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	Not For New Designs
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/mc705p6acdwe

Chapter 15
Mechanical Specifications

Chapter 16
Ordering Information

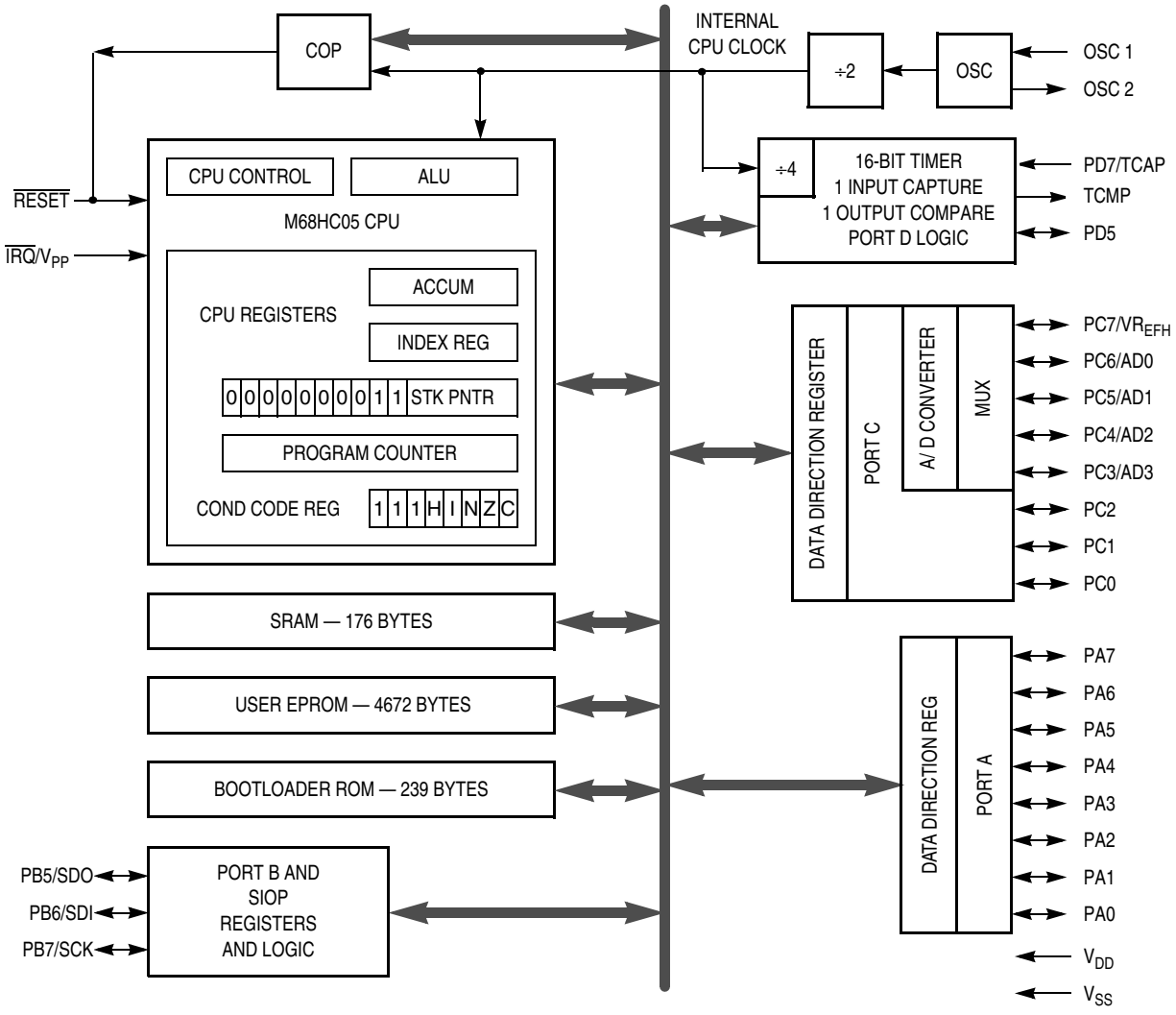


Figure 1-1. MC68HC705P6A Block Diagram

NOTE

A line over a signal name indicates an active low signal. For example, RESET is active high and $\overline{RESET}$ is active low.

Any reference to voltage, current, or frequency specified in the following sections will refer to the nominal values. The exact values and their tolerances or limits are specified in Chapter 14 Electrical Specifications.

