MC68HC705P9/D
REV. 3

MC68HC705P9

HCMOS Microcontroller Unit

TECHNICAL DATA

MOTOROLA

MOTOROLA

CSIC

MICROCONTROLLERS

List of Sections

List of Modules

All M68HC05 microcontroller units (MCUs) are customer-specified modular designs. To meet customer requirements, Motorola is constantly designing new modules and new versions of existing modules. The following table shows the version levels of the modules in the MC68HC705P9 MCU.

Module	Version
Central Processor Unit (CPU)	HC05CPU
Timer	TIM1IC1OC_A
Serial Input/Output Port (SIOP)	SIOP_A
Computer Operating Properly Watchdog (COP)	COP0COP
Analog-to-Digital Converter (ADC)	ATD4X8NVRL

Revision History

The following table summarizes differences between this revision and the previous revision of this Technical Data manual.

Previous Revision	2.0
Current Revision	3.0
Date	11/95
Changes	Format and organizational changes
Location	Throughout

Table of Contents

Table of Contents

Introduction

Contents

Features

- Four Peripheral Modules
  - 16-Bit Input Capture/Output Compare Timer
  - Synchronous Serial I/O Port (SIOP)
  - 4-Channel, 8-Bit Analog-to-Digital Converter (ADC)
  - Computer Operating Properly (COP) Watchdog
- 20 Bidirectional I/O Port Pins and One Input-Only Port Pin
- On-Chip Oscillator with Connections for:
  - Crystal
  - Ceramic Resonator
  - External Clock
- 2104 Bytes of EPROM/OTPROM
  - 48 Bytes of Page Zero EPROM/OTPROM
  - Eight Locations for User Vectors
- 128 Bytes of User RAM
- Bootloader ROM
- Memory-Mapped Input/Output (I/O) Registers
- Fully Static Operation with No Minimum Clock Speed
- Power-Saving Stop, Wait, and Data-Retention Modes

2-mc68hc705p9

Structure

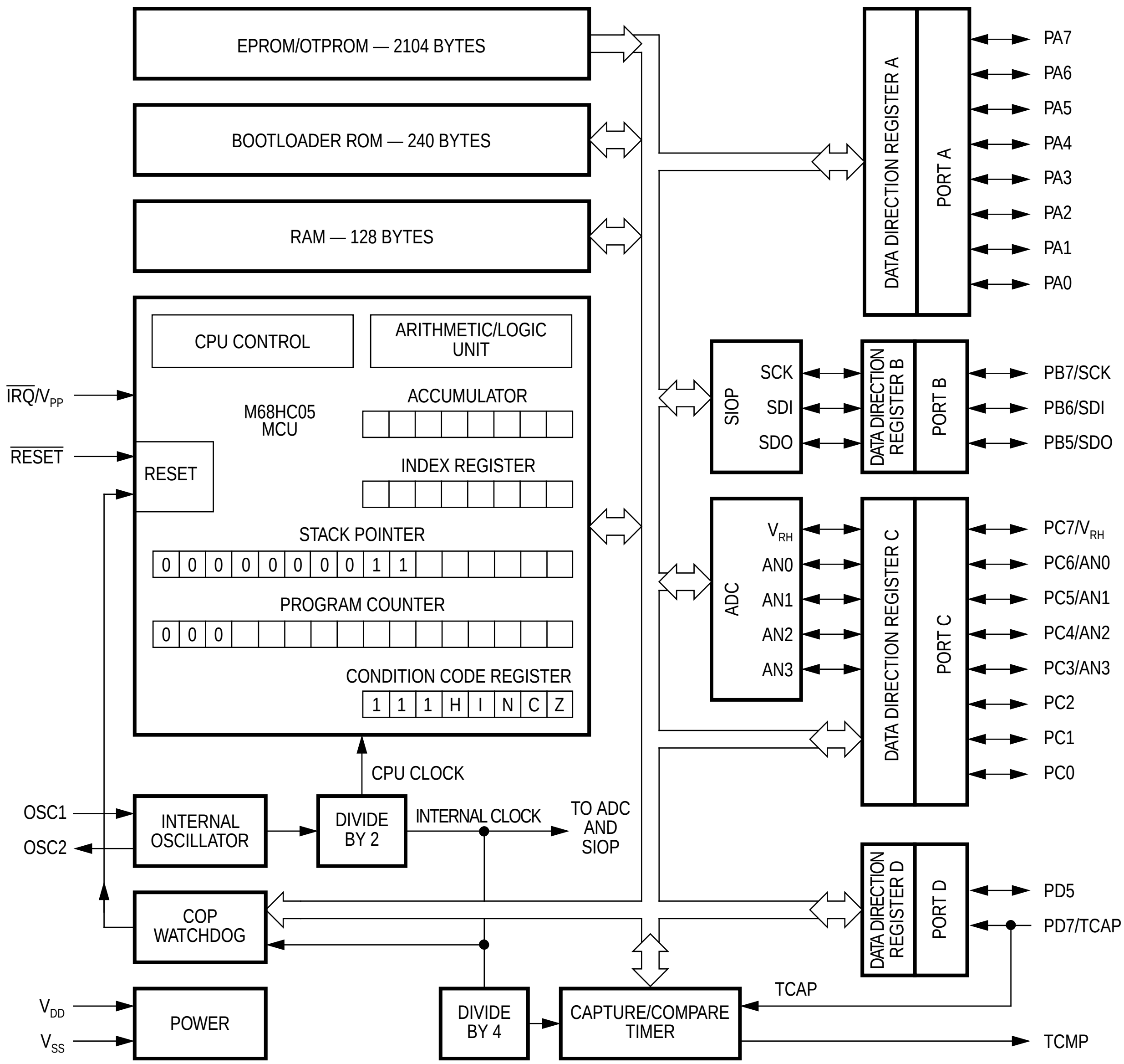


Figure 1. MC68HC705P9 Block Diagram

3-mc68hc705p9

Package Types and Order Numbers

Table 1. Order Numbers

Package Type	Case Outline	Pin Count	Operating Temperature	Order Number
Plastic DIP[1]	710	28	0 to +70 °C –40 to +85 °C –40 to +105 °C –40 to +125 °C	MC68HC705P9P MC68HC705P9CP MC68HC705P9VP MC68HC705P9MP
SOIC[2]	733	28	0 to +70 °C –40 to +85 °C –40 to +105 °C –40 to +125 °C	MC68HC705P9DW MC68HC705P9CDW MC68HC705P9VDW MC68HC705P9MDW
CERDIP[3]	751F	28	0 to +70 °C –40 to +85 °C –40 to +105 °C –40 to +125 °C	MC68HC705P9S MC68HC705P9CS MC68HC705P9VS MC68HC705P9MS

1. DIP = dual in-line package
2. SOIC = small outline integrated circuit
3. CERDIP = ceramic DIP

Programmable Options

The options in **Table 2** are programmable in the mask option register.

Table 2. Programmable Options

Feature	Option
COP Watchdog	Enabled or Disabled
External Interrupt Pin Triggering	Negative-Edge Triggering Only or Negative-Edge and Low-Level Triggering
SIOP Data Format	MSB First or LSB First

4-mc68hc705p9

Pin Descriptions

Contents

1-mc68hc705p9

Pin Assignments

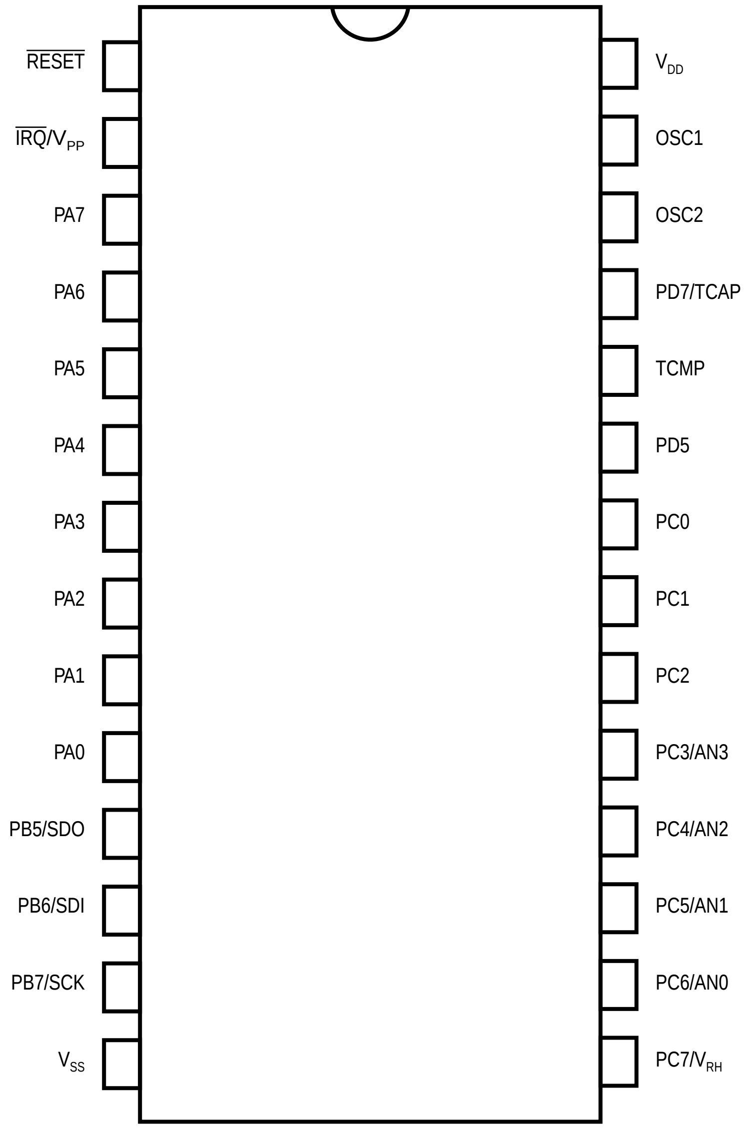


Figure 1. Pin Assignments

Pin Functions

V_{DD} and V_{SS}

V_{DD} and V_{SS} are the power supply and ground pins. The MCU operates from a single 5-V power supply.

Very fast signal transitions occur on the MCU pins, placing high short-duration current demands on the power supply. To prevent noise problems, take special care to provide good power supply bypassing at the MCU as **Figure 2** shows. Place the bypass capacitors as close as possible to the MCU. C2 is an optional bulk current bypass capacitor for use in applications that require the port pins to source high current levels.

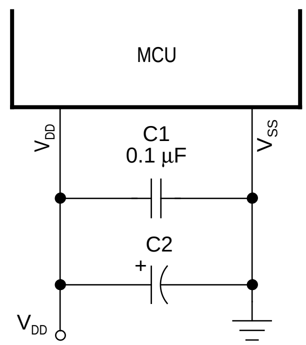


Figure 2. Bypassing Recommendation

OSC1 and OSC2

The OSC1 and OSC2 pins are the connections for the on-chip oscillator. The oscillator can be driven by any of the following:

- Crystal
- Ceramic resonator
- External clock signal

The frequency of the on-chip oscillator is f_{OSC}. The MCU divides the internal oscillator output by two to produce the internal clock with a frequency of f_{OP}.

Crystal Connections

The circuit in **Figure 3** shows a typical crystal oscillator circuit for an AT-cut, parallel resonant crystal. Follow the crystal supplier's recommendations, as the crystal parameters determine the external component values required to provide reliable startup and maximum stability. The load capacitance values used in the oscillator circuit design should include all stray layout capacitances. To minimize output distortion, mount the crystal and capacitors as close as possible to the pins.

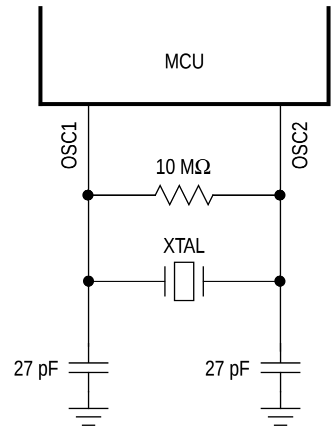


Figure 3. Crystal Connections

NOTE: *Use an AT-cut crystal. Do not use a strip or tuning fork crystal. The MCU may overdrive or have the incorrect characteristic impedance for a strip or tuning fork crystal.*

Ceramic Resonator Connections

To reduce cost, use a ceramic resonator in place of the crystal. **Figure 4** shows a ceramic resonator circuit. For the values of any external components, follow the recommendations of the resonator manufacturer. The load capacitance values used in the oscillator circuit design should include all stray layout capacitances. To minimize output distortion, mount the resonator and capacitors as close as possible to the pins.

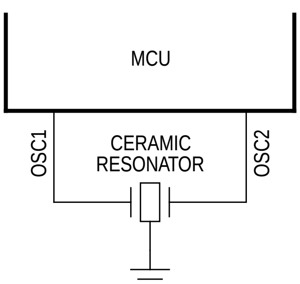


Figure 4. Ceramic Resonator Connections

4-mc68hc705p9

NOTE: *Because the frequency stability of ceramic resonators is not as high as that of crystal oscillators, using a ceramic resonator may degrade the performance of the ADC.*

External Clock Connections

An external clock from another CMOS-compatible device can drive the OSC1 input, with the OSC2 pin unconnected, as **Figure 5** shows.

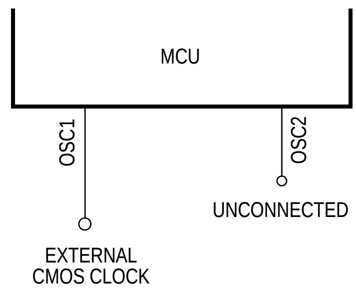


Figure 5. External Clock Connections

$\overline{\text{RESET}}$

A logic zero on the $\overline{\text{RESET}}$ pin forces the MCU to a known startup state. The $\overline{\text{RESET}}$ pin input circuit contains an internal Schmitt trigger to improve noise immunity.

$\overline{\text{IRQ}}/V_{PP}$

The $\overline{\text{IRQ}}/V_{PP}$ pin has the following functions:

- Applying asynchronous external interrupt signals
- Applying V_{PP}, the EPROM/OTPROM programming voltage

PA7–PA0

PA7–PA0 are general-purpose bidirectional I/O port pins. Use data direction register A to configure port A pins as inputs or outputs.

PB7/SCK–PB5/SDO

Port B is a 3-pin bidirectional I/O port that shares its pins with the SIOP. Use data direction register B to configure port B pins as inputs or outputs.

PC7/V_{RH}–PC0

Port C is an 8-pin bidirectional I/O port that shares five of its pins with the ADC. Use data direction register C to configure port C pins as inputs or outputs.

PD7/TCAP and PD5

Port D is a 2-pin I/O port that shares one of its pins with the capture/compare timer. Use data direction register D to configure port D pins as inputs or outputs.

TCMP

The TCMP pin is the output compare pin for the capture/compare timer.

Memory

Contents

Features

- 2104 Bytes of EPROM/OTPROM
  - 48 Bytes of Page Zero EPROM/OTPROM
  - Eight Locations for User Vectors
- 128 Bytes of User RAM
- Bootloader ROM

1-mc68hc705p9

Memory Map

Address	Contents
$0000 ↓ $001F	I/O Registers (32 Bytes)
$0020 ↓ $004F	Page Zero User EPROM (48 Bytes)
$0050 ↓ $007F	Unimplemented (48 Bytes)
$0080 ↓ $00FF	RAM (128 Bytes)
$0100 ↓ $08FF	User EPROM (2048 Bytes)
$0900	Mask Option Register
$0901 ↓ $1EFF	Unimplemented (5631 Bytes)
$1F00 ↓ $1FEF	Bootloader ROM (240 Bytes)
$1FF0	COP Control Register
$1FF1 ↓ $1FF7	Reserved
$1FF8 ↓ $1FFF	User Vector EPROM (8 Bytes)

Register	Address
Port A Data Register (PORTA)	$0000
Port B Data Register (PORTB)	$0001
Port C Data Register (PORTC)	$0002
Port D Data Register (PORTD)	$0003
Data Direction Register A (DDRA)	$0004
Data Direction Register B (DDRB)	$0005
Data Direction Register C (DDRC)	$0006
Data Direction Register D (DDRD)	$0007
Unimplemented	$0008
	$0009
SIOP Control Register (SCR)	$000A
SIOP Status Register (SSR)	$000B
SIOP Data Register (SDR)	$000C
Unimplemented	$000D
	$000E
	$000F
	$0010
	$0011
Timer Control Register (TCR)	$0012
Timer Status Register (TSR)	$0013
Input Capture Register High (ICRH)	$0014
Input Capture Register Low (ICRL)	$0015
Output Compare Register High (OCRH)	$0016
Output Compare Register Low (OCRL)	$0017
Timer Register High (TRH)	$0018
Timer Register Low (TRL)	$0019
Alternate Timer Register High (ATRH)	$001A
Alternate Timer Register Low (ATRL)	$001B
EPROM Programming Register (EPROG)	$001C
ADC Data Register (ADDR)	$001D
ADC Status/Control Register (ADSCR)	$001E
Reserved	$001F

Vector	Address
Timer Interrupt Vector High	$1FF8
Timer Interrupt Vector Low	$1FF9
External Interrupt Vector High	$1FFA
External Interrupt Vector Low	$1FFB
Software Interrupt Vector High	$1FFC
Software Interrupt Vector Low	$1FFD
Reset Vector High	$1FFE
Reset Vector Low	$1FFF

Figure 1. Memory Map

2-mc68hc705p9

Input/Output Register Summary

Addr.	Name	R/W	Bit 7	6	5	4	3	2	1	Bit 0
$0000	Port A Data Register (PORTA)	Read:	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
		Write:								
		Reset:	Unaffected by reset							
$0001	Port B Data Register (PORTB)	Read:	PB7	PB6	PB5	0	0	0	0	0
		Write:								
		Reset:	Unaffected by reset							
$0002	Port C Data Register (PORTC)	Read:	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
		Write:								
		Reset:	Unaffected by reset							
$0003	Port D Data Register (PORTD)	Read:	PD7	0	PD5	1	0	0	0	0
		Write:								
		Reset:	Unaffected by reset							
$0004	Data Direction Register A (DDRA)	Read:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0005	Data Direction Register B (DDRB)	Read:	DDRB7	DDRB6	DDRB5	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0006	Data Direction Register C (DDRC)	Read:	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0007	Data Direction Register D (DDRD)	Read:	0	0	DDRD5	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0008	Unimplemented									
$0009	Unimplemented									

= Unimplemented R = Reserved U = Unaffected

Figure 2. I/O Register Summary

3-mc68hc705p9

Addr.	Name	R/W	Bit 7	6	5	4	3	2	1	Bit 0
$000A	SIOP Control Register (SCR)	Read: Write:	0	SPE	0	MSTR	0	0	0	0
		Reset:	0	0	0	0	0	0	0	0
$000B	SIOP Status Register (SSR)	Read:	SPIF	DCOL	0	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000C	SIOP Data Register (SDR)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
		Reset:	Unaffected by reset							
$000D	Unimplemented									
$000E	Unimplemented									
$000F	Unimplemented									
$0010	Unimplemented									
$0011	Unimplemented									
$0012	Timer Control Register (TCR)	Read: Write:	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL
		Reset:	0	0	0	0	0	0	U	0
$0013	Timer Status Register (TSR)	Read:	ICF	OCF	TOF	0	0	0	0	0
		Write:								
		Reset:	Unaffected by reset			0	0	0	0	0
$0014	Input Capture Register High (ICRH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:								
$0015	Input Capture Register Low (ICRL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Unaffected by reset							
$0016	Output Compare Register High (OCRH)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
		Reset:	Unaffected by reset							

= Unimplemented R = Reserved U = Unaffected

Figure 2. I/O Register Summary (Continued)

4-mc68hc705p9

Addr.	Name	R/W	Bit 7	6	5	4	3	2	1	Bit 0
$0017	Output Compare Register Low (OCRL)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
		Reset:	Unaffected by reset							
$0018	Timer Register High (TRH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	Reset initializes TRH to $FF							
$0019	Timer Register Low (TRL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Reset initializes TRL to $FC							
$001A	Alternate Timer Register High (ATRH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	Reset initializes ATRH to $FF							
$001B	Alternate Timer Register Low (ATRL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Reset initializes ATRL to $FC							
$001C	EPROM Programming Register (EPROG)	Read:	0	0	0	0	0	LATCH	0	EPGM
		Write:	R	R	R	R	R		R	
		Reset:	Unaffected by reset							
$001D	ADC Data Register (ADDR)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Unaffected by reset							
$001E	ADC Status/Control Register (ADSCR)	Read:	CCF	ADRC	ADON	0	0	CH2	CH1	CH0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$001F	Reserved	Read: Write:	R	R	R	R	R	R	R	R
		Reset:								

= Unimplemented R = Reserved U = Unaffected

Figure 2. I/O Register Summary (Continued)

5-mc68hc705p9

Addr.	Name	R/W	Bit 7	6	5	4	3	2	1	Bit 0
$0900	Mask Option Register (MOR)	Read:	0	0	0	0	0	SIOP	IRQ	COPE
		Write:								
		Reset:	Unaffected by reset							0
$1FF0	COP Register (COPR)	Read:	R	R	R	R	R	R	R	COPC
		Write:								
		Reset:	Unaffected by reset							

= Unimplemented R = Reserved U = Unaffected

Figure 2. I/O Register Summary (Continued)

RAM

The 128 addresses from $0080–$00FF are RAM locations. The CPU uses the top 64 RAM addresses, $00C0–$00FF, as the stack. Before processing an interrupt, the CPU uses five bytes of the stack to save the contents of the CPU registers. During a subroutine call, the CPU uses two bytes of the stack to store the return address. The stack pointer decrements when the CPU stores a byte on the stack and increments when the CPU retrieves a byte from the stack.

NOTE: *Be careful when using nested subroutines or multiple interrupt levels. The CPU may overwrite data in the RAM during a subroutine or during the interrupt stacking operation.*

EPROM/OTPROM

An MCU with a quartz window has 2104 bytes of erasable, programmable ROM (EPROM). The quartz window allows EPROM erasure with ultraviolet light.

NOTE: *Keep the quartz window covered with an opaque material except when programming the MCU. Ambient light may affect MCU operation.*

In an MCU without the quartz window, the EPROM cannot be erased and serves as 2104 bytes of one-time programmable ROM (OTPROM). The following addresses are user EPROM/OTPROM locations:

- $0020–$004F
- $0100–$08FF
- $1FF8–$1FFF (reserved for user-defined interrupt and reset vectors)

The mask option register (MOR) is an EPROM/OTPROM location at address $0900.

EPROM/ OTPROM Programming

The two ways to program the EPROM/OTPROM are:

- Manipulating the control bits in the EPROM programming register to program the EPROM/OTPROM on a byte-by-byte basis
- Activating the bootloader ROM to download the contents of an external memory device to the on-chip EPROM/OTPROM

EPROM Programming Register

The EPROM programming register contains the control bits for programming the EPROM/OTPROM.

$001C	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	LATCH	0	EPGM
Write:	R	R	R	R	R		R	
Reset:	0	0	0	0	0	0	0	0

R = Reserved

Figure 3. EPROM Programming Register (EPROG)

LATCH — EPROM Bus Latch

This read/write bit latches the address and data buses for EPROM/OTPROM programming. Clearing the LATCH bit automatically clears the EPGM bit. EPROM/OTPROM data cannot be read while the LATCH bit is set. Resets clear the LATCH bit.

1 = Address and data buses configured for EPROM/OTPROM programming

0 = Address and data buses configured for normal operation

EPGM bit— EPROM Programming

This read/write bit applies the voltage from the $\overline{IRQ}/V_{PP}$ pin to the EPROM/OTPROM. To write the EPGM bit, the LATCH bit must already be set. Clearing the LATCH bit also clears the EPGM bit. Resets clear the EPGM bit.

1 = EPROM/OTPROM programming power switched on

0 = EPROM/OTPROM programming power switched off

NOTE: *Writing logic ones to both the LATCH and EPGM bits with a single instruction sets LATCH and clears EPGM. LATCH must be set first by a separate instruction.*

Bits 7–3 and Bit 1— Reserved

Bits 7–3 and bit 1 are factory test bits that always read as logic zeros.

Take the following steps to program a byte of EPROM/OTPROM:

1. Apply 16.5 V to the $\overline{IRQ}/V_{PP}$ pin.
2. Set the LATCH bit.
3. Write to any EPROM/OTPROM address.
4. Set the EPGM bit for a time, t_{EPGM}, to apply the programming voltage.
5. Clear the LATCH bit.

Bootloader ROM

The bootloader ROM, located at addresses $1F00–$1FEF, contains routines for copying an external EPROM to the on-chip EPROM/OTPROM.

The bootloader copies to the following EPROM/OTPROM addresses:

- $0020–$004F
- $0100–$0900
- $1FF0–$1FFF

The addresses of the code in the external EPROM must match the MC68HC705P9 addresses. The bootloader ignores all other addresses.

