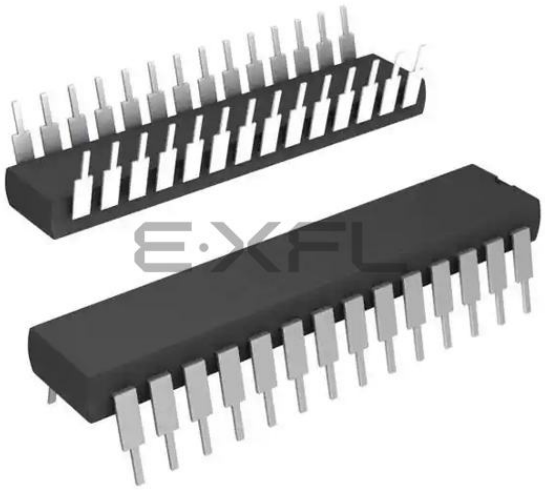


Welcome to E-XFL.COM

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/mc705p6ecpe

Chapter 4
Resets

Chapter 5
Interrupts

Chapter 6
Input/Output Ports

Chapter 7
Serial Input/Output Port (SIOP)

Chapter 1
General Description

1.1 Introduction

The MC68HC705P6A is an EPROM version of the MC68HC05P6 microcontroller. It is a low-cost combination of an M68HC05 Family microprocessor with a 4-channel, 8-bit analog-to-digital (A/D) converter, a 16-bit timer with output compare and input capture, a serial communications port (SIOP), and a computer operating properly (COP) watchdog timer. The M68HC05 CPU core contains 176 bytes of RAM, 4672 bytes of user EPROM, 239 bytes of bootloader ROM, and 21 input/output (I/O) pins (20 bidirectional, 1 input-only). This device is available in either a 28-pin plastic dual in-line (PDIP) or a 28-pin small outline integrated circuit (SOIC) package.

A functional block diagram of the MC68HC705P6A is shown in Figure 1-1.

1.2 Features

Features of the MC68HC705P6A include:

- Low cost
- M68HC05 core
- 28-pin SOIC, PDIP, or windowed DIP package
- 4672 bytes of user EPROM (including 48 bytes of page zero EPROM and 16 bytes of user vectors)
- 239 bytes of bootloader ROM
- 176 bytes of on-chip RAM
- 4-channel 8-bit A/D converter
- SIOP serial communications port
- 16-bit timer with output compare and input capture
- 20 bidirectional I/O lines and 1 input-only line
- PC0 and PC1 high-current outputs
- Single-chip, bootloader, and test modes
- Power-saving stop, halt, and wait modes
- Static EPROM mask option register (MOR) selectable options:
  - COP watchdog timer enable or disable
  - Edge-sensitive or edge- and level-sensitive external interrupt
  - SIOP most significant bit (MSB) or least significant bit (LSB) first
  - SIOP clock rates: OSC divided by 8, 16, 32, or 64
  - Stop instruction mode, STOP or HALT
  - EPROM security external lockout
  - Programmable keyscan (pullups/interrupts) on PA0–PA7

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$001B	Alternate Timer Register LSB (ATRL) See page 49.	Read:	ACRL7	ACRL6	ACRL5	ACRL4	ACRL3	ACRL2	ACRL1	ACRL0
		Write:								
		Reset:	1	1	1	1	1	1	0	0
$001C	EPROM Programming Register (EPROG) See page 58.	Read:	0	0	0	0	0	ELAT	0	EPGM
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$001D	A/D Conversion Value Data Register (ADC) See page 55.	Read:	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
		Write:								
		Reset:	Unaffected by reset							
$001E	A/D Status and Control Register (ADSC) See page 54.	Read:	CC	ADRC	ADON	0	0	CH2	CH1	CH0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$001F	Reserved for Test		R	R	R	R	R	R	R	R

= Unimplemented R = Reserved U = Undetermined

Figure 2-3. I/O and Control Register Summary (Sheet 3 of 3)

2.7 Mask Option Register

The mask option register (MOR) is a pair of EPROM bytes located at $1EFF and $1F00. It controls the programmable options on the MC68HC705P6A. See Chapter 11 Mask Option Register (MOR) for additional information.

