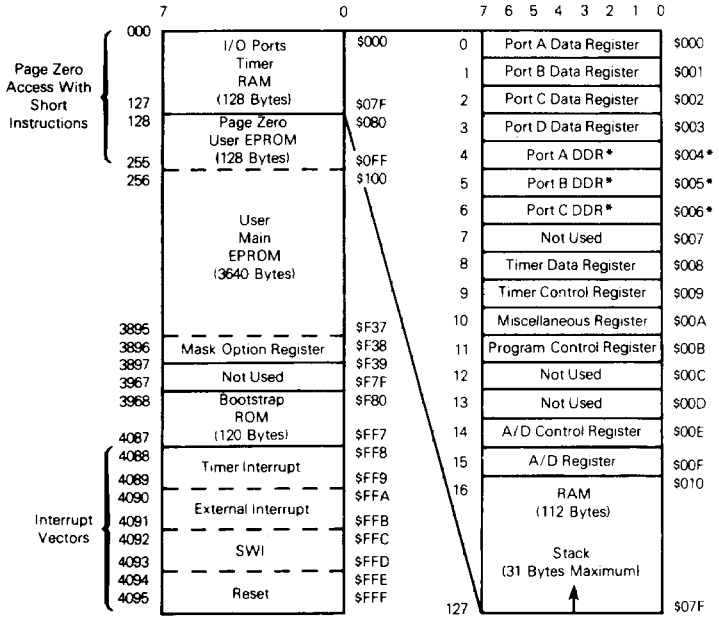
The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit value that may be added to an 8- or 16- bit immediate value to create an effective address. The index register may also be used as a temporary storage area.



PROGRAM COUNTER (PC)

The program counter is a 12-bit register that contains the address of the next byte to be fetched.





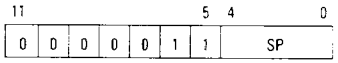
* Caution: Data direction registers (DDRs) are write-only; they read as $FF.

Figure 4. Memory Map

STACK POINTER (SP)

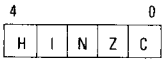
The stack pointer is a 12-bit register that contains the address of the next free location on the stack. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set at location $07F. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is pulled from the stack.

The seven most-significant bits of the stack pointer are permanently set at 0000011. Subroutines and interrupts may be nested down to location $061 (31 bytes maximum), which allows the programmer to use up to 15 levels of subroutine calls (less if interrupts are allowed).



CONDITION CODE REGISTER (CC)

The condition code register is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. These bits can be individually tested by a program, and specific actions can be taken as a result of their state. Each bit is explained in the following paragraphs.



Half Carry (H)

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

Interrupt (I)

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

Negative (N)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in the result is a logic 1).

Zero (Z)

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

Carry/Borrow (C)

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

RESETS

The MCU can be reset two ways: by initial power-up and by the external reset input ($\overline{\text{RESET}}$). The $\overline{\text{RESET}}$ input consists mainly of a Schmitt trigger that senses the line logic level.

POWER-ON-RESET (POR)

An internal reset is generated on power-up that allows the internal clock generator to stabilize. The power-on reset is used strictly for power turn-on conditions and should not be used to detect any drop in the power supply voltatge. A delay of t_{RHL} milliseconds is required before allowing $\overline{\text{RESET}}$ input to go high. Connecting a capacitor to the $\overline{\text{RESET}}$ input (Figure 5) typically provides sufficient delay.

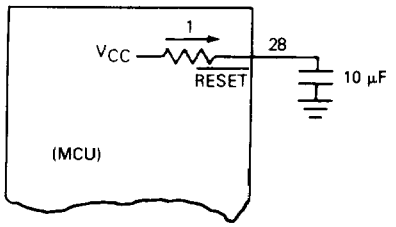


Figure 5. Power-Up $\overline{\text{RESET}}$ Delay Circuit

EXTERNAL $\overline{\text{RESET}}$ INPUT

The MCU is reset when a logic zero is applied to the $\overline{\text{RESET}}$ input for a period longer than one machine cycle (t_{cyc}). Under this type of reset, the Schmitt trigger switches off at V_{IRES-} to provide an internal reset voltage.

INTERRUPTS

The MCU can be interrupted four different ways: (1) through the external interrupt $\overline{\text{INT}}$ input pin, (2) with the internal timer interrupt request, (3) using the software interrupt instruction (SWI), or (4) the external Port D ($\overline{\text{INT2}}$) input pin.

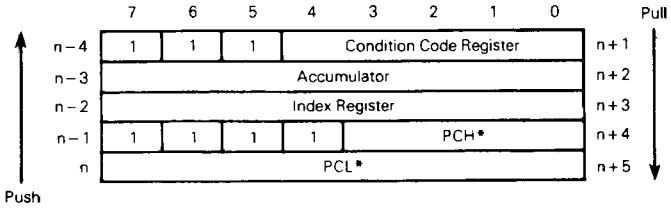
Interrupts cause the processor registers to be saved on the stack and the interrupt mask (I bit) set to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack after which normal processing resumes. The stacking order is shown in Figure 6.

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

NOTE

The current instruction is considered to be the one already fetched and being operated on.

When the current instruction is complete, the processor checks all pending hardward interrupts and, if unmasked (I bit clear), proceeds with interrupt processing; otherwise, the next instruction if fetched and executed. Masked interrupts are latched for later interrupt service. If the timer interrupt status bit is cleared before unmasking the interrupt, then the interrupt is not latched.



*For subroutine calls, only PCH and PCL are stacked

Figure 6. Interrupt Stacking Order

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed the same as any other instruction regardless of the setting of the I bit. Refer to Figure 7 for the reset and interrupt instruction processing sequence.

