MOTOROLA

SEMICONDUCTORS

EAST KILBRIDE, SCOTLAND

704-611

MC68705P5

Advance Information

HMOS

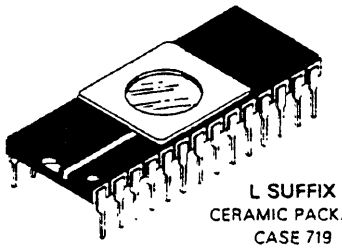
(HIGH-DENSITY, N-CHANNEL DEPLETION LOAD, 5 V EPROM PROCESS)

8-BIT EPROM MICROCOMPUTER

8-BIT EPROM MICROCOMPUTER UNIT

The MC68705P5 Microcomputer Unit (MCU) is an EPROM member of the M6805 Family of low-cost single-chip microcomputers. The user programmable EPROM allows program changes and lower volume applications in comparison to the factory mask programmable versions. The EPROM versions also reduce the development costs and turn-around time for prototype evaluation of the mask ROM versions. This 8-bit microcomputer contains a CPU, on-chip CLOCK, EPROM, bootstrap ROM, RAM, I/O, and a TIMER.

Because of these features, the MC68705P5 offers the user an economical means of designing an M6805 Family MCU into his system, either as a prototype evaluation, as a low-volume production run, or a pilot production run.



L SUFFIX
CERAMIC PACKAGE
CASE 719

S SUFFIX
CERDIP PACKAGE
ALSO AVAILABLE

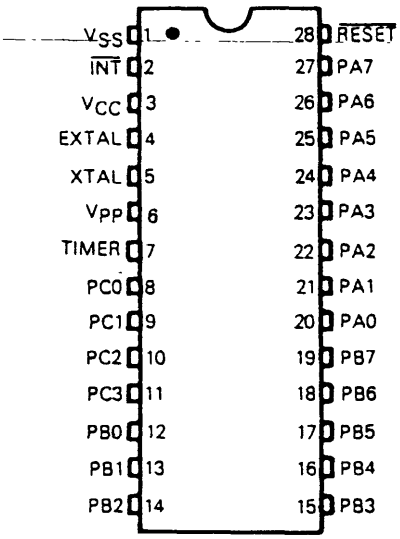
HARDWARE FEATURES

- 8-Bit Architecture
- 112 bytes of RAM
- Memory Mapped I/O
- 1804 Bytes of User EPROM
- Internal 8-Bit Timer with 7-Bit Prescaler
  - Programmable Prescaler
  - Programmable Timer Input Modes
  - External Timer Interrupt
- Vectored Interrupts — External, Timer, and Software
- Zero-Cross Detection on $\overline{INT}$ Input
- 20 TTL/CMOS Compatible Bidirectional I/O Lines (8 Lines are LED Compatible)
- On-Chip Clock Generator
- Master and Power-On Reset
- Complete Development System Support on EXORciser
- Emulates the MC6805P2 and MC6805P4 (Except for V_{SB})
- Bootstrap Program in ROM Simplifies EPROM Programming
- EPROM Security Features (Hardware and Software)

SOFTWARE FEATURES

- Similar to M6800 Family
- Byte Efficient Instruction Set
- Easy to Program
- True Bit Manipulation
- Bit Test and Branch Instructions
- Versatile Interrupt Handling
- Versatile Index Register
- Powerful Indexed Addressing for Tables
- Full Set of Conditional Branches
- Memory Usable as Registers/Flags
- Single Instruction Memory Examine/Change
- 10 Powerful Addressing Modes
- All Addressing Modes Apply to EPROM, RAM, and I/O

PIN ASSIGNMENT



ORDERING INFORMATION

(f = 1.0 MHz, T_A = 0 to 70°C)

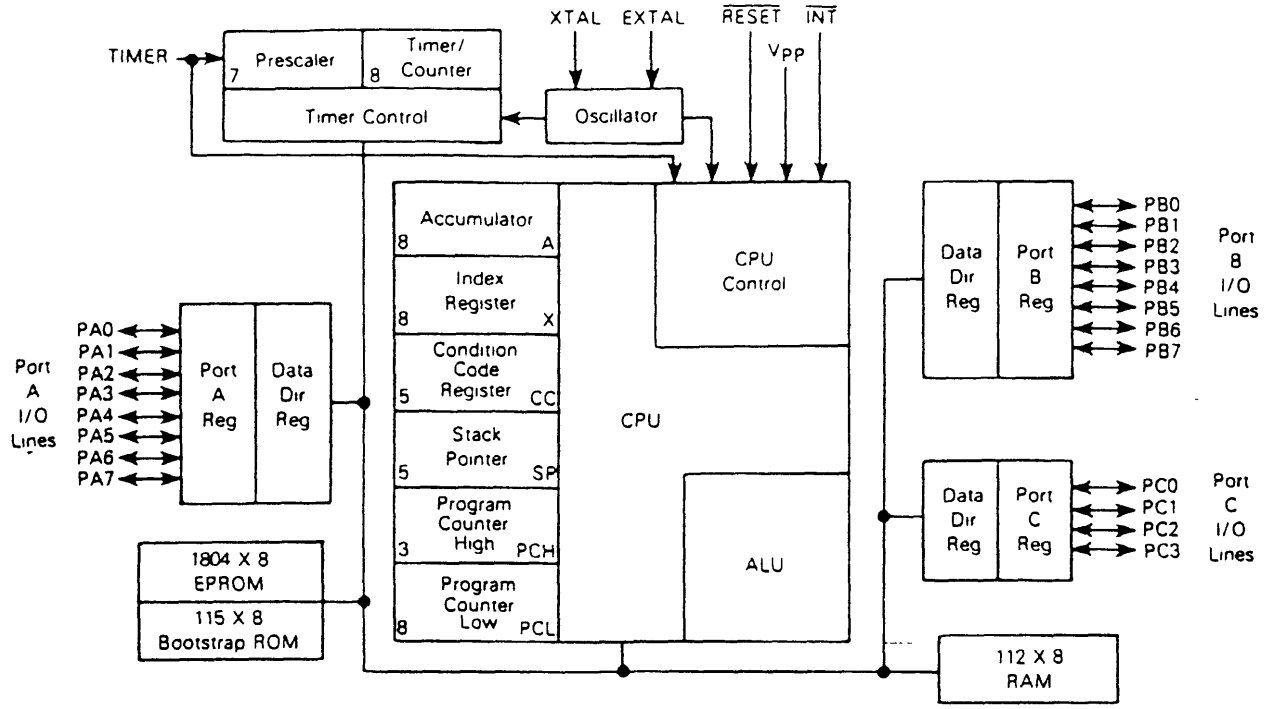
Package Type	Order Number
Ceramic L Suffix	MC68705P5L
Cerdip S Suffix	MC68705P5S

EXORciser is a registered trademark of Motorola Inc

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

©MOTOROLA INC., 1983

MC68706P5 HMOS MICROCOMPUTER BLOCK DIAGRAM



MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage			
EPROM Programming Voltage (Vpp Pin)	Vpp	−0.3 to +22.0	V
TIMER Pin			
Normal Mode	V_{in}	−0.3 to +7.0	V
Bootstrap Programming Mode	V_{in}	−0.3 to +15.0	V
All Others	V_{in}	−0.3 to +7.0	V
Operating Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C
Junction Temperature	T_J	+150	°C

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

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance			
Ceramic Package	θ_{JA}	50	°C/W

POWER CONSIDERATIONS

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

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

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:

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

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 Vdc, T_A = 20° to 30°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Programming Voltage (V_{PP} Pin)	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
Programming Oscillator Frequency	f_{oscp}	0.9	1.0	1.1	MHz
Bootstrap Programming Mode Voltage (TIMER Pin) I_{in} = 100 μA Max	V_{IHTP}	9.0	12.0	15.0	V

SWITCHING 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
Oscillator Frequency Normal	f_{osc}	0.4	–	4.2	MHz
Instruction Cycle Time ($4/f_{osc}$)	t_{cyc}	0.950	–	10	μs
$\overline{INT}$ or Timer Pulse Width (See Interrupt Section)	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) (See Figure 12)	–	40	50	60	%

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} 15.0	V
Input Low Voltage $\overline{RESET}$ $\overline{INT}$ All Other	V_{IL}	–0.3 –0.3 –0.3	– ** –	0.8 1.5 0.8	V
Internal Power Dissipation (No Port Loading, V_{CC} = 5.25 V, T_A = 0°C)	P_{INT}	–	450	TBD	mW
Input Capacitance XTAL All Other	C_{in}	– –	25 10	– –	pF
$\overline{INT}$ Zero-Crossing Voltage, through a Capacitor	V_{INT}	2.0	–	4.0	V_{acp-p}
$\overline{RESET}$ Hysteresis Voltage (See Figure 11) 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 Mode	V_{PP}*	20.0 4.0	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} Crystal Option) (V_{in} = 0.4 V Crystal Option) $\overline{RESET}$ (V_{in} = 0.8 V) (External Capacitor Changing Current)	I_{in}	– – – – –4.0	– 20 – – –	20 50 10 –1600 –40	μA

*V_{PP} is Pin 6 on the MC68705P5 and is connected to V_{CC} in the Normal Operating Mode. In the MC6805P2, Pin 6 is NUM and is connected to V_{SS} in the Normal Operating Mode. The user must allow for this difference when emulating the MC6805P2 ROM-based MCU.

**Due to internal biasing, this input when not used) floats to approximately 2.0 V.

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} + 0.7	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} + 0.7	V
Input Low Voltage	V_{IL}	V_{SS}	-	0.8	V
Hi-Z State Input Current	I_{TSI}	-	2	20	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} + 0.7	V
Input Low Voltage	V_{IL}	V_{SS}	-	0.8	V
Hi-Z State Input Current	I_{TSI}	-	2	20	µA

FIGURE 1 — TTL EQUIVALENT TEST LOAD (PORT B)

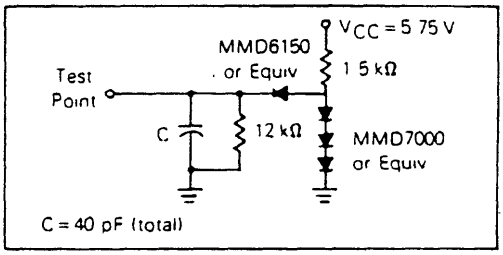


FIGURE 2 — CMOS EQUIVALENT TEST LOAD (PORT A)

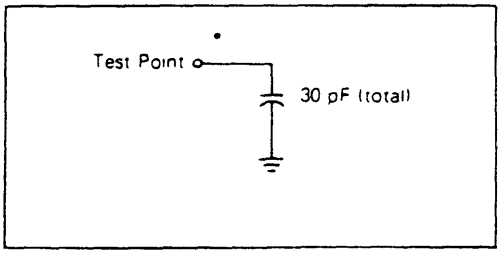
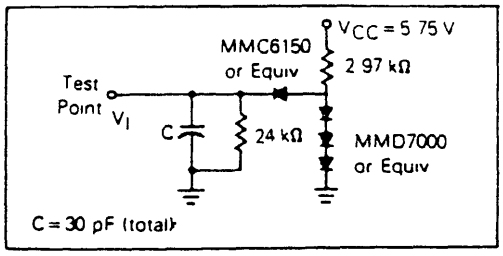


FIGURE 3 — TTL EQUIVALENT TEST LOAD (PORTS A AND C)



SIGNAL DESCRIPTION

The input and output signals for the MCU are described in the following paragraphs.

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

$\overline{INT}$ — This pin allows an external event to asynchronously interrupt the processor. It can also be used as a polled input using the BIL and BIH instructions. Refer to **INTERRUPTS** for additional information.