Figure 4 shows the circuit for downloading to the on-chip EPROM/OTPROM from a 2764 EPROM. The bootloader circuit includes an external 12-bit counter to address the external EPROM. Operation is fastest when unused external EPROM addresses contain $00. The bootloader function begins when a rising edge occurs on the $\overline{RESET}$ pin while the V_{PP} voltage is on the $\overline{IRQ}/V_{PP}$ pin, and the PD7/TCAP pin is at logic one.

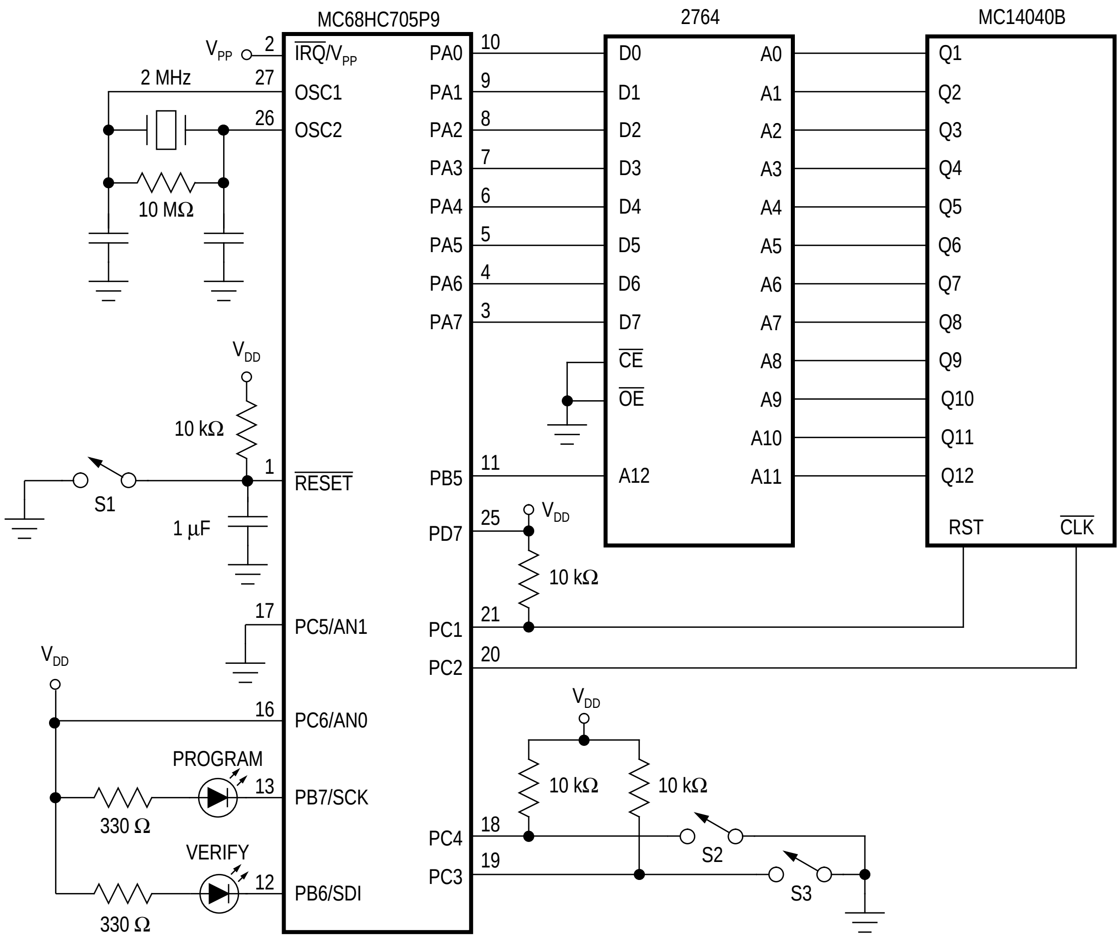


Figure 4. Bootloader Circuit

The logical states of the PC4/AN2 and PC3/AN3 pins select the bootloader function, as **Table 1** shows.

Table 1. Bootloader Function Selection

PC4/AN2	PC3/AN3	Function
1	1	Program and Verify
1	0	Verify Only

Complete the following steps to bootload the MCU:

1. Turn off all power to the circuit.
2. Install the EPROM containing the code to be downloaded.
3. Install the MCU.
4. Select the bootloader function:
   a. Open switches S2 and S3 to select the program and verify function.
   b. Open only switch S2 to select only the verify function.
5. Close switch S1.
6. Turn on the V_{DD} power supply.

CAUTION: *Turn on the V_{DD} power supply **before** turning on the V_{PP} power supply.*

7. Turn on the V_{PP} power supply.
8. Open switch S1. The bootloader code begins to execute. If the PROGRAM function is selected, the PROGRAM LED turns on during programming. If the VERIFY function is selected, the VERIFY LED turns on when verification is successful. The PROGRAM and VERIFY functions take about 10 seconds.
9. Close switch S1.
10. Turn off the V_{PP} power supply.

CAUTION: *Turn off the V_{PP} power supply **before** turning off the V_{DD} power supply.*

11. Turn off the V_{DD} power supply.

EPROM Erasing

The erased state of an EPROM bit is zero. Erase the EPROM by exposing it to 15 Ws/cm^2 of ultraviolet light with a wavelength of 2537 angstroms. Position the ultraviolet light source one inch from the EPROM. Do not use a shortwave filter.

Cerdip packages have a transparent window for erasing the EPROM with ultraviolet light. In the windowless PDIP and SOIC packages, the 2104 EPROM bytes function as one-time programmable ROM (OTPROM).

Mask Option Register

The mask option register (MOR) is an EPROM/OTPROM byte that is programmable only with the bootloader function. The MOR controls the following options:

- LSB first or MSB first SIOP data transfer
- Edge-triggered or edge- and level-triggered external interrupt pin
- Enabled or disabled COP watchdog

To program the MOR, use the 5-step procedure given in the section **EPROM Programming Register** on page 26. Write to address $0900 in step 3.

$0900	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	SIOP	IRQ	COPE
Write:								
Reset:	Unaffected by reset							
Erased:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 5. Mask Option Register (MOR)

SIOP — Serial I/O Port

The SIOP bit controls the shift direction into and out of the SIOP shift register.

1 = SIOP data transferred LSB first (bit 0 first)

0 = SIOP data transferred MSB first (bit 7 first)

IRQ — Interrupt Request

The IRQ bit makes the external interrupt function of the $\overline{\text{IRQ}}/V_{PP}$ pin level-triggered as well as edge-triggered.

1 = $\overline{\text{IRQ}}/V_{PP}$ pin negative-edge triggered and low-level triggered

0 = $\overline{\text{IRQ}}/V_{PP}$ pin negative-edge triggered only

COPE — COP Enable

COPE enables the COP watchdog. In applications that have wait cycles longer than the COP watchdog timeout period, the COP watchdog can be disabled by not programming the COPE bit to logic one.

1 = COP watchdog enabled
0 = COP watchdog disabled

Central Processor Unit
CPU

Contents

1-hc05cpu

Features

- 2.1-MHz Bus Frequency
- 8-Bit Accumulator
- 8-Bit Index Register
- 13-Bit Program Counter
- 6-Bit Stack Pointer
- Condition Code Register with Five Status Flags
- 62 Instructions
- Eight Addressing Modes
- Power-Saving Stop, Wait, and Data-Retention Modes

Introduction

The central processor unit (CPU) consists of a CPU control unit, an arithmetic/logic unit (ALU), and five CPU registers. The CPU control unit fetches and decodes instructions. The ALU executes the instructions. The CPU registers contain data, addresses, and status bits that reflect the results of CPU operations.

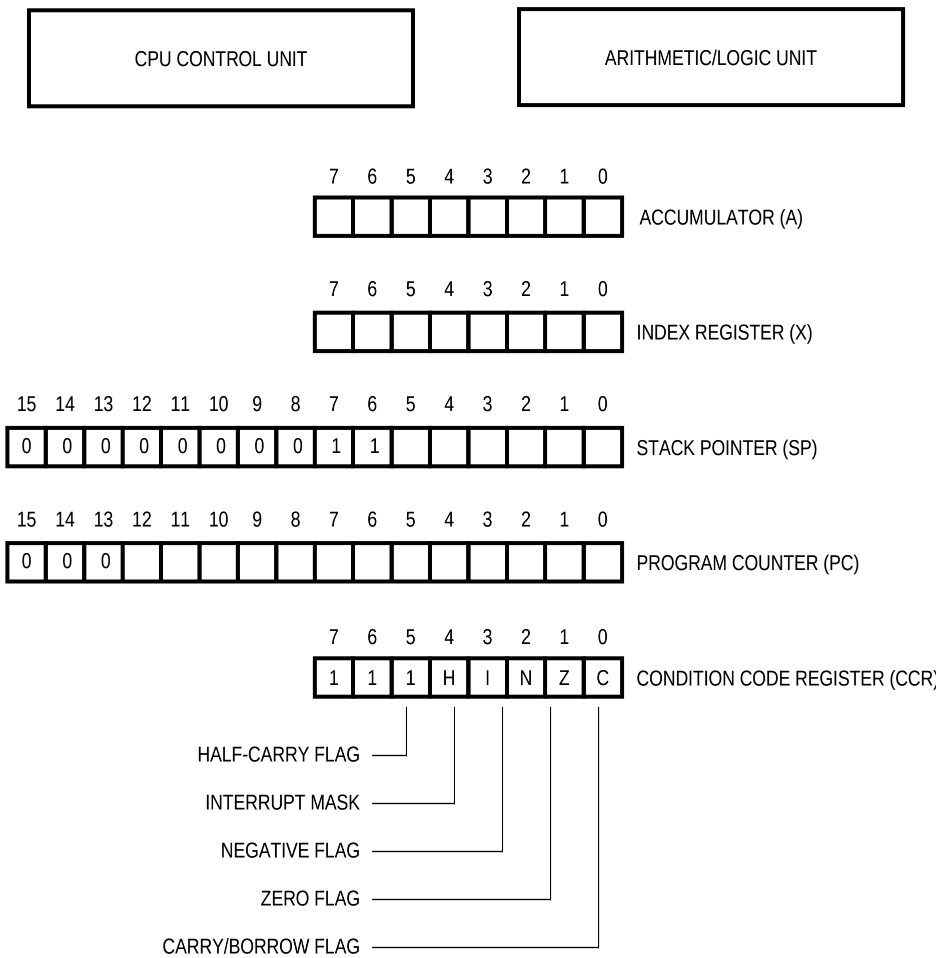


Figure 1. CPU Programming Model

3-hc05cpu

CPU Control Unit

The CPU control unit fetches and decodes instructions during program operation. The control unit selects the memory locations to read and write and coordinates the timing of all CPU operations.

Arithmetic/Logic Unit

The arithmetic/logic unit (ALU) performs the arithmetic, logic, and manipulation operations decoded from the instruction set by the CPU control unit. The ALU produces the results called for by the program and sets or clears status and control bits in the condition code register (CCR).

CPU Registers

The M68HC05 CPU contains five registers that control and monitor MCU operation:

- Accumulator
- Index register
- Stack pointer
- Program counter
- Condition code register

CPU registers are not memory mapped.

Accumulator

The accumulator is a general-purpose 8-bit register. The CPU uses the accumulator to hold operands and the results of arithmetic and logic operations.

	Bit 7	6	5	4	3	2	1	Bit 0
Read:								
Write:								
Reset:	Unaffected by reset							

Figure 2. Accumulator (A)

Index Register

The index register can be used for data storage or as a counter. In the indexed addressing modes, the CPU uses the byte in the index register to determine the effective address of the operand.

	Bit 7	6	5	4	3	2	1	Bit 0
Read:								
Write:								
Reset:	Unaffected by reset							

Figure 3. Index Register (X)

Stack Pointer

The stack pointer is a 16-bit register that contains the address of the next stack location to be used. During a reset or after the reset stack pointer instruction (RSP), the stack pointer is preset to $00FF. The address in the stack pointer decrements after a byte is stacked and increments before a byte is unstacked.

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	0	0	1	1						
Write:																
Reset:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

= Unimplemented

Figure 4. Stack Pointer (SP)

The 10 most significant bits of the stack pointer are permanently fixed at 0000000011, so the stack pointer produces addresses from $00C0 to $00FF. If subroutines and interrupts use more than 64 stack locations, the stack pointer wraps around to address $00FF and begins writing over the previously stored data. A subroutine uses two stack locations; an interrupt uses five locations.

Program Counter

The program counter is a 16-bit register that contains the address of the next instruction or operand to be fetched. The three most significant bits of the program counter are ignored internally and appear as 000.

Normally, the address in the program counter automatically increments to the next sequential memory location every time an instruction or operand is fetched. Jump, branch, and interrupt operations load the program counter with an address other than that of the next sequential location.

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
Read:																
Write:																
Reset:	0	0	0	Loaded with vector from $1FFE and $1FFF												

Figure 5. Program Counter (PC)

Condition Code Register

The condition code register is an 8-bit register whose three most significant bits are permanently fixed at 111. The condition code register contains the interrupt mask and four flags that indicate the results of the instruction just executed.

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	1	1	1	H	I	N	Z	C
Write:								
Reset:	1	1	1	U	1	U	U	U

= Unimplemented U = Unaffected

Figure 6. Condition Code Register (CCR)

6-hc05cpu

H — Half-Carry Flag

The CPU sets the half-carry flag when a carry occurs between bits 3 and 4 of the accumulator during an ADD or ADC operation. The half-carry flag is required for binary-coded decimal (BCD) arithmetic operations.

I — Interrupt Mask

Setting the interrupt mask disables interrupts. If an interrupt request occurs while the interrupt mask is logic zero, the CPU saves the CPU registers on the stack, sets the interrupt mask, and then fetches the interrupt vector. If an interrupt request occurs while the interrupt mask is set, the interrupt request is latched. Normally, the CPU processes the latched interrupt as soon as the interrupt mask is cleared again.

A return from interrupt (RTI) instruction pulls the CPU registers from the stack, restoring the interrupt mask to its cleared state. After any reset, the interrupt mask is set and can be cleared only by a software instruction.

N — Negative Flag

The CPU sets the negative flag when an arithmetic operation, logical operation, or data manipulation produces a negative result.

Z — Zero Flag

The CPU sets the zero flag when an arithmetic operation, logical operation, or data manipulation produces a result of $00.

C — Carry/Borrow Flag

The CPU sets the carry/borrow flag when an addition operation produces a carry out of bit 7 of the accumulator or when a subtraction operation requires a borrow. Some logical operations and data manipulation instructions also clear or set the carry/borrow flag.

Instruction Set

The MCU instruction set has 62 instructions and uses eight addressing modes. The instructions include all those of the M146805 CMOS Family plus one more: the unsigned multiply (MUL) instruction. The MUL instruction allows unsigned multiplication of the contents of the accumulator (A) and the index register (X). The high-order product is stored in the index register, and the low-order product is stored in the accumulator.

Addressing Modes

The CPU uses eight addressing modes for flexibility in accessing data. The addressing modes provide eight different ways for the CPU to find the data required to execute an instruction. The eight addressing modes are:

- Inherent
- Immediate
- Direct
- Extended
- Indexed, no offset
- Indexed, 8-bit offset
- Indexed, 16-bit offset
- Relative

Inherent

Inherent instructions are those that have no operand, such as return from interrupt (RTI) and stop (STOP). Some of the inherent instructions act on data in the CPU registers, such as set carry flag (SEC) and increment accumulator (INCA). Inherent instructions require no operand address and are one byte long.

Immediate

Immediate instructions are those that contain a value to be used in an operation with the value in the accumulator or index register. Immediate instructions require no operand address and are two bytes long. The opcode is the first byte, and the immediate data value is the second byte.

Direct

Direct instructions can access any of the first 256 memory locations with two bytes. The first byte is the opcode, and the second is the low byte of the operand address. In direct addressing, the CPU automatically uses $00 as the high byte of the operand address.

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 Motorola 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.

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.

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.

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 Motorola assembler determines the shortest form of indexed addressing.

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 Motorola 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.

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

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 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

Read-Modify-Write Instructions

These instructions read a memory location or a register, modify its contents, and write the modified value back to the memory location or to the register.

NOTE: *Do not use read-modify-write operations on write-only registers.*

Table 2. Read-Modify-Write Instructions

Instruction	Mnemonic
Arithmetic Shift Left (Same as LSL)	ASL
Arithmetic Shift Right	ASR
Bit Clear	BCLR(1)
Bit Set	BSET(1)
Clear Register	CLR
Complement (One’s Complement)	COM
Decrement	DEC
Increment	INC
Logical Shift Left (Same as ASL)	LSL
Logical Shift Right	LSR
Negate (Two’s Complement)	NEG
Rotate Left through Carry Bit	ROL
Rotate Right through Carry Bit	ROR
Test for Negative or Zero	TST(2)

1. Unlike other read-modify-write instructions, BCLR and BSET use only direct addressing.
2. TST is an exception to the read-modify-write sequence because it does not write a replacement value.

12-hc05cpu

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 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

Bit Manipulation Instructions

The CPU can set or clear any writable bit in the first 256 bytes of memory, which includes I/O registers and on-chip RAM locations. The CPU can also test and branch based on the state of any bit in any of the first 256 memory locations.

Table 4. Bit Manipulation Instructions

Instruction	Mnemonic
Bit Clear	BCLR
Branch if Bit Clear	BRCLR
Branch if Bit Set	BRSET
Bit Set	BSET

Control Instructions

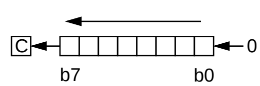
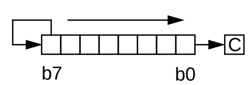
These instructions act on CPU registers and control CPU operation during program execution.

Table 5. Control Instructions

Instruction	Mnemonic
Clear Carry Bit	CLC
Clear Interrupt Mask	CLI
No Operation	NOP
Reset Stack Pointer	RSP
Return from Interrupt	RTI
Return from Subroutine	RTS
Set Carry Bit	SEC
Set Interrupt Mask	SEI
Stop Oscillator and Enable $\overline{\text{IRQ}}$ Pin	STOP
Software Interrupt	SWI
Transfer Accumulator to Index Register	TAX
Transfer Index Register to Accumulator	TXA
Stop CPU Clock and Enable Interrupts	WAIT

Instruction Set Summary

Table 6. Instruction Set Summary

Source Form	Operation	Description	Effect on CCR H	I	N	Z	C	Address Mode	Opcode	Operand	Cycles
ADC #*opr*	Add with Carry	A ← (A) + (M) + (C)	↕	—	↕	↕	↕	IMM	A9	ii	2
ADC *opr*								DIR	B9	dd	3
ADC *opr*								EXT	C9	hh ll	4
ADC *opr*,X								IX2	D9	ee ff	5
ADC *opr*,X								IX1	E9	ff	4
ADC ,X								IX	F9		3
ADD #*opr*	Add without Carry	A ← (A) + (M)	↕	—	↕	↕	↕	IMM	AB	ii	2
ADD *opr*								DIR	BB	dd	3
ADD *opr*								EXT	CB	hh ll	4
ADD *opr*,X								IX2	DB	ee ff	5
ADD *opr*,X								IX1	EB	ff	4
ADD ,X								IX	FB		3
AND #*opr*	Logical AND	A ← (A) ∧ (M)	—	—	↕	↕	—	IMM	A4	ii	2
AND *opr*								DIR	B4	dd	3
AN*D opr*								EXT	C4	hh ll	4
AND *opr*,X								IX2	D4	ee ff	5
AND *opr*,X								IX1	E4	ff	4
AND ,X								IX	F4		3
ASL *opr*	Arithmetic Shift Left (Same as LSL)		—	—	↕	↕	↕	DIR	38	dd	5
ASLA								INH	48		3
ASLX								INH	58		3
ASL *opr*,X								IX1	68	ff	6
ASL ,X								IX	78		5
ASR *opr*	Arithmetic Shift Right		—	—	↕	↕	↕	DIR	37	dd	5
ASRA								INH	47		3
ASRX								INH	57		3
ASR *opr*,X								IX1	67	ff	6
ASR ,X								IX	77		5
BCC *rel*	Branch if Carry Bit Clear	PC ← (PC) + 2 + *rel* ? C = 0	—	—	—	—	—	REL	24	rr	3
BCLR *n opr*	Clear Bit n	Mn ← 0	—	—	—	—	—	DIR (b0)	11	dd	5
								DIR (b1)	13	dd	5
								DIR (b2)	15	dd	5
								DIR (b3)	17	dd	5
								DIR (b4)	19	dd	5
								DIR (b5)	1B	dd	5
								DIR (b6)	1D	dd	5
								DIR (b7)	1F	dd	5
BCS *rel*	Branch if Carry Bit Set (Same as BLO)	PC ← (PC) + 2 + *rel* ? C = 1	—	—	—	—	—	REL	25	rr	3
BEQ *rel*	Branch if Equal	PC ← (PC) + 2 + *rel* ? Z = 1	—	—	—	—	—	REL	27	rr	3
BHCC *rel*	Branch if Half-Carry Bit Clear	PC ← (PC) + 2 + *rel* ? H = 0	—	—	—	—	—	REL	28	rr	3
BHCS *rel*	Branch if Half-Carry Bit Set	PC ← (PC) + 2 + *rel* ? H = 1	—	—	—	—	—	REL	29	rr	3

16-hc05cpu

Table 6. Instruction Set Summary (Continued)

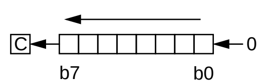
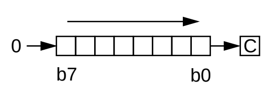
Source Form	Operation	Description	Effect on CCR H	I	N	Z	C	Address Mode	Opcode	Operand	Cycles
BHI *rel*	Branch if Higher	PC ← (PC) + 2 + *rel* ? C ∨ Z = 0	—	—	—	—	—	REL	22	rr	3
BHS *rel*	Branch if Higher or Same	PC ← (PC) + 2 + *rel* ? C = 0	—	—	—	—	—	REL	24	rr	3
BIH *rel*	Branch if IRQ Pin High	PC ← (PC) + 2 + *rel* ? IRQ = 1	—	—	—	—	—	REL	2F	rr	3
BIL *rel*	Branch if IRQ Pin Low	PC ← (PC) + 2 + *rel* ? IRQ = 0	—	—	—	—	—	REL	2E	rr	3
BIT #*opr*	Bit Test Accumulator with Memory Byte	(A) ∧ (M)	—	—	↕	↕	—	IMM	A5	ii	2
BIT *opr*								DIR	B5	dd	3
BIT *opr*								EXT	C5	hh ll	4
BIT *opr*,X								IX2	D5	ee ff	5
BIT *opr*,X								IX1	E5	ff	4
BIT ,X								IX	F5		3
BLO *rel*	Branch if Lower (Same as BCS)	PC ← (PC) + 2 + *rel* ? C = 1	—	—	—	—	—	REL	25	rr	3
BLS *rel*	Branch if Lower or Same	PC ← (PC) + 2 + *rel* ? C ∨ Z = 1	—	—	—	—	—	REL	23	rr	3
BMC *rel*	Branch if Interrupt Mask Clear	PC ← (PC) + 2 + *rel* ? I = 0	—	—	—	—	—	REL	2C	rr	3
BMI *rel*	Branch if Minus	PC ← (PC) + 2 + *rel* ? N = 1	—	—	—	—	—	REL	2B	rr	3
BMS *rel*	Branch if Interrupt Mask Set	PC ← (PC) + 2 + *rel* ? I = 1	—	—	—	—	—	REL	2D	rr	3
BNE *rel*	Branch if Not Equal	PC ← (PC) + 2 + *rel* ? Z = 0	—	—	—	—	—	REL	26	rr	3
BPL *rel*	Branch if Plus	PC ← (PC) + 2 + *rel* ? N = 0	—	—	—	—	—	REL	2A	rr	3
BRA *rel*	Branch Always	PC ← (PC) + 2 + *rel* ? 1 = 1	—	—	—	—	—	REL	20	rr	3
BRCLR *n opr rel*	Branch if Bit n Clear	PC ← (PC) + 2 + *rel* ? Mn = 0	—	—	—	—	↕	DIR (b0)	01	dd rr	5
								DIR (b1)	03	dd rr	5
								DIR (b2)	05	dd rr	5
								DIR (b3)	07	dd rr	5
								DIR (b4)	09	dd rr	5
								DIR (b5)	0B	dd rr	5
								DIR (b6)	0D	dd rr	5
								DIR (b7)	0F	dd rr	5
BRN *rel*	Branch Never	PC ← (PC) + 2 + *rel* ? 1 = 0	—	—	—	—	—	REL	21	rr	3
BRSET *n opr rel*	Branch if Bit n Set	PC ← (PC) + 2 + *rel* ? Mn = 1	—	—	—	—	↕	DIR (b0)	00	dd rr	5
								DIR (b1)	02	dd rr	5
								DIR (b2)	04	dd rr	5
								DIR (b3)	06	dd rr	5
								DIR (b4)	08	dd rr	5
								DIR (b5)	0A	dd rr	5
								DIR (b6)	0C	dd rr	5
								DIR (b7)	0E	dd rr	5
BSET *n opr*	Set Bit n	Mn ← 1	—	—	—	—	—	DIR (b0)	10	dd	5
								DIR (b1)	12	dd	5
								DIR (b2)	14	dd	5
								DIR (b3)	16	dd	5
								DIR (b4)	18	dd	5
								DIR (b5)	1A	dd	5
								DIR (b6)	1C	dd	5
								DIR (b7)	1E	dd	5

