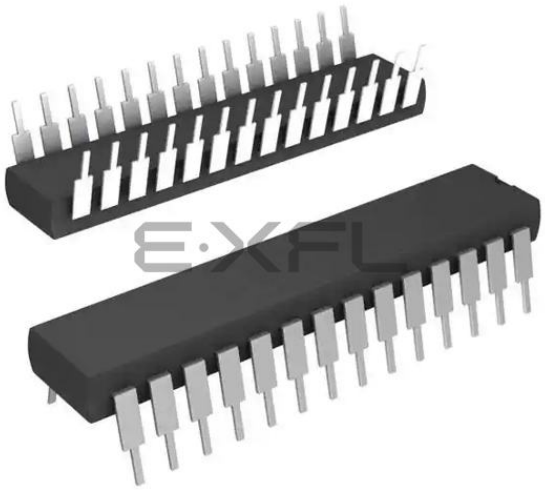


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	HC05
Core Size	8-Bit
Speed	2.1MHz
Connectivity	SIO
Peripherals	POR, WDT
Number of I/O	21
Program Memory Size	4.5KB (4.5K x 8)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	176 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 4x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.600", 15.24mm)
Supplier Device Package	28-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc705p6acpe

1.3.8 TCMP

This pin is the output from the 16-bit timer's output compare function. It is low after reset. Refer to Chapter 8 Capture/Compare Timer.

1.3.9 $\overline{IRQ}/V_{PP}$ (Maskable Interrupt Request)

This input pin drives the asynchronous interrupt function of the MCU in user mode and provides the V_{PP} programming voltage in bootloader mode. The MCU will complete the current instruction being executed before it responds to the $\overline{IRQ}$ interrupt request. When the $\overline{IRQ}/V_{PP}$ pin is driven low, the event is latched internally to signify an interrupt has been requested. When the MCU completes its current instruction, the interrupt latch is tested. If the interrupt latch is set and the interrupt mask bit (I bit) in the condition code register is clear, the MCU will begin the interrupt sequence.

Depending on the MOR LEVEL bit, the $\overline{IRQ}/V_{PP}$ pin will trigger an interrupt on either a negative edge at the $\overline{IRQ}/V_{PP}$ pin and/or while the $\overline{IRQ}/V_{PP}$ pin is held in the low state. In either case, the $\overline{IRQ}/V_{PP}$ pin must be held low for at least one t_{ILIH} time period. If the edge- and level-sensitive mode is selected (LEVEL bit set), the $\overline{IRQ}/V_{PP}$ input pin requires an external resistor connected to V_{DD} for wired-OR operation. If the $\overline{IRQ}/V_{PP}$ pin is not used, it must be tied to the V_{DD} supply. The $\overline{IRQ}/V_{PP}$ pin input circuitry contains an internal Schmitt trigger to improve noise immunity. Refer to Chapter 5 Interrupts.

NOTE

If the voltage level applied to the $\overline{IRQ}/V_{PP}$ pin exceeds V_{DD}, it may affect the MCU's mode of operation. See Chapter 3 Operating Modes.

Chapter 2
Memory

2.1 Introduction

The MC68HC705P6A utilizes 13 address lines to access an internal memory space covering 8 Kbytes. This memory space is divided into I/O, RAM, ROM, and EPROM areas.

2.2 User Mode Memory Map

When the MC68HC705P6A is in the user mode, the 32 bytes of I/O, 176 bytes of RAM, 4608 bytes of user EPROM, 48 bytes of user page zero EPROM, 239 bytes of bootloader ROM, and 16 bytes of user vectors EPROM are all active as shown in Figure 2-1.

2.3 Bootloader Mode Memory Map

Memory space is identical to the user mode. See Figure 2-1.

2.4 Input/Output and Control Registers

Figure 2-2 and Figure 2-3 briefly describe the I/O and control registers at locations $0000–$001F. Reading unimplemented bits will return unknown states, and writing unimplemented bits will be ignored.