XTAL and EXTAL — These pins provide connections to the on-chip clock oscillator circuit. A crystal, a resistor, or an external signal, depending on the CLK bit (see **MASK OPTIONS**), is connected to these pins to provide a system clock source with various stability/cost tradeoffs. Lead lengths and stray capacitance on these two pins should be minimized. Refer to **INTERNAL CLOCK GENERATOR OPTIONS** for recommendations about these inputs.

TIMER — This is used as an external input to control the internal timer/circuitry. This pin also detects a higher voltage level used to initiate the bootstrap program (see **PROGRAMMING FIRMWARE**). Refer to **TIMER** for additional information about the timer circuitry.

$\overline{RESET}$ — This pin has a Schmitt Trigger input and an on-chip pullup. The MCU can be reset by pulling $\overline{RESET}$ low. Refer to **RESETS** for additional information.

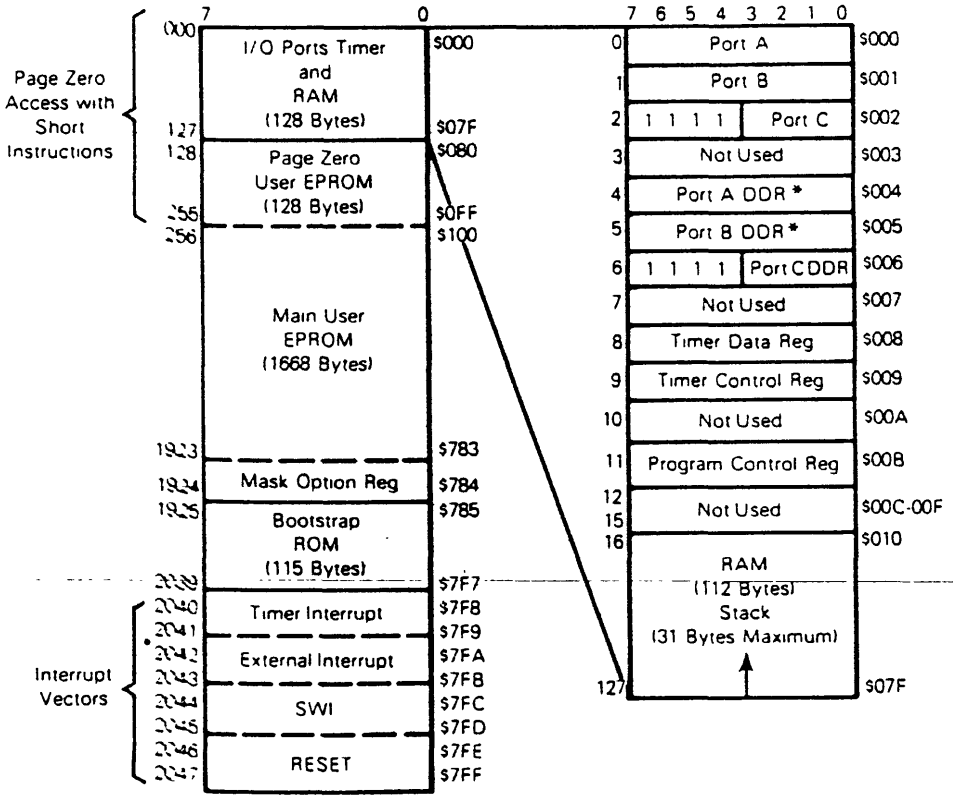
V_{PP} — This pin is used when programming the EPROM. By applying the programming voltage to this pin, one of the requirements is met for programming the EPROM. In normal operation, this pin is connected to V_{CC}. Refer to **PROGRAMMING FIRMWARE** and **ELECTRICAL CHARACTERISTICS.**

INPUT/OUTPUT LINES (PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7) — 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 (DDRs). Refer to **INPUT/OUTPUT** for additional information, being sure to observe the Caution.

MEMORY

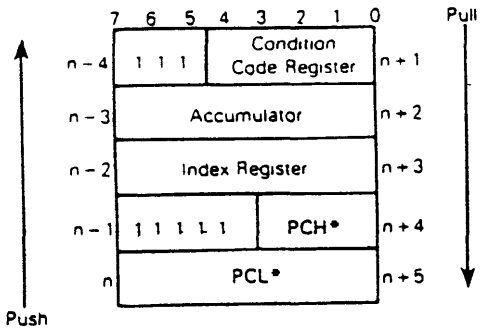
As shown in Figure 4, the MCU is capable of addressing 2048 bytes of memory and I/O registers with its program counter. The MC68705P5 MCU has implemented 2041 bytes of these locations. This consists of: 1804 bytes of user EPROM, 115 bytes of bootstrap ROM, 112 bytes of user RAM, an EPROM Mask Option Register (MOR), a Program Control Register (PCR), and eight bytes of I/O. The user EPROM is located in two areas. The main EPROM area is memory locations $080 to $783. The second area is reserved for eight interrupt/reset vector bytes at memory locations $7F8 to $7FF. The MCU uses nine of the lowest 16 memory locations for program control and I/O features such as ports, the port DDRs, and the timer. The Mask Option Register at memory location $784 completes the total. The 112 bytes of user RAM include up to 31 bytes for the stack.

FIGURE 4 — MC68705P5 MCU MEMORY CONFIGURATION



The stack area is used during the processing of interrupt and subroutine calls to save the processor state. The register contents are pushed onto the stack in the order shown in Figure 5. Because the stack pointer decrements during pushes, the low order byte (PCL) of the program counter is stacked first; then the high order three bits (PCH) are stacked. This ensures that the program counter is loaded correctly during pulls from the stack since the stack pointer increments during pulls. A subroutine call results in only the program counter (PCL, PCH) contents being pushed onto the stack; the remaining CPU registers are not pushed.

FIGURE 5 — INTERRUPT STACKING ORDER



*For subroutine calls, only PCH and PCL are stacked.

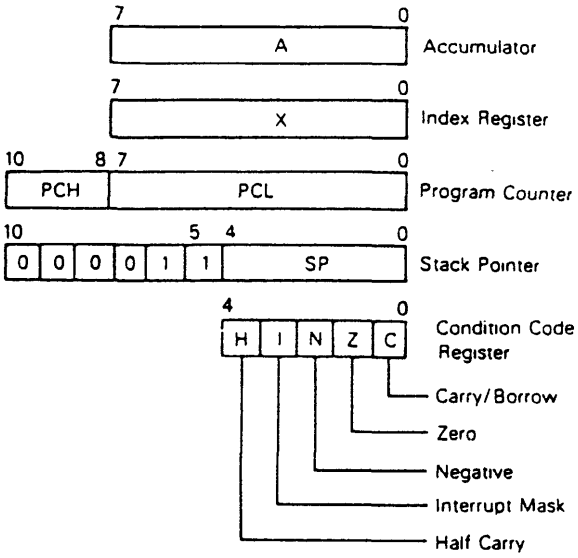
CENTRAL PROCESSING UNIT

The CPU of the M6805 Family is implemented independently from the I/O or memory configuration. Consequently, it can be treated as an independent central processor communicating with I/O and memory via internal address, data, and control buses.

REGISTERS

The M6805 Family CPU has five registers available to the programmer. They are shown in Figure 6 and are explained in the following paragraphs.

FIGURE 6 — PROGRAMMING MODEL



ACCUMULATOR (A) — The accumulator is a general purpose 8-bit register used to hold operands and results of the 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 instruction value to create an effective address. The index register can also be used for data manipulations using read-modify-write instructions. The index register may also be used as a temporary storage area.

PROGRAM COUNTER (PC) — The program counter is an 11-bit register that contains the address of the next instruction to be executed.

STACK POINTER (SP) — The stack pointer is an 11-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 to location $07F. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is then pulled from the stack. The six most significant bits of the stack pointer are permanently set to 000011. 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 action taken as a result of their state. Each of the five bits is explained below.

Half Carry (H) — 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 ($\overline{INT}$) are masked (disabled). If an 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 logical "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 logic unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions plus shifts and rotates.

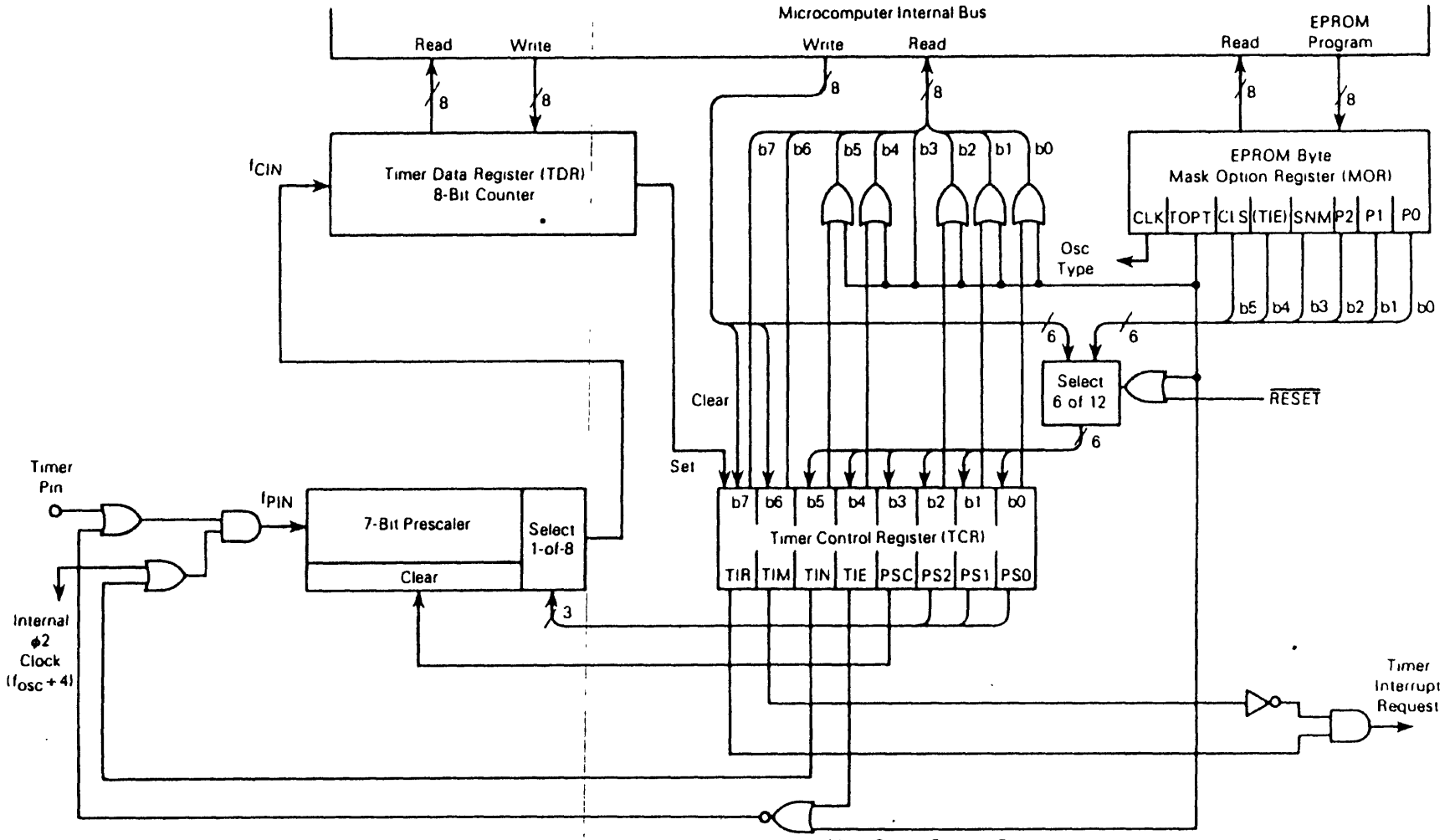
TIMER

The MC68705P5 MCU timer consists of an 8-bit software-programmable counter which is driven by a 7-bit software-programmable prescaler. Various timer clock sources may be selected ahead of the prescaler and counter. The timer selections are made via the Timer Control Register (TCR) and/or the Mask Option Register (MOR). The TCR also contains the interrupt control bits. The sections elsewhere entitled **TIMER CONTROL REGISTER** and **MASK OPTIONS** include additional details on controlling this timer.