Table 6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR H	I	N	Z	C	Address Mode	Opcode	Operand	Cycles
BSR *rel*	Branch to Subroutine	PC ← (PC) + 2; push (PCL) SP ← (SP) – 1; push (PCH) SP ← (SP) – 1 PC ← (PC) + *rel*	—	—	—	—	—	REL	AD	rr	6
CLC	Clear Carry Bit	C ← 0	—	—	—	—	0	INH	98		2
CLI	Clear Interrupt Mask	I ← 0	—	0	—	—	—	INH	9A		2
CLR *opr*	Clear Byte	M ← $00	—	—	0	1	—	DIR	3F	dd	5
CLRA		A ← $00						INH	4F		3
CLRX		X ← $00						INH	5F		3
CLR *opr*,X		M ← $00						IX1	6F	ff	6
CLR ,X		M ← $00						IX	7F		5
CMP #*opr*	Compare Accumulator with Memory Byte	(A) – (M)	—	—	↕	↕	↕	IMM	A1	ii	2
CMP *opr*								DIR	B1	dd	3
CMP *opr*								EXT	C1	hh ll	4
CMP *opr*,X								IX2	D1	ee ff	5
CMP *opr*,X								IX1	E1	ff	4
CMP ,X								IX	F1		3
COM *opr*	Complement Byte (One's Complement)	M ← $(\overline{M})$ = $FF – (M)	—	—	↕	↕	1	DIR	33	dd	5
COMA		A ← $(\overline{A})$ = $FF – (A)						INH	43		3
COMX		X ← $(\overline{X})$ = $FF – (X)						INH	53		3
COM *opr*,X		M ← $(\overline{M})$ = $FF – (M)						IX1	63	ff	6
COM ,X		M ← $(\overline{M})$ = $FF – (M)						IX	73		5
CPX #*opr*	Compare Index Register with Memory Byte	(X) – (M)	—	—	↕	↕	↕	IMM	A3	ii	2
CPX *opr*								DIR	B3	dd	3
CPX *opr*								EXT	C3	hh ll	4
CPX *opr*,X								IX2	D3	ee ff	5
CPX *opr*,X								IX1	E3	ff	4
CPX ,X								IX	F3		3
DEC *opr*	Decrement Byte	M ← (M) – 1	—	—	↕	↕	—	DIR	3A	dd	5
DECA		A ← (A) – 1						INH	4A		3
DECX		X ← (X) – 1						INH	5A		3
DEC *opr*,X		M ← (M) – 1						IX1	6A	ff	6
DEC ,X		M ← (M) – 1						IX	7A		5
EOR #*opr*	EXCLUSIVE OR Accumulator with Memory Byte	A ← (A) ⊕ (M)	—	—	↕	↕	—	IMM	A8	ii	2
EOR *opr*								DIR	B8	dd	3
EOR *opr*								EXT	C8	hh ll	4
EOR *opr*,X								IX2	D8	ee ff	5
EOR *opr*,X								IX1	E8	ff	4
EOR ,X								IX	F8		3
INC *opr*	Increment Byte	M ← (M) + 1	—	—	↕	↕	—	DIR	3C	dd	5
INCA		A ← (A) + 1						INH	4C		3
INCX		X ← (X) + 1						INH	5C		3
INC *opr*,X		M ← (M) + 1						IX1	6C	ff	6
INC ,X		M ← (M) + 1						IX	7C		5

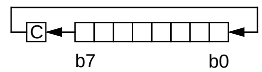
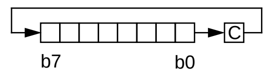
18-hc05cpu

Table 6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR H	I	N	Z	C	Address Mode	Opcode	Operand	Cycles
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
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		11
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

19-hc05cpu

Table 6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR H	I	N	Z	C	Address Mode	Opcode	Operand	Cycles
ROL *opr*	Rotate Byte Left through Carry Bit		—	—	↕	↕	↕	DIR	39	dd	5
ROLA								INH	49		3
ROLX								INH	59		3
ROL *opr*,X								IX1	69	ff	6
ROL ,X								IX	79		5
ROR *opr*	Rotate Byte Right through Carry Bit		—	—	↕	↕	↕	DIR	36	dd	5
RORA								INH	46		3
RORX								INH	56		3
ROR *opr*,X								IX1	66	ff	6
ROR ,X								IX	76		5
RSP	Reset Stack Pointer	SP ← $00FF	—	—	—	—	—	INH	9C		2
RTI	Return from Interrupt	SP ← (SP) + 1; Pull (CCR) SP ← (SP) + 1; Pull (A) SP ← (SP) + 1; Pull (X) SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)	↕	↕	↕	↕	↕	INH	80		9
RTS	Return from Subroutine	SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)	—	—	—	—	—	INH	81		6
SBC #*opr*	Subtract Memory Byte and Carry Bit from Accumulator	A ← (A) – (M) – (C)	—	—	↕	↕	↕	IMM	A2	ii	2
SBC *opr*								DIR	B2	dd	3
SBC *opr*								EXT	C2	hh ll	4
SBC *opr*,X								IX2	D2	ee ff	5
SBC *opr*,X								IX1	E2	ff	4
SBC ,X								IX	F2		3
SEC	Set Carry Bit	C ← 1	—	—	—	—	1	INH	99		2
SEI	Set Interrupt Mask	I ← 1	—	1	—	—	—	INH	9B		2
STA *opr*	Store Accumulator in Memory	M ← (A)	—	—	↕	↕	—	DIR	B7	dd	4
STA *opr*								EXT	C7	hh ll	5
STA *opr*,X								IX2	D7	ee ff	6
STA *opr*,X								IX1	E7	ff	5
STA ,X								IX	F7		4
STOP	Stop Oscillator and Enable IRQ Pin		—	0	—	—	—	INH	8E		2
STX *opr*	Store Index Register In Memory	M ← (X)	—	—	↕	↕	—	DIR	BF	dd	4
STX *opr*								EXT	CF	hh ll	5
STX *opr*,X								IX2	DF	ee ff	6
STX *opr*,X								IX1	EF	ff	5
STX ,X								IX	FF		4
SUB #*opr*	Subtract Memory Byte from Accumulator	A ← (A) – (M)	—	—	↕	↕	↕	IMM	A0	ii	2
SUB *opr*								DIR	B0	dd	3
SUB *opr*								EXT	C0	hh ll	4
SUB *opr*,X								IX2	D0	ee ff	5
SUB *opr*,X								IX1	E0	ff	4
SUB ,X								IX	F0		3

20-hc05cpu

Table 6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR: H	I	N	Z	C	Address Mode	Opcode	Operand	Cycles
SWI	Software Interrupt	PC ← (PC) + 1; Push (PCL) SP ← (SP) – 1; Push (PCH) SP ← (SP) – 1; Push (X) SP ← (SP) – 1; Push (A) SP ← (SP) – 1; Push (CCR) SP ← (SP) – 1; I ← 1 PCH ← Interrupt Vector High Byte PCL ← Interrupt Vector Low Byte	—	1	—	—	—	INH	83		10
TAX	Transfer Accumulator to Index Register	X ← (A)	—	—	—	—	—	INH	97		2
TST *opr* TSTA TSTX TST *opr*,X TST ,X	Test Memory Byte for Negative or Zero	(M) – $00	—	—	↕	↕	—	DIR INH INH IX1 IX	3D 4D 5D 6D 7D	dd ff 	4 3 3 5 4
TXA	Transfer Index Register to Accumulator	A ← (X)	—	—	—	—	—	INH	9F		2
WAIT	Stop CPU Clock and Enable Interrupts		—	0×	—	—	—	INH	8F		2

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

21-hc05cpu

Table 7. Opcode Map

	Bit Manipulation		Branch	Read-Modify-Write					Control		Register/Memory						
	DIR	DIR	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
MSB / LSB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	MSB / LSB
0	5 BRSET0 3 DIR	5 BSET0 2 DIR	3 BRA 2 REL	5 NEG 2 DIR	3 NEGA 1 INH	3 NEGX 1 INH	6 NEG 2 IX1	5 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	3 SUB 2 DIR	4 SUB 3 EXT	5 SUB 3 IX2	4 SUB 2 IX1	3 SUB 1 IX	0
1	5 BRCLR0 3 DIR	5 BCLR0 2 DIR	3 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	3 CMP 2 DIR	4 CMP 3 EXT	5 CMP 3 IX2	4 CMP 2 IX1	3 CMP 1 IX	1
2	5 BRSET1 3 DIR	5 BSET1 2 DIR	3 BHI 2 REL		11 MUL 1 INH						2 SBC 2 IMM	3 SBC 2 DIR	4 SBC 3 EXT	5 SBC 3 IX2	4 SBC 2 IX1	3 SBC 1 IX	2
3	5 BRCLR1 3 DIR	5 BCLR1 2 DIR	3 BLS 2 REL	5 COM 2 DIR	3 COMA 1 INH	3 COMX 1 INH	6 COM 2 IX1	5 COM 1 IX	10 SWI 1 INH		2 CPX 2 IMM	3 CPX 2 DIR	4 CPX 3 EXT	5 CPX 3 IX2	4 CPX 2 IX1	3 CPX 1 IX	3
4	5 BRSET2 3 DIR	5 BSET2 2 DIR	3 BCC 2 REL	5 LSR 2 DIR	3 LSRA 1 INH	3 LSRX 1 INH	6 LSR 2 IX1	5 LSR 1 IX			2 AND 2 IMM	3 AND 2 DIR	4 AND 3 EXT	5 AND 3 IX2	4 AND 2 IX1	3 AND 1 IX	4
5	5 BRCLR2 3 DIR	5 BCLR2 2 DIR	3 BCS/BLO 2 REL								2 BIT 2 IMM	3 BIT 2 DIR	4 BIT 3 EXT	5 BIT 3 IX2	4 BIT 2 IX1	3 BIT 1 IX	5
6	5 BRSET3 3 DIR	5 BSET3 2 DIR	3 BNE 2 REL	5 ROR 2 DIR	3 RORA 1 INH	3 RORX 1 INH	6 ROR 2 IX1	5 ROR 1 IX			2 LDA 2 IMM	3 LDA 2 DIR	4 LDA 3 EXT	5 LDA 3 IX2	4 LDA 2 IX1	3 LDA 1 IX	6
7	5 BRCLR3 3 DIR	5 BCLR3 2 DIR	3 BEQ 2 REL	5 ASR 2 DIR	3 ASRA 1 INH	3 ASRX 1 INH	6 ASR 2 IX1	5 ASR 1 IX		2 TAX 1 INH		4 STA 2 DIR	5 STA 3 EXT	6 STA 3 IX2	5 STA 2 IX1	4 STA 1 IX	7
8	5 BRSET4 3 DIR	5 BSET4 2 DIR	3 BHCC 2 REL	5 ASL/LSL 2 DIR	3 ASLA/LSLA 1 INH	3 ASLX/LSLX 1 INH	6 ASL/LSL 2 IX1	5 ASL/LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	3 EOR 2 DIR	4 EOR 3 EXT	5 EOR 3 IX2	4 EOR 2 IX1	3 EOR 1 IX	8
9	5 BRCLR4 3 DIR	5 BCLR4 2 DIR	3 BHCS 2 REL	5 ROL 2 DIR	3 ROLA 1 INH	3 ROLX 1 INH	6 ROL 2 IX1	5 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	3 ADC 2 DIR	4 ADC 3 EXT	5 ADC 3 IX2	4 ADC 2 IX1	3 ADC 1 IX	9
A	5 BRSET5 3 DIR	5 BSET5 2 DIR	3 BPL 2 REL	5 DEC 2 DIR	3 DECA 1 INH	3 DECX 1 INH	6 DEC 2 IX1	5 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	3 ORA 2 DIR	4 ORA 3 EXT	5 ORA 3 IX2	4 ORA 2 IX1	3 ORA 1 IX	A
B	5 BRCLR5 3 DIR	5 BCLR5 2 DIR	3 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	3 ADD 2 DIR	4 ADD 3 EXT	5 ADD 3 IX2	4 ADD 2 IX1	3 ADD 1 IX	B
C	5 BRSET6 3 DIR	5 BSET6 2 DIR	3 BMC 2 REL	5 INC 2 DIR	3 INCA 1 INH	3 INCX 1 INH	6 INC 2 IX1	5 INC 1 IX		2 RSP 1 INH		2 JMP 2 DIR	3 JMP 3 EXT	4 JMP 3 IX2	3 JMP 2 IX1	2 JMP 1 IX	C
D	5 BRCLR6 3 DIR	5 BCLR6 2 DIR	3 BMS 2 REL	4 TST 2 DIR	3 TSTA 1 INH	3 TSTX 1 INH	5 TST 2 IX1	4 TST 1 IX		2 NOP 1 INH	6 BSR 2 REL	5 JSR 2 DIR	6 JSR 3 EXT	7 JSR 3 IX2	6 JSR 2 IX1	5 JSR 1 IX	D
E	5 BRSET7 3 DIR	5 BSET7 2 DIR	3 BIL 2 REL						2 STOP 1 INH		2 LDX 2 IMM	3 LDX 2 DIR	4 LDX 3 EXT	5 LDX 3 IX2	4 LDX 2 IX1	3 LDX 1 IX	E
F	5 BRCLR7 3 DIR	5 BCLR7 2 DIR	3 BIH 2 REL	5 CLR 2 DIR	3 CLRA 1 INH	3 CLRX 1 INH	6 CLR 2 IX1	5 CLR 1 IX	2 WAIT 1 INH	2 TXA 1 INH		4 STX 2 DIR	5 STX 3 EXT	6 STX 3 IX2	5 STX 2 IX1	4 STX 1 IX	F

INH = Inherent
IMM = Immediate
DIR = Direct
EXT = Extended
REL = Relative
IX = Indexed, No Offset
IX1 = Indexed, 8-Bit Offset
IX2 = Indexed, 16-Bit Offset

MSB / LSB	0	MSB of Opcode in Hexadecimal
0	5 BRSET0 3 DIR	Number of Cycles Opcode Mnemonic Number of Bytes/Addressing Mode

LSB of Opcode in Hexadecimal

Resets and Interrupts

Contents

1-mc68hc705p9

Resets

A reset immediately stops the operation of the instruction being executed, initializes certain control bits, and loads the program counter with a user-defined reset vector address. The following sources can generate resets:

- Power-on reset (POR) circuit
- $\overline{\text{RESET}}$ pin
- COP watchdog

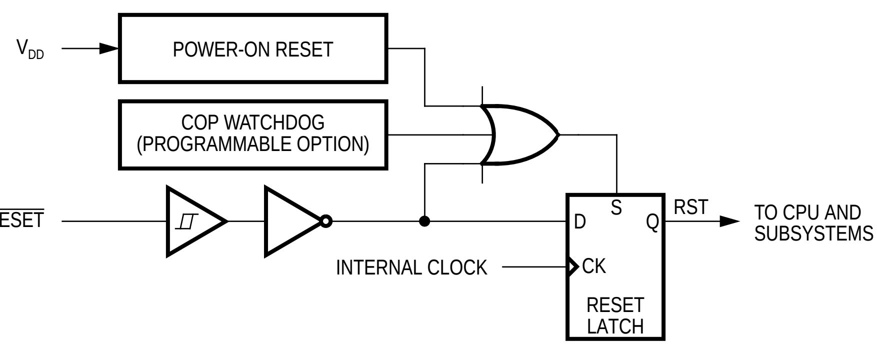


Figure 1. Reset Sources

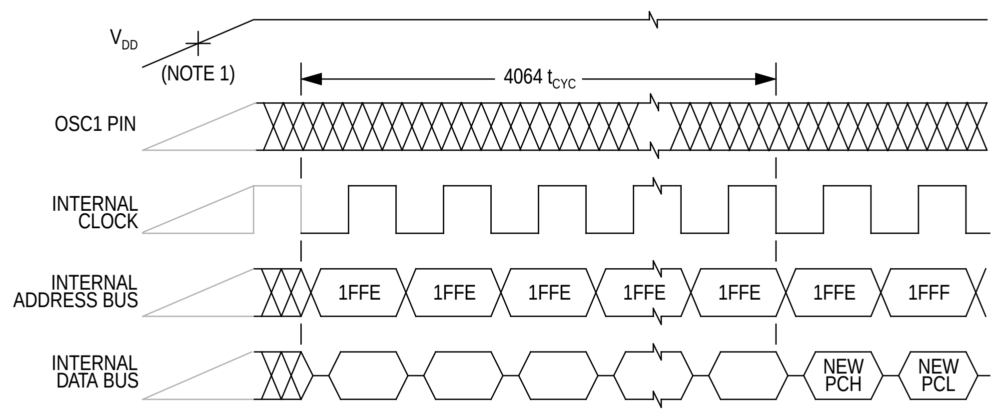
Power-On Reset

A positive transition on the V_{DD} pin generates a power-on reset.

NOTE: *The power-on reset is strictly for power-up conditions and cannot be used to detect drops in power supply voltage.*

A 4064 t_{CYC} (internal clock cycle) delay after the oscillator becomes active allows the clock generator to stabilize. If the $\overline{\text{RESET}}$ pin is at logic zero at the end of 4064 t_{CYC}, the MCU remains in the reset condition until the signal on the $\overline{\text{RESET}}$ pin goes to logic one.

2-mc68hc705p9

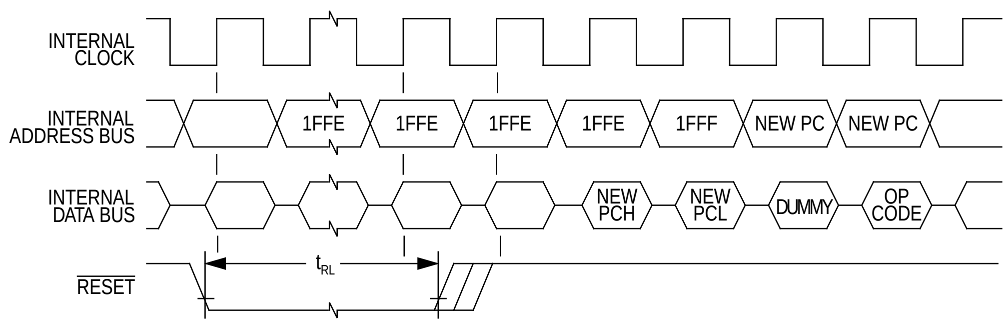


NOTES:
1. Power-on reset threshold is typically between 1 V and 2 V.
2. Internal clock, internal address bus, and internal data bus are not available externally.

Figure 2. Power-On Reset Timing

External Reset

A logic zero applied to the $\overline{\text{RESET}}$ pin for one and one-half t_{CYC} generates an external reset. A Schmitt trigger senses the logic level at the $\overline{\text{RESET}}$ pin.



NOTES:
1. Internal clock, internal address bus, and internal data bus are not available externally.
2. The next rising edge of the internal clock after the rising edge of $\overline{\text{RESET}}$ initiates the reset sequence.

Figure 3. External Reset Timing

Table 1. External Reset Timing

Characteristic	Symbol	Min	Max	Unit
$\overline{\text{RESET}}$ Pulse Width	t_{RL}	1.5	—	t_{CYC}

COP Watchdog Reset

A timeout of the COP watchdog generates a COP reset. The COP watchdog is part of a software error detection system and must be cleared periodically to start a new timeout period. To clear the COP watchdog and prevent a COP reset, write a logic zero to bit 0 (COPC) of the COP register at location $1FF0.

Low-Voltage Protection

A drop in power supply voltage below the minimum operating V_{DD} voltage is called a brownout condition. A brownout while the MCU is in a non-reset state can corrupt MCU operation and necessitate a power-on reset to resume operation.

The best protection against brownout is an undervoltage sensing circuit that pulls the $\overline{\text{RESET}}$ pin low when it detects a low-power supply voltage. The undervoltage sensing circuit may be made of discrete components or an integrated circuit can be used.

For information about brownout and the COP watchdog, see the **Computer Operating Properly Watchdog** section.

Interrupts

The following sources can generate interrupts:

- SWI instruction
- $\overline{IRQ}/V_{PP}$ pin
- Capture/compare timer

An interrupt temporarily stops normal program execution to process a particular event. An interrupt does not stop the operation of the instruction being executed, but takes effect when the current instruction completes its execution. Interrupt processing automatically saves the CPU registers on the stack and loads the program counter with a user-defined interrupt vector address.

Software Interrupt

The software interrupt (SWI) instruction causes a non-maskable interrupt.

External Interrupt

An interrupt signal on the $\overline{IRQ}/V_{PP}$ pin latches an external interrupt request. When the CPU completes its current instruction, it tests the IRQ latch. If the IRQ latch is set, the CPU then tests the I bit in the condition code register. If the I bit is clear, the CPU then begins the interrupt sequence.

The CPU clears the IRQ latch during interrupt processing, so that another interrupt signal on the $\overline{IRQ}/V_{PP}$ pin can latch another interrupt request during the interrupt service routine. As soon as the I bit is cleared during the return from interrupt, the CPU can recognize the new interrupt request. **Figure 4** shows the $\overline{IRQ}/V_{PP}$ pin interrupt logic.

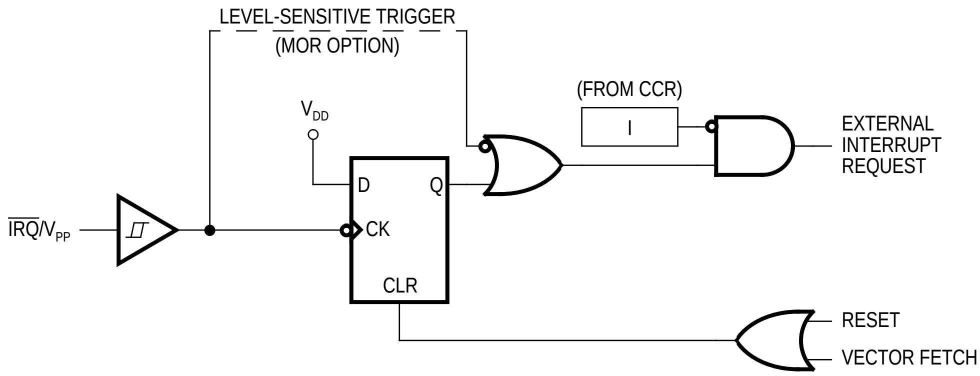


Figure 4. External Interrupt Logic

Setting the I bit in the condition code register disables external interrupts.

Interrupt triggering sensitivity of the $\overline{IRQ}/V_{PP}$ pin is a programmable option. The $\overline{IRQ}/V_{PP}$ pin can be negative-edge triggered or negative-edge- and low-level triggered. The level-sensitive triggering option allows multiple external interrupt sources to be wire-ORed to the $\overline{IRQ}/V_{PP}$ pin. An external interrupt request, shown in **Figure 5**, is latched as long as any source is holding the $\overline{IRQ}/V_{PP}$ pin low.

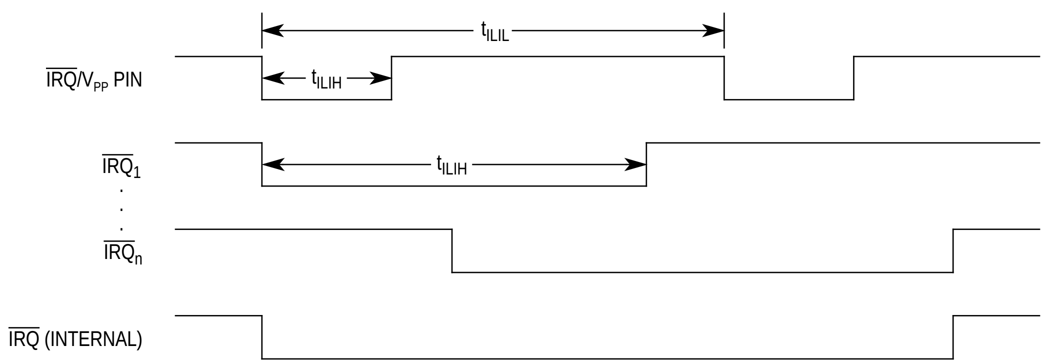


Figure 5. External Interrupt Timing

Table 2. External Interrupt Timing (V_{DD} = 5.0 Vdc)(1)

Characteristic	Symbol	Min	Max	Unit
Interrupt Pulse Width Low (Edge-Triggered)	t_{ILIH}	125	—	ns
Interrupt Pulse Period	t_{ILIL}	Note(2)	—	t_{CYC}

1. V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc, T_A = T_L to T_H
2. The minimum t_{ILIL} should not be less than the number of interrupt service routine cycles plus 19 t_{CYC}.

Table 3. External Interrupt Timing (V_{DD} = 3.3 Vdc)(1)

Characteristic	Symbol	Min	Max	Unit
Interrupt Pulse Width Low (Edge-Triggered)	t_{ILIH}	250	—	ns
Interrupt Pulse Period	t_{ILIL}	Note(2)	—	t_{CYC}

1. V_{DD} = 3.3 Vdc ±10%, V_{SS} = 0 Vdc, T_A = T_L to T_H
2. The minimum t_{ILIL} should not be less than the number of interrupt service routine cycles plus 19 t_{CYC}.