TIMER INTERRUPT

If the time mask bit (TCR6) is cleared, then, each time the timer decrements to zero (transitions from $01 to $00), an interrupt request is generated. The actual processor interrupt is generated only if the interrupt mask bit of the condition code register (CCR) is also cleared. When the interrupt is recognized, the current state of the machine is pushed onto the stack and the I bit in the CCR is set, masking further interrupts until the present one is serviced. The contents of the timer interrupt vector, containing the location of the timer interrupt service routine, is then loaded into the program counter. At the end of the timer interrupt service routine, the software normally executes an RTI instruction which restores the machine state and starts executing the interrupted program. The timer interrupt status bit can only be cleared by software.

EXTERNAL INTERRUPT

The external interrupt is internally synchronized and then latched on the falling edge of INT and INT2. Clearing the I bit enables the external interrupt. The INT2 interrupt has an interrupt request bit (bit 7) and a mask bit (bit 6) in the miscellaneous register (MR). The INT2 interrupt is inhibited when the mask bit is set. The INT2 is always read as a digital input on port D. The INT2 and timer interrupt request bits, if set, cause the MCU to process an interrupt when the condition code I bit is clear. The following paragraphs describe two typical external interrupt circuits.

Zero-Crossing Interrupt

A sinusoidal input signal (f_{INT} maximum) can be used to generate an external interrupt (see Figure 8a) for use as a zero-crossing detector (for negative transitions of the ac sinusoid). This type of circuit allows applications such as servicing time-of-day routines and engaging/disengaging ac power control devices. Off-chip, full-wave rectification provides an interrupt at every zero crossing of the ac signal and, thereby, provides a 2f clock.

Digital-Signal Interrupt

With this type of circuit (Figure 8b), the $\overline{INT}$ pin can be driven by a digital signal. The maximum frequency of a signal that can be recognized by the TIMER or $\overline{INT}$ pin logic is dependent on the parameter labeled t_{WL}, t_{WH}. Refer to **TIMER** for additional information.

SOFTWARE INTERRUPT (SWI)

The SWI is an executable instruction that is executed regardless of the state of the I bit in the CCR. The SWI execution is similar to the hardware interrupts.

MODES OF OPERATION

The MCU has two modes of operations. These modes are the normal and bootstrap. The following paragraphs describe the modes.

NORMAL MODE

This mode is a single-chip mode and is entered if the following conditions are met: (1) the $\overline{RESET}$ line is low, (2) the PC0 pin is within its normal operational range, and (3) the V_{PP} pin is connected to V_{CC}. The next rising edge of the $\overline{RESET}$ pin then causes the part to enter the normal mode.

BOOTSTRAP

The bootstrap mode is entered if the TIMER pin = 12 V. Refer to application note, *MC68705P3 R3 U3 8-Bit EPROM Microcomputer Programming Module* (AN-857 Rev.2).

TIMER

The MCU consists of an 8-bit software programmable counter driven by a 7-bit software programmable prescaler. The various timer sources are made via the timer control register (TCR) and/or the mask option register (MOR). The 8-bit counter may be loaded under program control and is decremented toward zero. When the timer reaches zero, the timer interrupt request bit (bit 7) in the timer control register (TCR) is set. Refer to Figure 9 for timer block diagram.

