



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	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc705p6acdwer

MC68HC705P6A

Advance Information Data Sheet

To provide the most up-to-date information, the revision of our documents on the World Wide Web will be the most current. Your printed copy may be an earlier revision. To verify you have the latest information available, refer to:

<http://www.freescale.com/>

The following revision history table summarizes changes contained in this document. For your convenience, the page number designators have been linked to the appropriate location.

Revision History

Date	Revision Level	Description	Page Number(s)
November, 2001	2.0	Format update to current publication standards	N/A
		Figure 11-1. Mask Option Register (MOR) — Definition of bit 6 corrected.	92
September, 2005	2.1	Updated to meet Freescale identity guidelines.	Throughout

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc.

© Freescale Semiconductor, Inc., 2005. All rights reserved.



List of Chapters

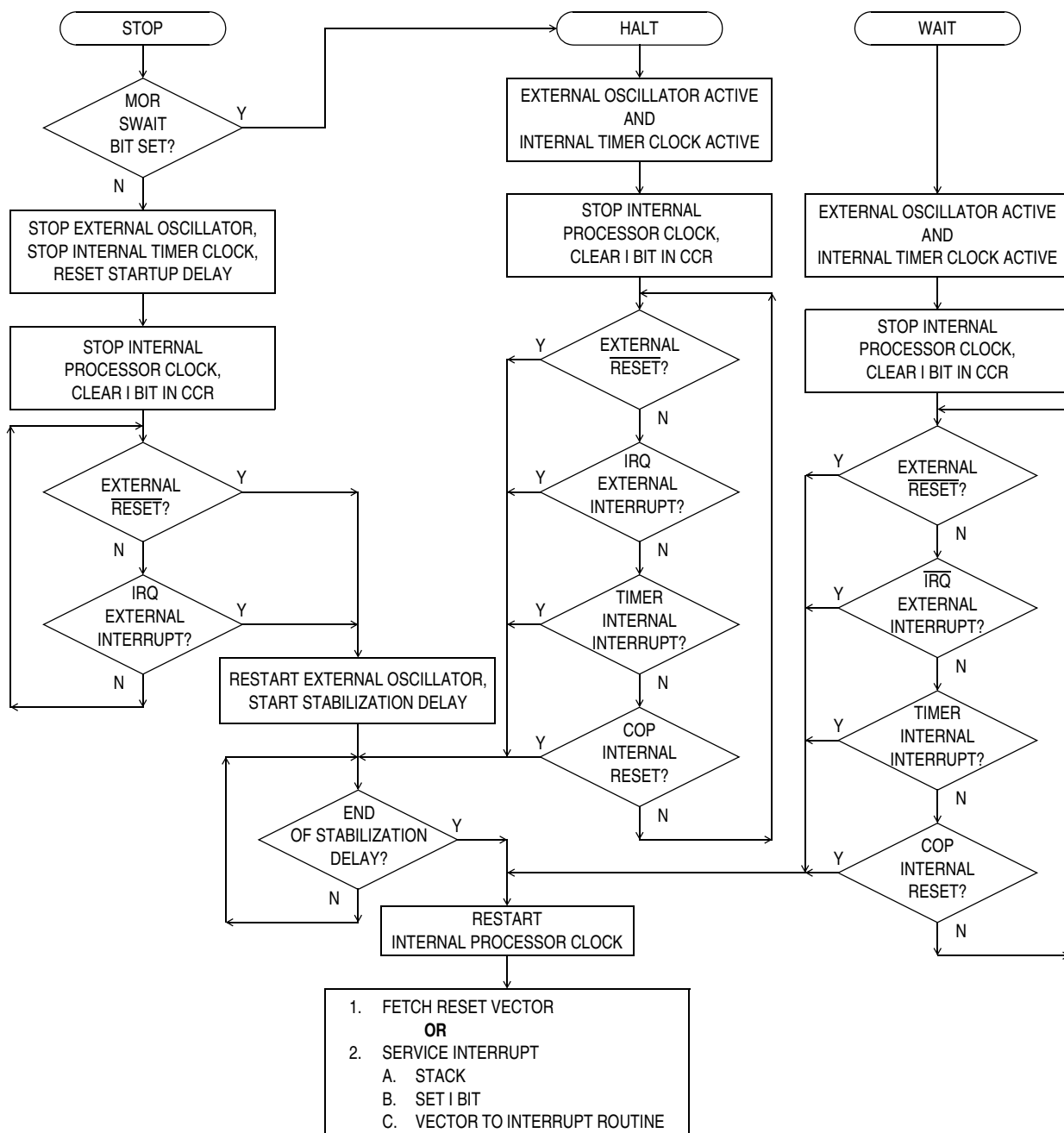


Figure 3-2. STOP/WAIT Flowcharts

3.4.1.2 Halt Mode

NOTE

*Halt mode is **NOT** designed for intentional use. Halt mode is only provided to keep the COP watchdog timer active in the event a STOP instruction is executed inadvertently. This mode of operation is usually achieved by invoking wait mode.*

