HC705MC4GRS/D
REV 1.5

68HC705MC4

SPECIFICATION
(General Release)

CSIC System Design Group
Austin, Texas

The MC68HC705MC4 is an MCU device in a 28-pin DIP or SOIC package with the HC05 CPU core, a 16-bit timer including an output compare and two input captures, an 8-bit A/D converter with a 6 channel input multiplexer, a dual channel pulse width modulator (PWM), an SCI and a COP watchdog timer. The 4 K byte memory map has 3584 bytes of user ROM/EPROM and 176 bytes of RAM.

TABLE OF CONTENTS

SECTION 1 INTRODUCTION 1

1.1 FEATURES (all timing based on 3 MHz bus) 1
1.2 MASK OPTIONS 3
1.3 FUNCTIONAL PIN DESCRIPTION 3
1.3.1 V_{DD} AND V_{SS} 3
1.3.2 OSC1, OSC2 3
1.3.2.1 Crystal with Internal Components 4
1.3.2.2 Ceramic Resonator 4
1.3.2.3 External Clock 5
1.3.3 $\overline{\text{RESET}}$ 5
1.3.4 PA0, PA1/PWMA1, PA2/PWMB1, PA3/PWMA2, PA4/PWMB2, PA5/PWMA3, PA6/PWMB3, PA7 5
1.3.5 PB4/TDO, PB5/RDI, PB6, PB7 5
1.3.6 PC0:5/AD0:5, PC6/Vrefh, PC7/Vrefl 5
1.3.7 PD6/TCAP1/TCMP, PD7/TCAP2 5
1.3.8 $\overline{\text{IRQ}}$ (MASKABLE INTERRUPT REQUEST) / V_{PP} 5
1.4 CPU CORE 6

SECTION 2 OPERATING MODES 7

2.1 USER MODE 7
2.2 BOOTLOADER MODE 8
2.2.1 Bootloader Functions 8
2.2.2 EPROM PROGRAMMING REGISTER 11
2.2.2.1 LATCH - EPROM Latch Control 12
2.2.2.2 EPGM - EPROM Program Control 12
2.2.3 MASK OPTION REGISTER (MOR) $0F00 12
2.2.3.1 COP - COP enable/disable 12
2.3 SELF-CHECK 13
2.4 LOW-POWER MODES 13
2.4.1 STOP INSTRUCTION 13
2.4.1.1 Stop Mode 13
2.4.2 WAIT INSTRUCTION 14
2.5 COP WATCHDOG TIMER CONSIDERATIONS 15

SECTION 3 RESETS 17

3.1 EXTERNAL RESET (RESET) 17
3.2 INTERNAL RESETS 17
3.2.1 ILLEGAL ADDRESS RESET 17
3.2.2 POWER-ON RESET (POR) 18
3.2.3 COMPUTER OPERATING PROPERLY (COP) RESET 18

SECTION 4 INTERRUPTS 19
4.1 RESET INTERRUPT SEQUENCE 21
4.2 SOFTWARE INTERRUPT (SWI) 22
4.3 HARDWARE INTERRUPTS 22
4.3.1 EXTERNAL INTERRUPT (IRQ) 22
4.3.1.1 IRQM - IRQ Enable Mask 23
4.3.1.2 IRQS - IRQ Sensitivity 23
4.3.1.3 EDGE - IRQ Active Edge Select 23
4.3.1.4 REQ - IRQ Interrupt Request 23
4.3.1.5 ACK - IRQ Interrupt Request Acknowledge 23
4.3.2 TIMER INTERRUPTS 24
4.3.2.1 INPUT CAPTURE INTERRUPTS 24
4.3.2.2 OUTPUT COMPARE INTERRUPT 25
4.3.2.3 TIMER OVERFLOW INTERRUPT 25
4.3.3 SCI INTERRUPT 25
4.3.4 CORE TIMER (CTIMER) INTERRUPT 25

SECTION 5 MEMORY 27
5.1 USER MODE MEMORY MAP 27
5.2 BOOTSTRAP/SELF-CHECK MODE MEMORY MAP 28
5.3 I/O AND CONTROL REGISTERS 28
5.4 RAM 32
5.5 ROM/EPROM 32

SECTION 6 INPUT/OUTPUT PORTS 33
6.1 PORT A 33
6.1.1 MCU LINE INTERFACE RECOMMENDATIONS 33
6.2 PORT B 34
6.3 PORT C 35
6.4 PORT D 36
6.5 I/O PORT PROGRAMMING 36