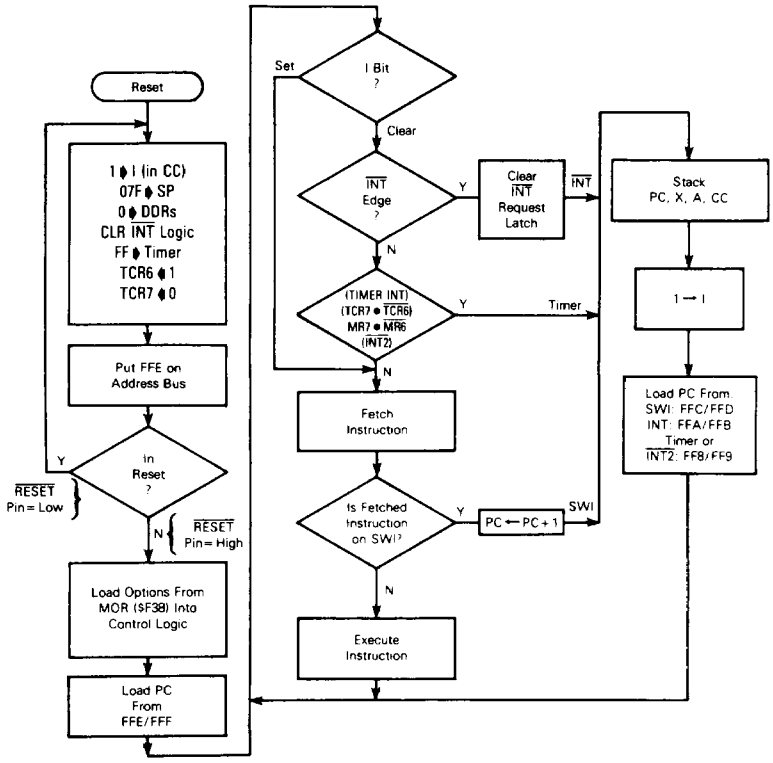


Figure 7. Reset and Interrupt Processing Flowchart

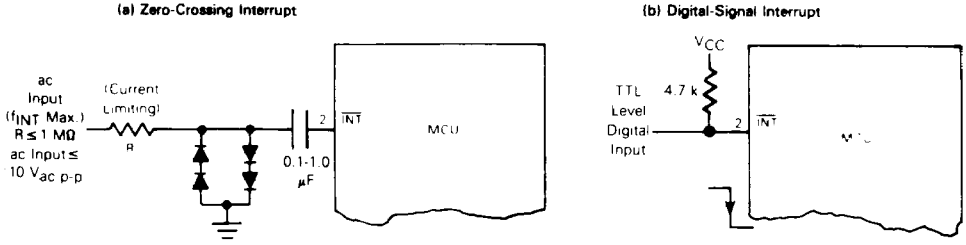
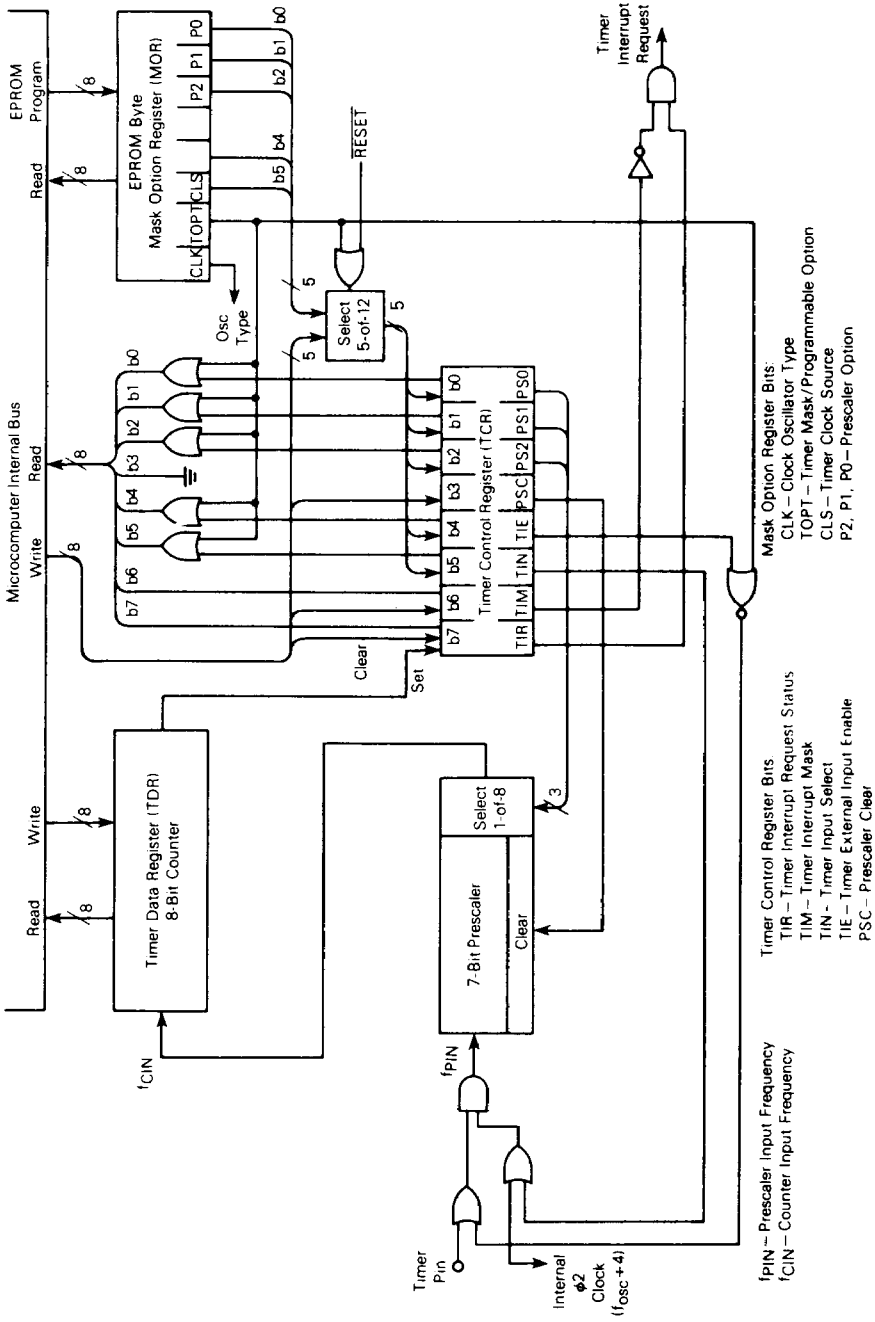


Figure 8. Typical Interrupt Circuits

The timer interrupt can be masked (disabled) by setting the timer interrupt mask bit (bit 6) in the TCR. When the I bit in the condition code register is cleared and the TCR bit 6 is cleared, the processor receives the interrupt. The MCU responds to this interrupt by (1) saving the present CPU state on the stack, (2) fetching the timer interrupt vector, and (3) executing the interrupt routine. The timer interrupt request bit must be cleared by software. Refer to **RESETS** and **INTERRUPTS** for additional information.

The prescaler is a 7-bit divider which is used to extend the maximum length of the timer. To avoid truncation errors, the prescaler is cleared when TCR bit 3 is set to a logic one; however, the TCR bit 3 always reads as a logic zero to ensure proper operation with read-modify-write instructions.



NOTE The TOPT bit in the mask option register selects whether the timer as software programmable via the timer control register or emulates the mask programmable parts via the MOR EPROM byte

Figure 9. Timer Block Diagram

3

The timer continues to count past zero, falling from $00 through $FF, and continues the countdown. The counter can be read at any time by reading the timer data register (TDR). This allows a program to determine the length of time since a timer interrupt has occurred without disturbing the counting process. The TDR is unaffected by reset.

SOFTWARE CONTROLLED MODE

This mode is selected when TOPT (bit 6) in the MOR is programmed to zero. The timer prescaler input can be configured for three different operating modes plus a disable mode, depending on the value written to TCR control bits 4 and 5 (TIE and TIN). The following paragraphs describe the different modes.

Timer Input Mode 1

When TIE and TIN are both programmed to zero, the timer input is from the internal clock (phase two) and the timer input pin is disabled. The internal clock mode can be used for periodic interrupt generation as well as a reference for frequency and event measurement.

Timer Input Mode 2

When TIE = 1 and TIN = 0, the internal clock and the timer input signals are ANDed to form the timer input. This mode can be used to measure external pulse widths. The active high, external pulse gates in the internal clock for the duration of the external pulse. The accuracy of the count is ± 1.