Execution of the STOP instruction when the SWAIT bit in the MOR is set places the MCU in this low-power mode. Halt mode consumes the same amount of power as wait mode (both halt and wait modes consume more power than stop mode).

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

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

3.4.2 WAIT Instruction

The WAIT instruction places the MCU in a low-power mode which consumes more power than stop mode. In wait mode, the internal clock is halted, suspending all processor and internal bus activity. Internal timer clocks remain active, permitting interrupts to be generated from the 16-bit timer and reset to be generated from the COP watchdog timer. Execution of the WAIT instruction automatically clears the I bit in the condition code register, enabling the $\overline{\text{IRQ}}$ external interrupt. All other registers, memory, and input/output lines remain in their previous state.

If the 16-bit timer interrupt is enabled, it will cause the processor to exit wait mode and resume normal operation. The 16-bit timer may be used to generate a periodic exit from wait mode. Wait mode may also be exited when an $\overline{\text{IRQ}}$ external interrupt or $\overline{\text{RESET}}$ occurs.

3.5 COP Watchdog Timer Considerations

The COP watchdog timer is active in user mode of operation when the COP bit in the MOR is set. Executing the STOP instruction when the SWAIT bit in the MOR is clear will cause the COP to be disabled. Therefore, it is recommended that the STOP instruction be modified to produce halt mode (set bit SWAIT in the MOR) if the COP watchdog timer is required to function at all times.

Furthermore, it is recommended that the COP watchdog timer be disabled for applications that will use the wait mode for time periods that will exceed the COP timeout period.

5.2 Interrupt Types

The interrupts fall into three categories: reset, software, and hardware.

5.2.1 Reset Interrupt Sequence

The reset function is not in the strictest sense an interrupt; however, it is acted upon in a similar manner as shown in [Figure 5-1](#). A low-level input on the $\overline{\text{RESET}}$ pin or internally generated RST signal causes the program to vector to its starting address which is specified by the contents of memory locations \$1FFE and \$1FFF. The I bit in the condition code register is also set. The MCU is configured to a known state during this type of reset as previously described in [Chapter 4 Resets](#).

5.2.2 Software Interrupt (SWI)

The SWI is an executable instruction. It is also a non-maskable interrupt since it is executed regardless of the state of the I bit in the CCR. As with any instruction, interrupts pending during the previous instruction will be serviced before the SWI opcode is fetched. The interrupt service routine address for the SWI instruction is specified by the contents of memory locations \$1FFC and \$1FFD.

5.2.3 Hardware Interrupts

All hardware interrupts are maskable by the I bit in the CCR. If the I bit is set, all hardware interrupts (internal and external) are disabled. Clearing the I bit enables the hardware interrupts. Four hardware interrupts are explained in the following subsections.

5.2.3.1 External Interrupt ($\overline{\text{IRQ}}$)

The $\overline{\text{IRQ}}/\text{V}_{\text{PP}}$ pin drives an asynchronous interrupt to the CPU. An edge detector flip-flop is latched on the falling edge of $\overline{\text{IRQ}}/\text{V}_{\text{PP}}$. If either the output from the internal edge detector flip-flop or the level on the $\overline{\text{IRQ}}/\text{V}_{\text{PP}}$ pin is low, a request is synchronized to the CPU to generate the IRQ interrupt. If the LEVEL bit in the mask option register is clear (edge-sensitive only), the output of the internal edge detector flip-flop is sampled and the input level on the $\overline{\text{IRQ}}/\text{V}_{\text{PP}}$ pin is ignored. The interrupt service routine address is specified by the contents of memory locations \$1FFA and \$1FFB. If the port A interrupts are enabled by the MOR, they generate external interrupts identically to the $\overline{\text{IRQ}}/\text{V}_{\text{PP}}$ pin.

NOTE

The internal interrupt latch is cleared nine internal clock cycles after the interrupt is recognized (immediately after location \$1FFA is read). Therefore, another external interrupt pulse could be latched during the $\overline{\text{IRQ}}$ service routine.

