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 ~ 125°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/mc705p6amdwe

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

Table of Contents

Chapter 1
General Description

Chapter 2
Memory

Chapter 3
Operating Modes

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

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.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$000D	Reserved for Test		R	R	R	R	R	R	R	R
$000E	Unimplemented									
$000F	Unimplemented									
$0010	Unimplemented									
$0011	Unimplemented									
$0012	Timer Control Register (TCR) See page 47.	Read:	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL
		Write:								
		Reset:	0	0	0	0	0	0	U	0
$0013	Timer Status Register (TSR) See page 48.	Read:	ICF	OCF	TOF	0	0	0	0	0
		Write:								
		Reset:	U	U	U	0	0	0	0	0
$0014	Input Capture Register MSB (ICRH) See page 50.	Read:	ICRH7	ICRH6	ICRH5	ICRH4	ICRH3	ICRH2	ICRH1	ICRH0
		Write:								
		Reset:	Unaffected by reset							
$0015	Input Capture Register LSB (ICRL) See page 50.	Read:	ICRL7	ICRL6	ICRL5	ICRL4	ICRL3	ICRL2	ICRL1	ICRL0
		Write:								
		Reset:	Unaffected by reset							
$0016	Output Compare Register MSB (OCRH) See page 50.	Read:	OCRH7	OCRH6	OCRH5	OCRH4	OCRH3	OCRH2	OCRH1	OCRH0
		Write:								
		Reset:	Unaffected by reset							
$0017	Output Compare Register LSB (OCRL) See page 50.	Read:	OCRL7	OCRL6	OCRL5	OCRL4	OCRL3	OCRL2	OCRL1	OCRL0
		Write:								
		Reset:	Unaffected by reset							
$0018	Timer Register MSB (TRH) See page 49.	Read:	TRH7	TRH6	TRH5	TRH4	TRH3	TRH2	TRH1	TRH0
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$0019	Timer Register LSB (TRL) See page 49.	Read:	TRL7	TRL6	TRL5	TRL4	TRL3	TRL2	TRL1	TRL0
		Write:								
		Reset:	1	1	1	1	1	1	0	0
$001A	Alternate Timer Register MSB (ATRH) See page 49.	Read:	ACRH7	ACRH6	ACRH5	ACRH4	ACRH3	ACRH2	ACRH1	ACRH0
		Write:								
		Reset:	1	1	1	1	1	1	1	1

= Unimplemented R = Reserved U = Undetermined

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

Chapter 5
Interrupts

5.1 Introduction

The MCU can be interrupted six different ways:

1. Non-maskable software interrupt instruction (SWI)
2. External asynchronous interrupt ($\overline{\text{IRQ}}$)
3. Input capture interrupt (TIMER)
4. Output compare interrupt (TIMER)
5. Timer overflow interrupt (TIMER)
6. Port A interrupt (if selected via mask option register)

Interrupts cause the processor to save the register contents on the stack and to set the interrupt mask (I bit) to prevent additional interrupts. Unlike reset, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction is completed.

When the current instruction is completed, the processor checks all pending hardware interrupts. If interrupts are not masked (I bit in the condition code register is clear) and the corresponding interrupt enable bit is set, the processor proceeds with interrupt processing. Otherwise, the next instruction is fetched and executed. The SWI is executed the same as any other instruction, regardless of the I-bit state.

When an interrupt is to be processed, the CPU puts the register contents on the stack, sets the I bit in the CCR, and fetches the address of the corresponding interrupt service routine from the vector table at locations $1FF8 through $1FFF. If more than one interrupt is pending when the interrupt vector is fetched, the interrupt with the highest vector location shown in Table 5-1 will be serviced first.