$1EFF	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PA7PU	PA6PU	PA5PU	PA4PU	PA3PU	PA2PU	PA1PU	PA0PU
Erased State:	0	0	0	0	0	0	0	0

$1F00	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	SECURE		SWAIT	SPR1	SPR0	LSBF	LEVEL	COP
Erased State:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 2-4. Mask Option Register (MOR)

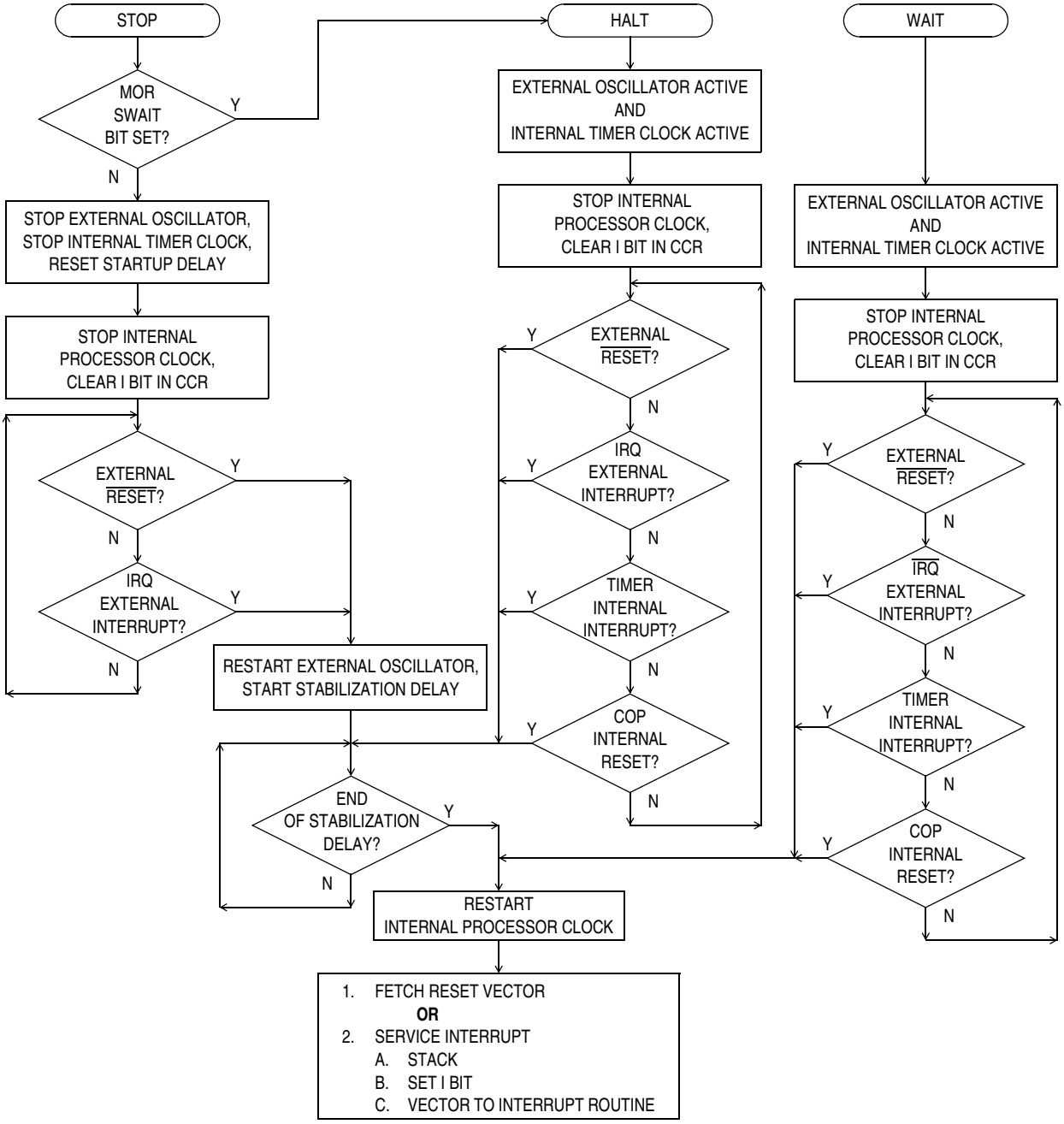


Figure 3-2. STOP/WAIT Flowcharts

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 ($\overline{IRQ}$)

The $\overline{\text{IRQ}}/V_{PP}$ pin drives an asynchronous interrupt to the CPU. An edge detector flip-flop is latched on the falling edge of $\overline{\text{IRQ}}/V_{PP}$. If either the output from the internal edge detector flip-flop or the level on the $\overline{\text{IRQ}}/V_{PP}$ 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 $\overline{\text{IRQ}}/V_{PP}$ 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 $\overline{\text{IRQ}}/V_{PP}$ 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 $\overline{IRQ}$ service routine.

Another interrupt will be serviced if the $\overline{IRQ}$ 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.

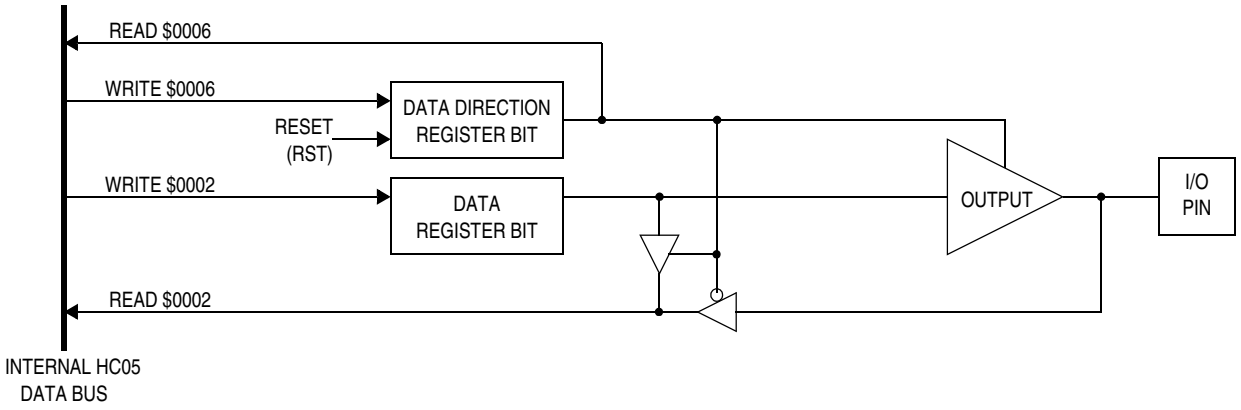


Figure 6-3. Port C I/O Circuitry

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.