Another interrupt will be serviced if the $\overline{\text{IRQ}}$ pin is still in a low state when the RTI in the service routine is executed.

5.2.3.2 Input Capture Interrupt

The input capture interrupt is generated by the 16-bit timer as described in [Chapter 8 Capture/Compare Timer](#). The input capture interrupt flag is located in register TSR and its corresponding enable bit can be found in register TCR. The I bit in the CCR must be clear for the input capture interrupt to be enabled. The interrupt service routine address is specified by the contents of memory locations \$1FF8 and \$1FF9.

5.2.3.3 Output Compare Interrupt

The output compare interrupt is generated by a 16-bit timer as described in [Chapter 8 Capture/Compare Timer](#). The output compare interrupt flag is located in register TSR and its corresponding enable bit can be found in register TCR. The I bit in the CCR must be clear for the output compare interrupt to be enabled. The interrupt service routine address is specified by the contents of memory locations \$1FF8 and \$1FF9.

5.2.3.4 Timer Overflow Interrupt

The timer overflow interrupt is generated by the 16-bit timer as described in [Chapter 8 Capture/Compare Timer](#). The timer overflow interrupt flag is located in register TSR and its corresponding enable bit can be found in register TCR. The I bit in the CCR must be clear for the timer overflow interrupt to be enabled. This internal interrupt will vector to the interrupt service routine located at the address specified by the contents of memory locations \$1FF8 and \$1FF9.

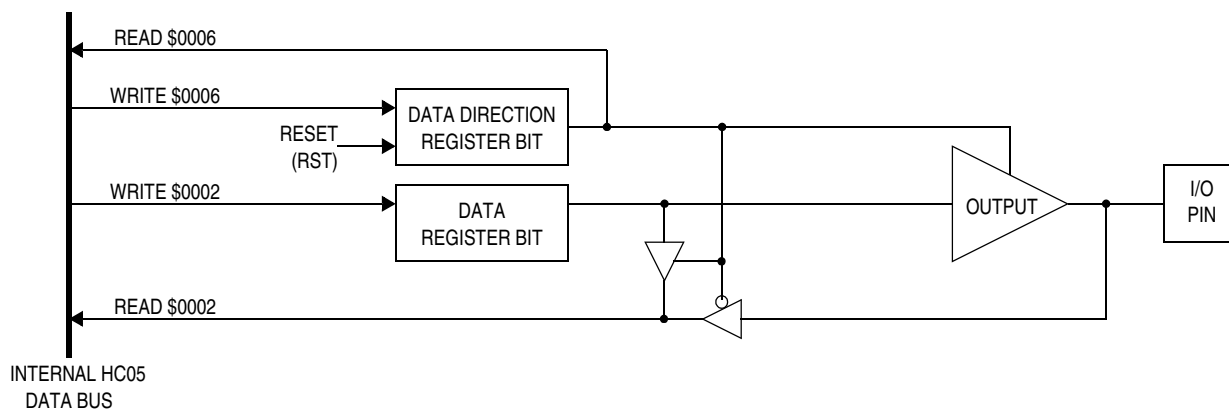


Figure 6-3. Port C I/O Circuitry

6.5 Port D

Port D is a 2-bit port with one bidirectional pin (PD5) and one input-only pin (PD7). Pin PD7 is shared with the 16-bit timer. The port D data register is located at address \$0003 and its data direction register (DDR) is located at address \$0007. The contents of the port D data register are indeterminate at initial powerup and must be initialized by user software. Reset does not affect the data registers, but clears the DDRs, thereby setting PD5 to input mode. Writing a 1 to DDR bit 5 sets PD5 to output mode (see [Figure 6-4](#)).

Port D may be used for general I/O applications regardless of the state of the 16-bit timer. Since PD7 is an input-only line, its state can be read from the port D data register at any time.