Table 5-1. Vector Addresses for Interrupts and Reset

Register	Flag Name	Interrupts	CPU Interrupt	Vector Address
N/A	N/A	Reset	RESET	$1FFE–$1FFF
N/A	N/A	Software	SWI	$1FFC–$1FFD
N/A	N/A	External Interrupt	IRQ	$1FFA–$1FFB
TSR	ICF	Timer Input Capture	TIMER	$1FF8–$1FF9
TSR	OCF	Timer Output Compare	TIMER	$1FF8–$1FF9
TSR	TOF	Timer Overflow	TIMER	$1FF8–$1FF9

An RTI instruction is used to signify when the interrupt software service routine is completed. The RTI instruction causes the CPU state to be recovered from the stack and normal processing to resume at the next instruction that was to be executed when the interrupt took place. Figure 5-1 shows the sequence of events that occurs during interrupt processing.

6.3 Port B

Port B is a 3-bit bidirectional port which can share pins PB5–PB7 with the SIOP communications subsystem. The port B data register is located at address $0001 and its data direction register (DDR) is located at address $0005. The contents of the port B 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 all of the port pins to input mode. Writing a 1 to a DDR bit sets the corresponding port pin to output mode (see Figure 6-2).

Port B may be used for general I/O applications when the SIOP subsystem is disabled. The SPE bit in register SPCR is used to enable/disable the SIOP subsystem. When the SIOP subsystem is enabled, port B registers are still accessible to software. Writing to either of the port B registers while a data transfer is under way could corrupt the data. See Chapter 7 Serial Input/Output Port (SIOP) for a discussion of the SIOP subsystem.

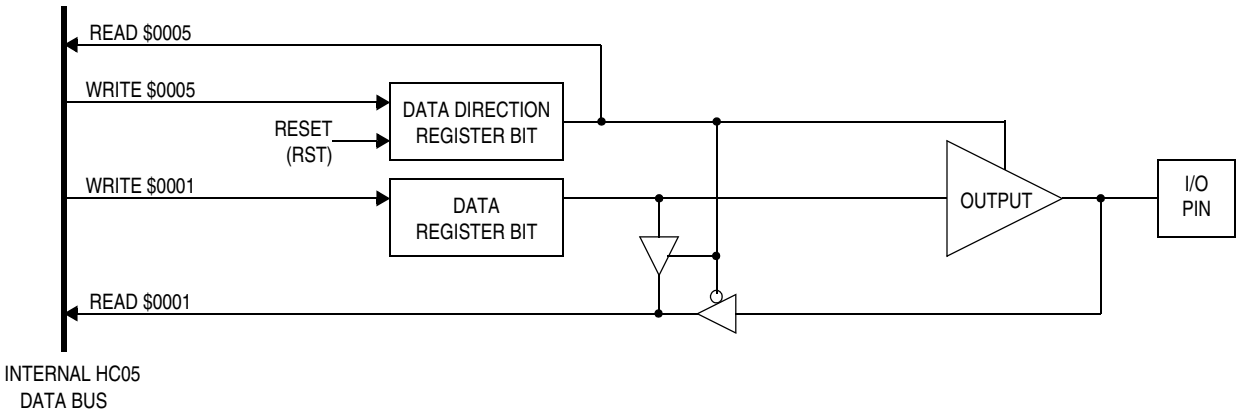

Figure 6-2. Port B I/O Circuitry

6.4 Port C

Port C is an 8-bit bidirectional port which can share pins PC3–PC7 with the A/D subsystem. The port C data register is located at address $0002 and its data direction register (DDR) is located at address $0006. The contents of the port C 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 all of the port pins to input mode. Writing a 1 to a DDR bit sets the corresponding port pin to output mode (see Figure 6-3).

Port C may be used for general I/O applications when the A/D subsystem is disabled. The ADON bit in register ADSC is used to enable/disable the A/D subsystem. Care must be exercised when using pins PC0–PC2 while the A/D subsystem is enabled. Accidental changes to bits that affect pins PC3–PC7 in the data or DDR registers will produce unpredictable results in the A/D subsystem. See Chapter 9 Analog Subsystem.

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.