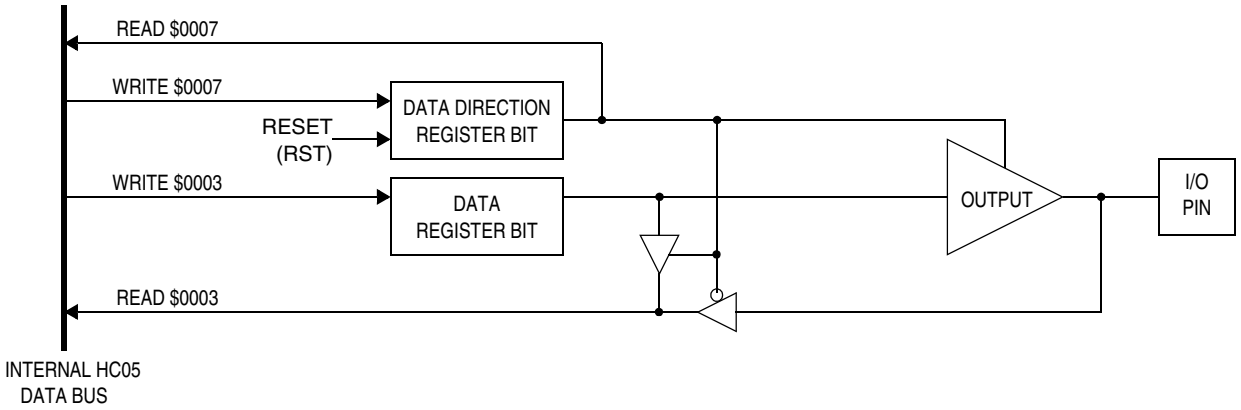


Figure 6-4. Port D I/O Circuitry

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.

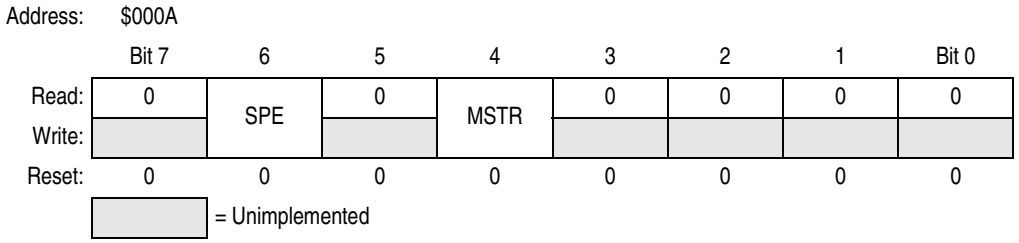


Figure 7-3. SIOP Control Register (SCR)

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.

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.

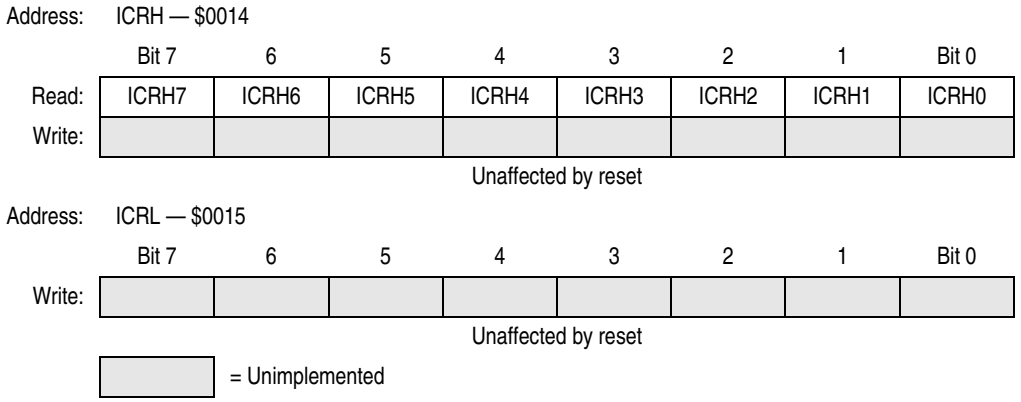


Figure 8-6. Input Capture Registers (ICRH and ICRL)

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).