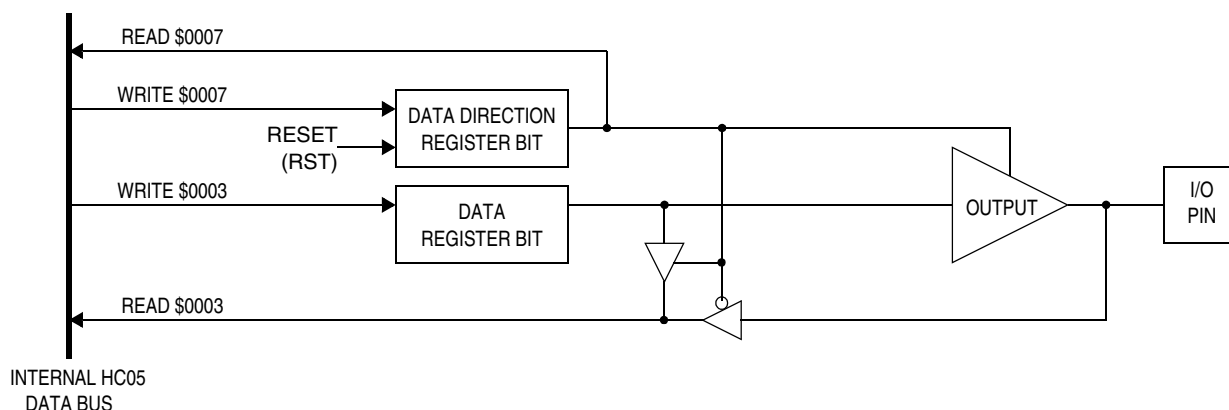


Figure 6-4. Port D I/O Circuitry

6.6 I/O Port Programming

Each pin on port A through port D (except pin 7 of port D) can be programmed as an input or an output under software control as shown in [Table 6-1](#), [Table 6-2](#), [Table 6-3](#), and [Table 6-4](#). The direction of a pin is determined by the state of its corresponding bit in the associated port data direction register (DDR). A pin is configured as an output if its corresponding DDR bit is set to a logic 1. A pin is configured as an input if its corresponding DDR bit is cleared to a logic 0.

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.

Address: \$000A

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	SPE	0	MSTR	0	0	0	0
Write:								
Reset:	0	0	0	0	0	0	0	0


 = Unimplemented

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:	SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0
Write:								
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.

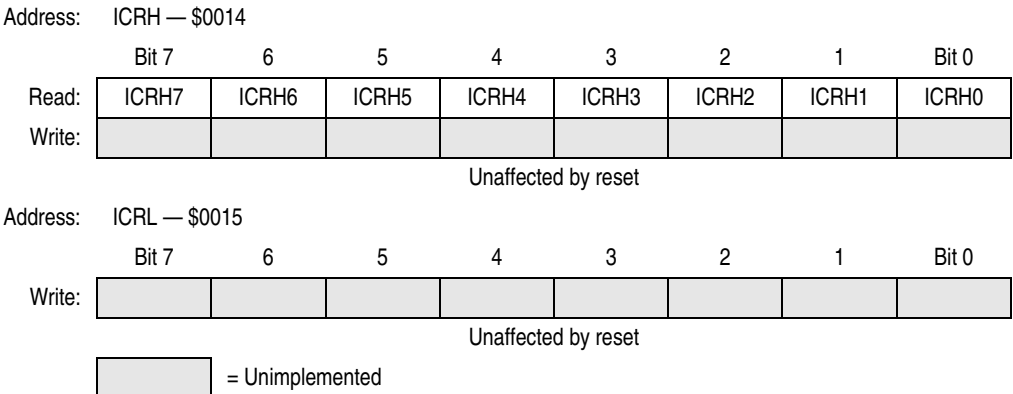


Figure 8-6. Input Capture Registers (ICRH and ICRL)

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).

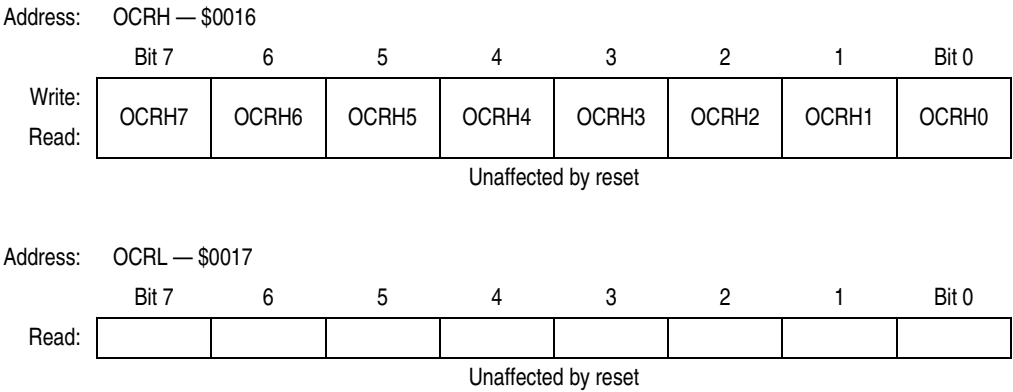


Figure 8-7. Output Compare Registers (OCRH and OCRL)

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.