Timer Interrupts

The capture/compare timer can generate the following interrupts:

- Input capture interrupt
- Output compare interrupt
- Timer overflow interrupt

Setting the I bit in the condition code register disables timer interrupts.

Input Capture Interrupt

An input capture interrupt request occurs if the input capture flag, ICF, becomes set while the input capture interrupt enable bit, ICIE, is also set. ICF is in the timer status register, and ICIE is in the timer control register.

Output Compare Interrupt

An output compare interrupt request occurs if the output compare flag, OCF, becomes set while the output compare interrupt enable bit, OCIE, is also set. OCF is in the timer status register, and OCIE is in the timer control register.

Timer Overflow Interrupt

A timer overflow interrupt request occurs if the timer overflow flag, TOF, becomes set while the timer overflow interrupt enable bit, TOIE, is also set. TOF is in the timer status register, and TOIE is in the timer control register.

Interrupt Processing

The CPU takes the following actions to begin servicing an interrupt:

- Stores the CPU registers on the stack in the order shown in **Figure 6**
- Sets the I bit in the condition code register to prevent further interrupts
- Loads the program counter with the contents of the appropriate interrupt vector locations:
  - $1FFC and $1FFD (software interrupt vector)
  - $1FFA and $1FFB (external interrupt vector)
  - $1FF8 and $1FF9 (timer interrupt vector)

The return from interrupt (RTI) instruction causes the CPU to recover the CPU registers from the stack as shown in **Figure 6**.

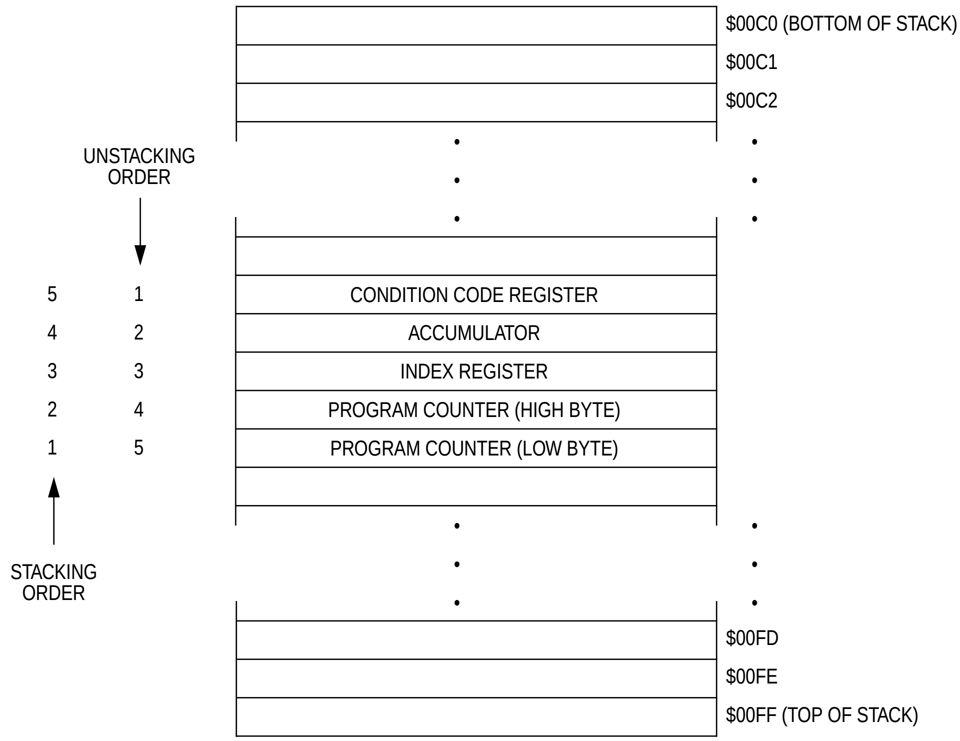


Figure 6. Interrupt Stacking Order

Table 4. Reset/Interrupt Vector Addresses

Function	Source	Local Mask	Global Mask	Priority (1 = Highest)	Vector Address
Reset	Power-On $\overline{\text{RESET}}$ Pin COP Watchdog(1)	None	None None None	1 1 1	\$1FFE–\$1FFF
Software Interrupt (SWI)	User Code	None	None	Same Priority as Instruction	\$1FFC–\$1FFD
External Interrupt	$\overline{\text{IRQ}}/V_{PP}$ Pin	None	I Bit	2	\$1FFA–\$1FFB
Timer Interrupts	ICF Bit OCF Bit TOF Bit	ICIE Bit OCIE Bit TOIE Bit	I Bit	3	\$1FF8–\$1FF9

1. The COP watchdog is programmable in the mask option register.

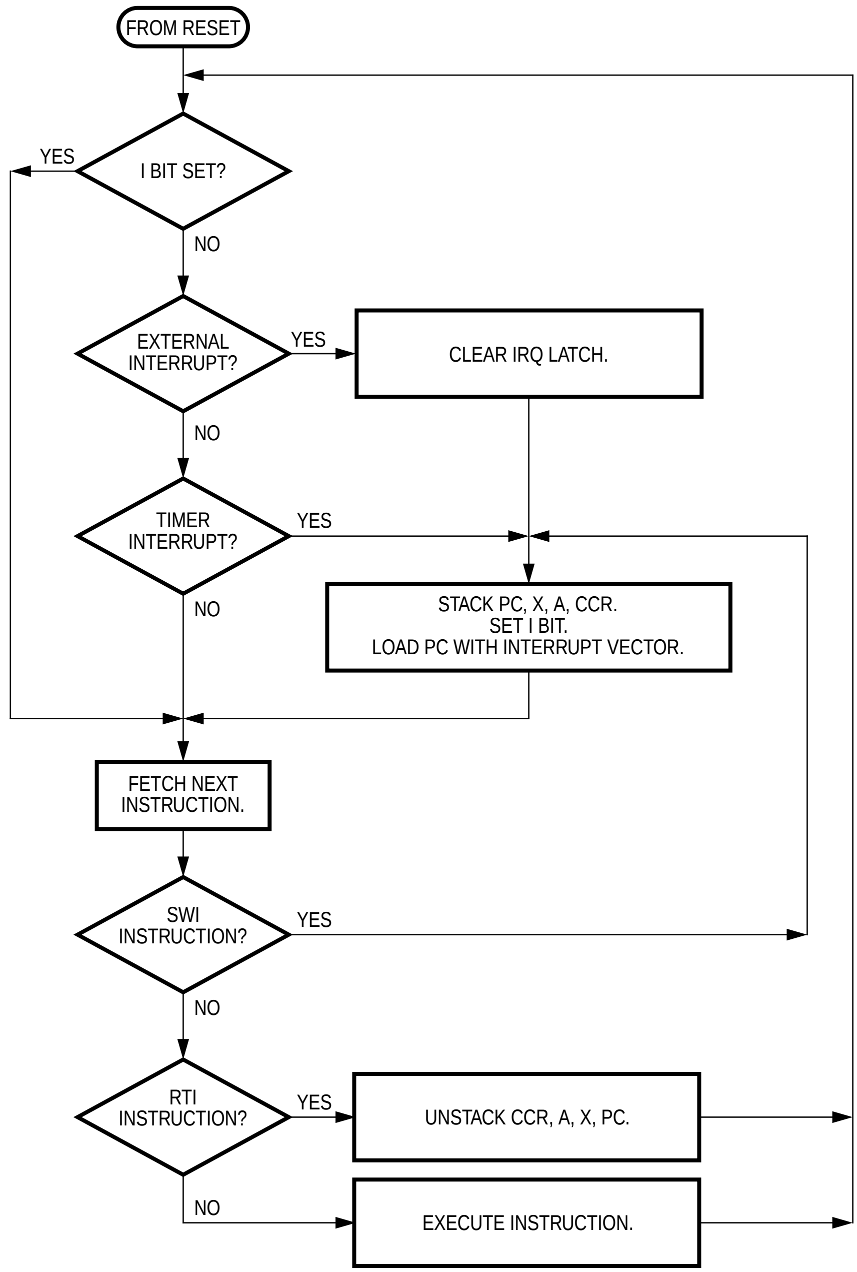


Figure 7. Interrupt Flowchart

10-mc68hc705p9

Low-Power Modes

Contents

Stop Mode

The STOP instruction puts the MCU in its lowest power-consumption mode and has the following effects on the MCU:

- Stops the internal oscillator, the CPU clock, and the internal clock, turning off the capture/compare timer, the COP watchdog, the SIOP, and the ADC
- Clears the I bit in the condition code register, enabling external interrupts
- Clears the ICIE, OCIE, and TOIE bits in the timer control register, disabling further timer interrupts

The STOP instruction does not affect any other registers or any I/O lines.

The following events bring the MCU out of stop mode:

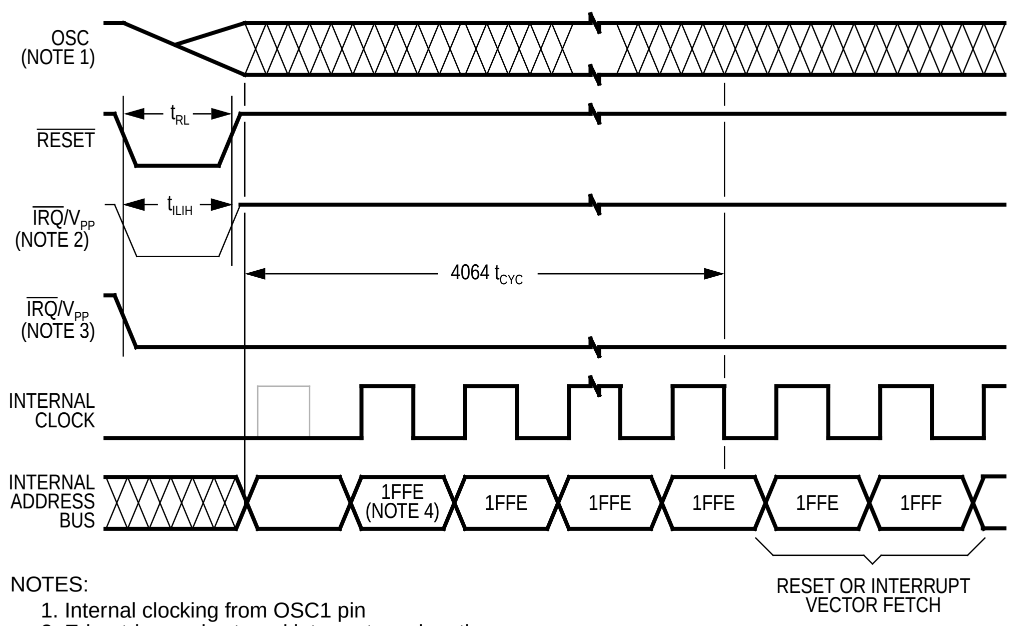
- An external interrupt signal on the $\overline{IRQ}/V_{PP}$ pin — A high-to-low transition on the $\overline{IRQ}/V_{PP}$ pin loads the program counter with the contents of locations \$1FFA and \$1FFB. The timer resumes counting from the last value before the STOP instruction.
- External reset — A logic zero on the $\overline{RESET}$ pin resets the MCU and loads the program counter with the contents of locations \$1FFE and \$1FFF. The timer begins counting from \$FFFC.

When the MCU exits stop mode, processing resumes after a stabilization delay of 4064 oscillator cycles.

An active edge on the PD7/TCAP pin during stop mode sets the ICF flag when an external interrupt brings the MCU out of stop mode. An external interrupt also latches the value in the timer registers into the input capture registers.

If a reset brings the MCU out of stop mode, then an active edge on the PD7/TCAP pin during stop mode has no effect on the ICF flag or the input capture registers.

See **Figure 1** for stop recovery timing information.



NOTES:
1. Internal clocking from OSC1 pin
2. Edge-triggered external interrupt mask option
3. Edge- and level-triggered external interrupt mask option
4. Reset vector shown as example

Figure 1. Stop Recovery Timing

Figure 2 shows the sequence of events caused by the STOP instruction.

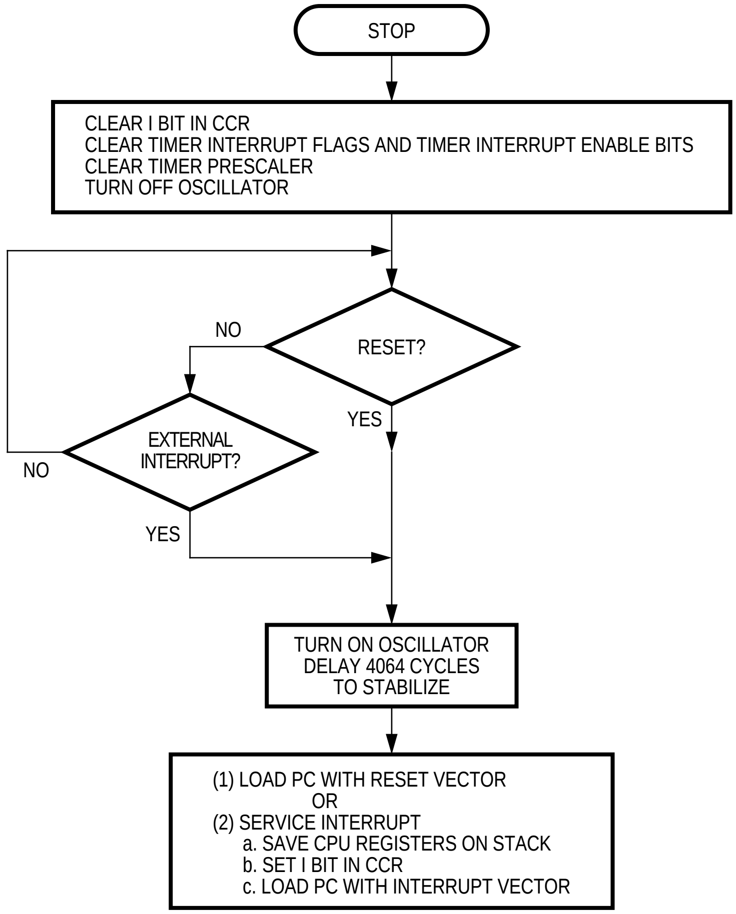


Figure 2. STOP Instruction Flowchart

Wait Mode

The WAIT instruction puts the MCU in an intermediate power-consumption mode and has the following effects on the MCU:

- Clears the I bit in the condition code register, enabling interrupts
- Stops the CPU clock, but allows the internal clock to drive the capture/compare timer, the COP watchdog, and the ADC

The WAIT instruction does not affect any other registers or any I/O lines.

The following conditions restart the CPU clock and bring the MCU out of wait mode:

- External interrupt — A high-to-low transition on the $\overline{IRQ}/V_{PP}$ pin loads the program counter with the contents of locations $1FFA and $1FFB.
- Timer interrupt — Input capture, output compare, and timer overflow interrupt requests load the program counter with the contents of locations $1FF8 and $1FF9.
- COP watchdog reset — A timeout of the COP watchdog resets the MCU and loads the program counter with the contents of locations $1FFE and $1FFF. Software can enable timer interrupts so that the MCU can periodically exit wait mode to reset the COP watchdog.
- External reset — A logic zero on the $\overline{RESET}$ pin resets the MCU and loads the program counter with the contents of locations $1FFE and $1FFF.

4-mc68hc705p9

Figure 3 shows the sequence of events caused by the WAIT instruction.

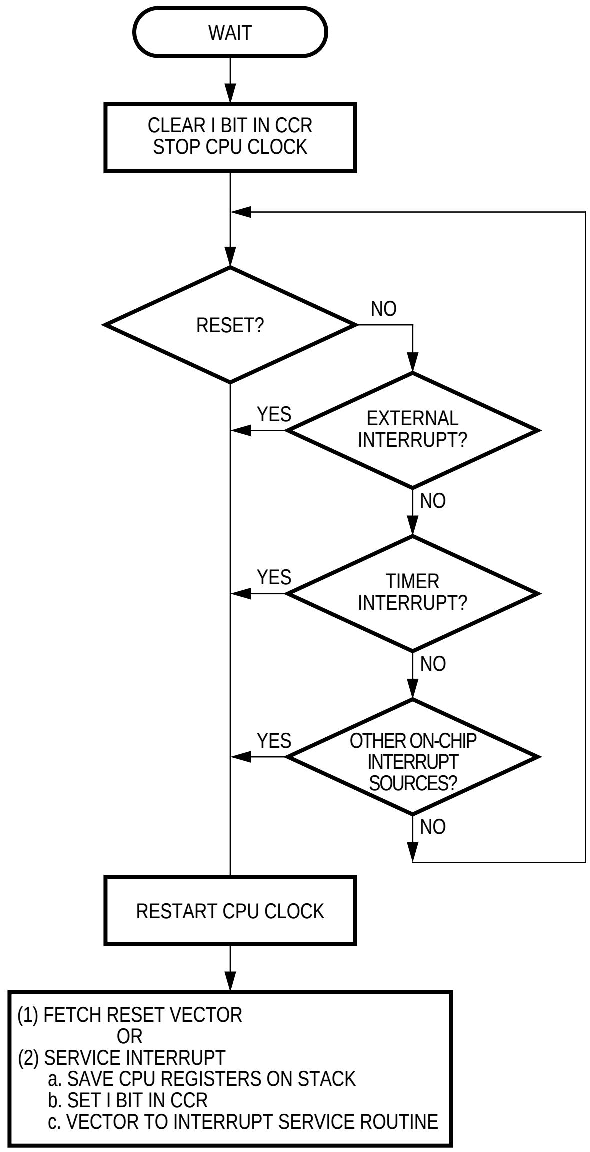


Figure 3. WAIT Instruction Flowchart

Figure 4 shows the effect of the STOP and WAIT instructions on the CPU clock and the timer clock.

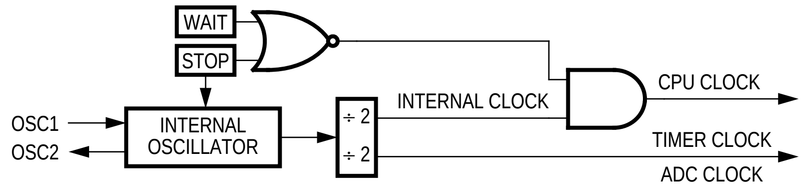


Figure 4. STOP/WAIT Clock Logic

Data-Retention Mode

In data-retention mode, the MCU retains RAM contents and CPU register contents at V_{DD} voltages as low as 2.0 Vdc. The data-retention feature allows the MCU to remain in a low-power consumption state during which it retains data, but the CPU cannot execute instructions.

To put the MCU in data-retention mode:

1. Drive the $\overline{\text{RESET}}$ pin to logic zero.
2. Lower the V_{DD} voltage. The $\overline{\text{RESET}}$ pin must remain low continuously during data-retention mode.

To take the MCU out of data-retention mode:

1. Return V_{DD} to normal operating voltage.
2. Return the $\overline{\text{RESET}}$ pin to logic one.

6-mc68hc705p9

Parallel I/O Ports

Contents

1-mc68hc705p9

Introduction

Twenty bidirectional pins and one input-only pin form four parallel input/output (I/O) ports. All the bidirectional port pins are programmable as inputs or outputs.

NOTE: *Connect any unused I/O pins to an appropriate logic level, either V_{DD} or V_{SS}. Although the I/O ports do not require termination for proper operation, termination reduces excess current consumption and the possibility of electrostatic damage.*

Addr.	Name:	R/W	Bit 7	6	5	4	3	2	1	Bit 0
$0000	Port A Data Register (PORTA)	Read:	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
		Write:								
		Reset:	Unaffected by reset							
$0001	Port B Data Register (PORTB)	Read:	PB7	PB6	PB5	0	0	0	0	0
		Write:								
		Reset:	Unaffected by reset							
$0002	Port C Data Register (PORTC)	Read:	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
		Write:								
		Reset:	Unaffected by reset							
$0003	Port D Data Register (PORTD)	Read:	PD7	0	PD5	1	0	0	0	0
		Write:								
		Reset:	Unaffected by reset							
$0004	Data Direction Register A (DDRA)	Read:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0005	Data Direction Register B (DDRB)	Read:	DDRB7	DDRB6	DDRB5	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 1. Parallel I/O Port Register Summary

2-mc68hc705p9

Addr.	Name:	R/W	Bit 7	6	5	4	3	2	1	Bit 0
$0006	Data Direction Register C (DDRC)	Read: Write:	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
		Reset:	0	0	0	0	0	0	0	0
$0007	Data Direction Register D (DDRD)	Read:	0	0	DDRD5	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 1. Parallel I/O Port Register Summary (Continued)

Port A

Port A is an 8-bit general-purpose I/O port.

Port A Data Register (PORTA)

The port A data register contains a latch for each of the eight port A pins.

$0000	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
Reset:	Unaffected by reset							

Figure 2. Port A Data Register (PORTA)

PA[7:0] — Port A Data Bits

These read/write bits are software programmable. Data direction of each port A pin is under the control of the corresponding bit in data direction register A. Reset has no effect on port A data.

3-mc68hc705p9

Data Direction Register A (DDRA)

Data direction register A determines whether each port A pin is an input or an output.

$0004	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
Reset:	0	0	0	0	0	0	0	0

Figure 3. Data Direction Register A (DDRA)

DDRA[7:0] — Data Direction Register A Bits

These read/write bits control port A data direction. Reset clears DDRA[7:0], configuring all eight port A pins as inputs.

1 = Corresponding port A pin configured as output

0 = Corresponding port A pin configured as input

NOTE: *Avoid glitches on port A pins by writing to the port A data register before changing data direction register A bits from 0 to 1.*

Figure 4 shows the I/O logic of port A.

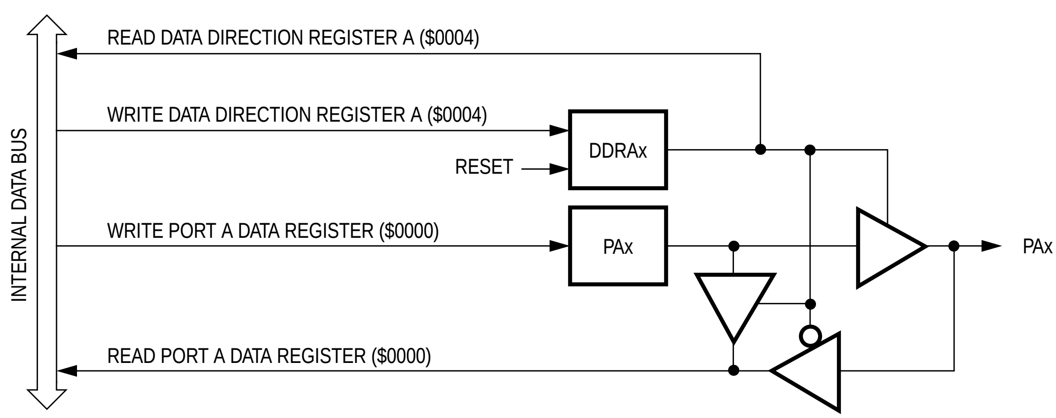


Figure 4. Port A I/O Circuit

4-mc68hc705p9

Writing a logic one to a DDRA bit enables the output buffer for the corresponding port A pin; a logic zero disables the output buffer.

When bit DDRAx is a logic one, reading address $0000 reads the PAx data latch. When bit DDRAx is a logic zero, reading address $0000 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. **Table 1** summarizes the operation of the port A pins.

Table 1. Port A Pin Operation

Data Direction Bit	I/O Pin Mode	Accesses to Data Bit	
		Read	Write
0	Input, Hi-Z(1)	Pin	Latch(2)
1	Output	Latch	Latch

1. Hi-Z = high impedance
2. Writing affects data register, but does not affect input.

Port B

Port B is a 3-bit I/O port that shares its pins with the serial I/O port (SIOP).

NOTE: *Do not use port B for general-purpose I/O while the SIOP is enabled.*

Port B Data Register (PORTB)

The port B data register contains a latch for each of the three port B pins.

$0001	Bit 7	6	5	4	3	2	1	Bit 0
Read:	PB7	PB6	PB5	0	0	0	0	0
Write:								
Reset:	Unaffected by reset							
Alternate Function:	SCK	SDI	SDO					

= Unimplemented

Figure 5. Port B Data Register (PORTB)

PB[7:5] — Port B Data Bits

These read/write bits are software programmable bits. Data direction of each port B pin is under the control of the corresponding bit in data direction register B. Reset has no effect on port B data.

NOTE: *Writing to data direction register B does not affect the data direction of port B pins that are being used by the SIOP. However, data direction register B always determines whether reading port B returns the states of the latches or the states of the pins.*

SCK — Serial Clock

When the SIOP is enabled, SCK is the SIOP clock output (in master mode) or the SIOP clock input (in slave mode).

6-mc68hc705p9

SDI — Serial Data Input

When the SIOP is enabled, SDI is the SIOP data input.