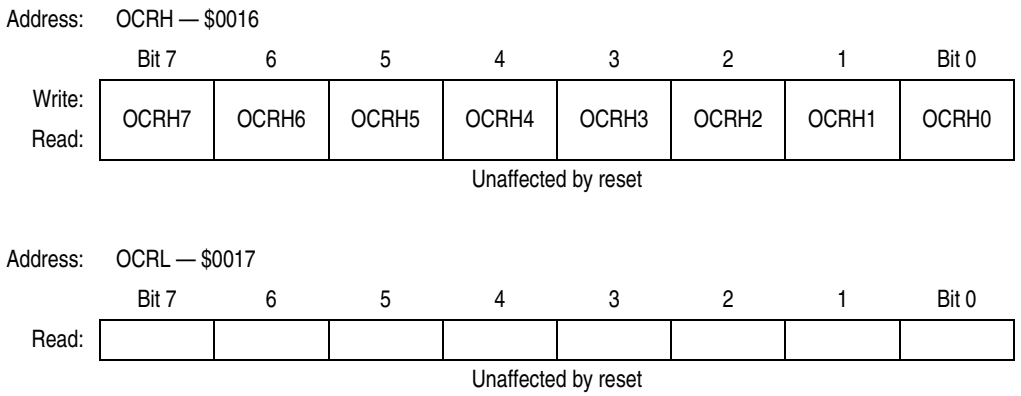


Figure 8-7. Output Compare Registers (OCRH and OCRL)

To prevent OCF from being set between the time it is read and the time the output compare registers are updated, use this procedure:

1. Disable interrupts by setting the I bit in the condition code register.
2. Write to OCRH. Compares are now inhibited until OCRL is written.
3. Clear bit OCF by reading timer status register (TSR).
4. Enable the output compare function by writing to OCRL.
5. Enable interrupts by clearing the I bit in the condition code register.

8.4 Timer During Wait/Halt Mode

The CPU clock halts during the wait (or halt) mode, but the timer remains active. If interrupts are enabled, a timer interrupt will cause the processor to exit the wait mode.

8.5 Timer During Stop Mode

In the stop mode, the timer stops counting and holds the last count value if STOP is exited by an interrupt. If STOP is exited by RESET, the counters are forced to $FFFC. During STOP, if at least one valid input capture edge occurs at the TCAP pins, the input capture detect circuit is armed. This does not set any timer flags or wake up the MCU, but if an interrupt is used to exit stop mode, there is an active input capture flag and data from the first valid edge that occurred during the stop mode. If reset is used to exit stop mode, then no input capture flag or data remains, even if a valid input capture edge occurred.

ADRC — RC Oscillator Control

When ADRC is set, the A/D subsystem operates from the internal RC oscillator instead of the internal clock. The RC oscillator requires a time, t_{RCON}, to stabilize before accurate conversion results can be obtained. See 9.2.2 Reference Voltage (VREFH) for more information.

ADON — A/D Subsystem On

When the A/D subsystem is turned on (ADON = 1), it requires a time, t_{ADON}, to stabilize before accurate conversion results can be attained.

CH2–CH0 — Channel Select Bits

CH2, CH1, and CH0 form a 3-bit field which is used to select an input to the A/D converter. Channels 0–3 correspond to port C input pins PC6–PC3. Channels 4–6 are used for reference measurements. Channel 7 is reserved. If a conversion is attempted with channel 7 selected, the result will be $00. Table 9-1 lists the inputs selected by bits CH0-CH3.

If the ADON bit is set and an input from channels 0–4 is selected, the corresponding port C pin's DDR bit will be cleared (making that port C pin an input). If the port C data register is read while the A/D is on and one of the shared input channels is selected using bit CH0–CH2, the corresponding port C pin will read as a logic 0. The remaining port C pins will read normally. To digitally read a port C pin, the A/D subsystem must be disabled (ADON = 0), or input channels 5–7 must be selected.

Table 9-1. A/D Multiplexer Input Channel Assignments

Channel	Signal
0	AD0 — port C, bit 6
1	AD1 — port C, bit 5
2	AD2 — port C, bit 4
3	AD3 — port C, bit 3
4	V_{REFH} — port C, bit 7
5	$(V_{REFH} + V_{SS})/2$
6	V_{SS}
7	Reserved for factory test

9.6 A/D Conversion Data Register (ADC)

This register contains the output of the A/D converter. See Figure 9-2.