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.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0000	Port A Data Register (PORTA) See page 37.	Read:	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
		Write:								
		Reset:	Unaffected by reset							
$0001	Port B Data Register (PORTB) See page 38.	Read:	PB7	PB6	PB5	0	0	0	0	0
		Write:								
		Reset:	Unaffected by reset							
$0002	Port C Data Register (PORTC) See page 38.	Read:	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
		Write:								
		Reset:	Unaffected by reset							
$0003	Port D Data Register (PORTD) See page 39.	Read:	PD7	0	PD5	1	0	0	0	0
		Write:								
		Reset:	Unaffected by reset							
$0004	Port A Data Direction Register (DDRA) See page 37.	Read:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0005	Port B Data Direction Register (DDRB) See page 38.	Read:	DDRB7	DDRB6	DDRB5	1	1	1	1	1
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0006	Port C Data Direction Register (DDRC) See page 38.	Read:	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0007	Port D Data Direction Register (DDRD) See page 39.	Read:	0	0	DDRD5	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0008	Unimplemented									
$0009	Unimplemented									
$000A	SIOP Control Register (SCR) See page 43.	Read:	0	SPE	0	MSTR	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000B	SIOP Status Register (SSR) See page 44.	Read:	SPIF	DCOL	0	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000C	SIOP Data Register (SDR) See page 44.	Read:	SDR7	SDR6	SDR5	SDR4	SDR3	SSDR2	SDR1	SDR0
		Write:								
		Reset:	Unaffected by reset							

= Unimplemented R = Reserved U = Undetermined

Figure 2-3. I/O and Control Register Summary (Sheet 1 of 3)

2.8 Computer Operating Properly (COP) Clear Register

The computer operating properly (COP) watchdog timer is located at address $1FF0. Writing a logical 0 to bit zero of this location will clear the COP watchdog counter as described in 4.3.2 Computer Operating Properly (COP) Reset.

$1FF0	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	0	0
Write:								COPR
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 2-5. COP Watchdog Timer Location

The POR will generate the RST signal and reset the MCU. If any other reset function is active at the end of this 4064 internal clock cycle delay, the RST signal will remain active until the other reset condition(s) end.

4.3.2 Computer Operating Properly (COP) Reset

When the COP watchdog timer is enabled (COP bit in the MOR is set), the internal COP reset is generated automatically by a timeout of the COP watchdog timer. This timer is implemented with an 18-stage ripple counter that provides a timeout period of 65.5 ms when a 4-MHz oscillator is used. The COP watchdog counter is cleared by writing a logical 0 to bit zero at location $1FF0.

The COP watchdog timer can be disabled by clearing the COP bit in the MOR or by applying 2 x V_{DD} to the $\overline{IRQ}/V_{PP}$ pin (for example, during bootloader). When the $\overline{IRQ}/V_{PP}$ pin is returned to its normal operating voltage range (between V_{SS}–V_{DD}), the COP watchdog timer's output will be restored if the COP bit in the mask option register (MOR) is set.

The COP register is shared with the least significant byte (LSB) of an unused vector address as shown in Figure 4-2. Reading this location will return the programmed value of the unused user interrupt vector, usually 0. Writing to this location will clear the COP watchdog timer.

Address: $1FF0

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	0	0
Write:								COPR

= Unimplemented

Figure 4-2. Unused Vector and COP Watchdog Timer

When the COP watchdog timer expires, it will generate the RST signal and reset the MCU. If any other reset function is active at the end of the COP reset signal, the RST signal will remain in the reset condition until the other reset condition(s) end. When the reset condition ends, the MCU's operating mode will be selected (see Table 3-1. Operating Mode Conditions After Reset).