2.5 RAM

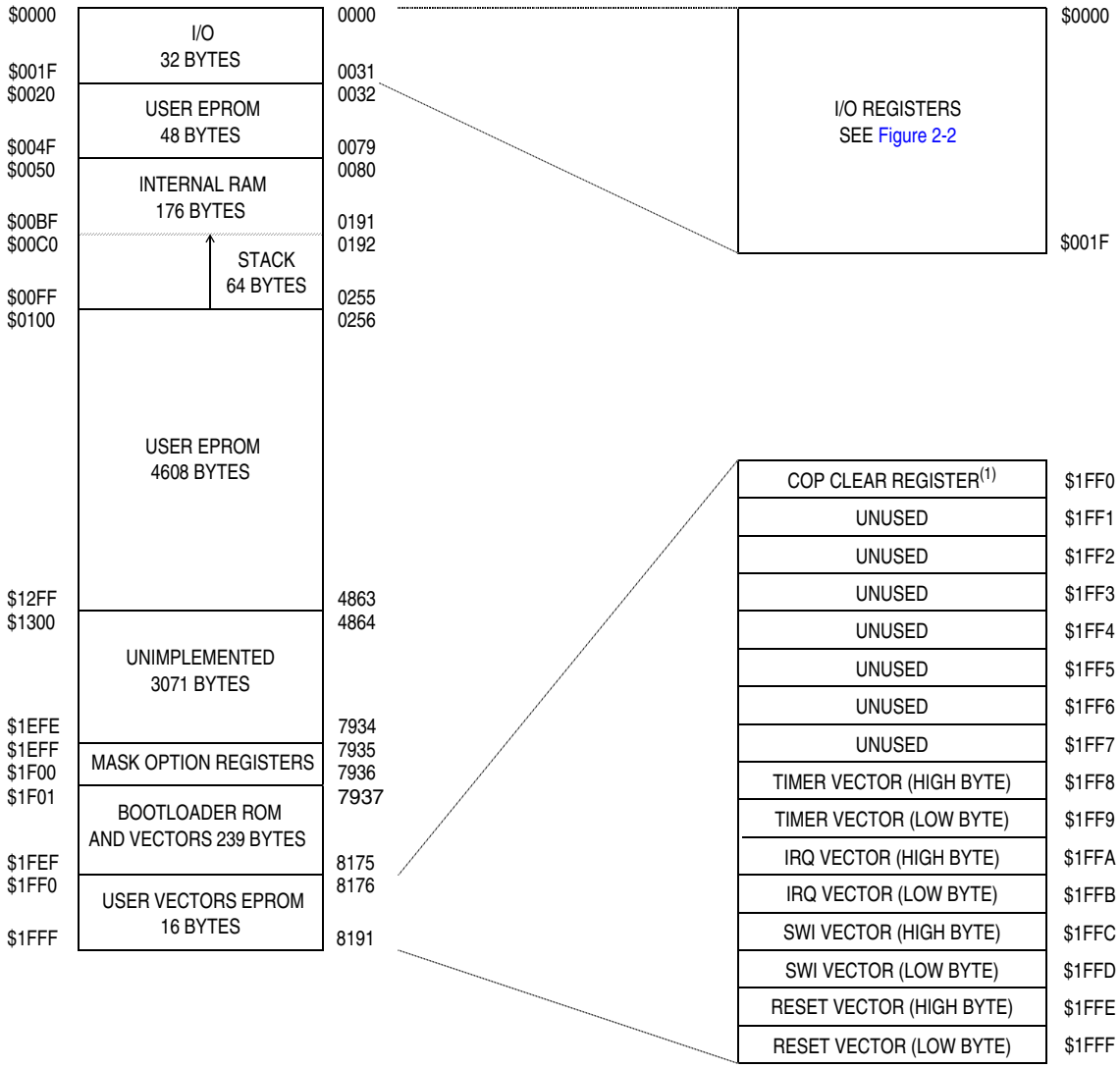
The user RAM consists of 176 bytes (including the stack) at locations $0050 through $00FF. The stack begins at address $00FF. The stack pointer can access 64 bytes of RAM from $00FF to $00C0.

NOTE

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

2.6 EPROM/ROM

There are 4608 bytes of user EPROM at locations $0100 through $12FF, plus 48 bytes in user page zero locations $0020 through $004F, and 16 additional bytes for user vectors at locations $1FF0 through $1FFF. The bootloader ROM and vectors are at locations $1F01 through $1FEF.



Note 1. Writing zero to bit 0 of $1FF0 clears the COP watchdog timer. Reading $1FF0 returns user EPROM data.

Figure 2-1. MC68HC705P6A User Mode Memory Map

Chapter 3
Operating Modes

3.1 Introduction

The MC68HC705P6A has two modes of operation that affect the pinout and architecture of the MCU: user mode and bootloader mode. The user mode is normally used for the application and the bootloader mode is used for programming the EPROM. The conditions required to enter each mode are shown in Table 3-1. The mode of operation is determined by the voltages on the $\overline{IRQ}/V_{PP}$ and PD7/TCAP pins on the rising edge of the external $\overline{RESET}$ pin.

Table 3-1. Operating Mode Conditions After Reset

$\overline{RESET}$ Pin	$\overline{IRQ}/V_{PP}$	PD7/TCAP	Mode
	V_{SS} to V_{DD}	V_{SS} to V_{DD}	Single chip
	V_{PP}	V_{DD}	Bootloader

The mode of operation is also determined whenever the internal computer operating properly (COP) watchdog timer resets the MCU. When the COP timer expires, the voltage applied to the $\overline{IRQ}/V_{PP}$ pin controls the mode of operation while the voltage applied to PD7/TCAP is ignored. The voltage applied to PD7/TCAP during the last rising edge on $\overline{RESET}$ is stored in a latch and used to determine the mode of operation when the COP watchdog timer resets the MCU.

3.2 User Mode

The user mode allows the MCU to function as a self-contained microcontroller, with maximum use of the pins for on-chip peripheral functions. All address and data activity occurs within the MCU and are not available externally. User mode is entered on the rising edge of $\overline{RESET}$ if the $\overline{IRQ}/V_{PP}$ pin is within the normal operating voltage range. The pinout for the user mode is shown in Figure 3-1.