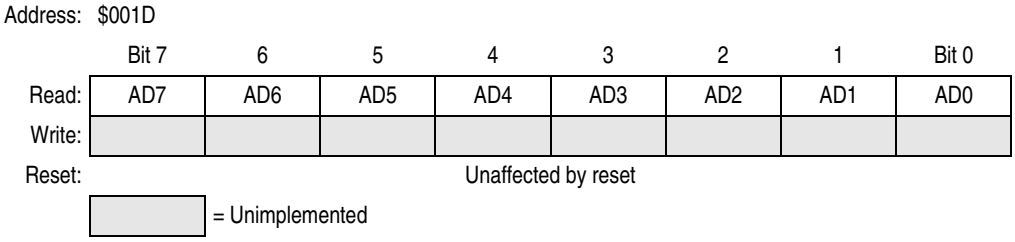


Figure 9-2. A/D Conversion Value Data Register (ADC)

13.2.4 Extended

Extended instructions use three bytes and can access any address in memory. The first byte is the opcode; the second and third bytes are the high and low bytes of the operand address.

When using the Freescale assembler, the programmer does not need to specify whether an instruction is direct or extended. The assembler automatically selects the shortest form of the instruction.

13.2.5 Indexed, No Offset

Indexed instructions with no offset are 1-byte instructions that can access data with variable addresses within the first 256 memory locations. The index register contains the low byte of the effective address of the operand. The CPU automatically uses $00 as the high byte, so these instructions can address locations $0000–$00FF.

Indexed, no offset instructions are often used to move a pointer through a table or to hold the address of a frequently used RAM or I/O location.

13.2.6 Indexed, 8-Bit Offset

Indexed, 8-bit offset instructions are 2-byte instructions that can access data with variable addresses within the first 511 memory locations. The CPU adds the unsigned byte in the index register to the unsigned byte following the opcode. The sum is the effective address of the operand. These instructions can access locations $0000–$01FE.

Indexed 8-bit offset instructions are useful for selecting the kth element in an n-element table. The table can begin anywhere within the first 256 memory locations and could extend as far as location 510 ($01FE). The k value is typically in the index register, and the address of the beginning of the table is in the byte following the opcode.

13.2.7 Indexed,16-Bit Offset

Indexed, 16-bit offset instructions are 3-byte instructions that can access data with variable addresses at any location in memory. The CPU adds the unsigned byte in the index register to the two unsigned bytes following the opcode. The sum is the effective address of the operand. The first byte after the opcode is the high byte of the 16-bit offset; the second byte is the low byte of the offset.

Indexed, 16-bit offset instructions are useful for selecting the kth element in an n-element table anywhere in memory.

As with direct and extended addressing, the Freescale assembler determines the shortest form of indexed addressing.

13.2.8 Relative

Relative addressing is only for branch instructions. If the branch condition is true, the CPU finds the effective branch destination by adding the signed byte following the opcode to the contents of the program counter. If the branch condition is not true, the CPU goes to the next instruction. The offset is a signed, two’s complement byte that gives a branching range of –128 to +127 bytes from the address of the next location after the branch instruction.

When using the Freescale assembler, the programmer does not need to calculate the offset, because the assembler determines the proper offset and verifies that it is within the span of the branch.

Chapter 14
Electrical Specifications

14.1 Introduction

This section contains the electrical and timing specifications.

14.2 Maximum Ratings

Maximum ratings are the extreme limits to which the MCU can be exposed without permanently damaging it.

The MCU contains circuitry to protect the inputs against damage from high static voltages; however, do not apply voltages higher than those shown in the table below. Keep V_{In} and V_{Out} within the range $V_{SS} \le (V_{In}$ or $V_{Out}) \le V_{DD}$. Connect unused inputs to the appropriate voltage level, either V_{SS} or V_{DD}.