11.3 MOR Programming

The contents of the MOR should be programmed in bootloader mode using the hardware shown in [Figure 10-2. MC68HC705P6A EPROM Programming Flowchart](#). In order to allow programming, all the implemented bits in the MOR are essentially read-write bits in bootloader mode as shown in [Figure 11-1](#).

The programming of the MOR is the same as user EPROM.

1. Set the ELAT bit in the EPROG register.
2. Write the desired data to the desired MOR address.
3. Set the EPGM bit in the EPROG.
4. Wait for the programming time (t_{EPGM}).
5. Clear the ELAT and EPGM bits in the EPROG.
6. Remove the programming voltage from the \overline{IRQ}/V_{PP} pin.

A sample routine to program a byte of EPROM is shown in [Table 11-2](#).

Once the MOR bits have been programmed, the options are not loaded into the MOR registers until the part is reset.

Table 11-2. MOR Programming Routine

001C		EPROG	EQU	\$1C	PROGRAMMING REG
00FF		DATA2	EQU	\$FF	SAMPLE MOR VALUES
0023		DATA1	EQU	#23	
1EFF		MOR2	EQU	\$1EFF	MOPR ADDRESSES
1F00		MOR1	EQU	\$1F00	
0000		EPGM	EQU	\$00	EPGM BIT IN EPROG REG
00E0			ORG	\$E0	
00E0	A6 04	LDA	#\$04		SET ELAT BIT
00E2	B7 1C	STA	EPROG		IN EPGM REG AT \$1C
00E4	A6 FF	LDA	#DATA2		DATA BYTE
00E6	C7 1E FF	STA	MOR2		WRITE IT TO MOR LOC
00E9	12 1C	BSET	EPGM,EPROG		TURN ON PGM VOLTAGE
00EB	AD 03	BSR	DELAY		WAIT 4 ms MINIMUM
00ED	3F 1C	CLR	EPROG		CLR EPGM REGISTER
00EF	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.



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	$SP \leftarrow (SP) + 1$; Pull (CCR) $SP \leftarrow (SP) + 1$; Pull (A) $SP \leftarrow (SP) + 1$; Pull (X) $SP \leftarrow (SP) + 1$; Pull (PCH) $SP \leftarrow (SP) + 1$; Pull (PCL)	†	†	†	†	†	INH	80		9
RTS	Return from Subroutine	$SP \leftarrow (SP) + 1$; Pull (PCH) $SP \leftarrow (SP) + 1$; 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	$A \leftarrow (A) - (M) - (C)$	—	—	†	†	†	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	$C \leftarrow 1$	—	—	—	—	1	INH	99		2
SEI	Set Interrupt Mask	$I \leftarrow 1$	—	1	—	—	—	INH	9B		2
STA opr STA opr STA opr,X STA opr,X STA ,X	Store Accumulator in Memory	$M \leftarrow (A)$	—	—	†	†	—	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	$M \leftarrow (X)$	—	—	†	†	—	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	$A \leftarrow (A) - (M)$	—	—	†	†	†	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	$PC \leftarrow (PC) + 1$; Push (PCL) $SP \leftarrow (SP) - 1$; Push (PCH) $SP \leftarrow (SP) - 1$; Push (X) $SP \leftarrow (SP) - 1$; Push (A) $SP \leftarrow (SP) - 1$; Push (CCR) $SP \leftarrow (SP) - 1$; $I \leftarrow 1$ PCH ← Interrupt Vector High Byte PCL ← Interrupt Vector Low Byte	—	1	—	—	—	INH	83		1 0
TAX	Transfer Accumulator to Index Register	$X \leftarrow (A)$	—	—	—	—	—	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