The MCU timer circuitry is shown in Figure 7. The 8-bit counter may be loaded under program control and is decremented toward zero by the f_{CIN} counter input (output of the prescaler option selection). Once the 8-bit counter has decremented to zero, it sets the TIR (Timer Interrupt Request) bit 7 (b7 of TCR). The TIM (Timer Interrupt Mask) bit (b6) can be software set to inhibit the interrupt request, or software cleared to pass the interrupt request to the processor. When the I-bit in the Condition Code Register is cleared, the processor receives the Timer Interrupt. The MCU responds to this interrupt by saving the present CPU state on the stack, fetching the timer interrupt vector from locations $7F8 and $7F9 and executing the interrupt routine. The processor is sensitive to the level of the timer interrupt request line; therefore if the interrupt is masked, the TIR bit may be cleared by software (e.g., BCLR) without generating an interrupt. When servicing a timer interrupt, the TIR bit MUST be cleared by the timer interrupt service routine software in order to clear the timer interrupt request.

The counter continues to count (decrement) after falling through to $FF from zero. Thus, the counter can be read at any time by the processor without disturbing the count. This allows a program to determine the length of time since the occurrence of a timer interrupt and does not disturb the counting process.

FIGURE 7 — MC68705P5 TIMER FUNCTIONAL BLOCK DIAGRAM



f_{PIN} – Prescaler Input Frequency
f_{CIN} – Counter Input Frequency

Timer Control Register Bits:
- TIR – Timer Interrupt Request Status
- TIM – Timer Interrupt Mask
- TIN – Timer Input Select
- TIE – Timer External Input Enable
- PSC – Prescaler Clear
- PS2, PS1, PS0 – Prescaler Select

Mask Option Register Bits:
- CLK – Clock Oscillator Type
- TOPT – Timer Mask/Programmable Option
- CLS – Timer Clock Source
- (TIE) – (Timer External Input Enable)
- SNM – Secure/Non-Secure Mode Option
- P2, P1, P0 – Prescaler Option

NOTES: The TOPT bit in the mask option register selects whether the timer is software programmable via the timer control register or emulates the mask programmable parts via the MOR PROM byte.

The TIE bit in the mask option register is not used if MOR TOPT = 1 (MC6805P2 emulation). It sets the intial value of TCR TIE if MOR TOPT = 0.

The internal circuit connected to the $\overline{\text{RESET}}$ pin consists of a Schmitt trigger which senses the $\overline{\text{RESET}}$ line logic level. The Schmitt trigger provides an internal reset voltage when it senses logical "0" on the $\overline{\text{RESET}}$ pin. During power-up, the Schmitt trigger switches on (removes reset) when the $\overline{\text{RESET}}$ pin voltage rises to V_{IRES+}. When the $\overline{\text{RESET}}$ pin voltage falls to a logical "0" for a period longer than one t_{cyc}, the Schmitt trigger switches off to provide an internal reset voltage. The "switch off" voltage occurs at V_{IRES-}. A typical reset Schmitt trigger hysteresis curve is shown in Figure 9. See Figure 13 under **INTERRUPTS** for the complete reset sequence.

INTERNAL CLOCK GENERATOR OPTIONS

The internal clock generator circuit is designed to require a minimum of external components. A crystal, a resistor, a jumper wire, or an external signal may be used to generate a system clock with various stability/cost tradeoffs. The Mask Option Register (EPROM) is programmed to select crystal or resistor operation. The oscillator frequency is internally divided by four to produce the internal system clocks.

The different connection methods are shown in Figure 10.

FIGURE 9 — TYPICAL RESET SCHMITT TRIGGER HYSTERESIS

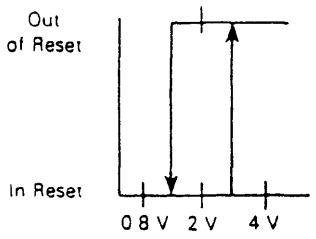
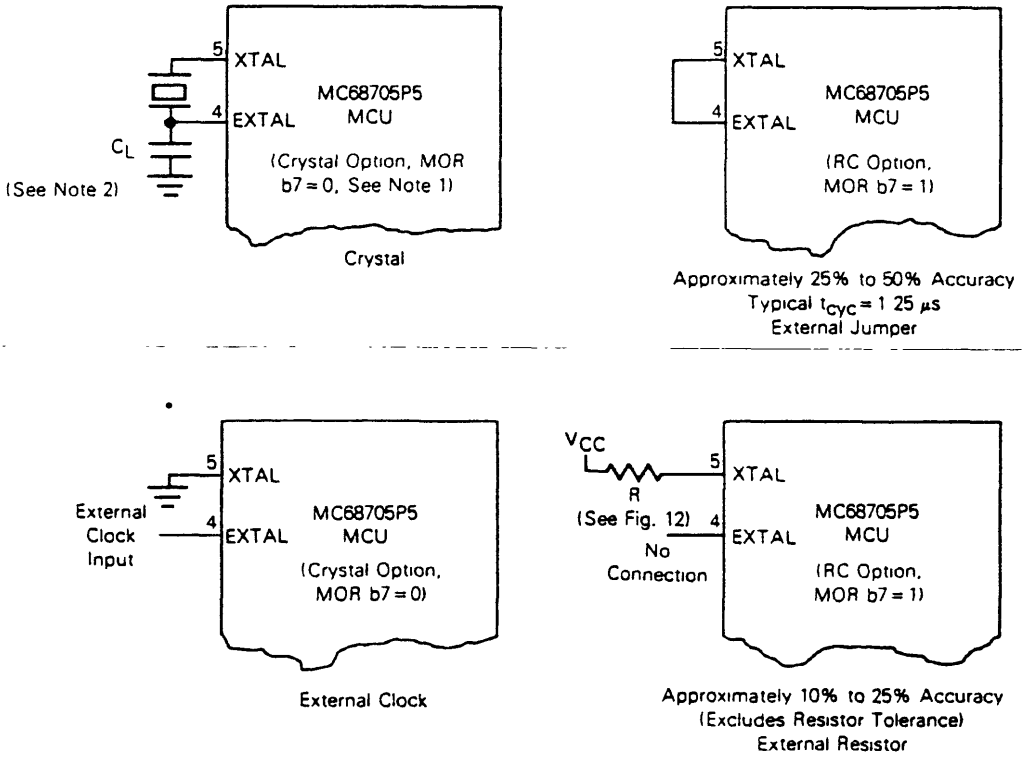


FIGURE 10 — CLOCK GENERATOR OPTIONS



NOTES:

1. 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.

The clock input to the timer can be from an external source (decrementing the counter occurs on a positive transition of the external source) applied to the TIMER input pin, or it can be the internal $\phi 2$ signal. The maximum frequency of a signal that can be recognized by the TIMER pin logic is dependent on the parameter labeled t_{WL}, t_{WH}. The pin logic that recognizes the high state on the pin must also recognize the low state on the pin in order to "re-arm" the internal logic. Therefore, the period can be calculated as follows (assumes 50/50 duty cycle for a given period):

$$t_{cyc} \times 2 + 250\ ns = period = \frac{1}{freq}$$

The period is not simply $t_{WL} + t_{WH}$. This computation is allowable, but it does reduce the maximum allowable frequency by defining an unnecessarily longer period (250 ns twice).

When the $\phi 2$ signal is used as the source, it can be gated by an input applied to the TIMER pin allowing the user to easily perform pulse-width measurements. (Note: When the MOR TOPT bit is set and the CLS bit is clear, an ungated $\phi 2$ clock input is obtained by tying the TIMER pin to V_{CC}.) The source of the clock input is selected via the TCR or the MOR as described later.

A prescaler option can be applied to the clock input that extends the timing interval up to a maximum of 128 counts before decrementing the counter. This prescaling option selects one of eight outputs on the 7-bit binary divider; one output bypasses prescaling. To avoid truncation errors, the prescaler is cleared when bit b3 of the TCR is written to a logic "1", when in the software controlled mode (TOPT = 0, more on these modes in later paragraphs), however, TCR bit b3 reads as a logic "0" when TOPT = 0 and as a "1" when TOPT = 1 to ensure proper operation with read-modify-write instructions (bit set and clear, for example).

At Reset, the prescaler and counter are initialized to an all "1s" condition, the Timer Interrupt Request bit (TCR, b7) is cleared and the Timer Interrupt Request mask (TCR, b6) is set. TCR bits b0, b1, b2, b4, and b5 are initialized by the corresponding Mask Option Register (MOR) bits at Reset. They are then software selectable after Reset if TOPT = 0.

Note that the timer block diagram in Figure 7 reflects two separate timer control configurations, a) software controlled mode via the Timer Control Register (TCR), and b) MOR controlled mode to emulate a mask ROM version with the Mask Option Register. In the software controlled mode, all TCR bits are read/write, except bit b3 which is write-only (always reads as a logic "0"). In the MOR controlled mode, TCR bits b7 and b6 are read/write, the other six have no effect on a write and read as logic "1s". The two configurations provide the user with the capability to freely select timer options as well as accurately emulate the MC6805P2 and MC6805P4 mask ROM version. In the following paragraphs refer to Figure 9 as well as the TIMER CONTROL REGISTER and MASK OPTIONS sections.

The TOPT (Timer Option) bit (b6) in the Mask Option Register is EPROM programmed to a logical "0" to select the software controlled mode, which is described first. TCR bits b5, b4, b3, b2, b1, and b0 give the program direct control of the prescaler and input selection options.

The Timer Prescaler input (f_{PIN}) can be configured for three different operating modes, plus a disable mode, depending upon the value written to TCR control bits b4 and b5 (TIE and TIN).

When the TIE and TIN bits are programmed to "0", the timer input is from the internal clock ($\phi 2$) and 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.

When TIE = 1 and TIN = 0, the internal clock and the TIMER input pin signals are ANDed to form the timer input f_{PIN}. This mode can be used to measure external pulse widths. The external pulse simply gates in the internal clock for the duration of the pulse. The accuracy of the count in this mode is ± one count.

When TIE = 0 and TIN = 1, no f_{PIN} input is applied to the prescaler and the timer is disabled.

When TIE and TIN are both programmed to a "1", the timer is from the external clock. The external clock can be used to count external events as well as provide an external frequency for generating periodic interrupts.

Bits b0, b1, and b2 in the TCR are program controlled to choose the appropriate prescaler output. The prescaling divides the f_{PIN} frequency by 1, 2, 4, etc. in binary multiples to 128 producing f_{CIN} frequency to the counter. The processor cannot write into or read from the prescaler, however, the prescaler is set to all "1s" by writing b3 of TCR to a "1", which allows for truncation-free counting.

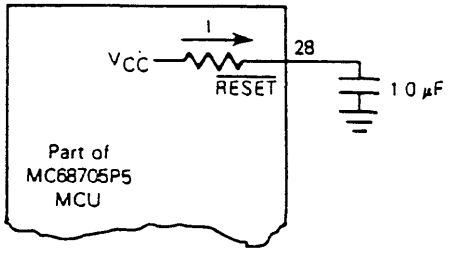
The MOR controlled mode of the timer is selected when the TOPT (Timer Option) bit (b6) in the MOR is programmed to a logical "1" to emulate the mask-programmable prescaler of the MC6805P2 and MC6805P4. The timer circuits are the same as described above; however, the Timer Control Register (TCR) is configured differently, as discussed below.