SDO — Serial Data Output

When the SIOP is enabled, SDO is the SIOP data output.

Data Direction Register B (DDRB)

Data direction register B determines whether each port B pin is an input or an output.

NOTE: *Enabling and then disabling the SIOP configures data direction register B for SIOP operation and can also change the port B data register. After disabling the SIOP, initialize data direction register B and the port B data register as your application requires.*

$0005	Bit 7	6	5	4	3	2	1	Bit 0
Read:	DDRB7	DDRB6	DDRB5	0	0	0	0	0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 6. Data Direction Register B (DDRB)

DDRB[7:5] — Data Direction Register B Bits

These read/write bits control port B data direction. Reset clears DDRB[7:5], configuring all three port B pins as inputs.

1 = Corresponding port B pin configured as output
0 = Corresponding port B pin configured as input

NOTE: *Avoid glitches on port B pins by writing to the port B data register before changing data direction register B bits from 0 to 1.*

Figure 7 shows the I/O logic of port B.

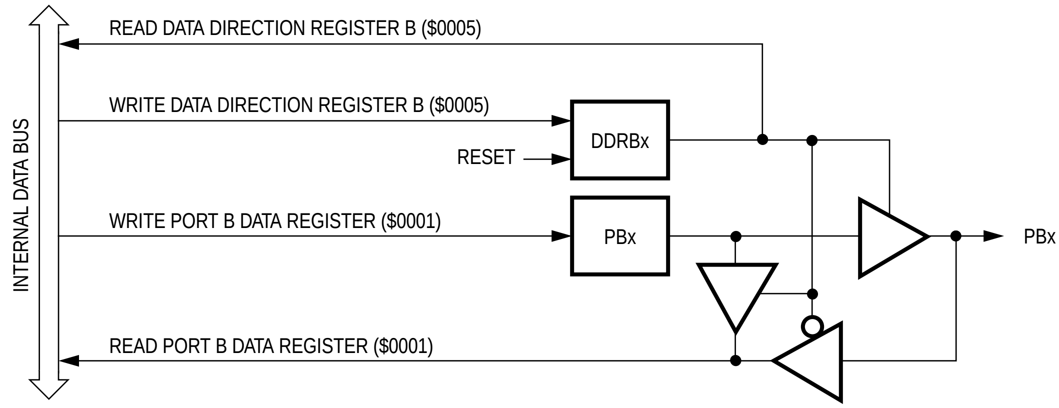


Figure 7. Port B I/O Logic

Writing a logic one to a DDRB bit enables the output buffer for the corresponding port B pin; a logic zero disables the output buffer.

When bit DDRBx is a logic one, reading address $0001 reads the PBx data latch. When bit DDRBx is a logic zero, reading address $0001 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. **Table 1** summarizes the operation of the port B pins.

Table 2. Port B Pin Operation

Data Direction Bit	I/O Pin Mode	Accesses to Data Bit	
		Read	Write
0	Input, Hi-Z(1)	Pin	Latch(2)
1	Output	Latch	Latch

1. Hi-Z = high impedance
2. Writing affects data register, but does not affect input.

8-mc68hc705p9

Port C

Port C is an 8-bit I/O port that shares five of its pins with the A/D converter (ADC). The five shared pins are available for general-purpose I/O functions when the ADC is disabled.

Port C Data Register (PORTC)

The port C data register contains a latch for each of the eight port C pins.

\$0002	Bit 7	6	5	4	3	2	1	Bit 0
Read: / Write:	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Reset:	Unaffected by reset							
Alternate Function:	V_{RH}	AN0	AN1	AN2	AN3			

Figure 8. Port C Data Register (PORTC)

PC[7:0] — Port C Data Bits

These read/write bits are software programmable. Data direction of each port C pin is under the control of the corresponding bit in data direction register C. Reset has no effect on port C data.

V_{RH} — Voltage Reference High Bit

When the ADC is turned on, the PC7/V_{RH} pin is the positive ADC reference voltage.

AN[3:0] — Analog Input Bits

When the ADC is turned on, the AN0–AN3 pins are software-selectable analog inputs. Unused analog inputs can be used as digital inputs, but pins PC3/AN3, PC4/AN2, PC5/AN1, and PC6/AN0 cannot be used as digital outputs while the ADC is on. Only pins PC0, PC1, and PC2 can be used as digital outputs when the ADC is on.

The port C data register reads normally while the ADC is on, except that the bit corresponding to the currently selected ADC input pin reads as logic zero.

Writing to bits PC7–PC3 while the ADC is on can produce unpredictable ADC results.

Data Direction Register C (DDRC)

Data direction register C determines whether each port C pin is an input or an output.

$0006	Bit 7	6	5	4	3	2	1	Bit 0
Read: / Write:	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
Reset:	0	0	0	0	0	0	0	0

Figure 9. Data Direction Register C (DDRC)

DDRC[7:0] — Data Direction Register C Bits

These read/write bits control port C data direction. Reset clears DDRC[7:0], configuring all port C pins as inputs.

1 = Corresponding port C pin configured as output

0 = Corresponding port C pin configured as input

NOTE: *Avoid glitches on port C pins by writing to the port C data register before changing data direction register C bits from 0 to 1.*

Writing to bits DDRC7–DDRC3 while the ADC is on can produce unpredictable ADC results.

Figure 10 shows the I/O logic of port C.

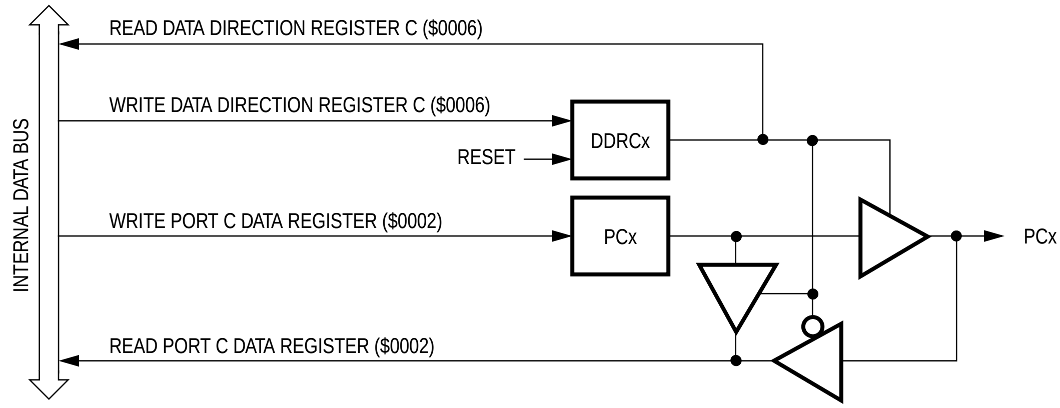


Figure 10. Port C I/O Logic

Writing a logic one to a DDRC bit enables the output buffer for the corresponding port C pin; a logic zero disables the output buffer.

When bit DDRCx is a logic one, reading address $0002 reads the PCx data latch. When bit DDRCx is a logic zero, reading address $0002 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. **Table 1** summarizes the operation of the port C pins.

Table 3. Port C Pin Operation

Data Direction Bit	I/O Pin Mode	Accesses to Data Bit	
		Read	**Write**
0	Input, Hi-Z[1]	Pin	Latch[2]
1	Output	Latch	Latch

1. Hi-Z = high impedance
2. Writing affects data register, but does not affect input.

Port D

Port D is a 2-bit port with one I/O pin and one input-only pin. Port D shares the input-only pin, PD7/TCAP, with the capture/compare timer. PD7/TCAP is the timer input capture pin. The PD7/TCAP pin can always be a general-purpose input, even if input capture interrupts are enabled.

Port D Data Register (PORTD)

The port D data register contains a latch for each of the two port D pins.

$0003	Bit 7	6	5	4	3	2	1	Bit 0
Read:	PD7	0	PD5	1	0	0	0	0
Write:								
Reset:	Unaffected by reset							
Alternate Function:	TCAP							

= Unimplemented

Figure 11. Port D Data Register (PORTD)

PD7 and PD5 — Port D Data Bits

These read/write bits are software programmable. Data direction of each port D pin is under the control of the corresponding bit in data direction register D. Reset has no effect on port D data.

TCAP — Timer Capture

TCAP is the input capture pin for the timer.

Data Direction Register D (DDRD)

Data direction register D determines whether each port D pin is an input or an output.

$0007	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	DDRD5	0	0	0	0	0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 12. Data Direction Register D (DDRD)

DDRD5 — Data Direction Register D Bit

This read/write bit controls the data direction of pin PD5. Reset clears DDRD5, configuring PD5 as an input.

1 = PD5 configured as output
0 = PD5 configured as input

NOTE: *Avoid glitches on port D pins by writing to the port D data register before changing data direction register D bits from 0 to 1.*

Figure 13 shows the I/O logic of port D.

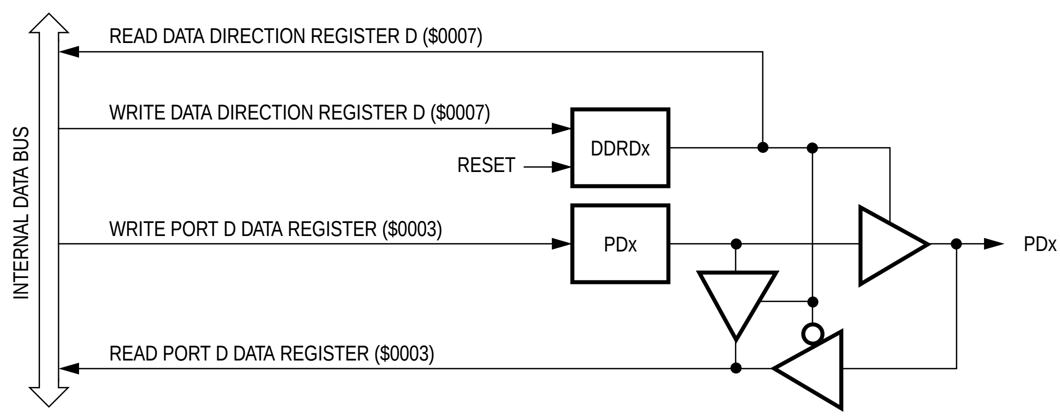


Figure 13. Port D I/O Logic

Writing a logic one to a DDRD bit enables the output buffer for the corresponding port D pin; a logic zero disables the output buffer.

When bit DDRDx is a logic one, reading address $0003 reads the PDx data latch. When bit DDRDx is a logic zero, reading address $0003 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. **Table 1** summarizes the operation of the port D pins.

Table 4. Port D Pin Operation

Data Direction Bit	I/O Pin Mode	Accesses to Data Bit	
		Read	Write
0	Input, Hi-Z[1]	Pin	Latch[2]
1	Output	Latch	Latch

1. Hi-Z = high impedance
2. Writing affects data register, but does not affect input.

Computer Operating Properly Watchdog COP

Contents

Features

- Protection from runaway software
- 65.5-ms timeout period (with 2-MHz bus frequency)
- Wait mode operation

1-cop0cop

Introduction

The purpose of the computer operating properly (COP) watchdog is to reset the MCU in case of software failure. Software that is operating properly periodically services the COP watchdog and prevents the reset from occurring. The COP watchdog function is programmable in the mask option register.

Operation

COP Watchdog Timeout

The COP watchdog is a 16-bit counter that generates a reset if allowed to time out. Periodically clearing the counter starts a new timeout period and prevents the COP from resetting the MCU. A COP watchdog timeout indicates that the software is not executing instructions in the correct sequence.

NOTE: *The internal clock drives the COP watchdog. Therefore, the COP watchdog cannot generate a reset for errors that cause the internal clock to stop.*

The COP watchdog also depends on a power supply voltage at or above a minimum specification and is not guaranteed to protect against brownout. For information about brownout protection, see the ***Resets and Interrupts*** *section.*

COP Watchdog Timeout Period

Use the following formula to calculate the COP timeout period:

$$\text{COP Timeout Period} = \frac{131{,}072\ \text{cycles}}{f_{BUS}}$$

where

$$f_{BUS} = \frac{\text{crystal frequency}}{2}$$

2-cop0cop

Clearing the COP Watchdog

To clear the COP watchdog and prevent a COP reset, write a logic zero to bit 0 (COPC) of the COP register at location $1FF0.

If the main program executes within the COP timeout period, the clearing routine needs to be executed only once. If the main program takes longer than the COP timeout period, the clearing routine must be executed more than once.

NOTE: *Place the clearing routine in the main program and not in an interrupt routine. Clearing the COP watchdog in an interrupt routine might prevent COP watchdog timeouts even though the main program is not operating properly.*

Interrupts

The COP watchdog does not generate interrupts.

COP Register

The COP register is a write-only register that returns the contents of EPROM location $1FF0 when read.

$1FF0	Bit 7	6	5	4	3	2	1	Bit 0
Read:	D7	D6	D5	D4	D3	D2	D1	D0
Write:								COPC
Reset:	U	U	U	U	U	U	U	0

= Unimplemented U = Unaffected

Figure 1. COP Register (COPR)

COPC — COP Clear

COPC is a write-only bit. Periodically writing a logic zero to COPC prevents the COP watchdog from resetting the MCU. Reset clears the COPC bit.

3-cop0cop

Low-Power Modes

The STOP and WAIT instructions put the MCU in low-power consumption standby modes.

Stop Mode

The STOP instruction clears the COP watchdog counter. Upon exit from stop mode by external reset:

- The counter begins counting from $0000.
- The counter is cleared again after the 4064-cycle oscillator stabilization delay.

Upon exit from stop mode by external interrupt:

- The counter begins counting from $0000.
- The counter is *not* cleared again after the oscillator stabilization delay and has a count of 4064 when the program resumes.

Wait Mode

The COP watchdog continues to operate normally after a WAIT instruction. Software should periodically take the MCU out of wait mode and write to the COPC bit to prevent a COP watchdog timeout.

Timer

Contents

1-tim1ic1oc_a

Features

- Programmable Polarity of Input Capture Edge
- Programmable Polarity of Output Compare Signal
- Alternate Counter Registers
- 16-Bit Counter
- Interrupt-Driven Operation with Three Maskable Interrupt Flags:
  - Input Capture
  - Output Compare
  - Timer Overflow

Introduction

The timer provides a timing reference for MCU operations. The input capture and output compare functions provide a means to latch the times at which external events occur, to measure input waveforms, and to generate output waveforms and timing delays. **Figure 1** shows the structure of the timer module.

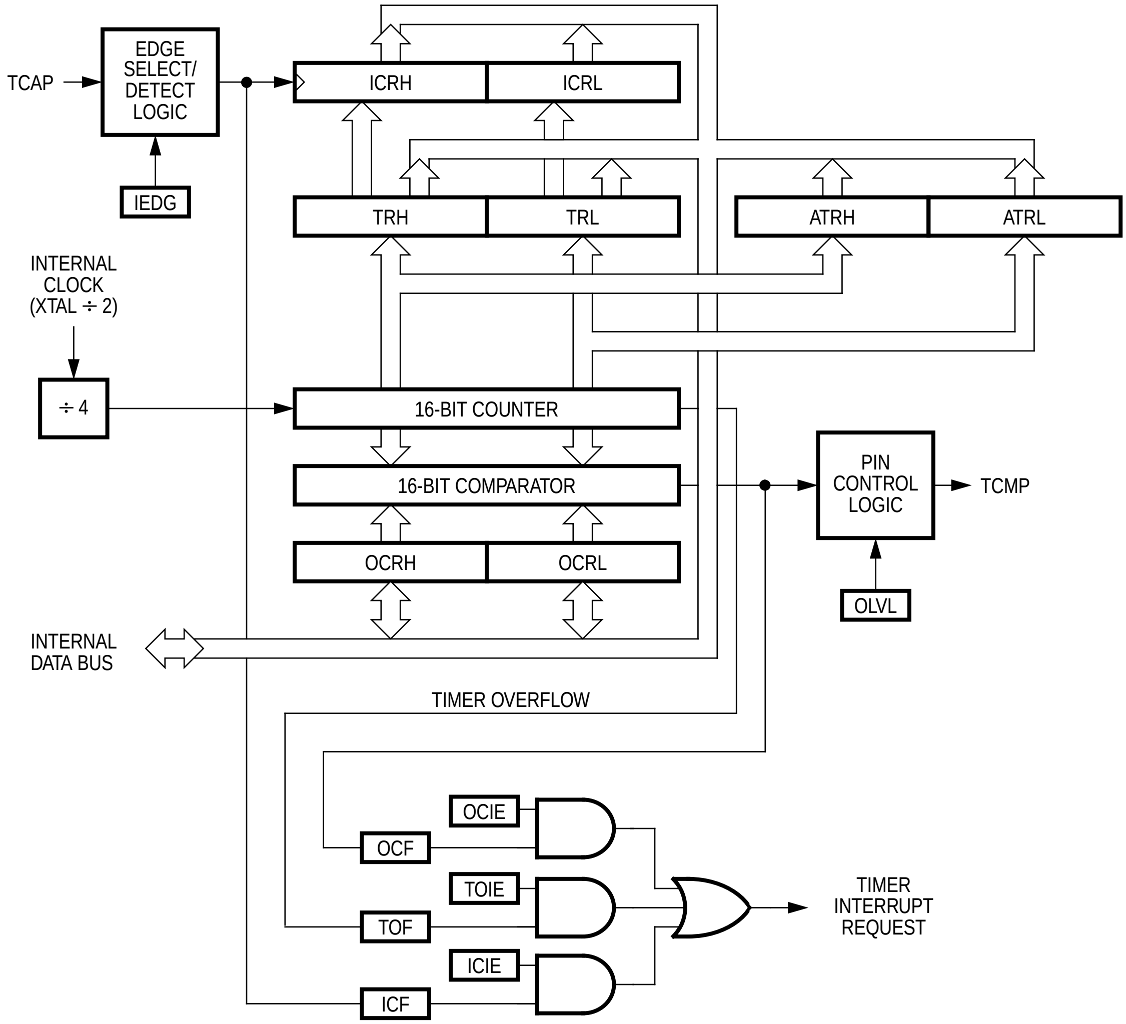


Figure 1. Timer Block Diagram

3-tim1ic1oc_a

Addr.	Name	R/W	Bit 7	6	5	4	3	2	1	Bit 0
$0012	Timer Control Register (TCR)	Read: Write:	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL
		Reset:	0	0	0	0	0	0	U	0
$0013	Timer Status Register (TSR)	Read:	ICF	OCF	TOF	0	0	0	0	0
		Write:								
		Reset:	U	U	U	0	0	0	0	0
$0014	Input Capture Register High (ICRH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	Unaffected by reset							
$0015	Input Capture Register Low (ICRL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Unaffected by reset							
$0016	Output Compare Register High (OCRH)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
		Reset:	Unaffected by reset							
$0017	Output Compare Register Low (OCRL)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
		Reset:	Unaffected by reset							
$0018	Timer Register High (TRH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	Reset initializes TRH to $FF							
$0019	Timer Register Low (TRL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Reset initializes TRL to $FC							
$001A	Alternate Timer Register High (ATRH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	Reset initializes ATRH to $FF							
$001B	Alternate Timer Register Low (ATRL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Reset initializes ATRL to $FC							

= Unimplemented   U = Unaffected

Figure 2. Timer I/O Register Summary

4-tim1ic1oc_a

Operation

The timing reference for the input capture and output compare functions is a 16-bit free-running counter. The counter is preceded by a divide-by-four prescaler and rolls over every 2^{18} cycles. Timer resolution with a 4-MHz crystal is 2 µs. Software can read the value in the counter at any time without affecting the counter sequence.

Because of the 16-bit timer architecture, the I/O registers for the input capture and output compare functions are pairs of 8-bit registers.

Pin Functions

The timer uses two pins.

PD7/TCAP

PD7/TCAP is the input capture pin. When an active edge occurs on PD7/TCAP, the timer transfers the current counter value to the input capture registers. PD7/TCAP is also an I/O port pin.

TCMP

TCMP is the output-only output compare pin. When the counter value matches the value written in the output compare registers, the timer transfers the output level bit, OLVL, to the TCMP pin.

Input Capture

The input capture function is a means to record the time at which an external event occurs. When the input capture circuitry detects an active edge on the PD7/TCAP pin, it latches the contents of the timer registers into the input capture registers. The polarity of the active edge is programmable.

Latching values into the input capture registers at successive edges of the same polarity measures the period of the input signal on the PD7/TCAP pin. Latching the counter values at successive edges of opposite polarity measures the pulse width of the signal. **Figure 3** shows the logic of the input capture function.

5-tim1ic1oc_a

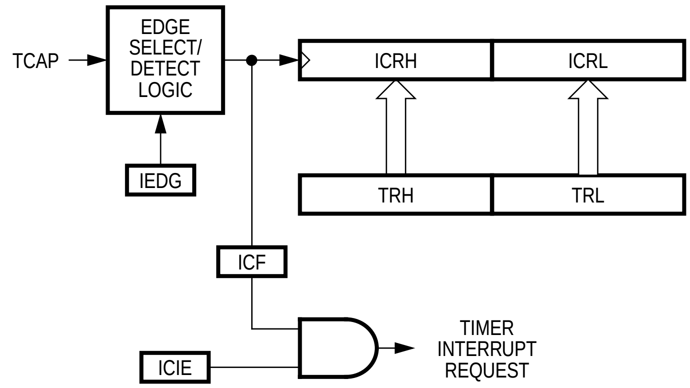


Figure 3. Input Capture Operation

Output Compare

The output compare function is a means of generating an output signal when the 16-bit counter reaches a selected value. Software writes the selected value into the output compare registers. On every fourth internal clock cycle the output compare circuitry compares the value of the counter to the value written in the output compare registers. When a match occurs, the timer transfers the programmable output level bit (OLVL) from the timer control register to the TCMP pin.

Software can use the output compare register to measure time periods, to generate timing delays, or to generate a pulse of specific duration or a pulse train of specific frequency and duty cycle on the TCMP pin. Figure 4 shows the logic of the output compare function.

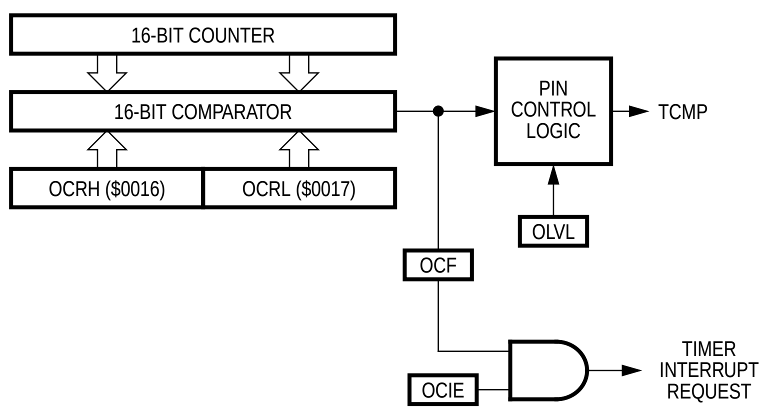


Figure 4. Output Compare Operation

6-tim1ic1oc_a

Timing

Table 1. Timer Characteristics (V_{DD} = 5.0 Vdc)[1]

Characteristic	Symbol	Min	Max	Unit
Timer Resolution[2]	t_{RESL}	4.0	—	t_{CYC}
Input Capture Pulse Width	t_H, t_L	125	—	ns
Input Capture Pulse Period	t_{TLTL}	Note (3)	—	t_{CYC}

1. V_{DD} = 5.0 Vdc ± 10%, $T_A = T_L$ to T_H unless otherwise noted.
2. A 2-bit prescaler in the timer is the limiting factor as it counts 4 t_{CYC}.
3. The minimum t_{TLTL} should not be less than the number of interrupt service routine cycles plus 19 t_{CYC}.

Table 2. Timer Characteristics (V_{DD} = 3.3 Vdc)[1]

Characteristic	Symbol	Min	Max	Unit
Timer Resolution[2]	t_{RESL}	4.0	—	t_{CYC}
Input Capture Pulse Width	t_H, t_L	250	—	ns
Input Capture Pulse Period	t_{TLTL}	Note (3)	—	t_{CYC}

1. V_{DD} = 3.3 Vdc ± 10%, $T_A = T_L$ to T_H unless otherwise noted.
2. A 2-bit prescaler in the timer is the limiting factor as it counts 4 t_{CYC}.
3. The minimum t_{TLTL} should not be less than the number of interrupt service routine cycles plus 19 t_{CYC}.