Timer Input Mode 3

When TIE = 0 and TIN = 1, no prescaler input frequency is applied to the prescaler and the timer is disabled.

Timer Input Mode 4

When TIE and TIN are both one, the timer input is from the external clock. The external clock can be used to count external events as well as to provide an external frequency for generating periodic interrupts. Frequency of external input must be $\leq f_{osc}/8$.

MOR CONTROLLED MODE

This mode is selected when TOPT (bit 6) in the MOR is programmed to logic one. The timer circuits are the same as described in **SOFTWARE CONTROLLED MODE**. The logic levels of TCR bits 0, 1, 2, and 5 are determined during EPROM programming by the same bits in the MOR. Therefore, bits 0, 1, 2, and 5 in the MOR control the prescaler division and the timer clock selection. TIE (bit 4) and PSC (bit 3) in the TCR are set to a logic one when in the MOR controlled mode. TIM (bit 6) and TIR (bit 7) are controlled by the counter and software.

TIMER CONTROL REGISTER (TCR) $009

This is an 8-bit register that controls various functions such as configuring operation mode, setting ratio of the prescaler, and generating timer interrupt request signal. All bits are read/write except bit 3. The configuration of the TCR is determined by the TOPT (bit 6) in the MOR.

When TOPT = 1, the TCR emulates the MC6805R2; when TOPT = 0, the TCR is controlled by software.

TCR with MOR TOPT = 1

7	6	5	4	3	2	1	0
TIR	TIM	*	1	PSC	*	*	*

TCR with MOR TOPT = 0

7	6	5	4	3	2	1	0
TIR	TIM	TIN	TIE	PSC	PS2	PS1	PS0
RESET:							
0	1	U	U	U	U	U	U

*The value of corresponding bits in MOR is written during $\overline{RESET}$ rising edge. These bits always read "one".

TIR — Timer Interrupt Request
Used to indicate the timer interrupt when it is logic one
1 = Set when the timer data register changes to zeros
0 = Cleared by external reset, power-on reset, or under program control

TIM — Timer Interrupt Mask
Used to inhibit the timer interrupt
1 = Interrupt inhibited
0 = Interrupt enabled

TIN — External or Internal
Selects input clock source
1 = External clock selected
0 = Internal clock selected ($f_{osc}/4$)

TIE — TIMER External Enable
Used to enable external TIMER pin. When TOPT = 1 TIE is always a logical "one".
1 = Enables external timer pin
0 = Disables external timer pin

PSC — Prescaler Clear
Write only bit. Writing a one to this bit resets the prescaler to zero. A read of this location always indicates a zero when TOPT = 0. When TOPT = 1, this bit will read a logical "one" and has no effect on the prescaler.

PS2, PS1, PS0 — Prescaler Clear
Decoded to select one of eight outputs of the prescaler

Prescaler

PS2	PS1	PS0	Divide By
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

NOTES

When changing the PS bits in software, the PSC bit should be written to a "one" in the same write cycle to clear the prescaler. Changing the PS bits without clearing the prescaler may cause prescaler truncation.

MASK OPTION REGISTER (MOR) $F38

The MOR is implemented in EPROM. This register contains all zeros prior to programming and is not affected by reset. The MOR bits are described in the following paragraphs.

7	6	5	4	3	2	1	0
CLK	TOPT	CLS			P2	P1	P0

CLK — Clock (oscillator type)
1 = Resistor Capacitor (RC)
0 = Crystal

TOPT — Timer Option
1 = MC6805R2 type timer prescaler. All bits except 6 and 7, of the TCR are invisible to the user. Bits 5, 2, 1, and 0 of the MOR determine the equivalent MC6805R2 mask options.
0 = All TCR bits are implemented as a software programmable timer. The state of MOR bits 5, 4, 2, 1, and 0 sets the initial values of their respective TCR bits.

CLS — Timer/Prescaler Clock Source
1 = External TIMER pin
0 = Internal clock

Bit 4
Not used if TOPT = 1. Sets the initial value of TIE in the TCR if TOPT = 0.
1 = Not used
0 = Sets initial value of TIE in the TCR

Bit 3
Not used

P2, P1, P0
The logical levels of these bits, when decoded, select one of eight outputs on the timer prescaler.

Prescaler

P2	P1	P0	Divide By
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

PROGRAMMING CONTROL REGISTER (PCR) $00B

The PCR is an 8-bit register which provides the necessary control bits to program the EPROM. The bootstrap program manipulates the PCR when programming so the user need not be concerned with PCR in most applications.

7	6	5	4	3	2	1	0
1	1	1	1	1	$\overline{VPON}$	$\overline{PGE}$	$\overline{PLE}$
RESET:							
U	U	U	U	U	U	1	1

$\overline{PLE}$ — Programming Latch Enable
Controls address and data being latched into the EPROM. Set during reset, but may be cleared anytime.
1 = Read EPROM
0 = Latch address and data on EPROM

PGE — Program Enable
Enables programming of EPROM. Must be set when changing the address and data. Set during reset.
1 = Inhibit EPROM programming
0 = Enable EPROM programming (if $\overline{PLE}$ is low)

$\overline{VPON}$ — V_{PP} On
A read-only bit that indicates high voltage at the V_{PP} pin. When set to "one", disconnects PGE and PLE from the chip.
1 = No high voltage on V_{PP} pin
0 = High voltage on V_{PP} pin

NOTE

$\overline{VPON}$ being "zero" does not indicate that the V_{PP} level is correct for programming. It is used as a safety interlock for the user in the normal operating mode.