The logical level for the functions of bits b0, b1, b2, and b5 in the TCR are all determined at the time of EPROM programming. They are controlled by corresponding bits within the Mask Option Register (MOR, $784). The value programmed into MOR bits b0, b1, b2, and b5 controls the prescaler division and the timer clock selection. Bit b4 (TIE) and b3 (PSC) are set to a logical "1" in the MOR controlled mode. (When read by software, these six TCR bits always read as logical "1s".) As in the software programmable configuration, the TIM (b6) and TIR (b7) bits of the TCR are controlled by the counter and software as described above and in **TIMER CONTROL REGISTER.** The MOR controlled mode is designed to exactly emulate the MC6805P2 and MC6805P4 which has only TIM and TIR in the TCR and have the prescaler options defined as manufacturing mask options.

RESETS

The MCU can be reset in two ways: by initial power-up and by the external reset input ($\overline{RESET}$). Upon power-up, a delay of t_{RHL} is needed before allowing the $\overline{RESET}$ input to go high. This time allows the internal clock generator to stabilize. Connecting a capacitor to the $\overline{RESET}$ input, as shown in Figure 8, typically provides sufficient delay.

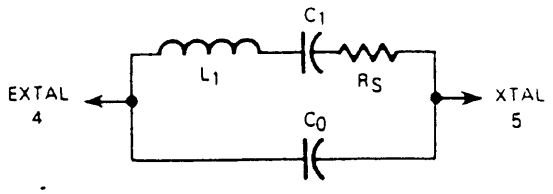
FIGURE 8 — POWER-UP RESET DELAY CIRCUIT



Crystal specifications and suggested PC board layouts are given in Figure 11. A resistor selection graph is given in Figure 12.

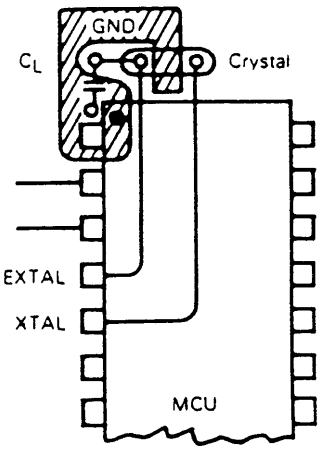
FIGURE 11 — CRYSTAL MOTIONAL-ARM PARAMETERS AND SUGGESTED PC BOARD LAYOUT

(a)

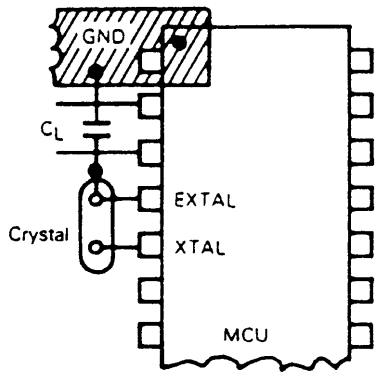


AT — Cut Parallel Resonance Crystal
C_O = 7 pF Max
Freq = 4.0 MHz @ C_L = 27 pF
R_S = 50 ohms Max

(b)

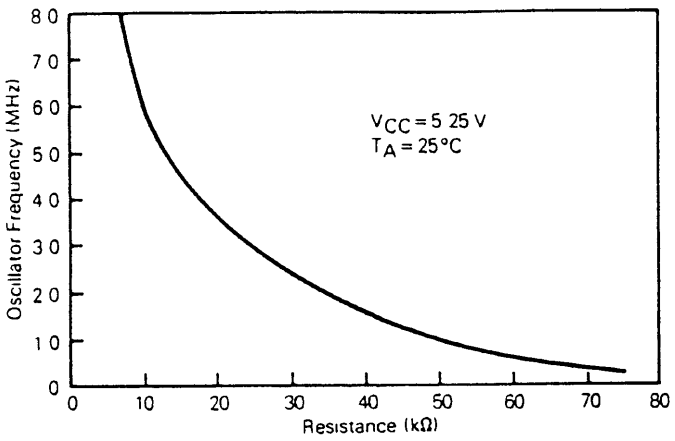


(c)



NOTE: Keep crystal leads and circuit connections as short as possible.

FIGURE 12 — TYPICAL FREQUENCY SELECTION FOR RESISTOR OSCILLATOR OPTION



The crystal oscillator start-up time is a function of many variables crystal parameters (especially R_S), oscillator load capacitances, IC parameters, ambient temperature, and supply voltage. To ensure rapid oscillator start-up neither the crystal characteristics nor the load capacitances should exceed recommendations.

BOOTSTRAP ROM

The bootstrap ROM contains a factory program which allows the MCU to fetch data from an external device and transfer it into the MC68705P5 EPROM. The bootstrap program provides: timing of programming pulses, timing of V_{PP} input, and verification after programming. See **PROGRAMMING FIRMWARE.**

MASK OPTION REGISTER (MOR)

The Mask Option Register is an 8-bit user programmed (EPROM) register. Bits in this register are used to select the type of system clock, the timer option, the timer/prescaler clock source, the prescaler option, and the secure mode. It is fully described in the **MASK OPTIONS.**

INTERRUPTS

The MC68705P5 MCU can be interrupted three different ways: through the external interrupt ($\overline{INT}$) input pin, the internal timer interrupt request, or the software interrupt instruction (SWI). When any interrupt occurs: the current instruction (including SWI) is completed, processing is suspended, the present CPU state is pushed onto the stack, the interrupt bit (I) in the Condition Code Register is set, the address of the interrupt routine is obtained from the appropriate interrupt vector address, and the interrupt routine is executed. Stacking the CPU registers, setting the I-bit, and vector fetching requires a total of 11 t_{cyc} periods for completion. A flowchart of the interrupt sequence is shown in

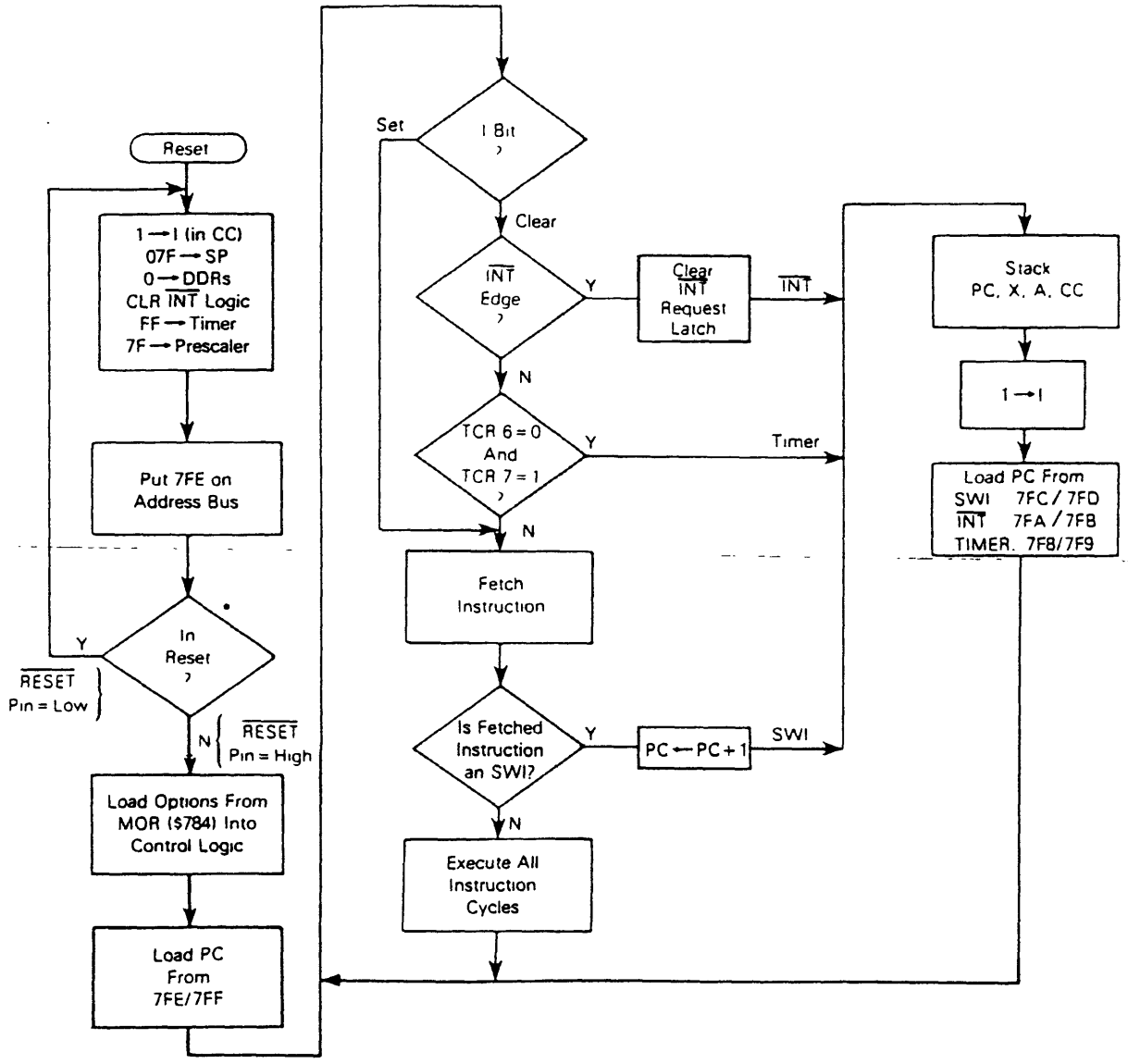
Figure 13. The interrupt service routine must end with a return from interrupt (RTI) instruction which allows the CPU to resume processing of the program prior to the interrupt (by unstacking the previous CPU state). Unlike $\overline{\text{RESET}}$, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction execution is complete.

When the current instruction is complete, the processor checks all pending hardware interrupts and, if unmasked, proceeds with interrupt processing, otherwise, the next instruction is fetched and executed. Note that masked interrupts are latched for later interrupt service.

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 as any other instruction.

The external interrupt is internally synchronized and then latched on the falling edge of $\overline{\text{INT}}$. A sinusoidal input signal

FIGURE 13 — RESET AND INTERRUPT PROCESSING FLOWCHART



(f_{INT} maximum) can be used to generate an external interrupt, as shown in Figure 14(a), for use as a Zero-Crossing Detector (for negative transitions of AC sinusoid). This 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.

For digital applications, 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}. The pin logic that recognizes the high (or low) state on the pin must also recognize the low (or high) state on the pin in order to "re-arm" the internal logic. Therefore, the period can be calculated as follows: (assumes 50/50 duty cycle for a given period)

$$t_{cyc} \times 2 + 250\ ns = period = \frac{1}{freq}$$

The period is not simply $t_{WL} + t_{WH}$. This computation is allowable, but it does reduce the maximum allowable frequency by defining an unnecessarily long period (250 ns twice). See Figure 14(b). For the $\overline{INT}$ function, the maximum allowable frequency is also determined by the software response of the $\overline{INT}$ service routine.

A software interrupt (SWI) is an executable instruction which is executed regardless of the state of the I-bit in the Condition Code Register. SWIs are usually used as breakpoints for debugging or as system calls.

INPUT/OUTPUT

There are 20 input/output pins. The $\overline{INT}$ pin may be polled with branch instructions to provide an additional input pin. All pins on Ports 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 "1" for output or a logic "0" for input. On Reset all the DDRs are initialized to a logic "0" state, placing the ports in the input mode. The port output registers are not initialized on Reset but may be written to before setting 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 15. When Port B is programmed for outputs, it is capable of sinking 10 mA and sourcing 1 mA on each pin.