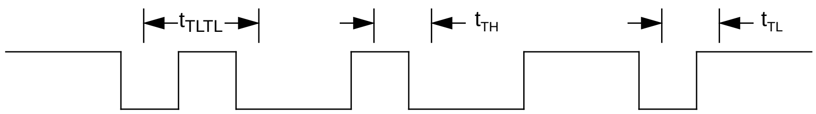


Figure 5. Input Capture Characteristics

7-tim1ic1oc_a

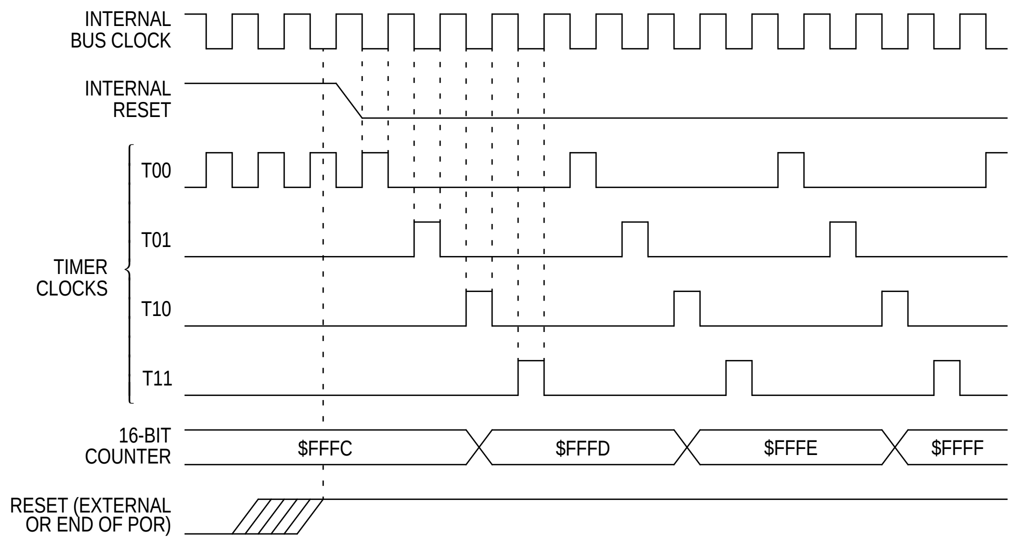
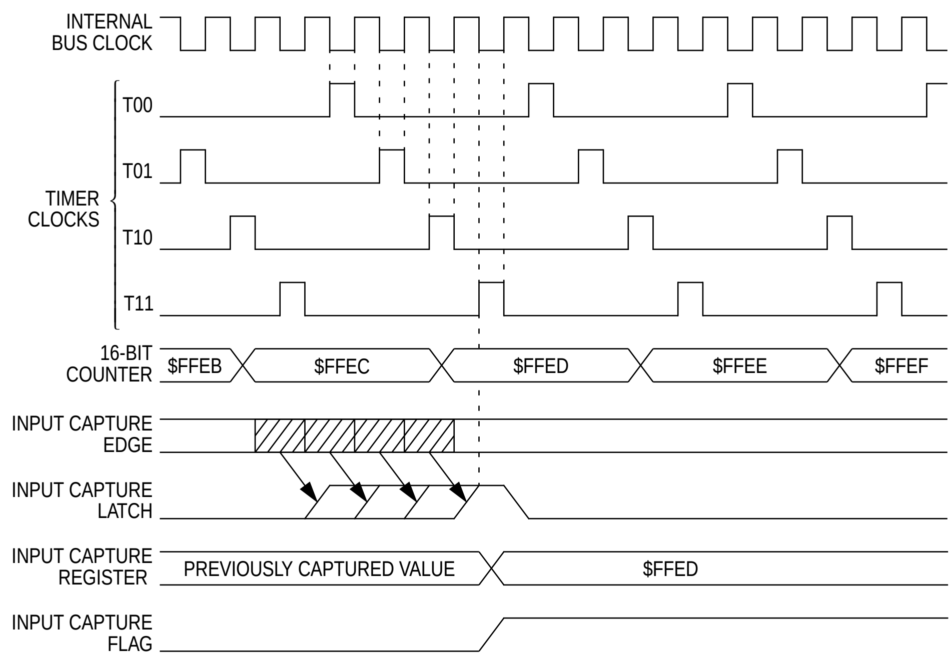


Figure 6. Timer Reset Timing

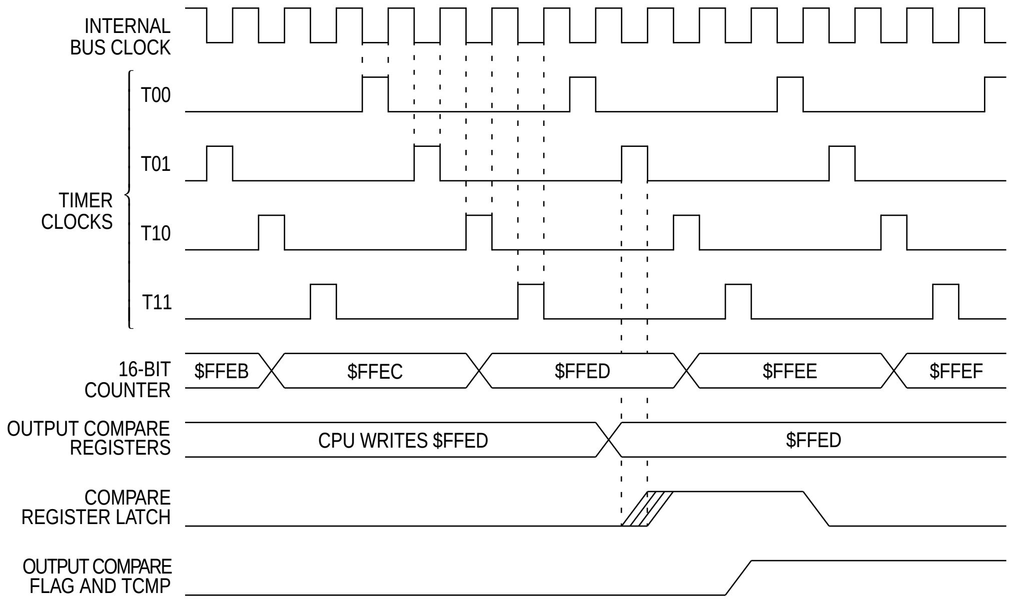


NOTE:

If the input capture edge occurs in the shaded area between T10 states, then the input capture flag becomes set during the next T11 state.

Figure 7. Input Capture Timing

8-tim1ic1oc_a



NOTES:

1. A write to the output compare registers may occur at any time, but a compare only occurs at timer state T01. Therefore, the compare may follow the write by up to four cycles.
2. The output compare flag is set at the timer state T11 that follows the comparison latch.

Figure 8. Output Compare Timing

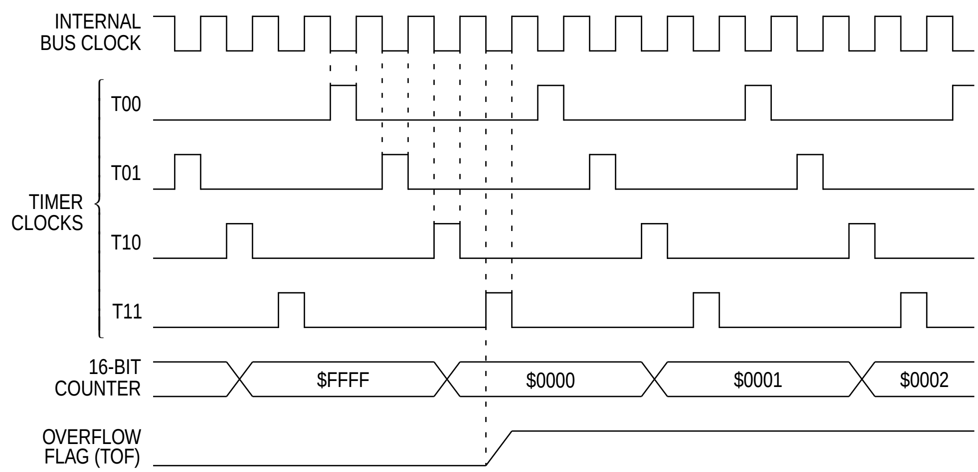


Figure 9. Timer Overflow Timing

Interrupts

The following timer sources can generate interrupts:

- Input capture flag (ICF) — The ICF bit is set when an edge of the selected polarity occurs on the input capture pin. The input capture interrupt enable bit, ICIE, enables ICF interrupt requests.
- Output compare flag (OCF) — The OCF bit is set when the counter value matches the value written in the output compare registers. The output compare interrupt enable bit, OCIE, enables OCF interrupt requests.
- Timer overflow flag (TOF) — The TOF bit is set when the counter value rolls over from $FFFF to $0000. The timer overflow enable bit (TOIE) enables timer overflow interrupt requests.

Table 3 summarizes the timer interrupt sources.

Table 3. Timer Interrupt Sources

Source	Local Mask	Global Mask	Priority (1 = Highest)
ICF Bit OCF Bit TOF Bit	ICIE Bit OCIE Bit TOIE Bit	I Bit	3

I/O Registers

The following registers control and monitor the operation of the timer:

- Timer control register (TCR)
- Timer status register (TSR)
- Timer registers (TRH and TRL)
- Alternate timer registers (ATRH and ATRL)
- Input capture registers (ICRH and ICRL)
- Output compare registers (OCRH and OCRL)

10-tim1ic1oc_a

Timer Control Register

The timer control register (TCR) performs the following functions:

- Enables input capture interrupts
- Enables output compare interrupts
- Enables timer overflow interrupts
- Controls the active edge polarity of the TCAP signal
- Controls the active level of the TCMP output

$0012	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL
Reset:	0	0	0	0	0	0	U	0

U = Unaffected

Figure 10. Timer Control Register (TCR)

ICIE — Input Capture Interrupt Enable

This read/write bit enables interrupts caused by an active signal on the PD7/TCAP pin. Reset clears the ICIE bit.
1 = Input capture interrupts enabled
0 = Input capture interrupts disabled

OCIE — Output Compare Interrupt Enable

This read/write bit enables interrupts caused by an active signal on the TCMP pin. Reset clears the OCIE bit.
1 = Output compare interrupts enabled
0 = Output compare interrupts disabled

TOIE — Timer Overflow Interrupt Enable

This read/write bit enables interrupts caused by a timer overflow. Reset clears the TOIE bit.
1 = Timer overflow interrupts enabled
0 = Timer overflow interrupts disabled

Bits 4–2 — Unused

These are read/write bits that always read as logic zeros.

IEDG — Input Edge

The state of this read/write bit determines whether a positive or negative transition on the PD7/TCAP pin triggers a transfer of the contents of the timer registers to the input capture registers. Reset has no effect on the IEDG bit.

1 = Positive edge (low-to-high transition) triggers input capture
0 = Negative edge (high-to-low transition) triggers input capture

OLVL — Output Level

The state of this read/write bit determines whether a logic one or a logic zero appears on the TCMP pin when a successful output compare occurs. Reset clears the OLVL bit.

1 = TCMP goes high on output compare
0 = TCMP goes low on output compare

Timer Status Register

The timer status register (TSR) contains flags for the following events:

- An active signal on the PD7/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 rollover from $FFFF to $0000

$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

(shaded) = Unimplemented U = Unaffected

Figure 11. Timer Status Register (TSR)

12-tim1ic1oc_a

ICF — Input Capture Flag

The ICF bit is automatically set when an edge of the selected polarity occurs on the PD7/TCAP pin. Clear the ICF bit by reading the timer status register with ICF set, and then reading the low byte of the input capture registers. Reset has no effect on ICF.

1 = Input capture
0 = No input capture

OCF — Output Compare Flag

The OCF bit is automatically set 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 of the output compare registers. Reset has no effect on OCF.

1 = Output compare
0 = No output compare

TOF — Timer Overflow Flag

The TOF bit is automatically set 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 of the timer registers. Reset has no effect on TOF.

1 = Timer overflow
0 = No timer overflow

Timer Registers

The read-only timer registers (TRH and TRL) contain the current high and low bytes of the 16-bit counter. Reading TRH before reading TRL causes TRL to be latched until TRL is read. Reading TRL after reading the timer status register clears the timer overflow flag (TOF). Writing to the timer registers has no effect.

$0018	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 15	14	13	12	11	10	9	Bit 8
Write:								
Reset:	Reset initializes TRH to $FF							

$0019	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 7	6	5	4	3	2	1	Bit 0
Write:								
Reset:	Reset initializes TRL to $FC							

= Unimplemented

Figure 12. Timer Registers (TRH and TRL)

Reading TRH returns the current value of the high byte of the counter and causes the low byte to be latched into a buffer. The buffer value remains fixed even if the high byte is read more than once. Reading TRL reads the transparent low byte buffer and completes the read sequence of the timer registers.

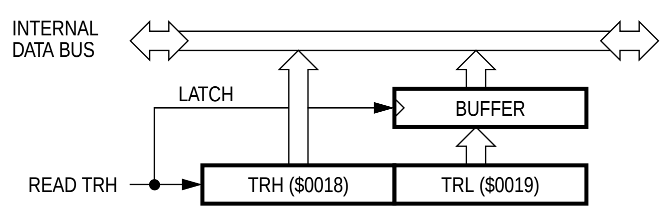


Figure 13. Timer Register Reads

NOTE: *To prevent interrupts from occurring between readings of TRH and TRL, set the interrupt mask (I bit) in the condition code register before reading TRH, and clear the mask after reading TRL.*

14-tim1ic1oc_a

Alternate Timer Registers

The read-only alternate timer registers (ATRH and ATRL) contain the current high and low bytes of the 16-bit counter. Reading ATRH before reading ATRL causes ATRL to be latched until ATRL is read. Reading does not affect the timer overflow flag (TOF). Writing to the alternate timer registers has no effect.

$001A	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 15	14	13	12	11	10	9	Bit 8
Write:								
Reset:	Reset initializes ATRH to $FF							
$001B	**Bit 7**	**6**	**5**	**4**	**3**	**2**	**1**	**Bit 0**
Read:	Bit 7	6	5	4	3	2	1	Bit 0
Write:								
Reset:	Reset initializes ATRL to $FC							

(shaded) = Unimplemented

Figure 14. Alternate Timer Registers (ATRH and ATRL)

Reading ATRH returns the current value of the high byte of the counter and causes the low byte to be latched into a buffer.

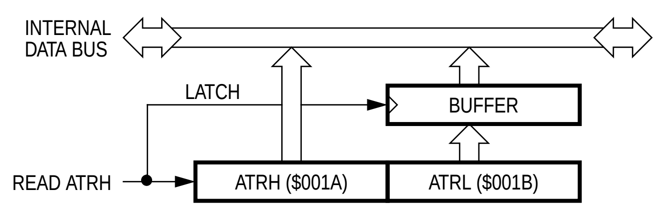


Figure 15. Alternate Timer Register Reads

NOTE: *To prevent interrupts between readings of ATRH and ATRL, set the interrupt mask (I bit) in the condition code register before reading ATRH, and clear the mask after reading ATRL.*

15-tim1ic1oc_a

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 read-only input capture registers (ICRH and ICRL). Reading ICRH before reading ICRL inhibits further captures until ICRL is read. Reading ICRL after reading the timer status register clears the input capture flag (ICF). Writing to the input capture registers has no effect.

$0014	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 15	14	13	12	11	10	9	Bit 8
Write:								
Reset:	Unaffected by reset							

$0015	7	6	5	4	3	2	1	0
Read:	Bit 7	6	5	4	3	2	1	Bit 0
Write:								
Reset:	Unaffected by reset							

= Unimplemented

Figure 16. Input Capture Registers (ICRH and ICRL)

NOTE: *To prevent interrupts between readings of ICRH and ICRL, set the interrupt mask (I bit) in the condition code register before reading ICRH, and clear the mask after reading ICRL.*

16-tim1ic1oc_a

Output Compare Registers

When the value of the 16-bit counter matches the value in the read/write output compare registers (OCRH and OCRL), 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 reading the timer status register clears the output compare flag (OCF).

$0016	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
Reset:	Unaffected by reset							

$0017	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
Reset:	Unaffected by reset							

Figure 17. 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 the following 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 the 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.

Low-Power Modes

The STOP and WAIT instructions put the MCU in low-power consumption standby modes.

Stop Mode

The STOP instruction suspends the timer counter. Upon exit from stop mode by external reset:

- The timer counter resumes counting from $FFFC.
- An input capture edge during stop mode does not affect the ICF bit or the input capture registers.

Upon exit from stop mode by external interrupt:

- The counter resumes counting from the suspended value.
- An input capture edge during stop mode sets the ICF bit and transfers the suspended timer counter value to the input capture registers.

Wait Mode

The timer remains active after a WAIT instruction. Any enabled timer interrupt request can bring the MCU out of wait mode.

Serial Input/Output Port SIOP

Contents

1-siop_a

Features

- Master or Slave Operation
- Programmable MSB-First or LSB-First Operation
- Interrupt-Driven Operation with Transfer Complete Flag
- Data Collision Flag
- Master Mode Frequency = Bus Frequency ÷ 4
- Maximum Slave Mode Frequency = Bus Frequency ÷ 4
- No Minimum Slave Mode Frequency

Introduction

The serial input/output port (SIOP) is a 3-wire master/slave communication port with serial clock, data input, and data output connections. The SIOP enables high-speed synchronous serial data transfer between the MCU and peripheral devices. Shift registers used with the SIOP can increase the number of parallel I/O pins controlled by the MCU. More powerful peripherals such as analog-to-digital converters and real-time clocks are also compatible with the SIOP. **Figure 1** shows the structure of the SIOP module.

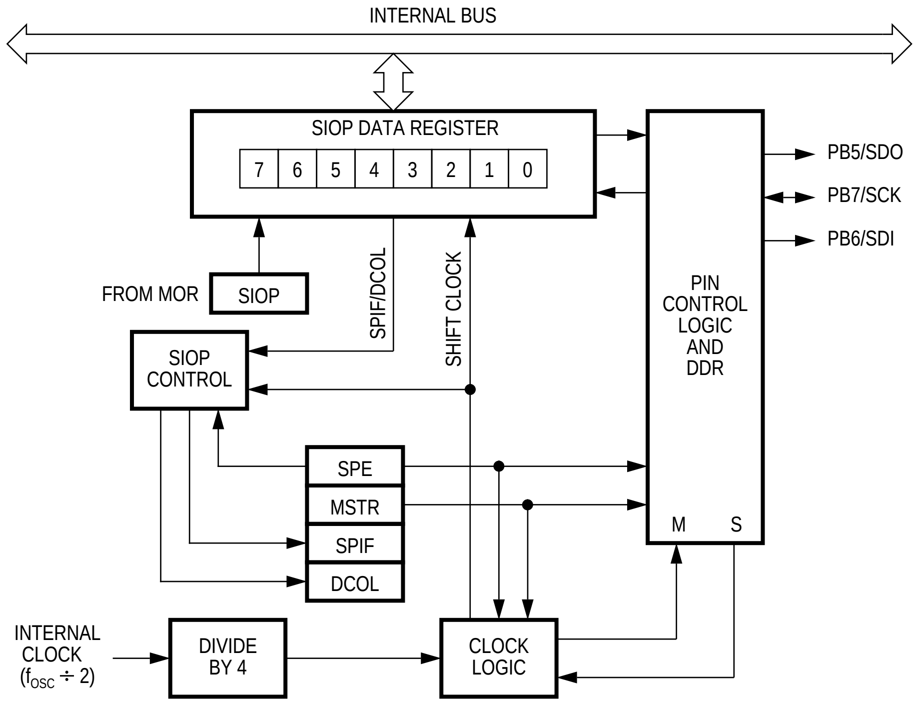


Figure 1. SIOP Block Diagram

Addr.	Name		Bit 7	6	5	4	3	2	1	Bit 0
$000A	SIOP Control Register (SCR)	Read: / Write:	0	SPE	0	MSTR	0	0	0	0
		Reset:	0	0	0	0	0	0	0	0
$000B	SIOP Status Register (SSR)	Read:	SPIF	DCOL	0	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000C	SIOP Data Register (SDR)	Read: / Write:	Bit 7	6	5	4	3	2	1	Bit 0
		Reset:	Unaffected by reset							

= Unimplemented

Figure 2. SIOP I/O Register Summary

3-siop_a

Operation

The master MCU initiates and controls the transfer of data to and from one or more slave peripheral devices. In master mode, a transmission is initiated by writing to the SIOP data register (SDR). Data written to the SDR is parallel-loaded and shifted out serially to the slave device(s).

Many simple slave devices are designed to only receive data from a master or to only supply data to a master. For example, when a serial-to-parallel shift register is used as an 8-bit port, the master MCU initiates transfers of 8-bit data values to the shift register. Since the serial-to-parallel shift register does not send any data to the master, the MCU ignores whatever it receives as a result of the transmission.

The SIOP is simpler than the serial peripheral interface (SPI) on some other Motorola MCUs in that:

- The polarity of the serial clock is fixed.
- There is no slave select pin.
- The direction of serial data does not automatically switch as on the SPI because the SIOP is not intended for use in multimaster systems. Most applications use one MCU as the master to initiate and control data transfer between one or more slave peripheral devices.

A programmable option allows the SIOP to transfer data MSB first or LSB first.

Pin Functions

The SIOP uses three pins and shares them with port B:

- PB7/SCK
- PB6/SDI
- PB5/SDO

NOTE: *Do not use the PB7/SCK, PB6/SDI, or PB5/SDO pins for general-purpose I/O while the SIOP is enabled.*

When bit 6 (SPE) of the SIOP control register (SCR) is set, the SIOP is enabled and the PB7/SCK, PB5/SDO, and PB6/SDI pins are dedicated to SIOP functions. Clearing SPE disables the SIOP and the SIOP pins become standard I/O port pins.

NOTE: *Enabling and then disabling the SIOP configures the data direction register bits associated with the SIOP pins for SIOP operation and can also change the associated port data register. After disabling the SIOP, initialize the data direction register and the port data register as the application requires.*

PB7/SCK

The PB7/SCK pin synchronizes the movement of data into and out of the MCU through the PB6/SDI and PB5/SDO pins. In master mode, the PB7/SCK pin is an output. The serial clock frequency in master mode is one-fourth the internal clock frequency.

In slave mode, the PB7/SCK pin is an input. The maximum serial clock frequency in slave mode is one-fourth the internal clock rate. Slave mode has no minimum serial clock frequency.

Figure 3 shows the timing relationships among the serial clock, data input, and data output. The state of the serial clock between transmissions is a logic one. The first falling edge on the PB7/SCK pin signals the beginning of a transmission, and data appears at the PB5/SDO pin. Data is captured at the PB6/SDI pin on the rising edge of the serial clock, and the transmission ends on the eighth rising edge of the serial clock.

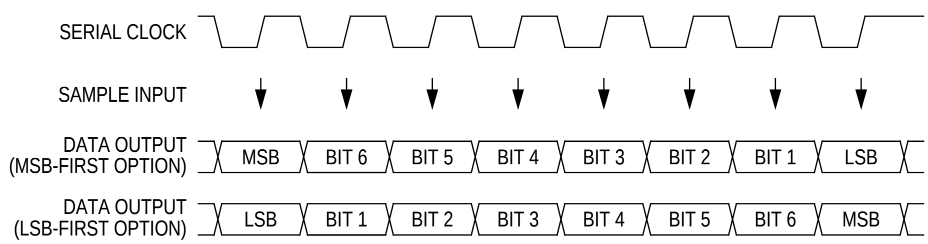


Figure 3. SIOP Data/Clock Timing

The first falling edge on PB7/SCK begins a transmission. At this time the first bit of received data is accepted at the PB6/SDI pin and the first bit of transmitted data is presented at the PB5/SDO pin.

PB5/SDO

The PB5/SDO pin is the SIOP data output. Between transfers, the state of the PB5/SDO pin reflects the value of the last bit shifted out on the previous transmission, if there was one. To preset the beginning state, write to the corresponding port data bit before enabling the SIOP. On the first falling edge on the PB7/SCK pin, the first data bit to be shifted out appears at the PB5/SDO pin.

After SPE is set, the PB5/SDO output driver can be disabled by writing a zero to the corresponding data direction register bit of the port, thereby configuring PB5/SDO as a high-impedance input.

PB6/SDI

The PB6/SDI pin is the SIOP data input. Valid SDI data must be present for an SDI setup time, t_S, before the rising edge of the serial clock and must remain valid for an SDI hold time, t_H, after the rising edge of the serial clock. (See **Table 1** and **Table 2**.)

Data Movement

Connecting the SIOP data register of a master MCU with the SIOP of a slave MCU forms a 16-bit circular shift register. During an SIOP transfer, the master shifts out the contents of its SIOP data register on its PB5/SDO pin. At the same time, the slave MCU shifts out the contents of its SIOP data register on its PB5/SDO pin. **Figure 4** shows how the master and slave exchange the contents of their data registers.