In the user mode, there is an 8-bit I/O port, a second 8-bit I/O port shared with the analog-to-digital (A/D) subsystem, one 3-bit I/O port shared with the serial input/output port (SIOP), and a 3-bit port shared with the 16-bit timer subsystem, which includes one general-purpose I/O pin.

3.3 Bootloader Mode

The bootloader mode provides a means to program the user EPROM from an external memory device or host computer. This mode is entered on the rising edge of $\overline{RESET}$ if V_{PP} is applied to the $\overline{IRQ}/V_{PP}$ pin and V_{DD} is applied to the PD7/TCAP pin. The user code in the external memory device must have data located in the same address space it will occupy in the internal MCU EPROM, including the mask option register (MOR) at $1EFF and $1F00.

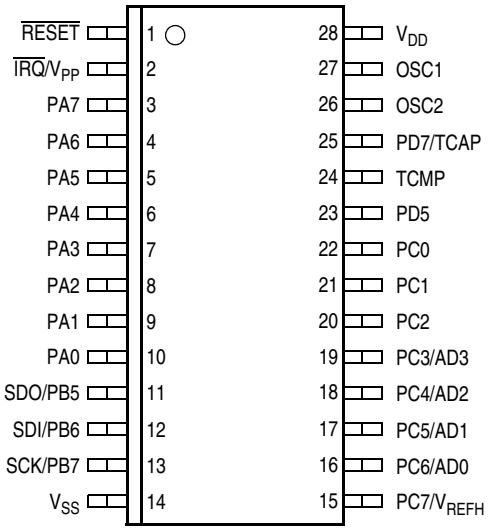


Figure 3-1. User Mode Pinout

3.4 Low-Power Modes

The MC68HC705P6A is capable of running in a low-power mode in each of its configurations. The WAIT and STOP instructions provide three modes that reduce the power required for the MCU by stopping various internal clocks and/or the on-chip oscillator. The SWAIT bit in the MOR is used to modify the behavior of the STOP instruction from stop mode to halt mode. The flow of the stop, halt, and wait modes is shown in Figure 3-2.

3.4.1 STOP Instruction

The STOP instruction can result in one of two modes of operation depending on the state of the SWAIT bit in the MOR. If the SWAIT bit is clear, the STOP instruction will behave like a normal STOP instruction in the M68HC05 Family and place the MCU in stop mode. If the SWAIT bit in the MOR is set, the STOP instruction will behave like a WAIT instruction (with the exception of a brief delay at startup) and place the MCU in halt mode.

3.4.1.1 Stop Mode

Execution of the STOP instruction when the SWAIT bit in the MOR is clear places the MCU in its lowest power consumption mode. In stop mode, the internal oscillator is turned off, halting *all* internal processing, including the COP watchdog timer. Execution of the STOP instruction automatically clears the I bit in the condition code register so that the $\overline{\text{IRQ}}$ external interrupt is enabled. All other registers and memory remain unaltered. All input/output lines remain unchanged.

The MCU can be brought out of stop mode only by an $\overline{\text{IRQ}}$ external interrupt or an externally generated $\overline{\text{RESET}}$. When exiting stop mode, the internal oscillator will resume after a 4064 internal clock cycle oscillator stabilization delay.

NOTE

Execution of the STOP instruction when the SWAIT bit in the MOR is clear will cause the oscillator to stop, and, therefore, disable the COP watchdog timer. To avoid turning off the COP watchdog timer, stop mode should be changed to halt mode by setting the SWAIT bit in the MOR. See 3.5 COP Watchdog Timer Considerations for additional information.

Chapter 4
Resets

4.1 Introduction

The MCU can be reset from three sources: one external input and two internal reset conditions. The $\overline{\text{RESET}}$ pin is a Schmitt trigger input as shown in Figure 4-1. The CPU and all peripheral modules will be reset by the RST signal which is the logical OR of internal reset functions and is clocked by PH1.

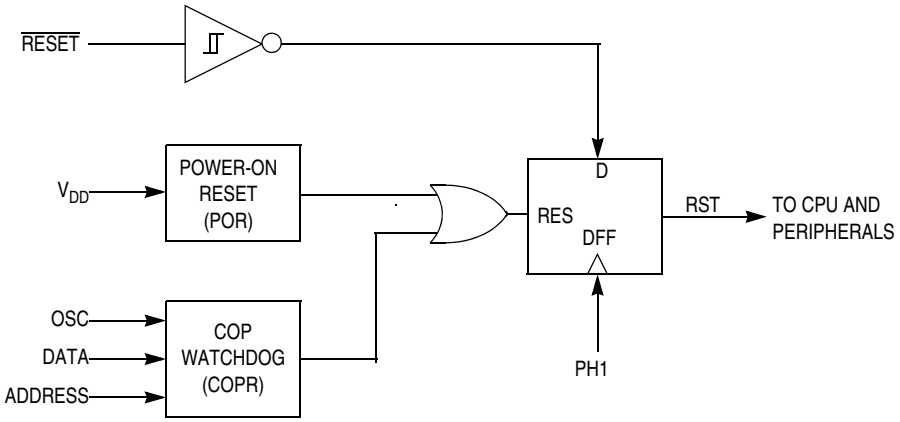


Figure 4-1. Reset Block Diagram

4.2 External Reset ($\overline{\text{RESET}}$)

The $\overline{\text{RESET}}$ input is the only external reset and is connected to an internal Schmitt trigger. The external reset occurs whenever the $\overline{\text{RESET}}$ input is driven below the lower threshold and remains in reset until the $\overline{\text{RESET}}$ pin rises above the upper threshold. The upper and lower thresholds are given in Chapter 14 Electrical Specifications.