Table 6-1. Port A I/O Functions

DDRA	I/O Pin Mode	Accesses to DDRA @ $0004	Accesses to Data Register @ $0000	
		Read/Write	Read	Write
0	IN, Hi-Z	DDRA0–DDRA7	I/O Pin	See Note
1	OUT	DDRA0–DDRA7	PA0–PA7	PA0–PA7

Note: Does not affect input, but stored to data register

Table 6-2. Port B I/O Functions

DDRB	I/O Pin Mode	Accesses to DDRB @ $0005	Accesses to Data Register @ $0001	
		Read/Write	Read	Write
0	IN, Hi-Z	DDRB5–DDRB7	I/O Pin	See Note
1	OUT	DDRB5–DDRB7	PB5–PB7	PB5–PB7

Note: Does not affect input, but stored to data register

Table 6-3. Port C I/O Functions

DDRC	I/O Pin Mode	Accesses to DDRC @ $0006	Accesses to Data Register @ $0002	
		Read/Write	Read	Write
0	IN, Hi-Z	DDRC0–DDRC7	I/O Pin	See Note
1	OUT	DDRC0–DDRC7	PC0–PC7	PC0–PC7

Note: Does not affect input, but stored to data register

Table 6-4. Port D I/O Functions

DDRD	I/O Pin Mode	Accesses to DDRD @ $0007	Accesses to Data Register @ $0003	
		Read/Write	Read	Write
0	IN, Hi-Z	DDRD5	I/O Pin	See Note 1
1	OUT	DDRD5	PD5	PD5

Notes:
1. Does not affect input, but stored to data register
2. PD7 is input only

NOTE

To avoid generating a glitch on an I/O port pin, data should be written to the I/O port data register before writing a logic 1 to the corresponding data direction register.

At power-on or reset, all DDRs are cleared, which configures all port pins as inputs. The DDRs are capable of being written to or read by the processor. During the programmed output state, a read of the data register will actually read the value of the output data latch and not the level on the I/O port pin.

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.

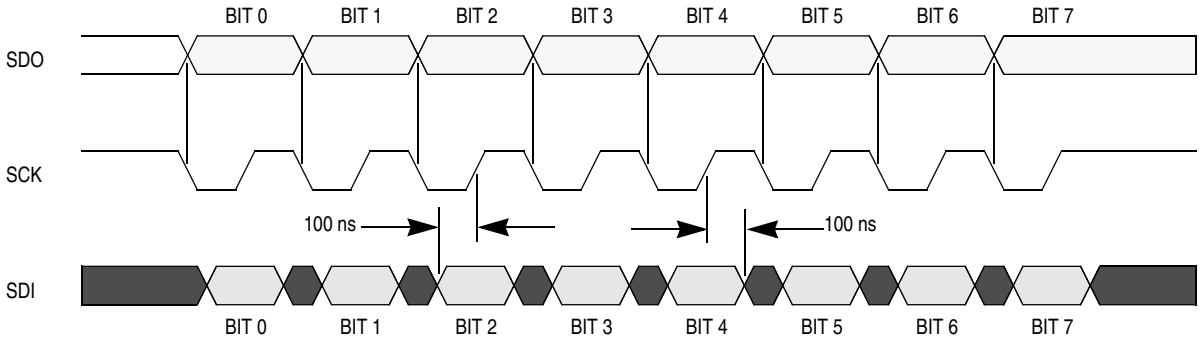


Figure 7-2. SIOP Timing Diagram

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.

9.4.1 Conversion Times

Each input conversion requires 32 internal clock cycles, which must be at a frequency equal to or greater than 1 MHz.

9.4.2 Internal versus External Oscillator

If the internal clock is 1 MHz or greater (i.e., external oscillator 2 MHz or greater), the internal RC oscillator must be turned off and the external oscillator used as the conversion clock.

If the MCU internal clock frequency is less than 1 MHz (2 MHz external oscillator), the internal RC oscillator (approximately 1.5 MHz) must be used for the A/D converter clock. The internal RC clock is selected by setting the ADRC bit in the ADSC register.

When the internal RC oscillator is being used, these limitations apply:

1. Since the internal RC oscillator is running asynchronously with respect to the internal clock, the conversion complete bit (CC) in register ADSC must be used to determine when a conversion sequence has been completed.
2. Electrical noise will slightly degrade the accuracy of the A/D converter. The A/D converter is synchronized to read voltages during the quiet period of the clock driving it. Since the internal and external clocks are not synchronized, the A/D converter will occasionally measure an input when the external clock is making a transition.

9.4.3 Multi-Channel Operation

An input multiplexer allows the A/D converter to select from one of four external analog signals. Port C pins PC3 through PC6 are shared with the inputs to the multiplexer.

9.5 A/D Status and Control Register (ADSC)

The ADSC register reports the completion of A/D conversion and provides control over oscillator selection, analog subsystem power, and input channel selection. See Figure 9-1.