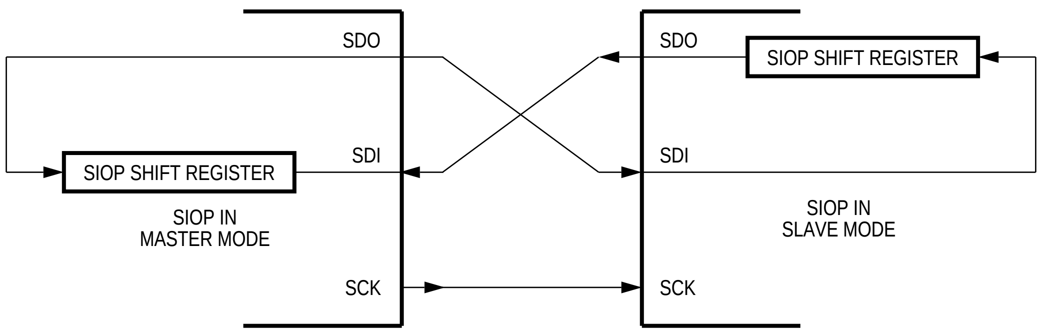


Figure 4. Master/Slave SIOP Shift Register Operation

7-siop_a

Timing

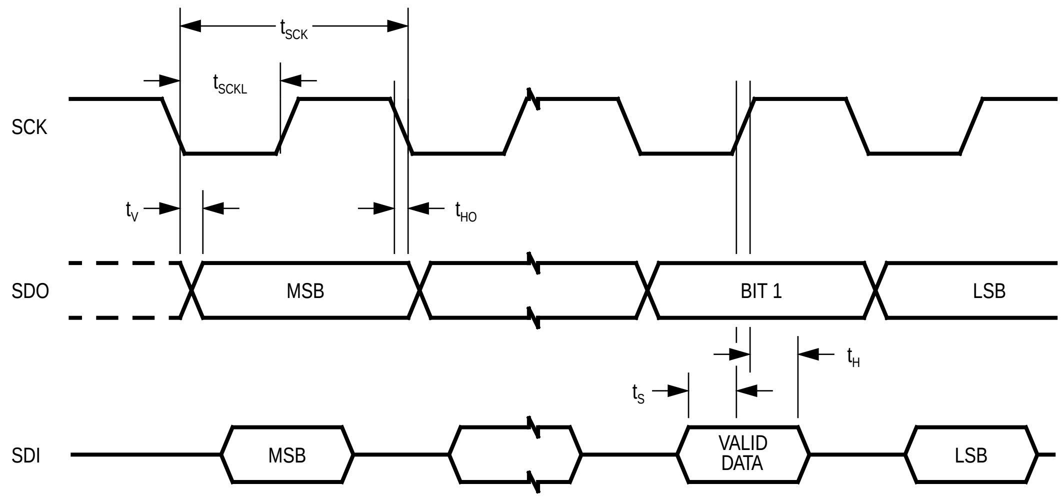


Figure 5. SIOP Timing

Table 1. SIOP Timing (V_{DD} = 5.0 Vdc)(1)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Master Slave	 $f_{SIOP(M)}$ $f_{SIOP(S)}$	 $f_{OSC}/64$ dc	 $f_{OSC}/8$ 525	 MHz kHz
Cycle Time Master Slave	 $t_{SCK(M)}$ $t_{SCK(S)}$	 4.0 —	 4.0 1920	 t_{CYC}(2) ns
Clock (SCK) Low Time (f_{OP} = 2.1 MHz)(3)(4)	t_{SCKL}	932	—	ns
SDO Data Valid Time	t_V	—	200	ns
SDO Hold Time	t_{HO}	0	—	ns
SDI Setup Time	t_S	100	—	ns
SDI Hold Time	t_H	100	—	ns

1. V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H unless otherwise noted.
2. $t_{CYC} = 1 \div f_{OP}$
3. f_{OSC} = crystal frequency; $f_{OP} = f_{OSC} \div 2$ = 2.1 MHz maximum
4. In master mode, the frequency of SCK is $f_{OP} \div 4$.

8-siop_a

Table 2. SIOP Timing (V_{DD} = 3.3 Vdc)[1]

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Master Slave	$f_{SIOP\ (M)}$ $f_{SIOP(S)}$	$f_{OSC}/64$ dc	$f_{OSC}/8$ 250	MHz kHz
Cycle Time Master Slave	$t_{SCK(M)}$ $t_{SCK(S)}$	4.0 —	4.0 4000	t_{CYC}[2]
Clock (SCK) Low Time (f_{OP} = 1.0 MHz)[3] [4]	t_{SCKL}	1980	—	ns
SDO Data Valid Time	t_V	—	400	ns
SDO Hold Time	t_{HO}	0	—	ns
SDI Setup Time	t_S	200	—	ns
SDI Hold Time	t_H	200	—	ns

1. V_{DD} = 3.3 Vdc ±10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H unless otherwise noted
2. $t_{CYC} = 1 \div f_{OP}$
3. f_{OSC} = crystal frequency; $f_{OP} = f_{OSC} \div 2$ = 1.0 MHz maximum
4. In master mode, the frequency of SCK is $f_{OP} \div 4$.

Interrupts

The SIOP does not generate interrupt requests.

I/O Registers

The following registers control and monitor SIOP operation:

- SIOP control register (SCR)
- SIOP status register (SSR)
- SIOP data register (SDR)

SIOP Control Register

The read/write SIOP control register (SCR) contains two bits. One bit enables the SIOP, and the other configures the SIOP for master mode or for slave mode.

$000A	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	0	SPE	0	MSTR	0	0	0	0
Reset:	0	0	0	0	0	0	0	0

Figure 6. SIOP Control Register (SCR)

SPE — SIOP Enable

This read/write bit enables the SIOP. Setting SPE initializes the data direction register as follows:

- The PB6/SDI pin is an input.
- The PB5/SDO pin is an output.
- The PB7/SCK pin is an input in slave mode and an output in master mode.

Clearing SPE disables the SIOP and returns the port to its normal I/O functions. The data direction register and the port data register remain in their SIOP-initialized state.

NOTE: *After clearing SPE, be sure to initialize the port for its intended I/O use.*

Clearing SPE during a transmission aborts the transmission, resets the bit counter, and returns the port to its normal I/O function. Reset clears SPE.

1 = SIOP enabled
0 = SIOP disabled

MSTR — Master Mode Select

This read/write bit configures the SIOP for master mode. Setting MSTR initializes the PB7/SCK pin as the serial clock output. Clearing MSTR initializes the PB7/SCK pin as the serial clock input. MSTR can be set at any time regardless of the state of SPE. Reset clears MSTR.

1 = Master mode selected
0 = Slave mode selected

SIOP Status Register

The read-only SIOP status register (SSR) contains two bits. One bit indicates that a SIOP transfer is complete, and the other indicates that an invalid access of the SIOP data register occurred while a transfer was in progress.

$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

Figure 7. SIOP Status Register (SSR)

SPIF — Serial Peripheral Interface Flag

This clearable, read-only bit is set automatically on the eighth rising edge on the PB7/SCK pin and indicates that a data transmission took place. SPIF does not inhibit further transmissions. Clear SPIF by reading the SIOP status register while SPIF is set and then reading or writing the SIOP data register. Reset clears SPIF.

1 = Transmission complete
0 = Transmission not complete

12-siop_a

DCOL — Data Collision Flag

This clearable, read-only bit is automatically set if the SIOP data register is accessed while a data transfer is in progress. Reading or writing the SIOP data register while a transmission is in progress causes invalid data to be transmitted or read. Clear DCOL by reading the SIOP status register with SPIF set and then accessing the SIOP data register. Because the clearing sequence accesses the SIOP data register, the sequence has to be completed before another transmission starts or DCOL is set again.

To clear DCOL when SPIF is not set, turn off the SIOP by writing a zero to SPE and then turn it back on by writing a one to SPE. Reset clears DCOL.

1 = Invalid access of SDR
0 = Valid access of SDR

SIOP Data Register

The SIOP data register (SDR) is both the transmit data register and the receive data register. To read or write the SIOP data register, the SPE bit in the SIOP control register must be set.

$000C	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
Reset:	Unaffected by reset							

Figure 8. SIOP Data Register (SDR)

With the SIOP configured for master mode, writing to the SIOP data register initiates a serial transfer. This register is not buffered. Writing to the SIOP data register overwrites the previous contents. Reading or writing to the SIOP data register while a transmission is in progress can cause invalid data to be transmitted or received.

Low-Power Modes

The WAIT and STOP instructions put the MCU in low-power consumption standby modes.

Stop Mode

The STOP instruction suspends the clock to the SIOP. When the MCU exits stop mode, processing resumes after the internal oscillator stabilization delay of 4064 oscillator cycles.

A STOP instruction in a master SIOP does not suspend the clock to slave SIOPs.

Wait Mode

The WAIT instruction suspends the clock to the SIOP. When the MCU exits wait mode, processing resumes immediately.

A WAIT instruction in a master SIOP does not suspend the clock to slave SIOPs.

Analog-to-Digital Converter ADC

Contents

Features

- 8-Bit Conversions with ± 1.5-LSB Precision
- Four External and Three Internal Analog Input Channels
- Wait Mode Operation

1-atd4x8nvrl

Introduction

The ADC consists of a single successive-approximation A/D converter, an input multiplexer to select one of four external or two internal channels, and control circuitry. **Figure 1** shows the structure of the ADC module.

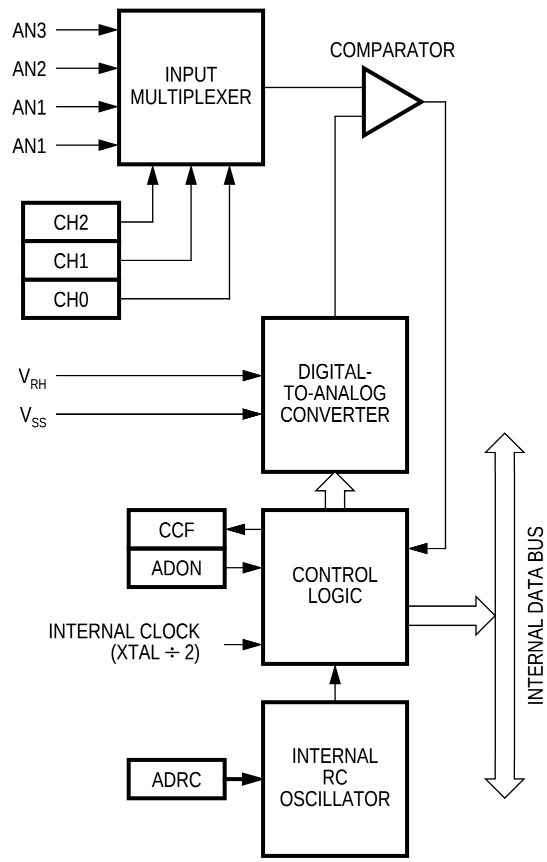


Figure 1. ADC Block Diagram

Table 1. ADC I/O Register Summary

Addr.	Name	R/W	Bit 7	6	5	4	3	2	1	Bit 0
$001D	ADC Data Register (ADDR)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Unaffected by reset							
$001E	ADC Status/Control Register (ADSCR)	Read:	CCF	ADRC	ADON	0	0	CH2	CH1	CH0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Operation

The A/D conversion process is ratiometric, using two reference voltages, V_{RH} and V_{SS}. Conversion accuracy is guaranteed only if V_{RH} is equal to V_{DD}.

Pin Functions

The ADC uses five pins and shares them with port C:

- PC7/V_{RH}
- PC6/AN0, PC5/AN1, PC4/AN2, and PC3/AN3

PC7/V_{RH}

The voltage reference high pin (PC7/V_{RH}) supplies the high reference voltage for the ratiometric conversion process. For ratiometric conversion, the supply voltage of the analog source should be the same as V_{RH} and be referenced to V_{SS}.

PC6/AN0–PC3/AN3

The multiplexer can select one of four external analog input channels (AN0, AN1, AN2, or AN3) for sampling. The conversion takes 32 cycles. The first 12 cycles sample the voltage on the selected input pin by charging an internal capacitor. In the last 20 cycles, a comparator successively compares the output of an internal D/A converter to the sampled analog input. Control logic changes the D/A converter input one bit at a time, starting with the MSB, until the D/A converter output matches the sampled analog input. The conversion is monotonic and has no missing codes. At the end of the conversion, the conversion complete flag (CCF) becomes set, and the CPU takes two cycles to move the result to the ADC data register.

NOTE: *To prevent excess power dissipation, do not simultaneously use an I/O port pin as a digital input and an analog input.*

While the ADC is on, the selected analog input reads as logic zero. The port C pins that are not selected read normally.

An analog input voltage equal to V_{RH} converts to digital $FF; an input voltage greater than V_{RH} converts to $FF with no overflow. An analog input voltage equal to V_{SS} converts to digital $00. For ratiometric conversion, the source of each analog input should use V_{RH} as the supply voltage and be referenced to V_{SS}.

The clock frequency must be equal to or greater than 1 MHz. If the internal clock frequency is less than 1 MHz, the internal RC oscillator (nominally 1.5 MHz) must be used for the ADC conversion clock. Make this selection by setting the ADRC bit to logic one in the ADC status and control register.

Interrupts

The ADC cannot generate interrupt requests.

Timing and Electrical Characteristics

Table 2. ADC Characteristics (V_{DD} = 5.0 Vdc)(1)

Characteristic	Min	Max	Unit
Resolution	8	8	Bit
Absolute Accuracy (4.0 > V_{RH} > V_{DD})(2)	—	±1.5	LSB
Conversion Range (PC7/V_{RH})	V_{SS}	V_{DD}	V
Conversion Time (Includes Sampling Time) External Clock Internal RC Oscillator (ADRC = 1)	 32 32	 32 32	 t_{AD}(3) µs
Monotonicity	Inherent (Within Total Error)		
Zero Input Reading (V_{IN} = 0 V)	00	01	Hex
Full-Scale Reading (V_{IN} = V_{RH})	FF	FF	Hex
Sample Acquisition Time(4) External Clock Internal RC Oscillator (ADRC = 1)	 12 —	 12 12	 t_{AD}(5) µs
Input Capacitance PC6/AN0, PC5/AN1, PC4/AN2, PC3/AN3	—	12	pF
Analog Input Voltage	V_{SS}	V_{RH}	V
Input Leakage(6) PC6/AN0, PC5/AN1, PC4/AN2, PC3/AN3 PC7/V_{RH}	 — —	 ±1 ±1	µA
ADC On Current Stabilization Time	—	100	µs

1. V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc
2. ADC accuracy may decrease proportionately as V_{RH} is reduced below 4.0 V.
3. t_{AD} = cycle time of the A/D converter
4. Source impedances more than 10 kΩ adversely affect internal RC charging time during input sampling.
5. $t_{AD} = t_{CYC} (1 \div f_{OP})$ if MCU clock is clock source
6. External system error caused by input leakage approximately equals R source times input current.

I/O Registers

The following registers control and monitor operation of the ADC:

- ADC status and control register (ADSCR)
- ADC data register (ADDR)

ADC Status and Control Register

The ADC status and control register (ADSCR) contains a conversion complete flag and four writable control bits. Writing to ADSCR clears the conversion complete flag and starts a new conversion sequence.

$001E	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CCF	ADRC	ADON	0	0	CH2	CH1	CH0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 2. ADC Status and Control Register (ADSCR)

CCF — Conversion Complete Flag

This read-only bit is automatically set when an analog-to-digital conversion is complete, and a new result can be read from the ADC data register. Clear the CCF bit by writing to the ADC status and control register or by reading the ADC data register. Resets clear the CCF bit.

1 = Conversion complete
0 = Conversion not complete

6-atd4x8nvrl

ADRC — ADC RC (Oscillator)

This read/write bit turns on the internal RC oscillator to drive the ADC. If the internal clock frequency (f_{OP}) is less than 1 MHz, ADRC must be set. When the RC oscillator is turned on, it requires a time, t_{ADRC}, to stabilize, and results can be inaccurate during this time. Resets clear the ADRC bit.

1 = Internal RC oscillator drives ADC
0 = Internal clock drives ADC

When the internal RC oscillator is being used as the ADC clock, two limitations apply:

- Because of the frequency tolerance of the RC oscillator and its asynchronism with the internal clock, the conversion complete flag must be used to determine when a conversion sequence is complete.
- The conversion process runs at the nominal 1.5-MHz rate, but the conversion results must be transferred to the ADC data register synchronously with the internal clock; therefore, the conversion process is limited to a maximum of one channel every internal clock cycle.

ADON — ADC On

This read/write bit turns on the ADC. When the ADC is on, it requires a time, t_{ADON}, for the current sources to stabilize. During this time, results can be inaccurate. Resets clear the ADON bit.

1 = ADC turned on
0 = ADC turned off

Bits 4–2 — Not used

Bits 4–2 always read as logic zeros.

CH[2:0] — Channel Select Bits

These read/write bits select one of eight ADC input channels as shown in **Table 3**. Channels 0–3 are the input pins, PC3/AN3, PC4/AN2, PC5/AN1, and PC6/AN0. Channels 4–6 can be used for reference measurements. Channel 7 is reserved for factory testing.

Table 3. ADC Input Channel Selection

CH[2:1:0]	Channel	Signal
000	0	AN0
001	1	AN1
010	2	AN2
011	3	AN3
100	4	V_{RH}
101	5	$(V_{RH} + V_{SS}) / 2$
110	6	V_{SS}
111	7	Reserved

To prevent excess power dissipation, do not use an ADC pin as an analog input and a digital input at the same time.

Using one of the port pins as the ADC input does not affect the ability to use the remaining port pins as digital inputs.

Reading a port pin that is selected as an analog input returns a logic zero.

ADC Data Register

The ADC data register (ADDR) is a read-only register that contains the result of the most recent analog-to-digital conversion.

$001D	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 7	6	5	4	3	2	1	Bit 0
Write:								
Reset:	Unaffected by reset							

Figure 3. ADC Data Register (ADDR)

Low-Power Modes

Stop Mode

The STOP instruction turns off the ADC and aborts any current and pending conversions.

Wait Mode

The ADC continues to operate normally after the WAIT instruction. To reduce power consumption in wait mode:

- If the ADC is not being used, clear both the ADON and ADRC bits before entering wait mode.
- If the ADC is being used and the internal clock rate is above 1 MHz, clear the ADRC bit before entering wait mode.

Specifications

Contents

1-mc68hc705p9

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 **Table 1**. Keep V_{IN} and V_{OUT} within the range $V_{SS} \leq (V_{IN}$ or $V_{OUT}) \leq V_{DD}$. Connect unused inputs to the appropriate voltage level, either V_{SS} or V_{DD}.

Table 1. Maximum Ratings

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	–0.3 to +7.0	V
Current Drain per Pin (Excluding V_{DD} and V_{SS})	I	25	mA
Input Voltage	V_{IN}	V_{SS} – 0.3 to V_{DD} + 0.3	V
EPROM Programming Voltage	V_{PP}	16.75	V
Storage Temperature Range	T_{STG}	–65 to +150	°C

NOTE: *This device is not guaranteed to operate properly at the maximum ratings. Refer to* ***5.0 V DC Electrical Characteristics*** on page 135 *and* ***3.3 V DC Electrical Characteristics*** on page 136 *for guaranteed operating conditions.*

2-mc68hc705p9

Operating Temperature Range

Table 2. Operating Temperature Range

Package Type	Symbol	Value	Unit
MC68HC705P9P[1], DW[2], S[3] (Standard) MC68HC705P9C[4]P, CDW, CS (Extended) MC68HC705P9V[5]P, VDW, VS (Automotive) MC68HC705P9M[6]P, MDW, MS (Automotive)	T_A	T_L to T_H 0 to 70 –40 to +85 –40 to +105 –40 to +125	°C

1. P = Plastic dual in-line package (PDIP)
2. DW = Small outline integrated circuit (SOIC)
3. S = Ceramic dual in-line package (Cerdip)
4. C = Extended temperature range (–40 to +85 °C)
5. V = Automotive temperature range (–40 to +105 °C)
6. M = Automotive temperature range (–40 to +125 °C)

Thermal Characteristics

Table 3. Thermal Characteristics

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic Dual In-Line Package (PDIP) Small Outline Integrated Circuit (SOIC) Ceramic Dual In-Line Package (Cerdip)	θ_{JA}	 60 60 60	°C/W

Power Considerations

The average chip junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \times \theta_{JA}) \qquad (1)$$

where:

T_A = ambient temperature in °C
θ_{JA} = package thermal resistance, junction to ambient in °C/W
$P_D = P_{INT} + P_{I/O}$
$P_{INT} = I_{CC} \times V_{CC}$ = chip internal power dissipation
$P_{I/O}$ = power dissipation on input and output pins (user-determined)

For most applications, $P_{I/O} \ll P_{INT}$ and can be neglected.

Ignoring $P_{I/O}$, the relationship between P_D and T_J is approximately:

$$P_D = \frac{K}{T_J + 273\,°C} \qquad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \times (T_A + 273\,°C) + \theta_{JA} \times (P_D)^2 \qquad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

4-mc68hc705p9

5.0 V DC Electrical Characteristics

Table 4. DC Electrical Characteristics (V_{DD} = 5.0 Vdc)(1)

Characteristic	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} = –0.8 mA) PA7–PA0, PB7/SCK–PB5/SDO, PC7/V_{RH}–PC0, PD5, TCMP	V_{OH}	V_{DD} – 0.8	—	—	V
Output Low Voltage (I_{LOAD} = 1.6 mA) PA7–PA0, PB7/SCK–PB5/SDO, PC7/V_{RH}–PC0, PD5, TCMP	V_{OL}	—	—	0.4	V
Input High Voltage PA7–PA0, PB7/SCK–PB5/SDO, PC7/V_{RH}–PC0, PD5, PD7/TCAP, $\overline{IRQ}$/V_{PP}, $\overline{RESET}$, OSC1	V_{IH}	0.7 × V_{DD}	—	V_{DD}	V
Input Low Voltage PA7–PA0, PB7/SCK–PB5/SDO, PC7/V_{RH}–PC0, PD5, PD7/TCAP, $\overline{IRQ}$/V_{PP}, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	0.2 × V_{DD}	V
Supply Current(3) (4) (5) (6) Run Mode Wait Mode (ADC On) Wait Mode (ADC Off) Stop Mode 25 °C 0 to 70 °C (Standard) –40 to 125 °C	I_{DD}	 — — — — — —	 4.7 2.1 1.3 2 — —	 6.5 2.9 1.9 30 50 100	 mA mA mA µA µA µA
I/O Ports Hi-Z Leakage Current PA7–PA0, PB7/SCK–PB5/SDO, PC7/V_{RH}–PC0, PD5	I_{IL}	—	—	±10	µA
ADC Ports Hi-Z Leakage Current	I_{OZ}	—	—	±1	µA
Input Current $\overline{RESET}$, $\overline{IRQ}$/V_{PP}, OSC1, PD7/TCAP	I_{IN}	—	—	±1	µA
Capacitance Ports (As Inputs or Outputs) $\overline{RESET}$, $\overline{IRQ}$/V_{PP}	 C_{OUT} C_{IN}	 — —	 — —	 12 8	 pF
Programming Voltage	V_{PP}	16.25	16.5	16.75	V
Programming Current	I_{PP}	—	5	10	mA
Programming Time per Byte	t_{EPGM}	4	—	—	ms

1. V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc, T_A = T_L to T_H unless otherwise noted
2. Typical values at midpoint of voltage range, 25 °C only
3. Run mode and wait mode I_{DD} 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 mode and stop mode I_{DD} measured with all ports configured as inputs; V_{IL} = 0.2 V; V_{IH} = V_{DD} – 0.2 V
5. Stop mode I_{DD} measured with OSC1 = V_{SS}
6. Wait mode I_{DD} affected linearly by OSC2 capacitance

3.3 V DC Electrical Characteristics

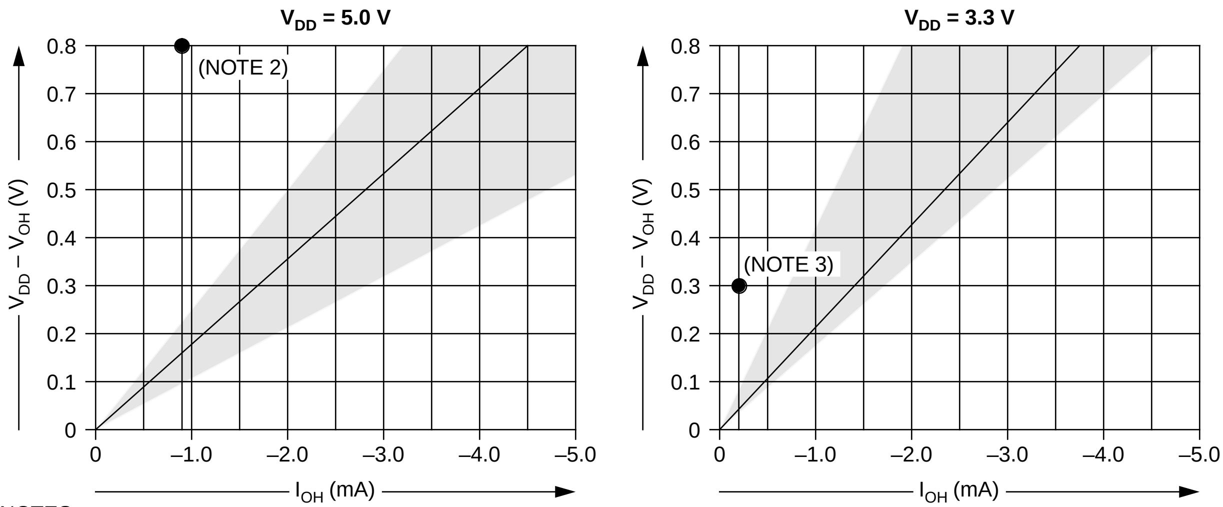
Table 5. DC Electrical Characteristics (V_{DD} = 3.3 Vdc)(1)