SECTION 7 ANALOG SUBSYSTEM 39
7.1 ANALOG SECTION 39
7.1.1 RATIOMETRIC CONVERSION 39
7.1.2 V_{REFH} and V_{REFL} 39
7.1.3 ACCURACY AND PRECISION 39
7.2 CONVERSION PROCESS 39
7.3 DIGITAL SECTION 39
7.3.1 CONVERSION TIMES 39
7.3.2 MULTI-CHANNEL OPERATION 40
7.3.3 UNUSED A/D INPUTS AS I/O 40
7.4 A/D STATUS AND CONTROL REGISTER (ADSCR) $25 40
7.4.1 COCO - CONVERSIONS COMPLETE 40
7.4.2 ADRC - A/D RC OSCILLATOR CONTROL 41
7.4.3 ADON - A/D ON 41

7.4.4 CH3:CH0 - CHANNEL SELECT BITS 41
7.5 A/D DATA REGISTER ($24) 42
7.6 A/D DURING WAIT MODE 42
7.7 A/D DURING STOP MODE 42

SECTION 8 16-BIT TIMER 43

8.1 TIMER 43
8.2 OUTPUT COMPARE 46
8.3 INPUT CAPTURES 48
8.4 TIMER CONTROL REGISTER (TCR) 49
8.4.1 ICIE2 - INPUT CAPTURE INTERRUPT ENABLE 2 50
8.4.2 ICIE1 - INPUT CAPTURE INTERRUPT ENABLE 1 50
8.4.3 TOIE - TIMER OVERFLOW INTERRUPT ENABLE 50
8.4.4 OCIE - OUTPUT COMPARE INTERRUPT ENABLE 50
8.4.5 TCMP/TCAP1 50
8.4.6 IEDG1 - INPUT CAPTURE EDGE SELECT 1 50
8.4.7 IEDG2 - INPUT CAPTURE EDGE SELECT 2 50
8.4.8 OLVL - OUTPUT COMPARE OUTPUT LEVEL SELECT .. 50
8.5 TIMER STATUS REGISTER (TSR) 50
8.5.1 ICF2 - INPUT CAPTURE 2 FLAG 51
8.5.2 ICF1 - INPUT CAPTURE 1 FLAG 51
8.5.3 TOF - TIMER OVERFLOW FLAG 51
8.5.4 OCF - OUTPUT COMPARE FLAG 51
8.6 TIMER OPERATION DURING WAIT/HALT MODES 52
8.7 TIMER OPERATION DURING STOP MODE 52

SECTION 9 PULSE WIDTH MODULATOR 53

9.1 PWM REGISTERS 54
9.2 PWM CONTROL BITS 55
9.2.1 POLx - POLARITY 55
9.2.2 Rx[3:0] - PWM OUTPUT RATE SELECT 56
9.2.3 CSx[3:1] - CHANNEL SELECT 57
9.2.4 MEx - MASK ENABLE 57
9.2.5 MSKx[3:1] - MASK 58
9.2.6 UPDATEx 58
9.3 PWM DATA BITS 59
9.4 PWM DURING RESETS 59
9.5 PWM OPERATION IN USER MODE 59
9.5.1 PWM DATA REGISTER 61
9.5.2 OPERATION WITH DIFFERENT PWM RATES 61
9.5.3 OPERATION WITH THE SAME PWM RATES 62
9.6 PWM DURING WAIT MODE 62
9.7 PWM DURING STOP MODE 63
9.8 APPLICATION EXAMPLES 63
9.8.1 DC BRUSH MOTOR INTERFACE 63
9.8.2 DC BRUSHLESS MOTOR INTERFACE 64

SECTION 10 SERIAL COMMUNICATIONS INTERFACE 67

10.1 INTRODUCTION 67
10.2 FEATURES 67
10.3 SCI DATA FORMAT 68
10.4 SCI OPERATION 68
10.4.1 TRANSMITTER 68
10.4.1.1 Character Length 68
10.4.1.2 Character Transmission 68
10.4.1.3 Break Characters 70
10.4.1.4 Idle Characters 70
10.4.1.5 Transmitter Interrupts 70
10.4.2 RECEIVER 70
10.4.2.1 Character Length 71
10.4.2.2 Character Reception 71
10.4.2.3 Receiver Wakeup 72
10.4.2.4 Receiver Noise Immunity 72
10.4.2.5 Framing Errors 72
10.4.2.6 Receiver Interrupts 72
10.5 SCI I/O REGISTERS 73
10.5.1 SCI Data Register (SCDR) 73
10.5.2 SCI Control Register 1 (SCCR1) 73
10.5.3 SCI Control Register 2 (SCCR2) 74
10.5.4 SCI Status Register (SCSR) 76
10.5.5 Baud Rate Register (BAUD) 78