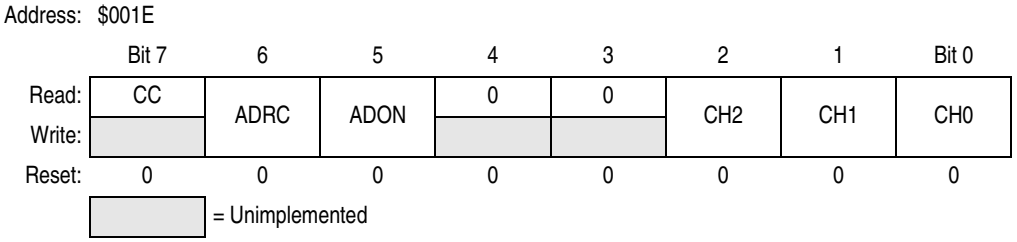


Figure 9-1. A/D Status and Control Register (ADSC)

CC — Conversion Complete

This read-only status bit is set when a conversion sequence has completed and data is ready to be read from the ADC register. CC is cleared when the ADSC is written to or when data is read from the ADC register. Once a conversion has been started, conversions of the selected channel will continue every 32 internal clock cycles until the ADSC register is written to again. During continuous conversion operation, the ADC register will be updated with new data, and the CC bit set every 32 internal clock cycles. Also, data from the previous conversion will be overwritten regardless of the state of the CC bit.

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.