Characteristic	Symbol	Min	Typ(2)	Max	Unit
Output Voltage ($I_{LOAD} \le 10.0\ \mu A$)	V_{OL} V_{OH}	— $V_{DD} - 0.1$	— —	0.1 —	V
Output High Voltage (I_{LOAD} = –0.2 mA) PA7–PA0, PB7/SCK–PB5/SDO, PC7/V_{RH}–PC0, PD5, TCMP	V_{OH}	$V_{DD} - 0.3$	—	—	V
Output Low Voltage (I_{LOAD} = 0.4 mA) PA7–PA0, PB7/SCK–PB5/SDO, PC7/V_{RH}–PC0, PD5, TCMP	V_{OL}	—	—	0.3	V
Input High Voltage PA7–PA0, PB7/SCK–PB5/SDO, PC7/V_{RH}–PC0, PD5, PD7/TCAP, $\overline{IRQ}/V_{PP}$, $\overline{RESET}$, OSC1	V_{IH}	$0.7 \times V_{DD}$	—	V_{DD}	V
Input Low Voltage PA7–PA0, PB7/SCK–PB5/SDO, PC7/V_{RH}–PC0, PD5, PD7/TCAP, $\overline{IRQ}/V_{PP}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	$0.2 \times V_{DD}$	V
Data-Retention Mode Supply Voltage	V_{RM}	2.0	—	—	V
Supply Current(3) (4) (5) (6) Run Mode Wait Mode (ADC On) Wait Mode (ADC Off) Stop Mode 25 °C 0 to 70 °C (Standard) –40 to 125 °C	I_{DD}	 — — — — — —	 1.6 0.9 0.4 1.0 — —	 2.3 1.3 0.6 20 40 50	 mA mA mA µA µA µA
I/O Ports Hi-Z Leakage Current PA7–PA0, PB7/SCK–PB5/SDO, PC7/V_{RH}–PC0, PD5	I_{IL}	—	—	±10	µA
Input Current $\overline{RESET}$, $\overline{IRQ}/V_{PP}$, OSC1, PD7/TCAP	I_{IN}	—	—	±1	µA
Capacitance Ports (As Inputs or Outputs) $\overline{RESET}$, $\overline{IRQ}/V_{PP}$	 C_{OUT} C_{IN}	 — —	 — —	 12 8	pF

1. V_{DD} = 3.3 Vdc ±10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H unless otherwise noted
2. Typical values at midpoint of voltage range, 25 °C only
3. Run mode and wait mode I_{DD} measured using external square wave clock source (f_{OSC} = 2.1 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 mode and stop mode I_{DD} measured with all ports configured as inputs; V_{IL} = 0.2 V; $V_{IH} = V_{DD} - 0.2$ V
5. Stop mode I_{DD} measured with OSC1 = V_{SS}
6. Wait mode I_{DD} affected linearly by OSC2 capacitance

6-mc68hc705p9

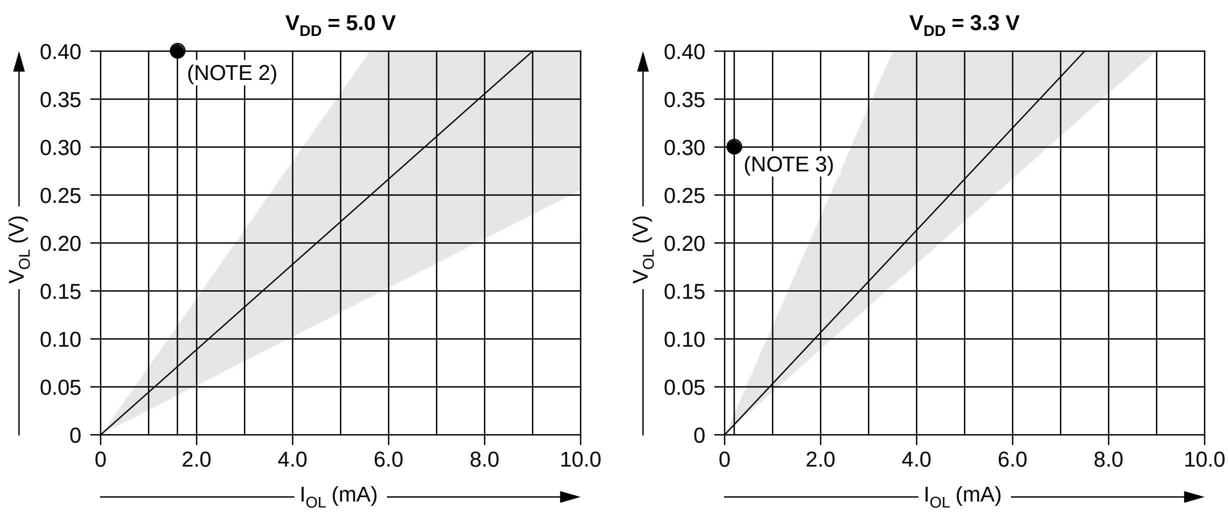
Driver Characteristics



NOTES:

1. Shaded area indicates variation in driver characteristics due to changes in temperature and for normal processing tolerances. Within the limited range of values shown, V vs I curves are approximately straight lines.
2. At V_{DD} = 5.0 V, devices are specified and tested for $V_{OL} \leq$ 800 mV @ I_{OL} = –0.8 mA.
3. At V_{DD} = 3.3 V, devices are specified and tested for $V_{OL} \leq$ 300 mV @ I_{OL} = –0.2 mA.

Figure 1. Typical High-Side Driver Characteristics



NOTES:

1. Shaded area indicates variation in driver characteristics due to changes in temperature and for normal processing tolerances. Within the limited range of values shown, V vs I curves are approximately straight lines.
2. At V_{DD} = 5.0 V, devices are specified and tested for $V_{OL} \leq$ 400 mV @ I_{OL} = 1.6 mA.
3. At V_{DD} = 3.3 V, devices are specified and tested for $V_{OL} \leq$ 300 mV @ I_{OL} = 0.4 mA.

Figure 2. Typical Low-Side Driver Characteristics

Typical Supply Current vs. Internal Clock Frequency

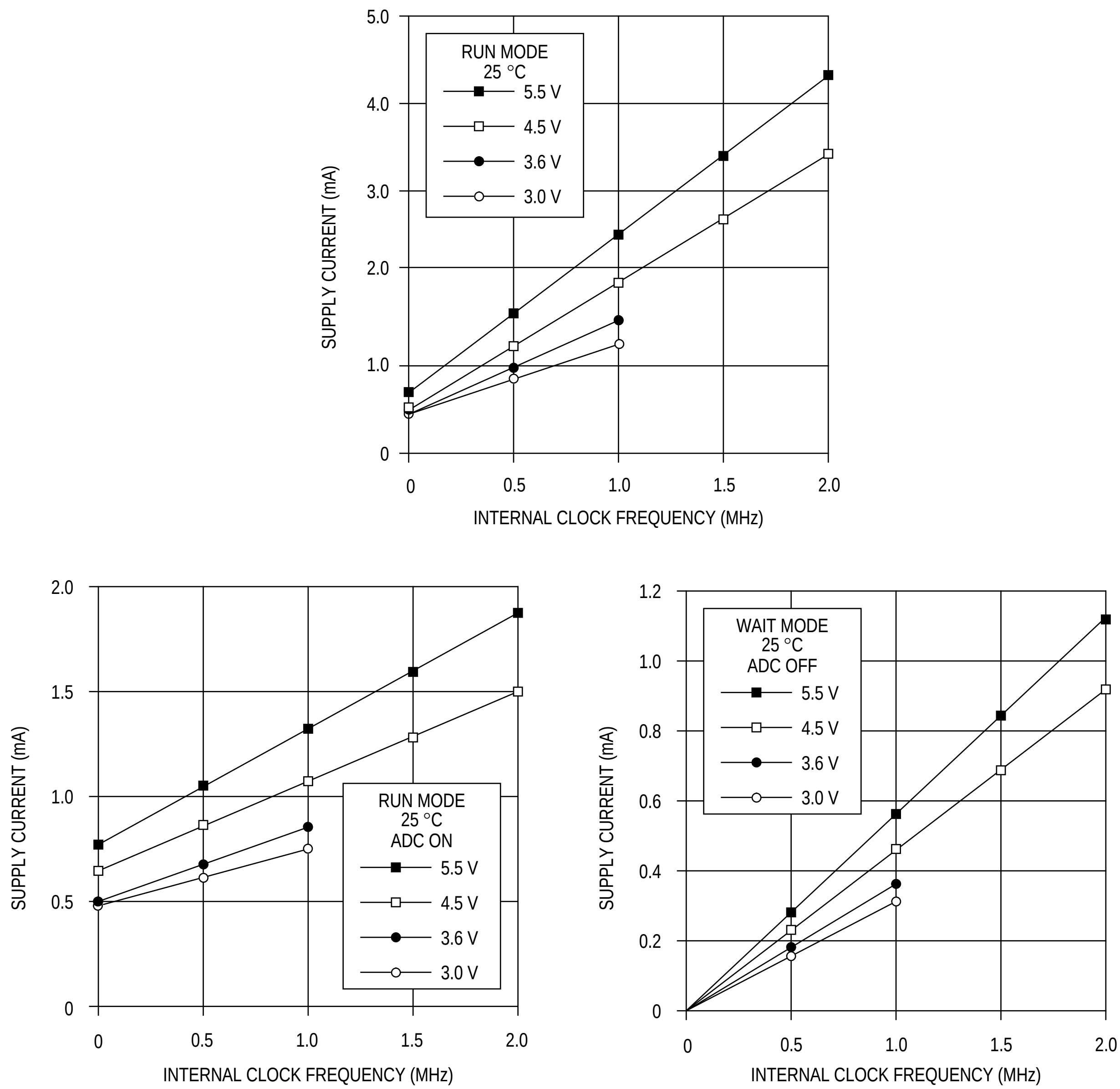


Figure 3. Typical Supply Current vs. Internal Clock Frequency

8-mc68hc705p9

Maximum Supply Current vs. Internal Clock Frequency

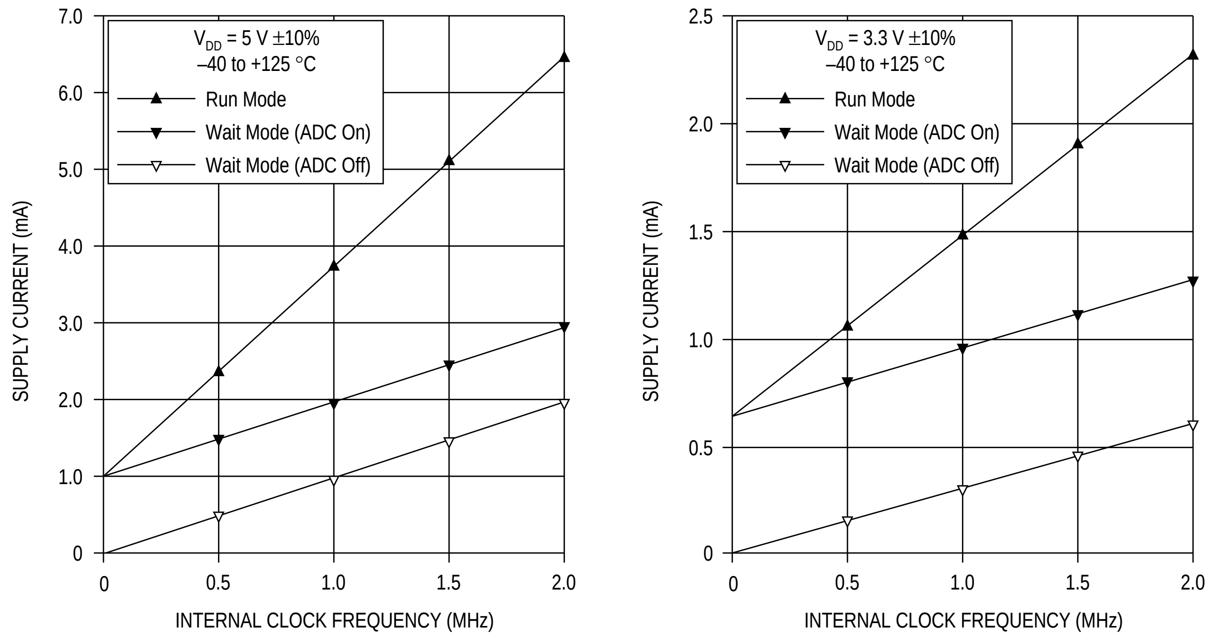


Figure 4. Maximum Supply Current vs. Internal Clock Frequency

5.0 V Control Timing

Table 6. Control Timing (V_{DD} = 5.0 Vdc)(1)

Characteristic	Symbol	Min	Max	Unit
Oscillator Frequency Crystal External Clock	f_{OSC}	— dc	4.2 4.2	MHz
Internal Operating Frequency ($f_{OSC} \div 2$) Crystal External Clock	f_{OP}	— dc	2.1 2.1	MHz
Cycle Time ($1 \div f_{OP}$)	t_{CYC}	480	—	ns
Crystal Oscillator Startup Time	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (Crystal Oscillator)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ Pulse Width	t_{RL}	1.5	—	t_{CYC}
Timer Resolution(2) Input Capture Pulse Width Input Capture Pulse Period	 t_{RESL} t_H, t_L t_{TLTL}	 4.0 125 Note(3)	 — — —	 t_{CYC} ns t_{CYC}
Interrupt Pulse Width Low (Edge-Triggered)	t_{ILIH}	125	—	ns
Interrupt Pulse Period	t_{ILIL}	Note(4)	—	t_{CYC}
OSC1 Pulse Width	t_{OH}, t_{OL}	90	—	ns
RC Oscillator Stabilization Time	t_{RCON}	—	5	µs
ADC On Current Stabilization Time	t_{ADON}	—	100	µs

1. V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H unless otherwise noted
2. A 2-bit prescaler in the timer is the limiting factor as it counts 4 t_{CYC}
3. The minimum t_{TLTL} should not be less than the number of interrupt service routine cycles plus 19 t_{CYC}
4. The minimum t_{ILIL} should not be less than the number of interrupt service routine cycles plus 19 t_{CYC}

10-mc68hc705p9

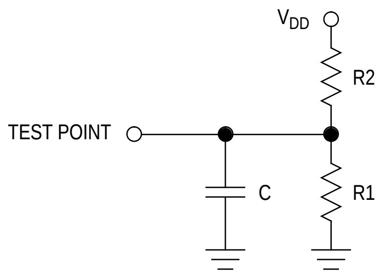
3.3 V Control Timing

Table 7. Control Timing (V_{DD} = 3.3 Vdc)[1]

Characteristic	Symbol	Min	Max	Unit
Oscillator Frequency Crystal External Clock	f_{OSC}	 — dc	 2.0 2.0	MHz
Internal Operating Frequency ($f_{OSC} \div 2$) Crystal External Clock	f_{OP}	 — dc	 1.0 1.0	MHz
Cycle Time ($1 \div f_{OP}$)	t_{CYC}	1	—	ms
Crystal Oscillator Startup Time	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (Crystal Oscillator)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ Pulse Width	t_{RL}	1.5	—	t_{CYC}
Timer Resolution[2] Input Capture Pulse Width Input Capture Pulse Period	 t_{RESL} t_H, t_L t_{TLTL}	 4.0 250 Note[3]	 — — —	 t_{CYC} ns t_{CYC}
Interrupt Pulse Width Low (Edge-Triggered)	t_{ILIH}	250	—	ns
Interrupt Pulse Period	t_{ILIL}	Note[4]	—	t_{CYC}
OSC1 Pulse Width	t_{OH}, t_{OL}	200	—	ns

1. V_{DD} = 3.3 Vdc ±10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H unless otherwise noted
2. A 2-bit prescaler in the timer is the limiting factor as it counts 4 t_{CYC}
3. The minimum t_{TLTL} should not be less than the number of interrupt service routine cycles plus 19 t_{CYC}
4. The minimum t_{ILIL} should not be less than the number of interrupt service routine cycles plus 19 t_{CYC}

Test Load



PINS	R1	R2	C
PA7–PA0	3.26 kΩ	2.38 kΩ	50 pF
PB7/SCK–PB5/SDO			
PC7/V_{RH}–PC0			

Figure 5. Test Load

Mechanical Specifications

The MC68HC705P9 is available in the following packages:

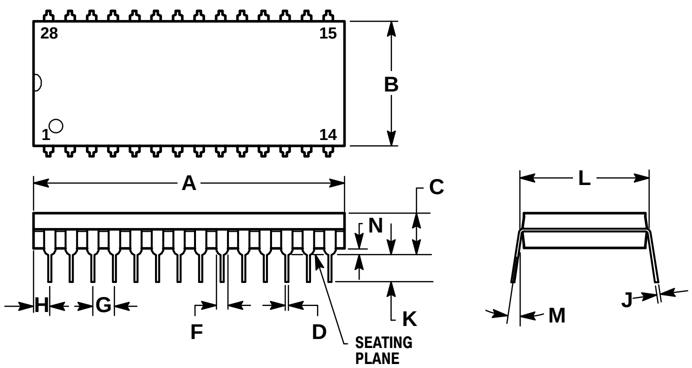
- 710 — Plastic dual in-line package (PDIP)
- 733 — Ceramic dual in-line package (Cerdip)
- 751F — Small outline integrated circuit (SOIC)

The following figures show the latest packages at the time of this publication. To make sure that you have the latest package specifications, contact one of the following:

- Local Motorola Sales Office
- Motorola Mfax
  - Phone 602-244-6609
  - EMAIL rmfax0@email.sps.mot.com
- Worldwide Web (wwweb) at http://design-net.com

Follow Mfax or wwweb on-line instructions to retrieve the current mechanical specifications.

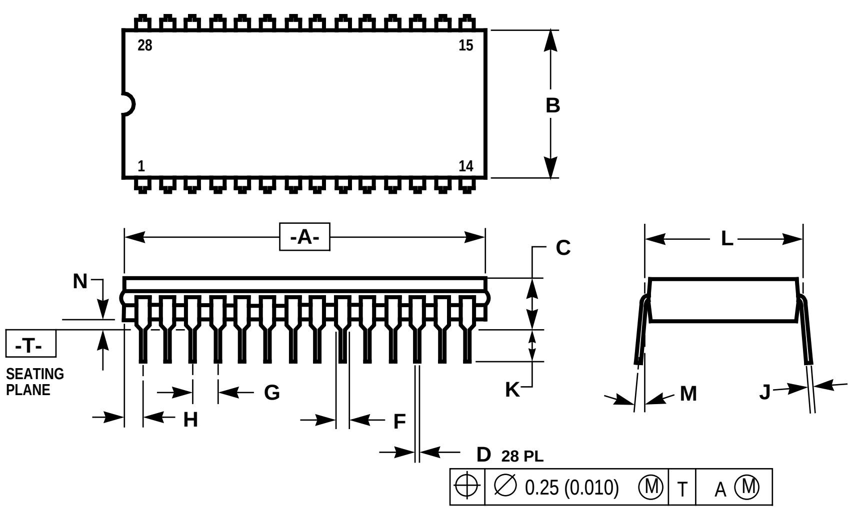
28-Pin PDIP — Case #710



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

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

28-Pin Cerdip — Case #733

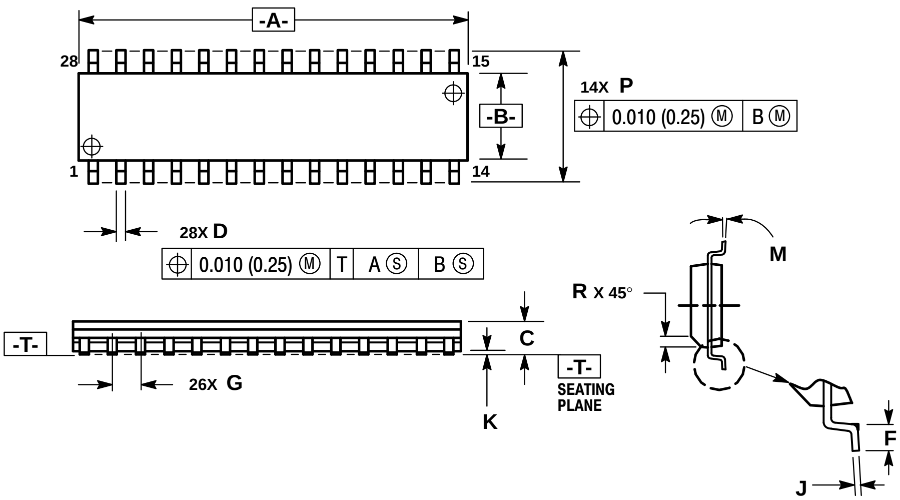


NOTES:
1. DIMENSIONS A AND B INCLUDES MENISCUS.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
4. CONTROLLING DIMENSION: INCH.

	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	1.435	1.490	36.45	37.84
B	0.500	0.605	12.70	15.36
C	0.160	0.230	4.06	5.84
D	0.015	0.022	0.38	0.55
F	0.050	0.065	1.27	1.65
G	0.100 BSC		2.54 BSC	
J	0.008	0.012	0.20	0.30
K	0.125	0.160	3.18	4.06
L	0.600 BSC		15.24 BSC	
M	0°	15°	0°	15°
N	0.020	0.050	0.51	1.27

13-mc68hc705p9

28-Pin SOIC — 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.

DIM	MILLIMETERS		INCHES	
	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

14-mc68hc705p9

Index

A

B

C

D

E

F

I

Q

R

S

Literature Updates

This document contains the latest data available at publication time. For updates, contact one of the following:

Literature Distribution Centers

Call or order literature by mail.

USA/Europe:

Motorola Literature Distribution
P.O. Box 20912
Phoenix, Arizona 85036
Phone 1 800 441-2447 or 602 303-5454

Japan:

Nippon Motorola Ltd.
Tatsumi-SPD-JLDC
Toshikatsu Otsuki
6F Seibu-Butsuryu Center
3-14-2 Tatsumi Koto-Ku
Tokyo 135, Japan
Phone 03-3521-8315

Hong Kong:

Motorola Semiconductors H.K. Ltd.
8B Tai Ping Industrial Park
51 Ting Kok Road
Tai Po, N.T., Hong Kong
Phone 852-26629298

Mfax

Call to access this worldwide faxing service.

RMFAX0@email.sps.mot.com
TOUCH-TONE 602 244-6609

Or, on the http://Design-NET.com home page, select the Mfax icon. Obtain a fax of complete, easy-to-use Mfax instructions by entering your FAX number and then pressing the 1 key.

Motorola SPS World Marketing World Wide Web Server

Use the Internet to access this World Wide Web Server. Use the following URL:

http://Design-net.com

CSIC Microcontroller Division's Web Site

Directly access the CSIC Microcontroller Division's web site with the following URL:

http://Design-net.com/csic/CSIC_home.html

MOTOROLA

MC68HC705P9 Rev. 3
Technical Data Book

Please help us to continue improving the quality and usefulness of our data books by filling out this form and sending your comments to us. You can return the form by mail, or FAX it to 512-891-3236.

Thank you for your help and continued support!

1. How do you rate the quality of this data book?

	High				Low
Organization	☐	☐	☐	☐	☐
Readability	☐	☐	☐	☐	☐
Accuracy	☐	☐	☐	☐	☐
Figures	☐	☐	☐	☐	☐

	High				Low
Tables	☐	☐	☐	☐	☐
Table of contents	☐	☐	☐	☐	☐
Page size/binding	☐	☐	☐	☐	☐
Overall impression	☐	☐	☐	☐	☐

2. Are you able to find the information you need easily?

Yes				No
☐	☐	☐	☐	☐

3. What are your recommendations for making this data book more useful?

4. What additional information would you like to see included in future data books?

PLEASE
PASTE
POSTAGE
HERE

Motorola
6501 William Cannon Drive West
Mail Stop OE17
Austin, Texas 78735-8598
USA

Attention: CSIC Publications Department

MOTOROLA
CSIC Microcontroller Division

First: cut along this line to remove

Second: fold back along this line

Please supply the following information (optional).

Name: ______________________________

Company Name: ______________________________

Title: ______________________________

Address: ______________________________

City: ______________________ State: _____ Zip: __________

Phone Number: ______________________________

MOTOROLA

CSIC

MICROCONTROLLERS

How to reach us:

MFAX: RMFAX0@email.sps.mot.com - TOUCHTONE (602) 244-6609

INTERNET: http://Design-net.com

USA/EUROPE: Motorola Literature Distribution; P.O. Box 20912; Phoenix, Arizona 85036. 1-800-441-2447

JAPAN: Nippon Motorola Ltd.; Tatsumi-SPD-JLDC, Toshikatsu Otsuki, 6F Seibu-Butsuryu-Center, 3-14-3 Tatsumi Koto-Ku, Tokyo 135, Japan. 03-3521-8315

HONG KONG: Motorola Semiconductors H.K. Ltd.; 8B Tai Ping Industrial Park, 51 Ting Tok Road, Tai Po, N.T., Hong Kong. 852-26629298