4.3 Internal Resets

The two internally generated resets are the initial power-on reset (POR) function and the computer operating properly (COP) watchdog timer function.

4.3.1 Power-On Reset (POR)

The internal POR is generated at power-up to allow the clock oscillator to stabilize. The POR is strictly for power turn-on conditions and should not be used to detect a drop in the power supply voltage. There is a 4064 internal clock cycle oscillator stabilization delay after the oscillator becomes active.

7.2 SIOP Signal Format

The SIOP subsystem is software configurable for master or slave operation. No external mode selection inputs are available (for instance, slave select pin).

7.2.1 Serial Clock (SCK)

The state of the SCK output normally remains a logic 1 during idle periods between data transfers. The first falling edge of SCK signals the beginning of a data transfer. At this time, the first bit of received data may be presented at the SDI pin and the first bit of transmitted data is presented at the SDO pin (see Figure 7-2). Data is captured at the SDI pin on the rising edge of SCK. The transfer is terminated upon the eighth rising edge of SCK.

The master and slave modes of operation differ only by the sourcing of SCK. In master mode, SCK is driven from an internal source within the MCU. In slave mode, SCK is driven from a source external to the MCU. The SCK frequency is dependent upon the SPR0 and SPR1 bits located in the mask option register. Refer to 11.2 Mask Option Register for a description of available SCK frequencies.

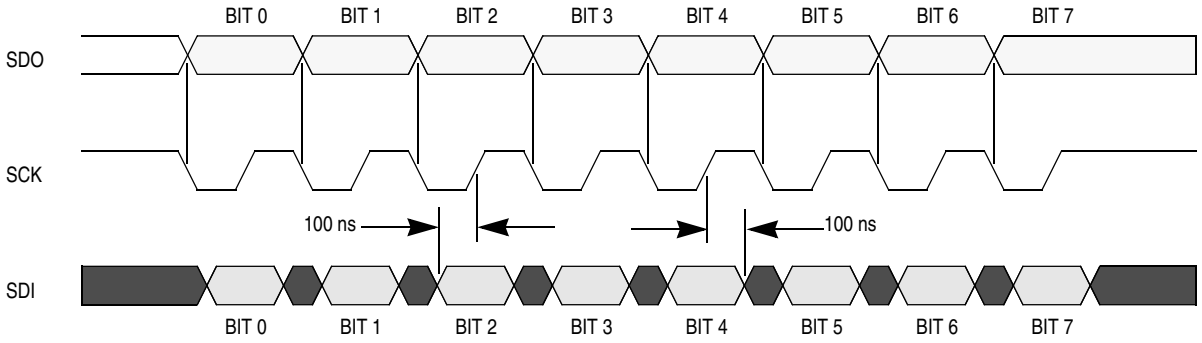


Figure 7-2. SIOP Timing Diagram

7.2.2 Serial Data Input (SDI)

The SDI pin becomes an input as soon as the SIOP subsystem is enabled. New data may be presented to the SDI pin on the falling edge of SCK.However, valid data must be present at least 100 nanoseconds before the rising edge of SCK and remain valid for 100 nanoseconds after the rising edge of SCK. See Figure 7-2.

7.2.3 Serial Data Output (SDO)

The SDO pin becomes an output as soon as the SIOP subsystem is enabled. Prior to enabling the SIOP, PB5 can be initialized to determine the beginning state. While the SIOP is enabled, PB5 cannot be used as a standard output since that pin is connected to the last stage of the SIOP serial shift register. Mask option register bit LSBF permits data to be transmitted in either the MSB first format or the LSB first format. Refer to 11.2 Mask Option Register for MOR LSBF programming information.

On the first falling edge of SCK, the first data bit will be shifted out to the SDO pin. The remaining data bits will be shifted out to the SDO pin on subsequent falling edges of SCK. The SDO pin will present valid data at least 100 nanoseconds before the rising edge of the SCK and remain valid for 100 nanoseconds after the rising edge of SCK. See Figure 7-2.

7.3 SIOP Registers

The SIOP is programmed and controlled by the SIOP control register (SCR) located at address $000A, the SIOP status register (SSR) located at address $000B, and the SIOP data register (SDR) located at address $000C.

7.3.1 SIOP Control Register (SCR)

This register is located at address $000A and contains two bits. Figure 7-3 shows the position of each bit in the register and indicates the value of each bit after reset.