7.3.2 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)

8.3.2 Timer Status Register

The timer status register (TSR), shown in Figure 8-3, contains flags to signal the following conditions:

- An active signal on the TCAP pin, transferring the contents of the timer registers to the input capture registers
- A match between the 16-bit counter and the output compare registers, transferring the OLVL bit to the TCMP pin
- A timer roll over from $FFFF to $0000

Address: $0013

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	ICF	OCF	TOF	0	0	0	0	0
Write:								
Reset:	U	U	U	0	0	0	0	0

= Unimplemented U = Undetermined

Figure 8-3. Timer Status Register (TSR)

ICF — Input Capture Flag

The ICF bit is set automatically when an edge of the selected polarity occurs on the TCAP pin. Clear the ICF bit by reading the timer status register with ICF set and then reading the low byte ($0015) of the input capture registers. Resets have no effect on ICF.

OCF — Output Compare Flag

The OCF bit is set automatically when the value of the timer registers matches the contents of the output compare registers. Clear the OCF bit by reading the timer status register with OCF set and then reading the low byte ($0017) of the output compare registers. Resets have no effect on OCF.

TOF — Timer Overflow Flag

The TOF bit is set automatically when the 16-bit counter rolls over from $FFFF to $0000. Clear the TOF bit by reading the timer status register with TOF set, and then reading the low byte ($0019) of the timer registers. Resets have no effect on TOF.

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.

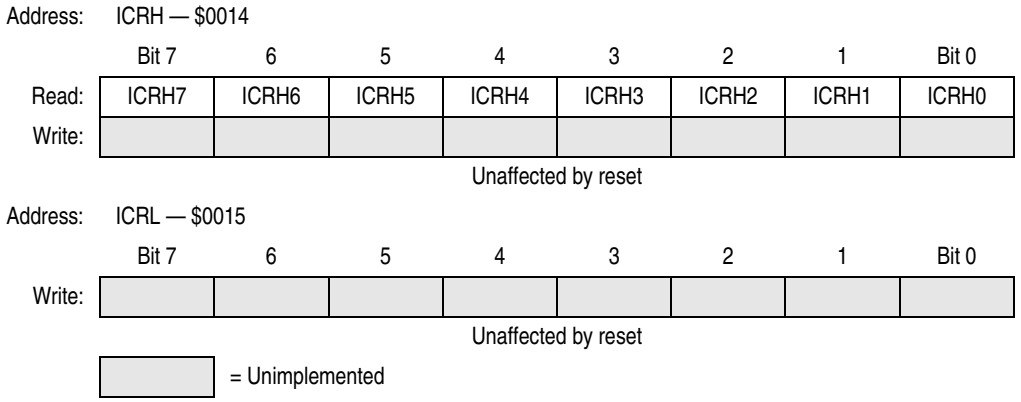

Figure 8-6. Input Capture Registers (ICRH and ICRL)

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).

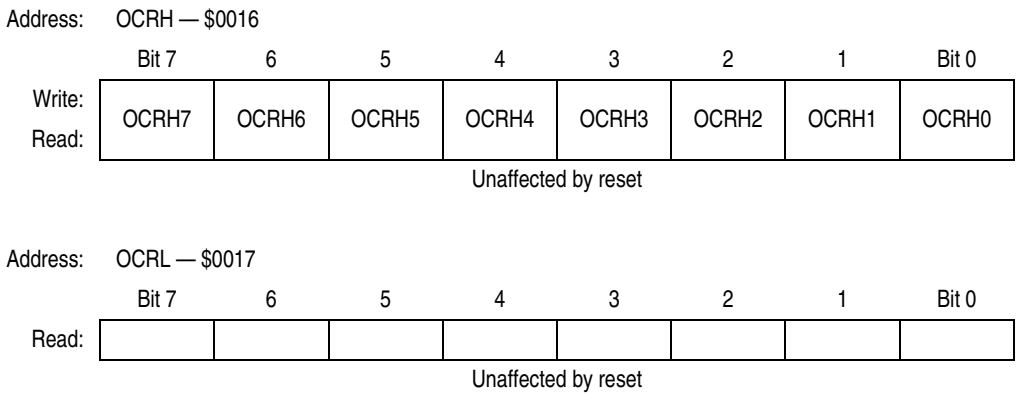

Figure 8-7. Output Compare Registers (OCRH and OCRL)

9.4.1 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.