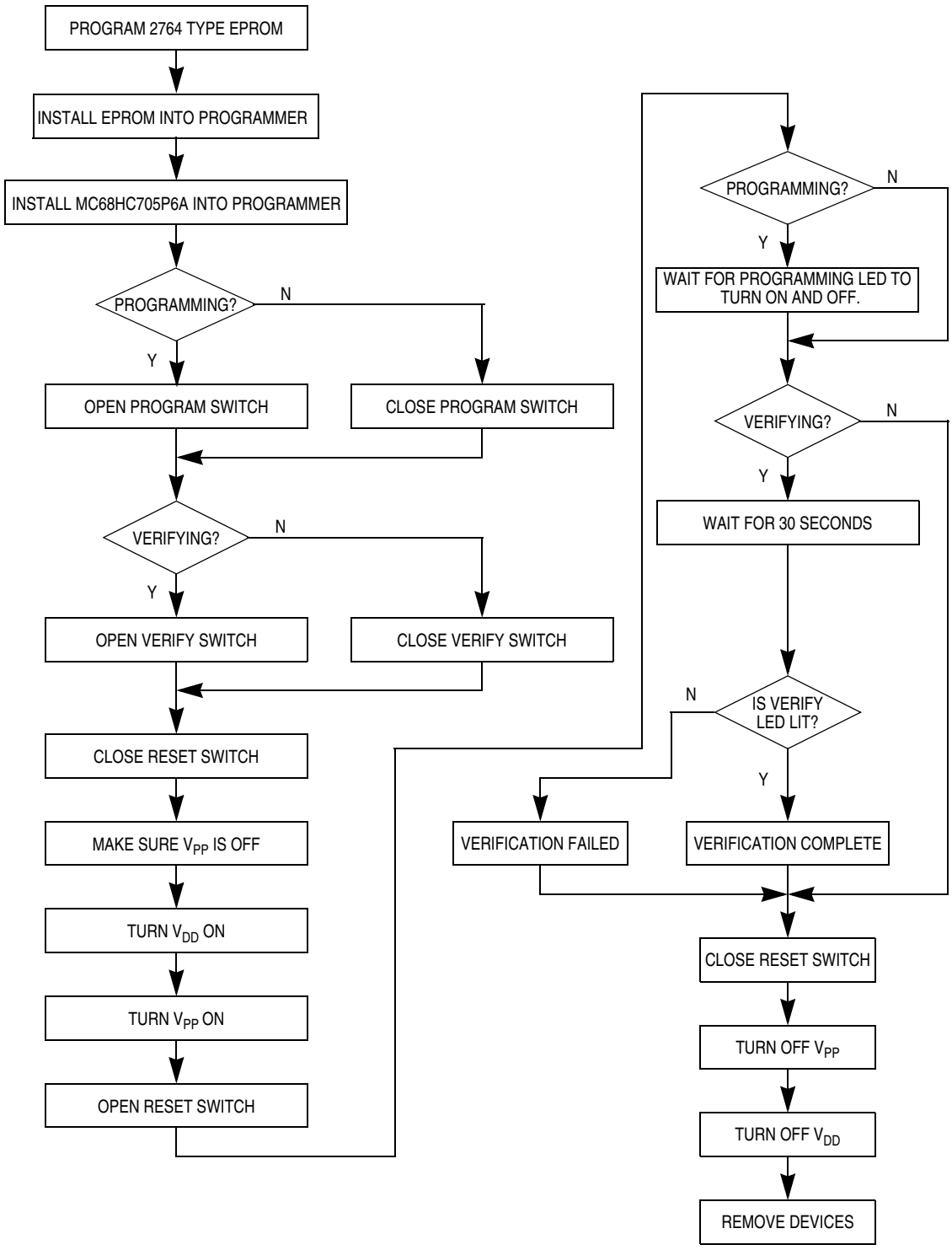


Figure 10-2. MC68HC705P6A EPROM Programming Flowchart

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
```

13.3 Instruction Types

The MCU instructions fall into the following five categories:

- Register/memory instructions
- Read-modify-write instructions
- Jump/branch instructions
- Bit manipulation instructions
- Control instructions

13.3.1 Register/Memory Instructions

These instructions operate on CPU registers and memory locations. Most of them use two operands. One operand is in either the accumulator or the index register. The CPU finds the other operand in memory.

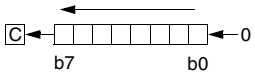
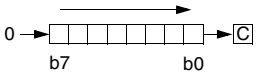
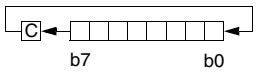
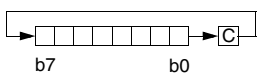
Table 13-1. Register/Memory Instructions

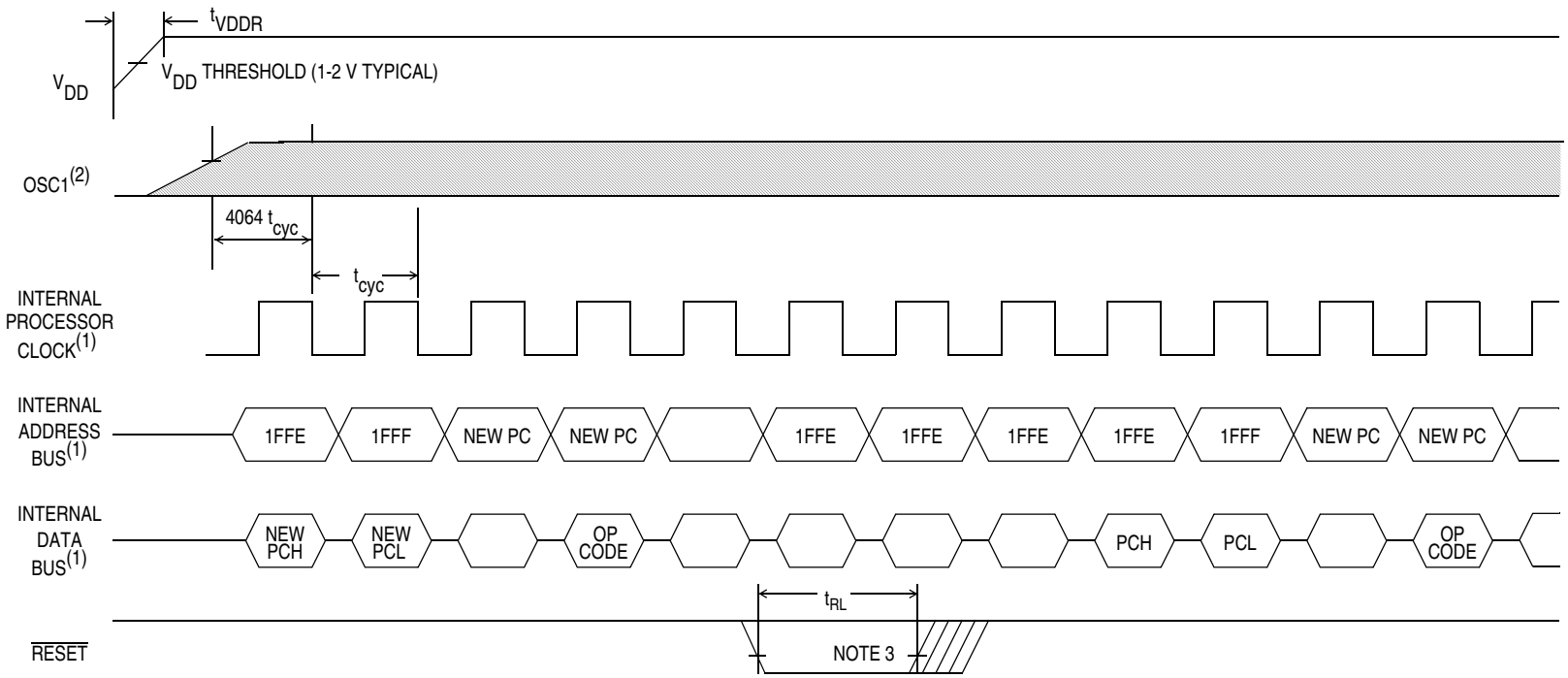
Instruction	Mnemonic
Add Memory Byte and Carry Bit to Accumulator	ADC
Add Memory Byte to Accumulator	ADD
AND Memory Byte with Accumulator	AND
Bit Test Accumulator	BIT
Compare Accumulator	CMP
Compare Index Register with Memory Byte	CPX
EXCLUSIVE OR Accumulator with Memory Byte	EOR
Load Accumulator with Memory Byte	LDA
Load Index Register with Memory Byte	LDX
Multiply	MUL
OR Accumulator with Memory Byte	ORA
Subtract Memory Byte and Carry Bit from Accumulator	SBC
Store Accumulator in Memory	STA
Store Index Register in Memory	STX
Subtract Memory Byte from Accumulator	SUB

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 4 of 6)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