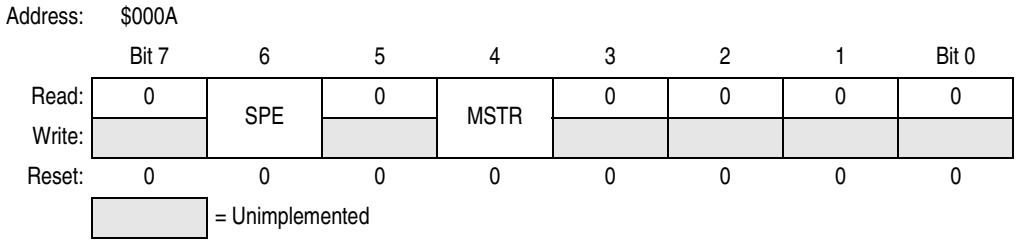


Figure 7-3. SIOP Control Register (SCR)

SPE — Serial Peripheral Enable

When set, the SPE bit enables the SIOP subsystem such that SDO/PB5 is the serial data output, SDI/PB6 is the serial data input, and SCK/PB7 is a serial clock input in the slave mode or a serial clock output in the master mode. Port B DDR and data registers can be manipulated as usual (except for PB5); however, these actions could affect the transmitted or received data.

The SPE bit is readable at any time. However, writing to the SIOP control register while a transmission is in progress will cause the SPIF and DCOL bits in the SIOP status register (see below) to operate incorrectly. Therefore, the SIOP control register should be written once to enable the SIOP and then not written to until the SIOP is to be disabled. Clearing the SPE bit while a transmission is in progress will 1) abort the transmission, 2) reset the serial bit counter, and 3) convert the port B/SIOP port to a general-purpose I/O port. Reset clears the SPE bit.

MSTR — Master Mode Select

When set, the MSTR bit configures the serial I/O port for master mode. A transfer is initiated by writing to the SDR. Also, the SCK pin becomes an output providing a synchronous data clock dependent upon the oscillator frequency. When the device is in slave mode, the SDO and SDI pins do not change function. These pins behave exactly the same in both the master and slave modes.

The MSTR bit is readable and writeable at any time regardless of the state of the SPE bit. Clearing the MSTR bit will abort any transfers that may have been in progress. Reset clears the MSTR bit as well as the SPE bit, disabling the SIOP subsystem.

7.3.2 SIOP Status Register (SSR)

This register is located at address $000B and contains two bits. Figure 7-4 shows the position of each bit in the register and indicates the value of each bit after reset.

Address:	$000B							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	SPIF	DCOL	0	0	0	0	0	0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 7-4. SIOP Status Register (SSR)

SPIF — Serial Port Interface Flag

SPIF is a read-only status bit that is set on the last rising edge of SCK and indicates that a data transfer has been completed. It has no effect on any future data transfers and can be ignored. The SPIF bit is cleared by reading the SSR followed by a read or write of the SDR. If the SPIF is cleared before the last rising edge of SCK, it will be set again on the last rising edge of SCK. Reset clears the SPIF bit.

DCOL — Data Collision

DCOL is a read-only status bit which indicates that an illegal access of the SDR has occurred. The DCOL bit will be set when reading or writing the SDR after the first falling edge of SCK and before SPIF is set. Reading or writing the SDR during this time will result in invalid data being transmitted or received.

The DCOL bit is cleared by reading the SSR (when the SPIF bit is set) followed by a read or write of the SDR. If the last part of the clearing sequence is done after another transfer has started, the DCOL bit will be set again. Reset clears the DCOL bit.

7.3.3 SIOP Data Register (SDR)

This register is located at address $000C and serves as both the transmit and receive data register. Writing to this register will initiate a message transmission if the SIOP is in master mode. The SIOP subsystem is not double buffered and any write to this register will destroy the previous contents. The SDR can be read at any time; however, if a transfer is in progress, the results may be ambiguous and the DCOL bit will be set. Writing to the SDR while a transfer is in progress can cause invalid data to be transmitted and/or received. Figure 7-5 shows the position of each bit in the register. This register is not affected by reset.

Address:	$000C							
	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0
Reset:	Unaffected by reset							

Figure 7-5. Serial Port Data Register (SDR)

8.3.5 Input Capture Registers

When a selected edge occurs on the TCAP pin, the current high and low bytes of the 16-bit counter are latched into the input capture registers. Reading ICRH before reading ICRL inhibits further capture until ICRL is read. Reading ICRL after reading the status register clears the input capture flag (ICF). Writing to the input capture registers has no effect.

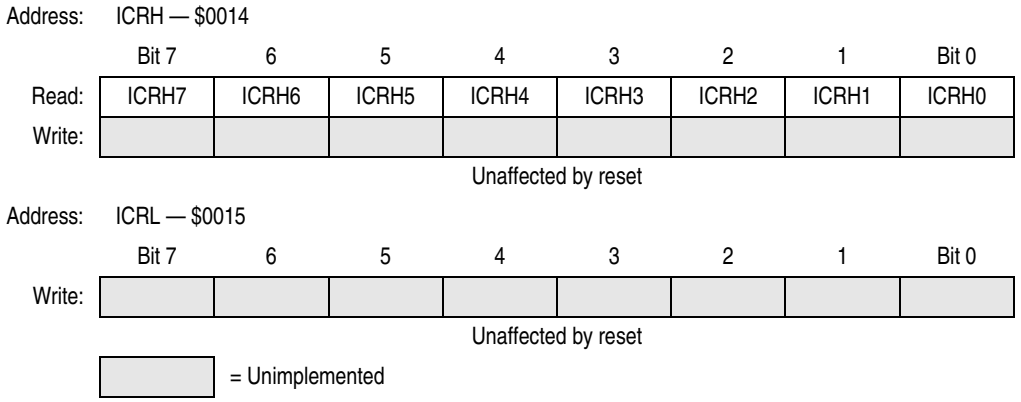


Figure 8-6. Input Capture Registers (ICRH and ICRL)