FIGURE 14 — TYPICAL INTERRUPT CIRCUITS

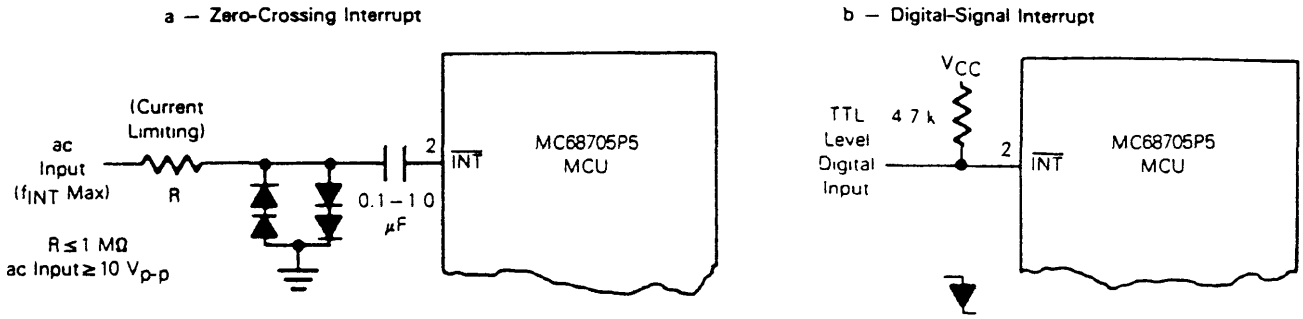
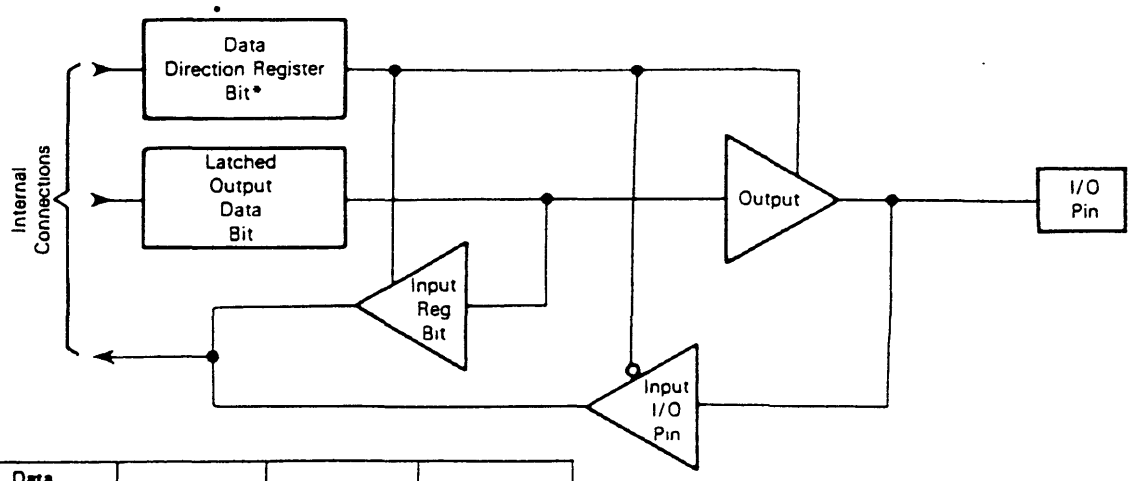


FIGURE 15 — TYPICAL PORT I/O CIRCUITRY



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

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

**Ports B and C are three-state ports. Port A has internal pullup devices to provide CMOS drive capability. See Electrical Characteristics tables for complete information.

All input/output lines are TTL compatible as both inputs and outputs. Port A lines are CMOS compatible as outputs while port B and C lines are CMOS compatible as inputs. The memory map in Figure 4 gives the address of data registers and DDRs. The Register configuration is provided in Figure 16. Figure 17 provides some examples of port connections.

Caution

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, and $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write functions they cannot be used to set or clear a single DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port must be written using a single-store instruction.

The latched output data bit (see Figure 15) 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 may be used to initialize 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").

FIGURE 16 — MCU REGISTER CONFIGURATION

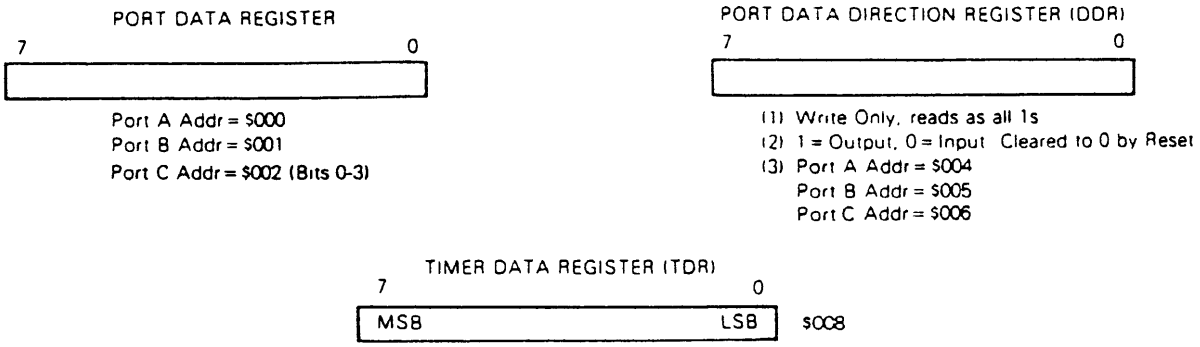
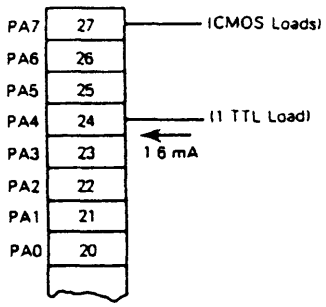
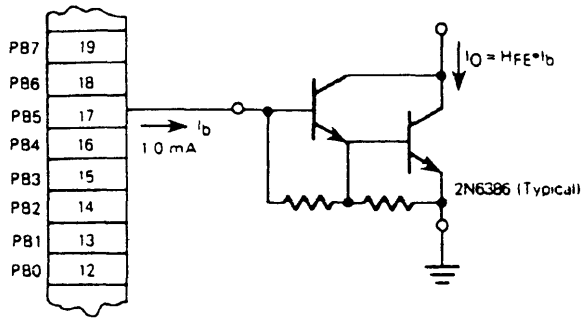


FIGURE 17 — TYPICAL PORT CONNECTIONS

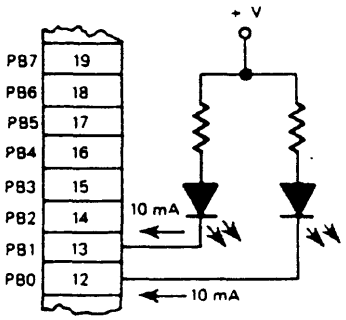
(a) Output Modes



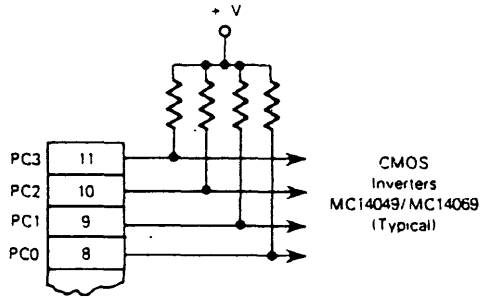
Port A, Bit 7 Programmed as Output, Driving CMOS Loads and Bit 4 Driving one TTL Load Directly (using CMOS output option)



Port B, Bit 5 Programmed as Output, Driving Darlington-Base Directly

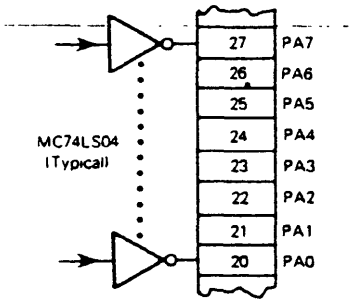


Port B, Bit 0 and Bit 1 Programmed as Output, Driving LEDs Directly

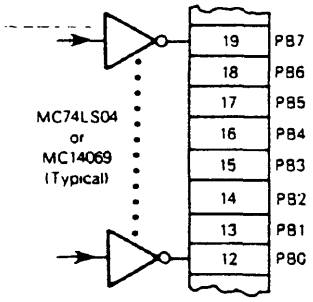


Port C, Bits 0-3 Programmed as Output, Driving CMOS Loads, Using External Pullup Resistors

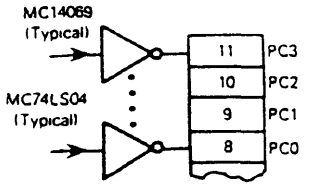
(b) Input Modes



TTL Driving Port A Directly



CMOS or TTL Driving Port B Directly



CMOS and TTL Driving Port C Directly

TIMER CONTROL REGISTER (TCR)

The configuration of the TCR is determined by the logic level of bit 6 (Timer Option, TOPT) in the Mask Option Register (MOR). Two configurations of the TCR are shown below, one for TOPT = 1 and the other for TOPT = 0. TOPT = 1 configures the TCR to emulate the MC6805P2 or MC6805P4. When TPOT = 0, it provides software control of the TCR. When TOPT = 1, the prescaler "mask" options are user programmable via the MOR. A description of each TCR bit is provided below (also see Figure 7 and **TIMER**).

b7	b6	b5	b4	b3	b2	b1	b0	
TIR	TIM	1	1	1	1	1	1	Timer Control Register $009

TCR with MOR TOPT = 1 (MC6805P2/P4 Emulation)

b7	b6	b5	b4	b3	b2	b1	b0	
TIR	TIM	TIN	TIE	PSC*	PS2	PS1	PS0	Timer Control Register $009

TCR with MOR TOPT = 0 (Software Programmable Timer)

* = write only, reads as a zero

b7, TIR — Timer Interrupt Request — Used to initiate the timer interrupt or signal a timer Data Register underflow when it is a logical "1".

1 = Set when the Timer Data Register changes to all zeros.
0 = Cleared by external reset, power-on reset, or under program control.

b6, TIM — Timer Interrupt Mask — Used to inhibit the timer interrupt, to the processor, when it is a logical "1".

1 = Set by an external reset, power-on reset, or under program control.
0 = Cleared under program control.

b5, TIN — External or Internal — Selects the input clock source to be either the external TIMER pin (7) or the internal $\phi 2$.

1 = Selects the external clock source (event count mode)
0 = Selects the internal $\phi 2$ ($f_{OSC} \div 4$) clock source

b4, TIE — External Enable — Used to enable the external TIMER pin (7) or to enable the internal clock (if TIN = 0) regardless of the external timer pin state (disables gated clock feature). When TOPT = 1, TIE is always a logical "1".

1 = Enables external timer pin.
0 = Disables external timer pin.

TIN-TIE Modes

TIN	TIE	CLOCK
0	0	Internal Clock ($\phi 2$)
0	1	Gated (AND) of External and Internal Clocks
1	0	No Clock
1	1	External Clock

b3, PSC — Prescaler Clear — This is a write-only bit. It reads as a logical "0" (when TOPT = 0) so the BSET and BCLR on the TCR function correctly. Writing a "1" into PSC generates a pulse which clears the prescaler. (When TOPT = 1 this bit is always read as a logical "1" and has no effect on the prescaler.)

b2, PS2
b1, PS1
b0, PS0 — Prescaler Select — These bits are decoded to select one of eight outputs on the timer prescaler. The table below shows the prescaler division resulting from decoding these bits.

PS2	PS1	PS0	Prescaler Division
0	0	0	1 (Bypass Prescaler)
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

Note

When changing the PS2-PS0 bits in software, the PSC bit should be written to a "1" in the same write cycle to clear the prescaler. Changing the PS bits without clearing the prescaler may cause an extraneous toggle of the Timer Data Register.

MASK OPTIONS

The MC68705P5 Mask Option Register is implemented in EPROM. Like all other EPROM bytes, the MOR contains all zeros prior to programming.

When used to emulate the MC6805P2 or MC6805P4, five of the eight MOR bits are used in conjunction with the prescaler. Of the remaining, the b7 bit is used to select the type of oscillator clock, b3 is the secure/non-secure mode option, and b4 is not used. Bits b0, b1, and b2 determine the division of the Timer prescaler. Bit b5 determines the Timer clock source. The value of the TOPT bit (b6) is programmed to configure the TCR (a logic "1" for MC6805P2/P4 emulation).