SECTION 11 CORE TIMER 81

11.1 CTIMER CONTROL AND STATUS REGISTER (CTCSR) $08 82
11.1.1 CTOF - Core Timer Overflow Flag 82
11.1.2 RTIF - Real Time Interrupt Flag 82
11.1.3 CTOIE - Core Timer Overflow Interrupt Enable 82
11.1.4 RTIE - Real Time Interrupt Enable 82
11.1.5 RT1:RT0 - Real Time Interrupt Rate Select 83
11.2 COMPUTER OPERATING PROPERLY (COP) WATCHDOG RESET 83
11.3 CTIMER COUNTER REGISTER (CTCR) $09 83
11.4 CORE TIMER DURING WAIT MODE 84
11.5 CORE TIMER DURING STOP MODE 84

SECTION 12 ELECTRICAL SPECIFICATIONS 85

12.1 MAXIMUM RATINGS 85
12.2 THERMAL CHARACTERISTICS 85
12.3 DC ELECTRICAL CHARACTERISTICS 86
12.4 A/D CONVERTER CHARACTERISTICS 87
12.5 CONTROL TIMING 88

LIST OF FIGURES

Figure 1-1 Block Diagram 2
Figure 1-2 Oscillator Connections 4

Figure 2-1 User Mode Pinout 8
Figure 2-2 Programmer Interface to Host 9
Figure 2-3 Bootloader Flowchart 10
Figure 2-4 Programming Circuit 11
Figure 2-5 EPROM Programming Register 11
Figure 2-6 Mask Option Register 12
Figure 2-7 STOP/WAIT Flowcharts 15

Figure 3-1 Reset Block Diagram 17
Figure 3-2 COP Watchdog Timer Register 18

Figure 4-1 Interrupt Processing Flowchart 21
Figure 4-2 Interrupt Status and Control Register 23
Figure 4-3 Interrupt Hardware Structure 24

Figure 5-1 User Mode Memory Map 28
Figure 5-2 I/O and Control Registers Memory Map 29
Figure 5-3 I/O and Control Registers $0000-$000F 30
Figure 5-4 I/O and Control Registers $0010-$001F 31
Figure 5-5 I/O and Control Registers $0020-$002F 32

Figure 6-1 Port A I/O Circuitry 33
Figure 6-2 Line Interface Circuitry 34
Figure 6-3 Port B I/O Circuitry 35
Figure 6-4 Port C I/O Circuitry 35
Figure 6-5 Port D Circuitry 36

Figure 7-1 A/D Status and Control Register 40
Figure 7-2 A/D Data Register 42

Figure 8-1 16-Bit Timer Block Diagram 44
Figure 8-2 Timer Registers (TMRH / TMRL) 45
Figure 8-3 Alternate Counter Registers (ACRH / ACRL) 45
Figure 8-4 State Timing Diagram for Timer Overflow 45
Figure 8-5 Output Compare Registers (OCRH / OCRL) 46
Figure 8-6 Output Compare Software Initialization Example 47

Figure 8-7 State Timing Diagram for Output Compare 47
Figure 8-8 Input Capture Registers (ICRH1 / ICRL1)..... 48
Figure 8-9 Input Capture Registers (ICRH2 / ICRL2)..... 48
Figure 8-10 State Timing Diagram for Input Capture 49
Figure 8-11 Timer Control Register (TCR) 49
Figure 8-12 Timer Status Register (TSR)..... 51

Figure 9-1 PWM Block Diagram..... 53
Figure 9-2 PWM CTL Registers 54
Figure 9-3 PWMA & PWMB Data Registers 54
Figure 9-4 PWM Waveforms (POLx = 1) 55
Figure 9-5 PWM Waveforms (POLx = 0) 56
Figure 9-6 PWM Output MUX Logic..... 58
Figure 9-7 PWM Control Example..... 59
Figure 9-8 PWM Interlock Mechanisms 60
Figure 9-9 DC Brush Motor Interface 63
Figure 9-10 PWM Register Structure 66