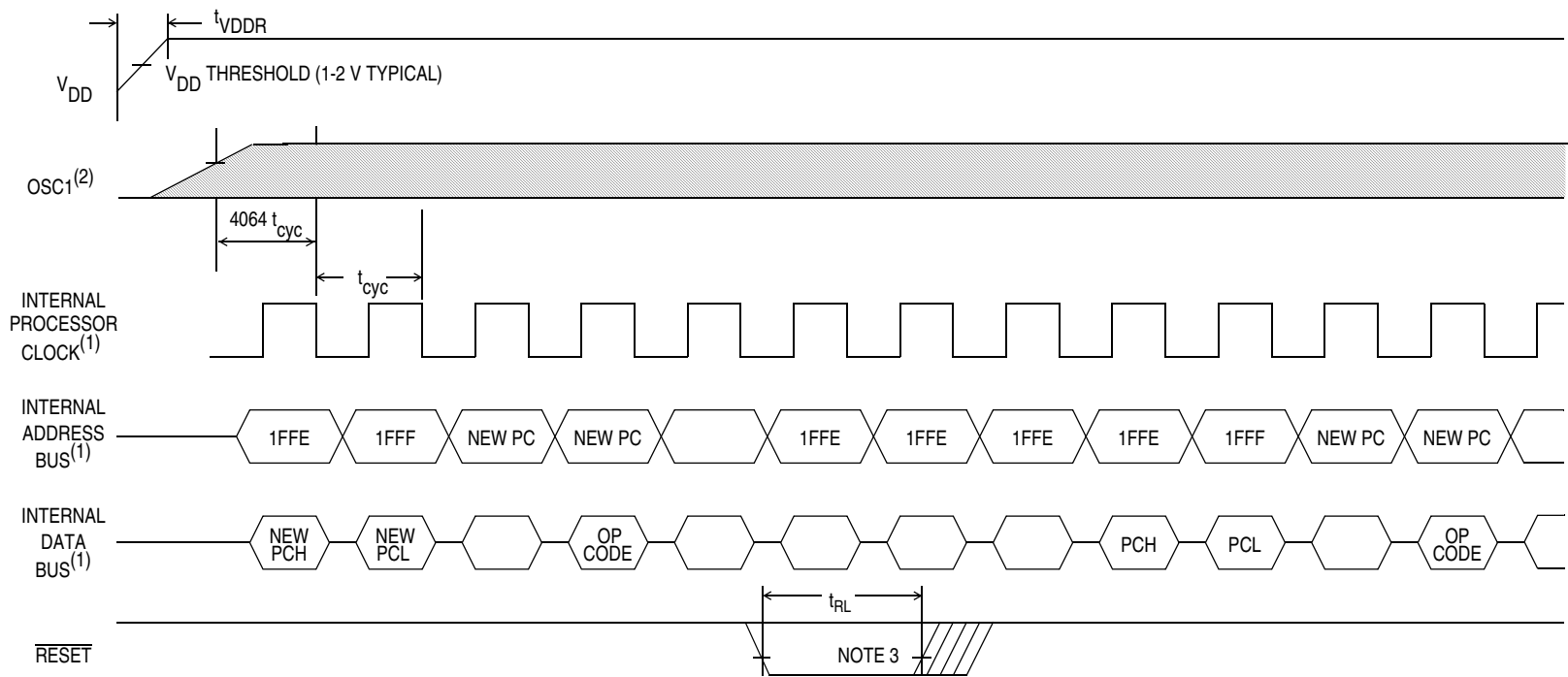
Table 13-6. Instruction Set Summary (Sheet 6 of 6)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
TXA	Transfer Index Register to Accumulator	$A \leftarrow (X)$	—	—	—	—	—	INH	9F		2
WAIT	Stop CPU Clock and Enable Interrupts		—	0	—	—	—	INH	8F		2

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

13.5 Opcode Map

See [Table 13-7](#).



Notes:

1. Internal timing signal and bus information are not available externally.
2. OSC1 line is not meant to represent frequency. It is only used to represent time.
3. The next rising edge of the internal clock following the rising edge of RESET initiates the reset sequence.

Figure 14-2. Power-On Reset and External Reset Timing Diagram



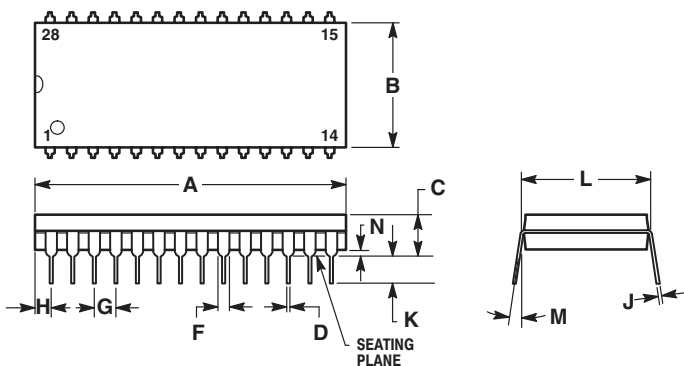
Chapter 15

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.

DIM	MILLIMETERS		INCHES	
	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