NOTE

To prevent interrupts from occurring between readings of ICRH and ICRL, set the interrupt flag in the condition code register before reading ICRH, and clear the flag after reading ICRL.

8.3.6 Output Compare Registers

When the value of the 16-bit counter matches the value in the output compare registers, the planned TCMP pin action takes place. Writing to OCRH before writing to OCRL inhibits timer compares until OCRL is written. Reading or writing to OCRL after the timer status register clears the output compare flag (OCF).

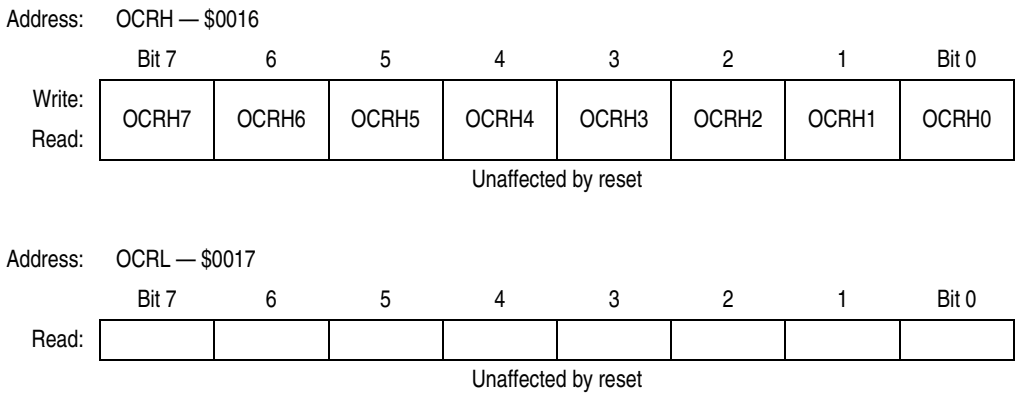


Figure 8-7. Output Compare Registers (OCRH and OCRL)

9.7 A/D Subsystem Operation during Halt/Wait Modes

The A/D subsystem continues normal operation during wait and halt modes. To decrease power consumption during wait or halt mode, the ADON and ADRC bits in the A/D status and control register should be cleared if the A/D subsystem is not being used.

9.8 A/D Subsystem Operation during Stop Mode

When stop mode is enabled, execution of the STOP instruction will terminate all A/D subsystem functions. Any pending conversion is aborted. When the oscillator resumes operation upon leaving stop mode, a finite amount of time passes before the A/D subsystem stabilizes sufficiently to provide conversions at its rated accuracy. The delays built into the MC68HC705P6A when coming out of stop mode are sufficient for this purpose. No explicit delays need to be added to the application software.

Address $001C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	ELAT	0	EPGM
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 10-1. EPROM Programming Register (EPROG)

EPGM — EPROM Program Control

If the EPGM bit is set, programming power is applied to the EPROM array. If the EPGM bit is cleared, programming power is removed from the EPROM array. The EPGM bit cannot be set unless the ELAT bit is set already.

Whenever the ELAT bit is cleared, the EPGM bit is cleared also. Both the EPGM and the ELAT bit cannot be set using the same write instruction. Any attempt to set both the EPGM and ELAT bit on the same write instruction cycle will result in the ELAT bit being set and the EPGM bit being cleared. The EPGM bit is a read-write bit and can be read at any time. The EPGM bit is cleared by reset.

ELAT— EPROM Latch Control

If the ELAT bit is set, the EPROM address and data bus are configured for programming to the array. If the ELAT bit is cleared, the EPROM address and data bus are configured for normal reading of data from the array. When the ELAT bit is set, the address and data bus are latched in the EPROM array when a subsequent write to the array is made. Data in the EPROM array cannot be read if the ELAT bit is set.

Whenever the ELAT bit is cleared, the EPGM bit is cleared also. Both the EPGM and the ELAT bit cannot be set using the same write instruction. Any attempt to set both the EPGM and ELAT bit on the same write instruction cycle will result in the ELAT bit being set and the EPGM bit being cleared. The ELAT bit is a read-write bit and can be read at any time. The ELAT bit is cleared by reset.

To program a byte of EPROM, manipulate the EPROG register as follows:

1. Set the ELAT bit in the EPROG register.
2. Write the desired data to the desired EPROM address.
3. Set the EPGM bit in the EPROG register for the specified programming time, t_{EPGM}.
4. Clear the ELAT and EPGM bits in the EPROG register.

This sequence is also shown in the sample program listing in Table 10-1.