Rating[1]	Symbol	Value	Unit
Supply voltage	V_{DD}	–0.3 to +7.0	V
Input voltage	V_{In}	V_{SS} –0.3 to V_{DD} +0.3	V
Bootloader mode ($\overline{IRQ}/V_{PP}$ pin only)	V_{In}	V_{SS} –0.3 to 2 x V_{DD} +0.3	V
Current drain per pin excluding V_{DD} and V_{SS}	I	25	mA
Storage temperature range	T_{stg}	–65 to +150	°C

1. Voltages are referenced to V_{SS}.

NOTE

This device is not guaranteed to operate properly at the maximum ratings. Refer to 14.5 5.0-Volt DC Electrical Characteristics and 14.6 3.3-Volt DC Electrical Charactertistics for guaranteed operating conditions.

14.3 Operating Temperature Range

Characteristic	Symbol	Value	Unit
Operating temperature range MC68HC705P6A (standard) MC68HC705P6AC (extended)	T_A	T_L to T_H 0 to +70 –40 to +85	°C

14.4 Thermal Characteristics

Characteristic	Symbol	Value	Unit
Thermal resistance PDIP SOIC	θ_{JA}	 60 60	°C/W

14.7 A/D Converter Characteristics

Characteristic[1]	Min	Max	Unit	Comments
Resolution	8	8	Bits	
Absolute accuacy ($V_{DD} \geq V_{REFH} > 4.0$)	—	± 1 1/2	LSB	Including quanitization
Conversion range V_{REFH}	V_{SS} V_{SS}	V_{REFH} V_{DD}	V	A/D accuracy may decrease proportionately as V_{REFH} is reduced below 4.0
Input leakage AD0, AD1, AD2, AD3 V_{REFH}	— —	± 1 ± 1	µA	
Conversion time MCU external oscillator Internal RC oscillator	— —	32 32	t_{cyc} µs	Includes sampling time
Monotonicity	Inherent (within total error)			
Zero input reading	00	01	Hex	V_{in} = 0 V
Full-scale reading	FE	FF	Hex	$V_{in} = V_{REFH}$
Sample time MCU external oscillator Internal RC oscillator	— —	12 12	t_{cyc} µs	
Input capacitance	—	12	pF	
Analog input voltage	V_{SS}	V_{REFH}	V	
A/D on current stabilization time	—	100	µs	t_{ADON}
A/D ports hi-z leakage current (PC3:7)	—	± 1	µA	I_{OZ}

1. V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = –40°C to +85°C, unless otherwise noted.

14.8 EPROM Programming Characteristics

Characteristic	Symbol	Min	Typ	Max	Unit
Programming voltage $\overline{IRQ}/V_{PP}$	V_{PP}	16.25	16.5	16.75	V
Programming current $\overline{IRQ}/V_{PP}$	I_{PP}	—	5.0	10	mA
Programming time per byte	t_{EPGM}	4	—	—	ms

14.9 SIOP Timing

Number	Characteristic	Symbol	Min	Max	Unit
	Operating frequency Master Slave	$f_{op(m)}$ $f_{op(s)}$	0.25 dc	0.25 0.25	f_{op}
1	Cycle time Master Slave	$t_{cyc(m)}$ $t_{cyc(s)}$	4.0 —	4.0 4.0	t_{cyc}
2	SCK low time	t_{cyc}	932	—	ns
3	SDO data valid time	t_v	—	200	ns
4	SDO hold time	t_{ho}	0	—	ns
5	SDI setup time	t_s	100	—	ns
6	SDI hold time	t_h	100	—	ns