If the MOR Timer Option (TOPT) bit is a 0, bits b5, b4, b2, b1, and b0 set the initial value of their respective TCR bits during reset. After initialization the TCR is software controllable.

A description of the MOR bits is as follows:

b7	b6	b5	b4	b3	b2	b1	b0	
CLK	TOPT	CLS		SNM	P2	P1	P0	Mask Option Register $784

b7, CLK Clock Oscillator Type
1 = RC
0 = Crystal

Note

V_{IHTP} on the TIMER pin (7) forces the crystal mode.

b6, TOPT Timer Option
1 = MC6805P2/P4 type timer/prescaler. All bits, except 6 and 7, of the Timer-Control Register (TCR) are invisible to the user. Bits 5, 2, 1, and 0 of the Mask Option Register determine the equivalent MC6805P2/P4 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 (TCR is then software contolled after initialization).

b5, CLS Timer/Prescaler Clock Source
1 = External TIMER pin.
0 = Internal $\phi 2$

b4, (TIE) Not used if MOR TOPT = 1 (MC6805P2/P4 emulation). Sets initial value of TCR TIE if MOR TOPT = 0.

b3, SNM When this bit is set, i.e., programmed to a "1", it is not possible to access the EPROM contents of the MC68705P5 externally. For further information see **PROGRAMMING FIRMWARE.**

b2, P2
b1, P1
b0, P0 Prescaler Option – the logical levels of these bits, when decoded, select one of eight outputs on the timer prescaler. The table below shows the division resulting from decoding combinations of these three bits.

P2	P1	P0	Prescaler Division
0	0	0	1 (Bypass Prescaler)
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

Two examples for programming the MOR are discussed below.

Example 1 To emulate an MC6805P2 to verify your program with an RC oscillator, and an event count input for the timer with no prescaling, the MOR would be set to "11111000". To write the MOR, it is simply programmed as any other EPROM byte.

Example 2 Suppose you wish to use the MC68705P5 programmable prescaler functions, and you wish the initial condition of the prescaler to be divided by 64, with the input disabled and an internal clock source. If the clock oscillator was to be in the crystal mode, the MOR would be set to "00001110".

ON-CHIP PROGRAMMING HARDWARE

The Programming Control Register (PCR) at location $00B is an 8-bit register which utilizes the three LSBs (the five MSBs are set to logic "1s"). This register provides the necessary control bits to allow programming the MC68705P5 EPROM. The bootstrap program manipulates the PCR when programming, so that users need not be concerned with the PCR in most applications. A description of each bit follows.

b7	b6	b5	b4	b3	b2	b1	b0	
1	1	1	1	1	$\overline{VPON}$	$\overline{PGE}$	$\overline{PLE}$	Program Control Register $00B

b0, $\overline{PLE}$ Programming Latch Enable – When cleared this bit allows the address and data to be latched into the EPROM. When this bit is set, data can be read from the EPROM.
1 = (set) read EPROM
0 = (clear) latch address and data into EPROM (read disabled)
$\overline{PLE}$ is set during a Reset, but may be cleared any time. However, its effect on the EPROM is inhibited if $\overline{VPON}$ is a logic "1".

b1, $\overline{PGE}$ Program Enable – When cleared, $\overline{PGE}$ enables programming of the EPROM. $\overline{PGE}$ can only be cleared if $\overline{PLE}$ is cleared. $\overline{PGE}$ must be set when changing the address and data; i.e., setting up the byte to be programmed.
1 = (set) inhibit EPROM programming
0 = (clear) enable EPROM programming (if $\overline{PLE}$ is low)

$\overline{PGE}$ is set during a Reset, however, it has no effect on EPROM circuits if $\overline{VPON}$ is a logic "1"

b2, $\overline{VPON}$ (Vpp ON) – $\overline{VPON}$ is a read-only bit and when at a logic "0" it indicates that a "high voltage" is present at the Vpp pin

1 = no "high voltage" on Vpp pin

0 = "high voltage" on Vpp pin

VPON being "1" "disconnects" PGE and PLE from the rest of the chip, preventing accidental clearing of these bits from effecting the normal operating mode

Note

$\overline{VPON}$ being "0" does not indicate that the Vpp level is correct for programming. It is used as a safety interlock for the user in the normal operating mode.

The Programming Control Register functions are shown below:

$\overline{VPON}$	$\overline{PGE}$	$\overline{PLE}$	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 iff $\overline{PLE}$ = 0
1	0	1	Invalid state, $\overline{PGE}$ = 0 iff $\overline{PLE}$ = 0
0	1	1	"High voltage" on Vpp
1	1	1	$\overline{PGE}$ and $\overline{PLE}$ disabled from system (Operating Mode)

ERASING THE EPROM

The MC68705P5 EPROM can be erased by exposure to high-intensity ultraviolet (UV) light with a wavelength of 2537 Å. The recommended integrated dose (UV intensity x exposure time) is a 25 Ws/cm^2. The lamps should be used without shortwave filters and the MC68705P5 should be positioned about one inch from the UV tubes. Ultraviolet erasure clears all bits of the MC68705P5 EPROM to the "0" state. Data is then entered by programming "1s" into the desired bit locations.

Caution

Be sure that the EPROM window is shielded from light except when erasing. This protects both the EPROM and light-sensitive nodes.

PROGRAMMING FIRMWARE

The MC68705P5 has 115 bytes of mask ROM containing a bootstrap program which can be used to program the MC68705P5 EPROM. The vector at addresses $7F6 and $7F7 is used to start executing the program. This vector is fetched when V_{IHTP} is applied to pin 7 (TIMER pin) of the MC68705P5 and the $\overline{RESET}$ pin is allowed to rise above V_{IRES+}. Figure 18 provides a schematic diagram of a circuit and a summary of programming steps which can be used to program the EPROM in the MC68705P5.

Note that the MC68705P5 will not execute the bootstrap program when in the secure mode. Therefore, the on-chip EPROM must be completely erased before programming. To enter the secure mode, bit 3 of the mask option register must be programmed to a logic one and memory locations $782 and $783 must be programmed with $20 and $FE respectively. After programming, the only way to revert the non-secure mode is by erasing the entire EPROM

PROGRAMMING STEPS

The MCM2716 UV EPROM must first be programmed with an exact duplicate of the information that is to be transferred to the MC68705P5. Non-EPROM addresses are ignored by the bootstrap. Since the MC68705P5 and the MCM2716 are to be inserted and removed from the circuit they should be mounted in sockets. In addition, the precaution below must be observed (refer to Figure 18)

Caution

Be sure S1 and S2 are closed and V_{CC} and +26 V are not applied when inserting the MC68705P5 and MCM2716 into their respective sockets. This ensures that $\overline{RESET}$ is held low while inserting the devices

When ready to program the MC68705P5 it is only necessary to provide V_{CC} and +26 V, open switch S2 (to apply Vpp and V_{IHTP}), and then open S1 (to remove Reset). Once the voltages are applied and both S2 and S1 are open, the CLEAR output control line (PB4) goes high and then low, then the 11-bit counter (MC14040B) is clocked by the PB3 output ($\overline{COUNT}$). The counter selects the MCM2716 EPROM byte which is to load the equivalent MC68705P5 EPROM byte selected by the bootstrap program. Once the EPROM location is loaded, $\overline{COUNT}$ clocks the counter to the next EPROM location. This continues until the MC68705P5 is completely programmed at which time the Programmed indicator LED is lit. The counter is cleared and the loop is repeated to verify the programmed data. The Verified indicator LED lights if the programming is correct.

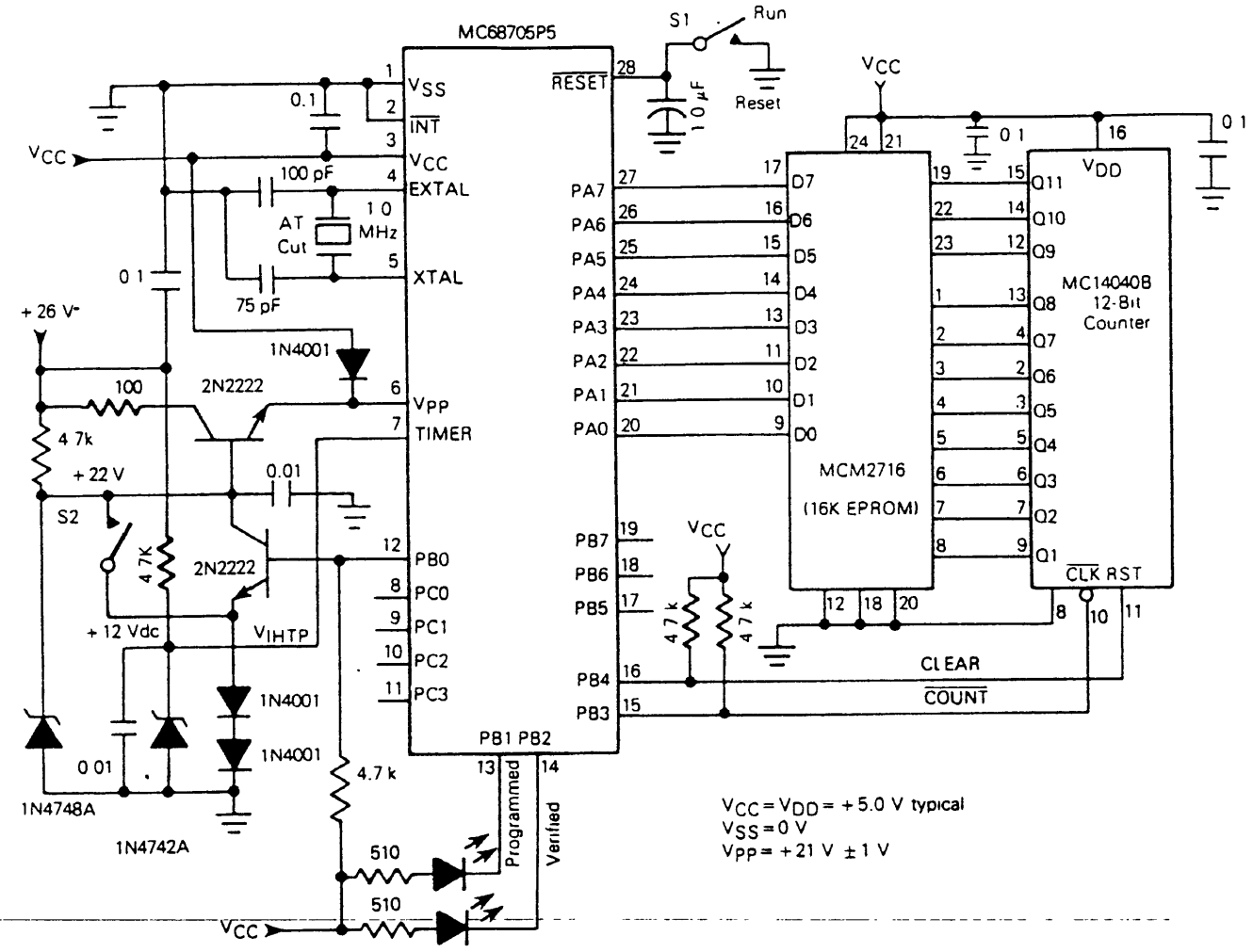
Once the MC68705P5 has been programmed and verified, close switch S2 (to remove Vpp and V_{IHTP}) and close switch S1 (to Reset). Disconnect +26 V and V_{CC}, then remove the MC68705P5 from its socket.