Table 10-1. EPROM Programming Routine

```
001C                    EPROG   EQU $1C     PROGRAMMING REG
0055                    DATA    EQU $55     DATA VALUE
0700                    EPROM   EQU $700    A SAMPLE EPROM ADX
0000                    EPGM    EQU $00     EPGM BIT IN EPROG REG
00D0                    ORG     $D0
00D0    A6 02                   LDA  #$04   SET LAT BIT IN EPROG
00D2    B7 1C                   STA  EPROG
00D4    A6 55                   LDA  #DATA  DATA BYTE
00D6    C7 07 00                STA  EPROM  WRITE IT TO EPROM LOC
00D9    10 1C                   BSET  EPGM, EPROG  TURN ON PGM VOLTAGE
00DB    AD 03                   BSR  DELAY  WAIT 4 ms MINIMUM
00DD    3F 1C                   CLR  EPROG  CLR LAT AND PGM BITS
00DF    81                      RTS
```

I — Interrupt Mask Bit

When the interrupt mask is set, the internal and external interrupts are disabled. Interrupts are enabled when the interrupt mask is cleared. When an interrupt occurs, the interrupt mask is automatically set after the CPU registers are saved on the stack, but before the interrupt vector is fetched. If an interrupt request occurs while the interrupt mask is set, the interrupt request is latched. Normally, the interrupt is processed as soon as the interrupt mask is cleared.

A return from interrupt (RTI) instruction pulls the CPU registers from the stack, restoring the interrupt mask to its state before the interrupt was encountered. After any reset, the interrupt mask is set and can only be cleared by the clear I bit (CLI), STOP, or WAIT instructions.

N — Negative Bit

The negative bit is set when the result of the last arithmetic operation, logical operation, or data manipulation was negative. (Bit 7 of the result was a logic one.)

The negative bit can also be used to check an often-tested flag by assigning the flag to bit 7 of a register or memory location. Loading the accumulator with the contents of that register or location then sets or clears the negative bit according to the state of the flag.

Z — Zero Bit

The zero bit is set when the result of the last arithmetic operation, logical operation, data manipulation, or data load operation was zero.

C — Carry/Borrow Bit

The carry/borrow bit is set when a carry out of bit 7 of the accumulator occurred during the last arithmetic operation, logical operation, or data manipulation. The carry/borrow bit is also set or cleared during bit test and branch instructions and during shifts and rotates. This bit is not set by an INC or DEC instruction.

13.3.3 Jump/Branch Instructions

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

The BRCLR and BRSET instructions cause a branch based on the state of any readable bit in the first 256 memory locations. These 3-byte instructions use a combination of direct addressing and relative addressing. The direct address of the byte to be tested is in the byte following the opcode. The third byte is the signed offset byte. The CPU finds the effective branch destination by adding the third byte to the program counter if the specified bit tests true. The bit to be tested and its condition (set or clear) is part of the opcode. The span of branching is from –128 to +127 from the address of the next location after the branch instruction. The CPU also transfers the tested bit to the carry/borrow bit of the condition code register.

Table 13-3. Jump and Branch Instructions

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

Table 13-6. Instruction Set Summary (Sheet 3 of 6)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
CLR *opr* CLRA CLRX CLR *opr*,X CLR ,X	Clear Byte	M ← $00 A ← $00 X ← $00 M ← $00 M ← $00	—	—	0	1	—	DIR INH INH IX1 IX	3F 4F 5F 6F 7F	dd ff 	5 3 3 6 5
CMP #*opr* CMP *opr* CMP *opr* CMP *opr*,X CMP *opr*,X CMP ,X	Compare Accumulator with Memory Byte	(A) – (M)	—	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A1 B1 C1 D1 E1 F1	ii dd hh ll ee ff ff 	2 3 4 5 4 3
COM *opr* COMA COMX COM *opr*,X COM ,X	Complement Byte (One's Complement)	M ← ($\overline{M}$) = $FF – (M) A ← ($\overline{A}$) = $FF – (A) X ← ($\overline{X}$) = $FF – (X) M ← ($\overline{M}$) = $FF – (M) M ← ($\overline{M}$) = $FF – (M)	—	—	↕	↕	1	DIR INH INH IX1 IX	33 43 53 63 73	dd ff 	5 3 3 6 5
CPX #*opr* CPX *opr* CPX *opr* CPX *opr*,X CPX *opr*,X CPX ,X	Compare Index Register with Memory Byte	(X) – (M)	—	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A3 B3 C3 D3 E3 F3	ii dd hh ll ee ff ff 	2 3 4 5 4 3
DEC *opr* DECA DECX DEC *opr*,X DEC ,X	Decrement Byte	M ← (M) – 1 A ← (A) – 1 X ← (X) – 1 M ← (M) – 1 M ← (M) – 1	—	—	↕	↕	—	DIR INH INH IX1 IX	3A 4A 5A 6A 7A	dd ff 	5 3 3 6 5
EOR #*opr* EOR *opr* EOR *opr* EOR *opr*,X EOR *opr*,X EOR ,X	EXCLUSIVE OR Accumulator with Memory Byte	A ← (A) ⊕ (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	A8 B8 C8 D8 E8 F8	ii dd hh ll ee ff ff 	2 3 4 5 4 3
INC *opr* INCA INCX INC *opr*,X INC ,X	Increment Byte	M ← (M) + 1 A ← (A) + 1 X ← (X) + 1 M ← (M) + 1 M ← (M) + 1	—	—	↕	↕	—	DIR INH INH IX1 IX	3C 4C 5C 6C 7C	dd ff 	5 3 3 6 5
JMP *opr* JMP *opr* JMP *opr*,X JMP *opr*,X JMP ,X	Unconditional Jump	PC ← Jump Address	—	—	—	—	—	DIR EXT IX2 IX1 IX	BC CC DC EC FC	dd hh ll ee ff ff 	2 3 4 3 2
JSR *opr* JSR *opr* JSR *opr*,X JSR *opr*,X JSR ,X	Jump to Subroutine	PC ← (PC) + n (n = 1, 2, or 3) Push (PCL); SP ← (SP) – 1 Push (PCH); SP ← (SP) – 1 PC ← Effective Address	—	—	—	—	—	DIR EXT IX2 IX1 IX	BD CD DD ED FD	dd hh ll ee ff ff 	5 6 7 6 5