$\overline{VPON}$	$\overline{PGE}$	$\overline{PLE}$	Programming Conditions
0	0	0	Programming mode (program EPROM byte)
1	0	0	$\overline{PGE}$ and $\overline{PLE}$ disabled from system
0	1	0	Programming disabled (latch address and data in EPROM)
1	1	0	$\overline{PGE}$ and $\overline{PLE}$ disabled from system
0	0	1	Invalid state; $\overline{PGE}$ 0 if $\overline{PLE}$ 0
1	0	1	Invalid state; $\overline{PGE}$ 0 if $\overline{PLE}$ 0
0	1	1	"High voltage" on V_{PP}
1	1	1	$\overline{PGE}$ and $\overline{PLE}$ disabled from system (operating mode)

EPROM PROGRAMMING

ERASING THE EPROM

The EPROM can be erased by exposure to high-intensity ultraviolet (UV) light with a wavelength of 2537 angstroms. The recommended integrated dose (UV intensity × exposure time) is 25Ws/cm^2. The lamps should be used without software filters, and the MCU should be positioned about one inch from the UV tubes. Ultraviolet erasure clears all bits of the MCU EPROM to the "zero" state. Data then can be entered by programming "ones" into the desired bit locations.

PROGRAMMING

The MCU bootstrap program can be used to program the MCU EPROM. The alternate vectoring used to implement the self-check is used to start execution of the bootstrap program.

A MCM2532 UV EPROM (other industry standard EPROMs may be used) must first be programmed with the same information that is to be transferred to the MCU EPROM. Refer to application note, *MC68705P3/R3/U3 8-bit EPROM Microcomputer Programming Module* (AN-857

Rev.2) for schematic diagrams and instructions on programming the MCU EPROM.

ANALOG-TO-DIGITAL CONVERTER

The chip resident 8-bit analog-to-digital (A D) converter uses a successive approximation technique as show in Figure 10. Four external analog inputs can be connected to the A/D via Port D. Four internal analog channels ($V_{RH} - V_{RL}$, $V_{RH} - V_{RL}/2$, $V_{RH} - V_{RL}/4$, and V_{RL}) may be selected for calibration. The accuracy of these internal channels may not meet the accuracy specifications of the external channels.

Multiplexer selection is controlled by the A/D control register (ACR) bits 0, 1, and 2. Refer to Table 2 for multiplexer selection. The ACR is shown in Figure 10. The converter uses 30 machine cycles to complete a conversion of a sampled analog input. When the conversion is complete, the digital value is placed in the A/D result register (ARR); the conversion flag is set; selected input is sampled again; and a new conversion begins. When ACR7 is cleared, the conversion in progress is aborted and the selected input, which is held internally, is sampled for five machine cycles.

The converter uses V_{RH} and V_{RL} as reference voltages. An input voltage equal to or greater than V_{RH} converts to $FF. An input voltage equal to or less than V_{RL}, but greater than V_{SS}, converts to $00. Maximum and minimum ratings must not be exceeded. Each analog input source should use V_{RH} as the supply voltage and should be referenced to V_{RL} for the ratiometric conversion. To maintain full accuracy of the A/D, three requirements should be followed: (1) V_{RH} should be equal to or less than V_{CC}, (2) V_{RL} should be equal to or greater than V_{SS} but less than maximum specifications, and (3) $V_{RH} - V_{RL}$ should be equal to or greater than 4 volts.

The A/D has a built-in 1/2 LSB offset intended to reduce the magnitude of the quantizing error to ± 1/2 LSB, rather than +0, −1 LSB with no offset. This implies that, ignoring errors, the transition point from $00 to $01 occurs at 1/2 LSB above V_{RL}. Similarly, the transition from $FE to $FF occurs 1-1/2 LSB below V_{RH}, ideally.

Table 2. A/D Input MUX Selection

A/D Control Register			Input Selected	A/D Output (Hex)		
ACR2	ACR1	ACR0		Min	Typ	Max
0	0	0	AN0			
0	0	1	AN1			
0	1	0	AN2			
0	1	1	AN3			
1	0	0	V_{RH}*	FE	FF	FF
1	0	1	V_{RL}*	00	00	01
1	1	0	$V_{RH/4}$*	3F	40	41
1	1	1	$V_{RH/2}$*	7F	80	81

*Internal (calibration) levels

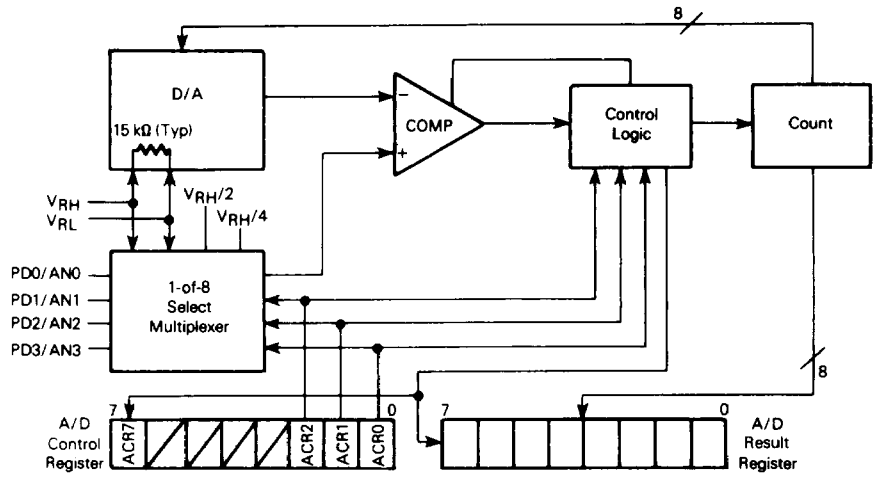


Figure 10. A/D Block Diagram

INSTRUCTION SET

The MCU has a set of 59 basic instructions which can be divided into five different types: register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type.