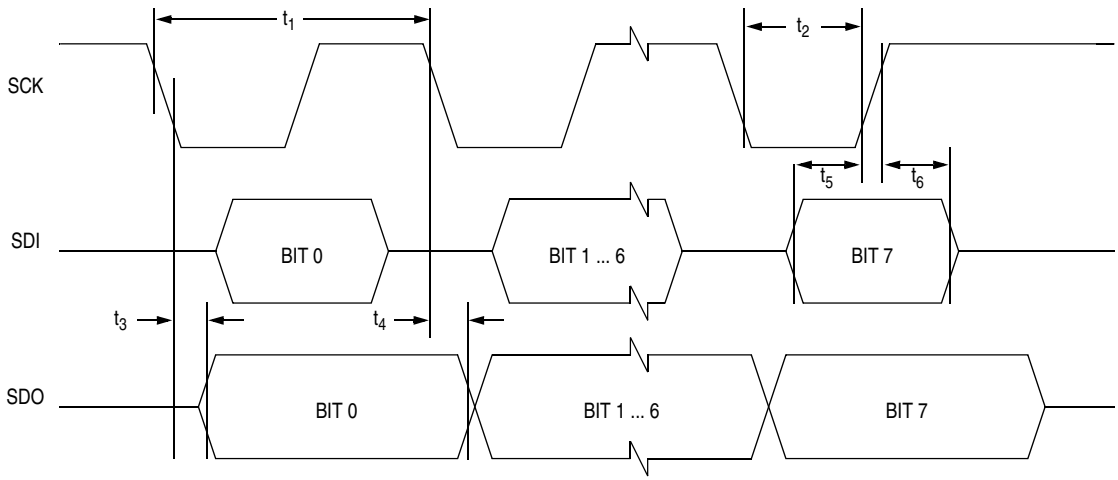
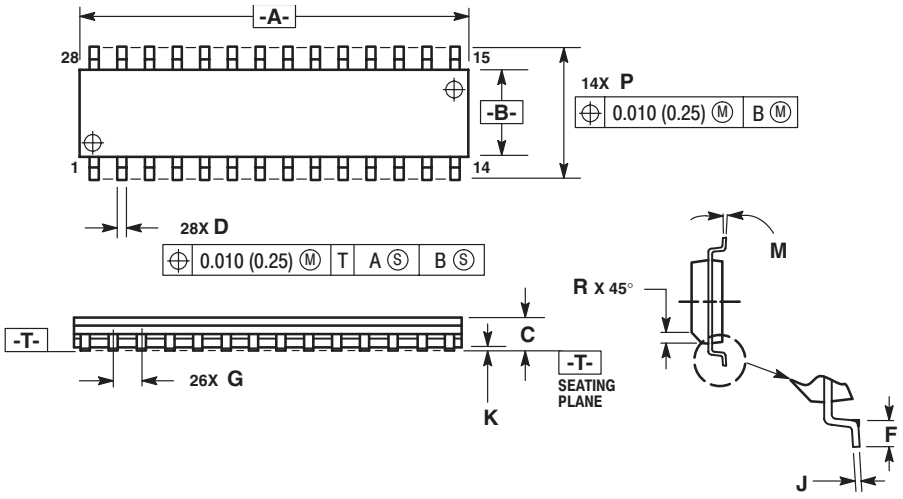


Figure 14-1. SIOP Timing Diagram

15.3 Small Outline Integrated Circuit Package (Case 751F)



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DIMENSION A AND B DO NOT INCLUDE MOLD PROTRUSION.
4. MAXIMUM MOLD PROTRUSION 0.15 (0.006) PER SIDE.
5. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.13 (0.005) TOTAL IN EXCESS OF D DIMENSION AT MAXIMUM MATERIAL CONDITION.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	17.80	18.05	0.701	0.711
B	7.40	7.60	0.292	0.299
C	2.35	2.65	0.093	0.104
D	0.35	0.49	0.014	0.019
F	0.41	0.90	0.016	0.035
G	1.27 BSC		0.050 BSC	
J	0.23	0.32	0.009	0.013
K	0.13	0.29	0.005	0.011
M	0°	8°	0°	8°
P	10.05	10.55	0.395	0.415
R	0.25	0.75	0.010	0.029