LDA #*opr* LDA *opr* LDA *opr* LDA *opr*,X LDA *opr*,X LDA ,X	Load Accumulator with Memory Byte	A ← (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	A6 B6 C6 D6 E6 F6	ii dd hh ll ee ff ff	2 3 4 5 4 3
LDX #*opr* LDX *opr* LDX *opr* LDX *opr*,X LDX *opr*,X LDX ,X	Load Index Register with Memory Byte	X ← (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	AE BE CE DE EE FE	ii dd hh ll ee ff ff	2 3 4 5 4 3
LSL *opr* LSLA LSLX LSL *opr*,X LSL ,X	Logical Shift Left (Same as ASL)		—	—	↕	↕	↕	DIR INH INH IX1 IX	38 48 58 68 78	dd ff 	5 3 3 6 5
LSR *opr* LSRA LSRX LSR *opr*,X LSR ,X	Logical Shift Right		—	—	0	↕	↕	DIR INH INH IX1 IX	34 44 54 64 74	dd ff 	5 3 3 6 5
MUL	Unsigned Multiply	X : A ← (X) × (A)	0	—	—	—	0	INH	42		1 1
NEG *opr* NEGA NEGX NEG *opr*,X NEG ,X	Negate Byte (Two's Complement)	M ← –(M) = $00 – (M) A ← –(A) = $00 – (A) X ← –(X) = $00 – (X) M ← –(M) = $00 – (M) M ← –(M) = $00 – (M)	—	—	↕	↕	↕	DIR INH INH IX1 IX	30 40 50 60 70	dd ff 	5 3 3 6 5
NOP	No Operation		—	—	—	—	—	INH	9D		2
ORA #*opr* ORA *opr* ORA *opr* ORA *opr*,X ORA *opr*,X ORA ,X	Logical OR Accumulator with Memory	A ← (A) ∨ (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	AA BA CA DA EA FA	ii dd hh ll ee ff ff	2 3 4 5 4 3
ROL *opr* ROLA ROLX ROL *opr*,X ROL ,X	Rotate Byte Left through Carry Bit		—	—	↕	↕	↕	DIR INH INH IX1 IX	39 49 59 69 79	dd ff 	5 3 3 6 5
ROR *opr* RORA RORX ROR *opr*,X ROR ,X	Rotate Byte Right through Carry Bit		—	—	↕	↕	↕	DIR INH INH IX1 IX	36 46 56 66 76	dd ff 	5 3 3 6 5
RSP	Reset Stack Pointer	SP ← $00FF	—	—	—	—	—	INH	9C		2



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 $\overline{\text{RESET}}$ initiates the reset sequence.

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