MC6805P2 AND MC6805P4 EMULATION

The MC68705P5 emulates the MC6805P2 and MC6805P4 "exactly." MC6805P2/P4 mask features are implemented in the Mask Option Register (MOR) EPROM byte on the MC68705P5. There are a few minor exceptions to the exactness of emulation with are listed below:

1. The MC6805P2 "future ROM" area is implemented in the MC68705P5 and these 704 bytes must be left unprogrammed to accurately simulate the MC6805P2/P4. (The MC6805P2/P4 reads all "0s" from this area.)
2. The reserved ROM areas in the MC6805P2/P4 and MC68705P5 have different data stored in them and this data is subject to change without notice. The MC6805P2 uses the reserved ROM for the self-check feature and the MC68705P5 uses this area for the bootstrap program.
3. The MC6805P2 reads all "1s" in its 48 byte "future RAM" area. This RAM is not implemented in the MC6805P2 mask ROM version, but is implemented in the MC68705P5 and MC68705P3.
4. The Vpp line (pin 6) in the MC68705P5 must be tied to V_{CC} for normal operation. In the MC6805P2, pin 6 is the NUM pin and is grounded in normal operation. The MC6805P4 uses pin 6 for V_{SB} which is normally tied to V_{CC}, as with the MC68705P5.
5. The function in the Non-User Mode is not identical to the MC6805P2/P4 version. Therefore, the MC68705P5

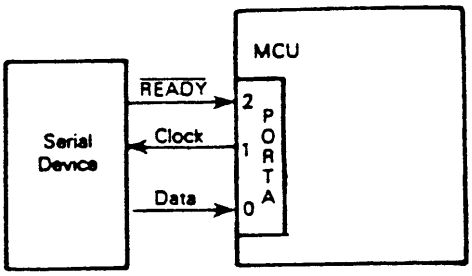
FIGURE 18 — PROGRAMMING CONNECTIONS SCHEMATIC DIAGRAM



Summary of Programming Steps:

1. When plugging in the MC68705P5 or the MCM2716, be sure that S1 and S2 are closed and that V_{CC} and +26 V are not applied.
2. To initiate programming, be sure S1 is closed; S2 is closed; and V_{CC} and +26 V are applied. Then open S2, followed by S1.
3. Before removing the MC68705P5, first close S2 and then close S1. Disconnect V_{CC} and +26 V; then remove the MC68705P5.

FIGURE 19 — BIT MANIPULATION EXAMPLE




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

does not function in the MEX6805 Support System. In normal operation, all pin functions are the same as on the MC6805P2/P4 version, except for pin 6 as previously noted.

6. The MC6805P4 provides a standby RAM feature which is not available on the MC68705P5

The operation of all other circuitry has been exactly duplicated or designed to function exactly the same in both devices including Interrupts, Timer, Data Ports, and Data Direction Registers (DDRs). A stated design goal has been to provide the user with a safe inexpensive way to verify his program and system design before committing to a factory programmed ROM.

SOFTWARE

BIT MANIPULATION

The MC68705P5 MCU has the ability to set or clear any single random-access memory or input/output bit (except the Data Direction Register, see Caution under **INPUT/OUTPUT**), with a single instruction (BSET, BCLR). Any bit in the page zero memory can be tested, using the BRSET and BRCLR instructions and the program branches as a result of its state. The Carry bit equals the value of the bit referenced by BRSET and BRCLR. A Rotate instruction may then be used to accumulate serial input data in a RAM location or register. This 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 19 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, LSB 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 a RAM location.

ADDRESSING MODES

The MC68705P5 MCU has 10 addressing modes which are explained briefly in the following paragraphs. For additional details and graphical illustrations, refer to the M6805 Family Users Manual.

The term "effective address" (EA) is used in describing the addressing modes. EA 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 which 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 2-byte instruction. This address area includes all on-chip RAM, I/O registers, and 128 bytes of EPROM. Direct addressing is an effective use of both memory and time.

EXTENDED — In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode. Instructions using extended addressing are capable of referencing arguments anywhere in memory with a single 3-byte instruction. When using the Motorola assembler, the programmer need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction.

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

INDEXED, NO OFFSET — In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. Thus, 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 unsigned byte following the opcode. This addressing mode is useful in selecting the kth element in an n element table. With this 2-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 address mode can be used in a manner similar to indexed, 8-bit offset, except that this 3-byte instruction allows tables to be anywhere in memory. As with Direct and Extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

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 read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single 2-byte instruction. See Caution under the **INPUT/OUTPUT**.

BIT TEST AND BRANCH — The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit which is to be tested and the condition (set or clear) is included in the opcode, and the

address of the byte to be tested is in the single byte immediately following the opcode byte The signed relative 8-bit offset is in the third byte and is added to the value of the PC, if the branch condition is true. This single 3-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of memory The span of branching is from −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 See Caution under **INPUT/OUTPUT.**

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 control instruction with no other arguments, are included in this mode These instructions are one byte long.

INSTRUCTION SET

The MC68705P5 MPU has a set of 59 basic instructions, which when combined with the 10 address modes produce 207 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.

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 operands Refer to Table 1

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 (see Caution under **INPUT/OUTPUT**). The test for negative or zero (TST) instruction is included in the read-modify-write instructions, though it does not perform the write. Refer to Table 2.

BRANCH INSTRUCTIONS — The branch instructions cause a branch from the program when a certain condition is met. Refer to Table 3

BIT MANIPULATION INSTRUCTIONS — These instructions are used on any bit in the first 256 bytes of the memory (see Caution under **INPUT/OUTPUT**). One group either sets or clears. The other group performs the bit and test branch operations. Refer to Table 4.

CONTROL INSTRUCTIONS — The control instructions control the MCU operations during program execution. Refer to Table 5.

ALPHABETICAL LISTING — The complete instruction set is given in alphabetical order in Table 6.

OPCODE MAP SUMMARY — Table 7 is an opcode map for the instructions used on the MCU.

TABLE 1 — REGISTER/MEMORY INSTRUCTIONS

		Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8 Bit Offset)			Indexed (16 Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	OP Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	4	C6	3	5	F6	1	4	E6	2	5	D6	3	6
Load X from Memory	LDX	AE	2	2	BE	2	4	CE	3	5	FE	1	4	EE	2	5	DE	3	6
Store A in Memory	STA				B7	2	5	C7	3	6	F7	1	5	E7	2	6	D7	3	7
Store X in Memory	STX				BF	2	5	CF	3	6	FF	1	5	EF	2	6	DF	3	7
Add Memory to A	ADD	AB	2	2	BB	2	4	CB	3	5	FB	1	4	EB	2	5	DB	3	6
Add Memory and Carry to A	ADC	A9	2	2	B9	2	4	C9	3	5	F9	1	4	E9	2	5	D9	3	6
Subtract Memory	SUB	A0	2	2	B0	2	4	C0	3	5	F0	1	4	E0	2	5	D0	3	6
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	4	C2	3	5	F2	1	4	E2	2	5	D2	3	6
AND Memory to A	AND	A4	2	2	B4	2	4	C4	3	5	F4	1	4	E4	2	5	D4	3	6
OR Memory with A	ORA	AA	2	2	BA	2	4	CA	3	5	FA	1	4	EA	2	5	DA	3	6
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	4	C8	3	5	F8	1	4	E8	2	5	D8	3	6
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	4	C1	3	5	F1	1	4	E1	2	5	D1	3	6
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	4	C3	3	5	F3	1	4	E3	2	5	D3	3	6
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	4	C5	3	5	F5	1	4	E5	2	5	D5	3	6
Jump Unconditional	JMP				BC	2	3	CC	3	4	FC	1	3	EC	2	4	DC	3	5
Jump to Subroutine	JSR				BD	2	7	CD	3	8	FD	1	7	ED	2	8	DD	3	9

TABLE 2 — READ-MODIFY-WRITE INSTRUCTION

		Addressing Modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (No Offset)			Indexed (8 Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	4	5C	1	4	3C	2	6	7C	1	6	6C	2	7
Decrement	DEC	4A	1	4	5A	1	4	3A	2	6	7A	1	6	6A	2	7
Clear	CLR	4F	1	4	5F	1	4	3F	2	6	7F	1	6	6F	2	7
Complement	COM	43	1	4	53	1	4	33	2	6	73	1	6	63	2	7
Negate (2 s Complement)	NEG	40	1	4	50	1	4	30	2	6	70	1	6	60	2	7
Rotate Left Thru Carry	ROL	49	1	4	59	1	4	39	2	6	79	1	6	69	2	7
Rotate Right Thru Carry	ROR	46	1	4	56	1	4	36	2	6	76	1	6	66	2	7
Logical Shift Left	LSL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Logical Shift Right	LSR	44	1	4	54	1	4	34	2	6	74	1	6	64	2	7
Arithmetic Shift Right	ASR	47	1	4	57	1	4	37	2	6	77	1	6	67	2	7
Test for Negative or Zero	TST	4D	1	4	5D	1	4	3D	2	6	7D	1	6	6D	2	7

TABLE 3 — BRANCH INSTRUCTIONS

Function	Mnemonic	Relative Addressing Mode		
		Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	4
Branch Never	BRN	21	2	4
Branch IFF Higher	BHI	22	2	4
Branch IFF Lower or Same	BLS	23	2	4
Branch IFF Carry Clear	BCC	24	2	4
(Branch IFF Higher or Same)	(BHS)	24	2	4
Branch IFF Carry Set	BCS	25	2	4
(Branch IFF Lower)	(BLO)	25	2	4
Branch IFF Not Equal	BNE	26	2	4
Branch IFF Equal	BEQ	27	2	4
Branch IFF Half Carry Clear	BHCC	28	2	4
Branch IFF Half Carry Set	BHCS	29	2	4
Branch IFF Plus	BPL	2A	2	4
Branch IFF Minus	BMI	2B	2	4
Branch IFF Interrupt Mask Bit is Clear	BMC	2C	2	4
Branch IFF Interrupt Mask Bit is Set	BMS	2D	2	4
Branch IFF Interrupt Line is Low	BIL	2E	2	4
Branch IFF Interrupt Line is High	BIH	2F	2	4
Branch to Subroutine	BSR	AD	2	8

TABLE 4 — BIT MANIPULATION INSTRUCTIONS

Function	Mnemonic	Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IFF Bit n is set	BRSET n (n = 0. ..7)	–	–	–	2 • n	3	10
Branch IFF Bit n is clear	BRCLR n (n = 0.. 7)	–	–	–	01 + 2 • n	3	10
Set Bit n	BSET n (n = 0. .7)	10 + 2 • n	2	7	–	–	–
Clear Bit n	BCLR n (n = 0... 7)	11 + 2 • n	2	7	–	–	–

TABLE 5 — CONTROL INSTRUCTIONS

Function	Mnemonic	Op Code	Inherent	
			# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	11
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2

TABLE 6 — INSTRUCTION SET

	Addressing Modes										Condition Codes				
Mnemonic	Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		X	X	X		X	X	X			Λ	•	Λ	Λ	Λ
ADD		X	X	X		X	X	X			Λ	•	Λ	Λ	Λ
AND		X	X	X		X	X	X			•	•	Λ	Λ	•
ASL	X		X			X	X				•	•	Λ	Λ	Λ
ASR	X		X			X	X				•	•	Λ	Λ	Λ
BCC					X						•	•	•	•	•
BCLR									X		•	•	•	•	•
BCS					X						•	•	•	•	•
BEQ					X						•	•	•	•	•
BHCC					X						•	•	•	•	•
BHCS					X						•	•	•	•	•
BHI					X						•	•	•	•	•
BHS					X						•	•	•	•	•
BIH					X						•	•	•	•	•
BIL					X						•	•	•	•	•
BIT		X	X	X		X	X	X			•	•	Λ	Λ	•
BLO					X						•	•	•	•	•
BLS					X						•	•	•	•	•
BMC					X						•	•	•	•	•
BMI					X						•	•	•	•	•
BMS					X						•	•	•	•	•
BNE					X						•	•	•	•	•
BPL					X						•	•	•	•	•
BRA					X						•	•	•	•	•
BRN					X						•	•	•	•	•
BRCLR										X	•	•	•	•	Λ
BRSET										X	•	•	•	•	Λ
BSET									X		•	•	•	•	•
BSR					X						•	•	•	•	•
CLC	X	•									•	•	•	•	0
CLI	X										•	0	•	•	•
CLR	X		X			X	X				•	•	0	1	•
CMP		X	X	X		X	X	X			•	•	Λ	Λ	Λ
COM	X		X			X	X				•	•	Λ	Λ	1
CPX		X	X	X		X	X	X			•	•	Λ	Λ	Λ