REGISTER/MEMORY INSTRUCTIONS

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

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

BIT MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any bit which resides in the first 256 bytes of the memory space where all port registers, port DDRs, timer, timer control, and on-chip RAM reside. An additional feature allows the software to test and branch on the state of any bit within these 256 locations. The bit set, bit clear and bit test, and branch functions are all implemented with a single instruction. For test and branch instructions, the value of the bit tested is also placed in the carry bit of the condition code register. Refer to the following list for bit manipulation instructions.

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

READ-MODIFY-WRITE INSTRUCTIONS

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

Function	Mnemonic
Increment	INC
Decrement	DEC
Clear	CLR
Complement	COM
Negate (2's Complement)	NEG
Rotate Left Thru Carry	ROL
Rotate Right Thru Carry	ROR
Logical Shift Left	LSL
Logical Shift Right	LSR
Arithmetic Shift Right	ASR
Test for Negative or Zero	TST

BRANCH INSTRUCTIONS

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

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

3

CONTROL INSTRUCTIONS

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

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

OPCODE MAP SUMMARY

Table 3 is an opcode map for the instructions used on the MCU.

ADDRESSING MODES

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

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

IMMEDIATE

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

DIRECT

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

EXTENDED

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

RELATIVE

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte (the offset) following the opcode is added to the PC if, and only if, the branch conditions are true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from −126 to +129 from the opcode address.

INDEXED, NO OFFSET

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

INDEXED, 8-BIT OFFSET

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

INDEXED, 16-BIT OFFSET

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

BIT SET/CLEAR

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

BIT TEST AND BRANCH

The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit to be tested, and its condition (set or clear), is included in the opcode. The address of the byte to be tested is in

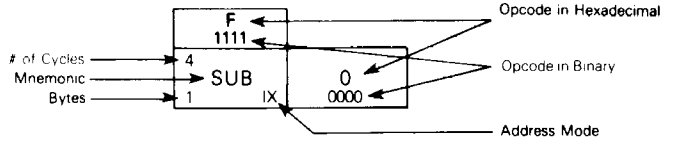
Table 3. Opcode Map

Hi / Low	Bit Manipulation BTB 0 0000	Bit Manipulation BSC 1 0001	Branch REL 2 0010	Read-Modify-Write DIR 3 0011	Read-Modify-Write INH 4 0100	Read-Modify-Write INH 5 0101	Read-Modify-Write IX1 6 0110	Read-Modify-Write IX 7 0111	Control INH 8 1000	Control INH 9 1001	Register/Memory IMM A 1010	Register/Memory DIR B 1011	Register/Memory EXT C 1100	Register/Memory IX2 D 1101	Register/Memory IX1 E 1110	Register/Memory IX F 1111	Hi / Low
0 0000	BRSET0	BSET0	BRA	NEG	NEG	NEG	NEG	NEG	RTI		SUB	SUB	SUB	SUB	SUB	SUB	0 0000
1 0001	BRCLR0	BCLR0	BRN						RTS		CMP	CMP	CMP	CMP	CMP	CMP	1 0001
2 0010	BRSET1	BSET1	BHI								SBC	SBC	SBC	SBC	SBC	SBC	2 0010
3 0011	BRCLR1	BCLR1	BLS	COM	COMA	COMX	COM	COM	SWI		CPX	CPX	CPX	CPX	CPX	CPX	3 0011
4 0100	BRSET2	BSET2	BCC	LSR	LSRA	LSRX	LSR	LSR			AND	AND	AND	AND	AND	AND	4 0100
5 0101	BRCLR2	BCLR2	BCS								BIT	BIT	BIT	BIT	BIT	BIT	5 0101
6 0110	BRSET3	BSET3	BNE	ROR	RORA	RORX	ROR	ROR			LDA	LDA	LDA	LDA	LDA	LDA	6 0110
7 0111	BRCLR3	BCLR3	BEQ	ASR	ASRA	ASRX	ASR	ASR		TAX		STA	STA	STA	STA	STA	7 0111
8 1000	BRSET4	BSET4	BHCC	LSL	LSLA	LSLX	LSL	LSL		CLC	EOR	EOR	EOR	EOR	EOR	EOR	8 1000
9 1001	BRCLR4	BCLR4	BHCS	ROL	ROLA	ROLX	ROL	ROL		SEC	ADC	ADC	ADC	ADC	ADC	ADC	9 1001
A 1010	BRSET5	BSET5	BPL	DEC	DECA	DECX	DEC	DEC		CLI	ORA	ORA	ORA	ORA	ORA	ORA	A 1010
B 1011	BRCLR5	BCLR5	BMI							SEI	ADD	ADD	ADD	ADD	ADD	ADD	B 1011
C 1100	BRSET6	BSET6	BMC	INC	INCA	INCX	INC	INC		RSP		JMP	JMP	JMP	JMP	JMP	C 1100
D 1101	BRCLR6	BCLR6	BMS	TST	TSTA	TSTX	TST	TST		NOP	BSR	JSR	JSR	JSR	JSR	JSR	D 1101
E 1110	BRSET7	BSET7	BIL								LDX	LDX	LDX	LDX	LDX	LDX	E 1110
F 1111	BRCLR7	BCLR7	BIH	CLR	CLRA	CLRX	CLR	CLR		TXA		STX	STX	STX	STX	STX	F 1111

Abbreviations for Address Modes

INH	Inherent
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND



the single byte immediately following the opcode byte. The signed relative 8-bit offset in the third byte is added to the PC if the specified bit is set or cleared in the specified memory location. This single three-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of memory. The span of branching is from −125 to +130 from the opcode address. The state of the tested bit is also transferred to the carry bit of the condition code register.