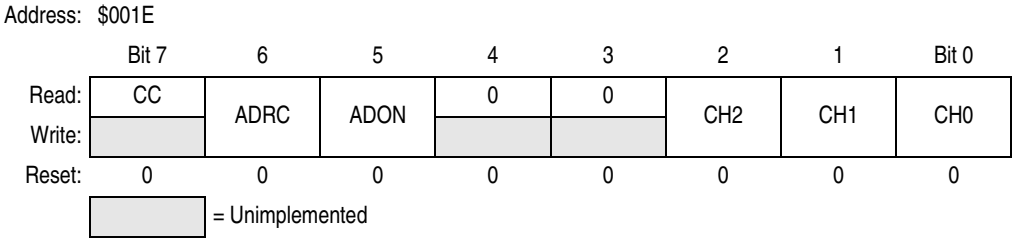

Figure 9-1. A/D Status and Control Register (ADSC)

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.

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.

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

14.6 3.3-Volt DC Electrical Charactertistics

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Output voltage I_{Load} = 10.0 µA I_{Load} = –10.0 µA	V_{OL} V_{OH}	— V_{DD} –0.1	— —	0.1 —	V
Output high voltage (I_{Load} = –0.2 mA) PA0:7, PB5:7, PC2:7, PD5, TCMP (I_{Load} = –1.2 mA) PC0:1	V_{OH}	V_{DD} –0.3 V_{DD} –0.3	— —	— —	V
Output low voltage (I_{Load} = 0.4 mA) PA0:7, PB5:7, PC2:7, PD5, TCMP (I_{Load} = 2.5 mA) PC0:1	V_{OL}	— —	— —	0.3 0.3	V
Input high voltage PA0:7, PB5:7, PC0:7, PD5, TCAP/PD7, $\overline{IRQ}$/V_{PP}, $\overline{RESET}$, OSC1	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input low voltage PA0:7, PB5:7, PC0:7, PD5, TCAP/PD7, $\overline{IRQ}$/V_{PP}, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	0.2 x V_{DD}	V
Supply current[3], [4] Run Wait[5] (A/D converter on) Wait[5] (A/D converter off) Stop[6] 25°C 0°C to +70°C (standard) –40°C to +85°C (extended)	I_{DD}	 — — — — — —	 1.8 1.0 0.6 2 — —	 2.5 1.4 1.0 20 40 50	 mA mA mA µA µA µA
I/O ports high-z leakage current PA0:7, PB5:7, PC0:7, PD5, TCAP/PD7	I_{IL}	—	—	±10.0	µA
A/D ports hi-z leakage current PC3:7	I_{OZ}	—	—	±1.0	µA
Input current $\overline{RESET}$, $\overline{IRQ}$/V_{PP}, OSC1, PD7/TCAP	I_{In}	—	—	±1.0	µA
Input pullup current PA0:7 (with pullup enabled)	I_{In}	75	175	350	µA
Capaitance Ports (as input or output) $\overline{RESET}$, $\overline{IRQ}$/V_{PP}	 C_{Out} C_{In}	 — —	 — —	 12 8	 pF

1. V_{DD} = 3.3 Vdc ± 0.3 Vdc, V_{SS} = 0 Vdc, T_A = –40°C to +85°C, unless otherwise noted. All values shown reflect pre-silicon estimates.
2. Typical values at midpoint of voltage range, 25°C only.
3. Run (Operating) I_{DD}, Wait I_{DD}: To be measured using external square wave clock source (f_{osc} = 4.2 MHz), all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2.
4. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 V, V_{IH} = V_{DD} –0.2 V.
5. Wait I_{DD} will be affected linearly by the OSC2 capacitance.
6. Stop I_{DD} to be measured with OSC1 = V_{SS}.

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

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

RoHS-compliant and/or Pb- free versions of Freescale products have the functionality and electrical characteristics of their non-RoHS-compliant and/or non-Pb- free counterparts. For further information, see http://www.freescale.com or contact your Freescale sales representative.

For information on Freescale.s Environmental Products program, go to http://www.freescale.com/epp.

MC68HC705P6A
Rev. 2.1, 9/2005

freescale™
semiconductor