Condition Code Symbols

H	Half Carry (From Bit 3)
I	Interrupt Mask
N	Negative (Sign Bit)
Z	Zero
C	Carry/Borrow
Λ	Test and Set if True, Cleared Otherwise
•	Not Affected
?	Load CC Register From Stack
1	Set
0	Clear

TABLE 6 — INSTRUCTION SET (CONTINUED)

Mnemonic	Addressing Modes										Condition Codes				
	Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
DEC	X		X			X	X				•	•	Λ	Λ	•
EOR		X	X	X		X	X	X			•	•	Λ	Λ	•
INC	X		X			X	X				•	•	Λ	Λ	•
JMP			X	X		X	X	X			•	•	•	•	•
JSR			X	X		X	X	X			•	•	•	•	•
LDA		X	X	X		X	X	X			•	•	Λ	Λ	•
LDX		X	X	X		X	X	X			•	•	Λ	Λ	•
LSL	X		X			X	X				•	•	Λ	Λ	Λ
LSR	X		X			X	X				•	•	0	Λ	Λ
NEG	X		X			X	X				•	•	Λ	Λ	Λ
NOP	X										•	•	•	•	•
ORA		X	X	X		X	X	X			•	•	Λ	Λ	•
ROL	X		X			X	X				•	•	Λ	Λ	Λ
RSP	X										•	•	•	•	•
RTI	X										?	?	?	?	?
RTS	X										•	•	•	•	•
SBC		X	X	X		X	X	X			•	•	Λ	Λ	Λ
SEC	X										•	•	•	•	1
SEI	X										•	1	•	•	•
STA			X	X		X	X	X			•	•	Λ	Λ	•
STX			X	X		X	X	X			•	•	Λ	Λ	•
SUB		X	X	X		X	X	X			•	•	Λ	Λ	Λ
SWI	X										•	1	•	•	•
TAX	X										•	•	•	•	•
TST	X		X			X	X				•	•	Λ	Λ	•
TXA	X										•	•	•	•	•

Condition Code Symbols

H	Half Carry (From Bit 3)
I	Interrupt Mask
N	Negative (Sign Bit)
Z	Zero
C	Carry/Borrow
Λ	Test and Set if True, Cleared Otherwise
•	Not Affected
?	Load CC Register From Stack
1	Set
0	Clear

TABLE 7 — M6805 HMOS FAMILY OPCODE MAP

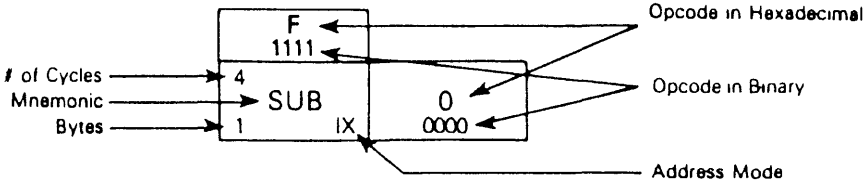
	Bit Manipulation		Branch	Read-Modify-Write					Control		Register/Memory						
	BTB 0 0000	BSC 1 0001	REL 2 0010	DIR 3 0011	INH 4 0100	INH 5 0101	IX1 6 0110	IX 7 0111	INH 8 1000	INH 9 1001	IMM A 1010	DIR B 1011	EXT C 1100	IX2 D 1101	IX1 E 1110	IX F 1111	
0 0000	10 BRSET0 3 BTB	7 BSET0 2 BSC	4 BRA 2 REL	6 NEG 2 DIR	4 NEG 1 INH	4 NEG 1 INH	7 NEG 2 IX1	6 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	4 SUB 2 DIR	5 SUB 3 EXT	6 SUB 3 IX2	5 SUB 2 IX1	4 SUB 1 IX	0 0000
1 0001	10 BRCLR0 3 BTB	7 BCLR0 2 BSC	4 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	4 CMP 2 DIR	5 CMP 3 EXT	6 CMP 3 IX2	5 CMP 2 IX1	4 CMP 1 IX	1 0001
2 0010	10 BRSET1 3 BTB	7 BSET1 2 BSC	4 BHI 2 REL								2 SBC 2 IMM	4 SBC 2 DIR	5 SBC 3 EXT	6 SBC 3 IX2	5 SBC 2 IX1	4 SBC 1 IX	2 0010
3 0011	10 BRCLR1 3 BTB	7 BCLR1 2 BSC	4 BLS 2 REL	6 COM 2 DIR	4 COMA 1 INH	4 COMX 1 INH	7 COM 2 IX1	6 COM 1 IX	11 SWI 1 INH		2 CPX 2 IMM	4 CPX 2 DIR	5 CPX 3 EXT	6 CPX 3 IX2	5 CPX 2 IX1	4 CPX 1 IX	3 0011
4 0100	10 BRSET2 3 BTB	7 BSET2 2 BSC	4 BCC 2 REL	6 LSR 2 DIR	4 LSRA 1 INH	4 LSRX 1 INH	7 LSR 2 IX1	6 LSR 1 IX			2 AND 2 IMM	4 AND 2 DIR	5 AND 3 EXT	6 AND 3 IX2	5 AND 2 IX1	4 AND 1 IX	4 0100
5 0101	10 BRCLR2 3 BTB	7 BCLR2 2 BSC	4 BCS 2 REL								2 BIT 2 IMM	4 BIT 2 DIR	5 BIT 3 EXT	6 BIT 3 IX2	5 BIT 2 IX1	4 BIT 1 IX	5 0101
6 0110	10 BRSET3 3 BTB	7 BSET3 2 BSC	4 BNE 2 REL	6 ROR 2 DIR	4 RORA 1 INH	4 RORX 1 INH	7 ROR 2 IX1	6 ROR 1 IX			2 LDA 2 IMM	4 LDA 2 DIR	5 LDA 3 EXT	6 LDA 3 IX2	5 LDA 2 IX1	4 LDA 1 IX	6 0110
7 0111	10 BRCLR3 3 BTB	7 BCLR3 2 BSC	4 BEQ 2 REL	6 ASR 2 DIR	4 ASRA 1 INH	4 ASRX 1 INH	7 ASR 2 IX1	6 ASR 1 IX		2 TAX 1 INH		5 STA 2 DIR	6 STA 3 EXT	7 STA 3 IX2	6 STA 2 IX1	5 STA 1 IX	7 0111
8 1000	10 BRSET4 3 BTB	7 BSET4 2 BSC	4 BHCC 2 REL	6 LSL 2 DIR	4 LSLA 1 INH	4 LSLX 1 INH	7 LSL 2 IX1	6 LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	4 EOR 2 DIR	5 EOR 3 EXT	6 EOR 3 IX2	5 EOR 2 IX1	4 EOR 1 IX	8 1000
9 1001	10 BRCLR4 3 BTB	7 BCLR4 2 BSC	4 BHCS 2 REL	6 ROL 2 DIR	4 ROLA 1 INH	4 ROLX 1 INH	7 ROL 2 IX1	6 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	4 ADC 2 DIR	5 ADC 3 EXT	6 ADC 3 IX2	5 ADC 2 IX1	4 ADC 1 IX	9 1001
A 1010	10 BRSET5 3 BTB	7 BSET5 2 BSC	4 BPL 2 REL	6 DEC 2 DIR	4 DECA 1 INH	4 DECX 1 INH	7 DEC 2 IX1	6 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	4 ORA 2 DIR	5 ORA 3 EXT	6 ORA 3 IX2	5 ORA 2 IX1	4 ORA 1 IX	A 1010
B 1011	10 BRCLR5 3 BTB	7 BCLR5 2 BSC	4 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	4 ADD 2 DIR	5 ADD 3 EXT	6 ADD 3 IX2	5 ADD 2 IX1	4 ADD 1 IX	B 1011
C 1100	10 BRSET6 3 BTB	7 BSET6 2 BSC	4 BMC 2 REL	6 INC 2 DIR	4 INCA 1 INH	4 INCX 1 INH	7 INC 2 IX1	6 INC 1 IX		2 RSP 1 INH		3 JMP 2 DIR	4 JMP 3 EXT	5 JMP 3 IX2	4 JMP 2 IX1	3 JMP 1 IX	C 1100
D 1101	10 BRCLR6 3 BTB	7 BCLR6 2 BSC	4 BMS 2 REL	6 TST 2 DIR	4 TSTA 1 INH	4 TSTX 1 INH	7 TST 2 IX1	6 TST 1 IX		2 NOP 1 INH	8 BSR 2 REL	7 JSR 2 DIR	8 JSR 3 EXT	9 JSR 3 IX2	8 JSR 2 IX1	7 JSR 1 IX	D 1101
E 1110	10 BRSET7 3 BTB	7 BSET7 2 BSC	4 BIL 2 REL								2 LDX 2 IMM	4 LDX 2 DIR	5 LDX 3 EXT	6 LDX 3 IX2	5 LDX 2 IX1	4 LDX 1 IX	E 1110
F 1111	10 BRCLR7 3 BTB	7 BCLR7 2 BSC	4 BIH 2 REL	6 CLR 2 DIR	4 CLRA 1 INH	4 CLRX 1 INH	7 CLR 2 IX1	6 CLR 1 IX		2 TXA 1 INH		5 STX 2 DIR	6 STX 3 EXT	7 STX 3 IX2	6 STX 2 IX1	5 STX 1 IX	F 1111

Hi / Low

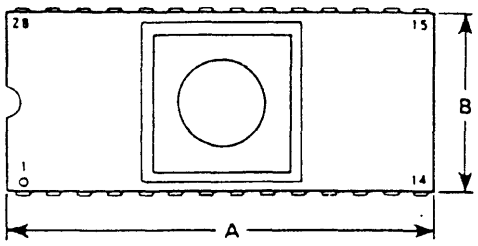
Abbreviations for Address Modes

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

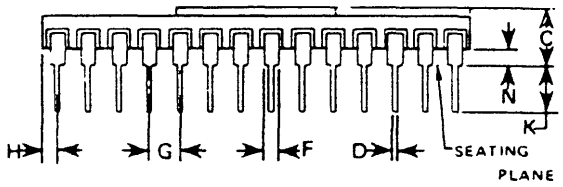


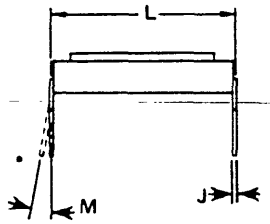
PACKAGE DIMENSIONS



L SUFFIX
CERAMIC PACKAGE
CASE 719

DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	35.20	35.92	1.386	1.414
B	14.73	15.34	0.580	0.604
C	3.18	5.08	0.125	0.200
D	0.38	0.53	0.015	0.021
F	0.76	1.40	0.030	0.055
G	2.54 BSC		0.100 BSC	
H	0.76	1.78	0.030	0.070
J	0.20	0.30	0.008	0.012
K	2.54	4.57	0.100	0.180
L	14.99	15.49	0.590	0.610
M	–	10°	–	10°
N	0.51	1.52	0.020	0.060





NOTES:
1. LEADS, TRUE POSITIONED WITHIN 0.25 mm (0.010) DIAMETER (AT SEATING PLANE) AT MAXIMUM MATERIAL CONDITION.
2. DIMENSION "L" TO CENTER OF LEADS WHEN FORMED PARALLEL.