INHERENT

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

ELECTRICAL SPECIFICATIONS

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to −7.0	V
Input Voltage EPROM Programming Voltage (V_{PP} Pin) TIMER Pin — Normal Mode TIMER Pin — Bootstrap Programming Mode All Others	 V_{PP} V_{in} V_{in} V_{in}	 −0.3 to +22.0 −0.3 to −7.0 −0.3 to +15.0 −0.3 to +7.0	V
Operating Temperature Range MC68705R3 MC68705R3C	T_A	T_L to T_H 0 to +70 −40 to 85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C
Junction Temperature Plastic Cerdip	T_J	 150 175	°C W

These devices contain circuity to protect the inputs against damage due to high static voltages or electrical 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 $V_{SS} \leq (V_{in}$ and $V_{out}) \leq V_{CC}$. 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_{CC}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic (P Suffix) Plastic (FN Suffix) Cerdip (S Suffix)	θ_{JA}	 50 100 60	°C W

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 = $P_{INT} + P_{PORT}$
P_{INT} = $I_{CC} \times V_{CC}$, Watts — Chip Internal Power
P_{PORT} = Port Power Dissipation, Watts — User Determined

For most applications $P_{PORT} < P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is:

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

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

PROGRAMMING OPERATION ELECTRICAL CHARACTERISTICS

(V_{CC} = 5.25 Vdc ±0.5%, V_{SS} = 0, T_A = 20 to 30°C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Programming Voltage	V_{PP}	20.0	21.0	22.0	V
V_{PP} Supply Current V_{PP} = 5.25 V V_{PP} = 21.0 V	I_{PP}	 — —	 — —	 8 30	mA
Oscillator Frequency	$f_{osc\ p}$	0.9	1.0	1.1	MHz
Bootstrap Programming Mode Voltage (TIMER Pin) @ I_{IHTP} = 100 µA Maximum	V_{IHTP}	9.0	12.0	15.0	V

ELECTRICAL CHARACTERISTICS

(V_{CC} = 5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, T_A = 0° to 70°C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Input High Voltage $\overline{RESET}$ (4.75 ≤ V_{CC} ≤ 5.75) (V_{CC} < 4.75) $\overline{INT}$ 4.75 ≤ V_{CC} < 5.75) (V_{CC} < 4.75) All Other	V_{IH}	 4.0 V_{CC} - 0.5 4.0 V_{CC} - 0.5 2.0	 — — ** ** —	 V_{CC} V_{CC} V_{CC} V_{CC} V_{CC}	V
Input High Voltage (TIMER Pin) Timer Mode Bootstrap Programming Mode	V_{IH}	 2.0 9.0	 — 12.0	 V_{CC} - 1.0 15.0	V
Input Low Voltage $\overline{RESET}$ $\overline{INT}$ All Other	V_{IL}	 V_{SS} V_{SS} V_{SS}	 — ** —	 0.8 1.5 0.8	V
$\overline{INT}$ Zero-Crossing Input Voltage — Through a Capacitor	V_{INT}	2.0	—	4.0	$V_{ac\ p-p}$
Internal Power Dissipation (No Port Loading, V_{CC} = 5.25 V for Steady-State Operation) T_A = 0°C T_A = 40°C	P_{INT}	— —	520 580	740 800	mW
Input Capacitance EXTAL All Other (See Note)	C_{in}	 — —	 25 10	 — —	pF
$\overline{RESET}$ Hysteresis Voltage Out of Reset Voltage Into Reset Voltage	 V_{IRES+} V_{IRES-}	 2.1 0.8	 — —	 4.0 2.0	V
Programming Voltage (V_{PP} Pin) Programming EPROM Operating Voltage	V_{PP}*	 20.0 4.75	 21.0 V_{CC}	 22.0 5.75	V
Input Current TIMER (V_{in} = 0.4 V) $\overline{INT}$ (V_{in} = 0.4 V) EXTAL (V_{in} = 2.4 V to V_{CC}) (V_{in} = 0.4 V) $\overline{RESET}$ (V_{in} = 0.8 V) (External Capacitor Changing Current)	 I_{in} I_{RES}	 — — — — -4.0	 — 20 — — —	 20 50 10 1600 40	µA

*V_{PP} (pin 7) is connected to V_{CC} in the normal operating mode.
**Due to internal biasing, this input (when not used) floats to approximately 2.0 V.

NOTE: Port D analog inputs, when selected, C_{in} = 25 pF for the first 5 out of 30 cycles.

3

SWITCHING CHARACTERISTICS

(V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, T_A = 0°C to 70°C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Oscillator Frequency Normal	f_{osc}	0.4	—	4.2	MHz
Instruction Cycle Time ($4/f_{osc}$)	t_{cyc}	0.950	—	10	µs
$\overline{INT}$, $\overline{INT2}$, or Timer Pulse Width	t_{WL}, t_{WH}	t_{cyc} + 250	—	—	ns
$\overline{RESET}$ Pulse Width	t_{RWL}	t_{cyc} + 250	—	—	ns
$\overline{RESET}$ Delay Time (External Cap = 1.0 µF)	t_{RHL}	—	100	—	ms
$\overline{INT}$ Zero Crossing Detection Input Frequency	f_{INT}	0.03	—	1.0	kHz
External Clock Duty Cycle (EXTAL)	—	40	50	60	%
Crystal Oscillator Start-Up Time	—	—	—	100	ms

A/D CONVERTER CHARACTERISTICS

(V_{CC} = +5.25 V ±0.5 Vdc, V_{SS} = 0 Vdc, T_A = 0 C to 70 C, unless otherwise noted)

Characteristic	Min	Typ	Max	Unit	Comments
Resolution	8	8	8	Bits	
Non-Linearity	—	—	±1/2	LSB	For V_{RH} = 4.0 to 5.0 V and V_{RL} = 0 V.
Quantizing Error	—	—	±1/2	LSB	
Conversion Range	V_{RL}	—	V_{RH}	V	
V_{RH} V_{RL}	— V_{SS}	— —	V_{CC} 0.2	V V	A/D accuracy may decrease proportionately as V_{RH} is reduced be below 4.0 V. The sum of V_{RH} and V_{RL} must not exceed V_{CC}.
Conversion Time	30	30	30	t_{cyc}	Includes sampling time
Monotonicity	Inherent (within total error)				
Zero Input Reading	00	00	01	hexadecimal	V_{in} = 0
Ratiometric Reading	FE	FF	FF	hexadecimal	V_{in} = V_{RH}
Sample Time	5	5	5	t_{cyc}	
Sample/Hold Capacitance, Input	—	—	25	pF	
Analog Input Voltage	V_{RL}	—	V_{RH}	V	Negative transients on any analog lines (pins 19-24) are not allowed at any time during conversion.