Table 13-6. Instruction Set Summary (Sheet 5 of 6)

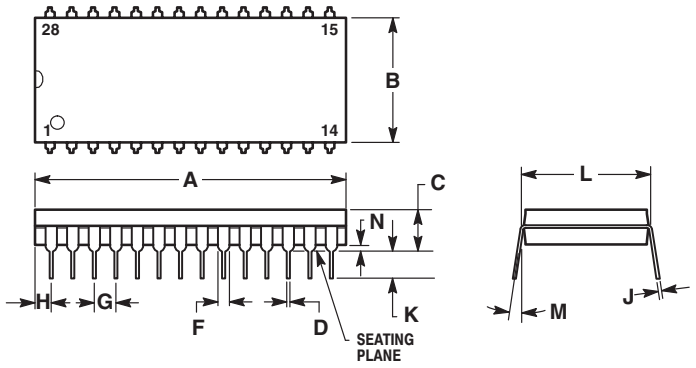
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
RTI	Return from Interrupt	SP ← (SP) + 1; Pull (CCR) SP ← (SP) + 1; Pull (A) SP ← (SP) + 1; Pull (X) SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)	↕	↕	↕	↕	↕	INH	80		9
RTS	Return from Subroutine	SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)	—	—	—	—	—	INH	81		6
SBC #*opr* SBC *opr* SBC *opr* SBC *opr*,X SBC *opr*,X SBC ,X	Subtract Memory Byte and Carry Bit from Accumulator	A ← (A) – (M) – (C)	—	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A2 B2 C2 D2 E2 F2	ii dd hh ll ee ff ff	2 3 4 5 4 3
SEC	Set Carry Bit	C ← 1	—	—	—	—	1	INH	99		2
SEI	Set Interrupt Mask	I ← 1	—	1	—	—	—	INH	9B		2
STA *opr* STA *opr* STA *opr*,X STA *opr*,X STA ,X	Store Accumulator in Memory	M ← (A)	—	—	↕	↕	—	DIR EXT IX2 IX1 IX	B7 C7 D7 E7 F7	dd hh ll ee ff ff	4 5 6 5 4
STOP	Stop Oscillator and Enable IRQ Pin		—	0	—	—	—	INH	8E		2
STX *opr* STX *opr* STX *opr*,X STX *opr*,X STX ,X	Store Index Register In Memory	M ← (X)	—	—	↕	↕	—	DIR EXT IX2 IX1 IX	BF CF DF EF FF	dd hh ll ee ff ff	4 5 6 5 4
SUB #*opr* SUB *opr* SUB *opr* SUB *opr*,X SUB *opr*,X SUB ,X	Subtract Memory Byte from Accumulator	A ← (A) – (M)	—	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A0 B0 C0 D0 E0 F0	ii dd hh ll ee ff ff	2 3 4 5 4 3
SWI	Software Interrupt	PC ← (PC) + 1; Push (PCL) SP ← (SP) – 1; Push (PCH) SP ← (SP) – 1; Push (X) SP ← (SP) – 1; Push (A) SP ← (SP) – 1; Push (CCR) SP ← (SP) – 1; I ← 1 PCH ← Interrupt Vector High Byte PCL ← Interrupt Vector Low Byte	—	1	—	—	—	INH	83		1 0
TAX	Transfer Accumulator to Index Register	X ← (A)	—	—	—	—	—	INH	97		2
TST *opr* TSTA TSTX TST *opr*,X TST ,X	Test Memory Byte for Negative or Zero	(M) – $00	—	—	↕	↕	—	DIR INH INH IX1 IX	3D 4D 5D 6D 7D	dd ff	4 3 3 5 4

Chapter 15
Mechanical Specifications

15.1 Introduction

The MC68HC705P6A is available in either a 28-pin plastic dual in-line (PDIP) or a 28-pin small outline integrated circuit (SOIC) package.

15.2 Plastic Dual In-Line Package (Case 710)



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

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