PORT ELECTRICAL CHARACTERISTICS

(V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, T_A = 0° to 70°C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Port A					
Output Low Voltage, I_{Load} = 1.6 mA	V_{OL}	—	—	0.4	V
Output High Voltage, I_{Load} = −100 µA	V_{OH}	2.4	—	—	V
Output High Voltage, I_{Load} = −10 µA	V_{OH}	V_{CC} − 1.0	—	—	V
Input High Voltage, I_{Load} = −300 µA (Max)	V_{IH}	2.0	—	V_{CC}	V
Input Low Voltage, I_{Load} = −500 µA (Max)	V_{IL}	V_{SS}	—	0.8	V
Hi-Z State Input Current (V_{in} = 2.0 V to V_{CC})	I_{IH}	—	—	300	µA
Hi-Z State Input Current (V_{in} = 0.4 V)	I_{IL}	—	—	−500	µA
Port B					
Output Low Voltage, I_{Load} = 3.2 mA	V_{OL}	—	—	0.4	V
Output Low Voltage, I_{Load} = 10 mA (Sink)	V_{OL}	—	—	1.0	V
Output High Voltage, I_{Load} = −200 µA	V_{OH}	2.4	—	—	V
Darlington Current Drive (Source), V_O = 1.5 V	I_{OH}	−1.0	—	−10	mA
Input High Voltage	V_{IH}	2.0	—	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS}	—	0.8	V
Hi-Z State Input Current	I_{TSI}	—	±2	10	µA
Port C					
Output Low Voltage, I_{Load} = 1.6 mA	V_{OL}	—	—	0.4	V
Output High Voltage, I_{Load} = −100 µA	V_{OH}	2.4	—	—	V
Input High Voltage	V_{IH}	2.0	—	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS}	—	0.8	V
Hi-Z State Input Current	I_{TSI}	—	<2	10	µA
Port D (Input Only)					
Input High Voltage	V_{IH}	2.0	—	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS}	—	0.8	V
Input Current	I_{in}	—	<1	5	µA

3

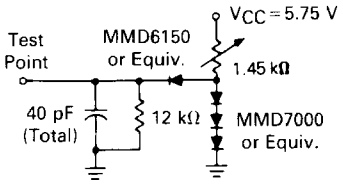


Figure 11. TTL Equivalent Test Load
(Port B)

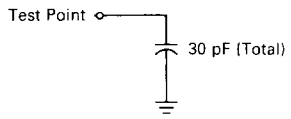


Figure 12. CMOS Equivalent Test Load
(Port A)

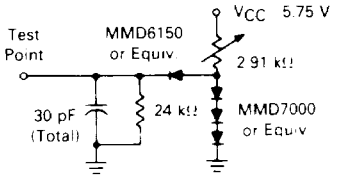


Figure 13. TTL Equivalent Test Load
(Ports A and C)

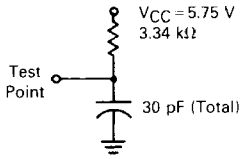


Figure 14. Open-Drain Equivalent Test Load
(Port C)

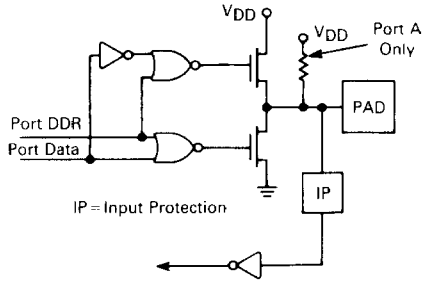


Figure 15. Ports A and C Logic Diagram

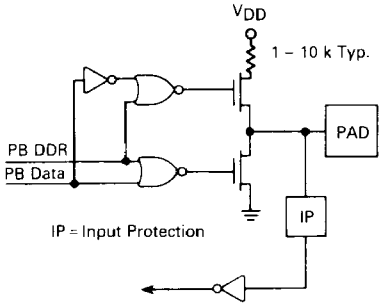
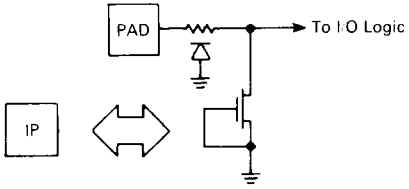


Figure 16. Port B Logic Diagram



Port 17. Typical Input Protection

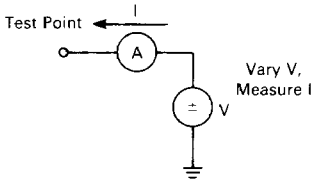


Figure 18. I/O Characteristic
Measurement Circuit

ORDERING INFORMATION

The following table provides generic information pertaining to the package type, temperature, and MC order numbers for the MC68705R3.

Table 4. Generic Information

Package Type	Temperature	Order Number
Cerdip S Suffix	0°C to 70°C −40° to −85°C	MC68705R3S MC68705R3CS
Plastic P Suffix	0°C to 70°C −40°C to 85°C	MC68705R3P MC68705R3CP
PLCC FN Suffix	−40°C to −85°C	MC68705R3CFN

MECHANICAL DATA

3

PIN ASSIGNMENTS