Figure 10-1 SCI Data Format..... 68
Figure 10-2 SCI Transmitter..... 69
Figure 10-3 SCI Receiver..... 71
Figure 10-4 SCI Data Register (SCDR) 73
Figure 10-5 SCI Control Register 1 (SCCR1) 73
Figure 10-6 SCI Control Register 2 (SCCR2) 74
Figure 10-7 SCI Status Register (SCSR) 76
Figure 10-8 Baud Rate Register (BAUD) 78

Figure 11-1 Core Timer Block Diagram..... 81
Figure 11-2 Core Timer Control and Status Register (CTCSR) 82
Figure 11-3 Core Timer Counter Register..... 84

Figure 12-1 Power-On Reset and External Reset Timing Diagram 89

LIST OF TABLES

Table 2-1	Operating Mode Conditions after Reset	7
Table 2-2	Bootloader Functions	9
Table 2-3	COP Watchdog Timer Recommendations	16
Table 4-1	Vector Addresses for Interrupts and Reset	20
Table 6-1	Port A I/O Pin Functions	37
Table 6-2	Port B I/O Pin Functions	37
Table 6-3	Port C I/O Pin Functions	37
Table 6-4	Port D I/O Pin Functions	37
Table 7-1	A/D Channel Assignments	42
Table 9-1	PWM Rate Select Table	57
Table 9-2	DC Brush Motor Truth Table	63
Table 9-3	Brushless DC Motor Commutation Sequence	65
Table 10-1	Baud Rate Generator Clock Prescaling	78
Table 10-2	Baud Rate Selection	79
Table 10-3	Baud Rate Selection Examples	80
Table 11-1	RTI Rates	83
Table 11-2	Minimum COP Reset Times	83

SECTION 1 INTRODUCTION

The Motorola MC68HC705MC4 microcontroller is a low-cost M68HC05 Family EPROM microprocessor intended for use in industrial motor control and power supply applications. It features a 2-channel, 8-bit high speed PWM module (including a commutation mux for brushless permanent magnet motor control), a 6 input, 8-bit A/D to accommodate analog feedback signals, and an SCI to support multi-controller networking. This device is available in the 28-pin PDIP and 28-pin SOIC package.

1.1 FEATURES (all timing based on 3 MHz bus)

- Low cost, HC05 Core running at 3 MHz bus speed (at V_{DD} = 5 V +/- 10%)
- 28-pin PDIP, SOIC or windowed ceramic package. Low EMI emission pinout
- 3584 Bytes of User EPROM (including eight User Vectors of 2 Bytes each)
- 176 Bytes of User RAM
- Dual channel, high speed PWM featuring:
  - 8-bit resolution
  - Independent prescaler frequency selection and period counters
  - 2 frequency ranges, 9 steps in each:
    A: 183 Hz to 23.4 kHz, B: 122 Hz to 15.6 kHz
  - S/W programmable PWM polarity
  - Dual S/W controllable PWM output mux (each PWM to 3 I/O)
- 8-bit +/- 1LSB A/D converter with 6 input mux. High and low references. Conversion rate = 10.7 μS
- 16-bit timer with 2 input captures or 1 input capture plus 1 output compare. Resolution = 1.33 μS. Input Capture active edge S/W selectable as rising, falling, or both
- 15-stage multi-function Core Timer with timer overflow, real-time interrupt and watchdog
- Asynchronous Serial Communications Interface (SCI)
- 22 general purpose I/O lines, some shared with peripheral functions
- One 8-bit high source current I/O port (10 mA/pin @ V_{DD}-2.0 V, 20 mA max/port) (Port A)
- One high sink current (10mA @ 1.0 V) output pin (PB7)

- Mask, Request, Acknowledge, Edge and Sensitivity (Edge- and Level-Sensitive or Edge-Sensitive Only) control/status bits for IRQ Interrupt
- On-Chip Oscillator for Crystal/Ceramic Resonator
- Mask Selectable COP Watchdog System
- Power Saving STOP and WAIT Mode Instructions (Mask Selectable STOP Instruction Disable)
- Illegal Address Reset
- Steering Diode on $\overline{\text{RESET}}$ Pin to V_{DD}

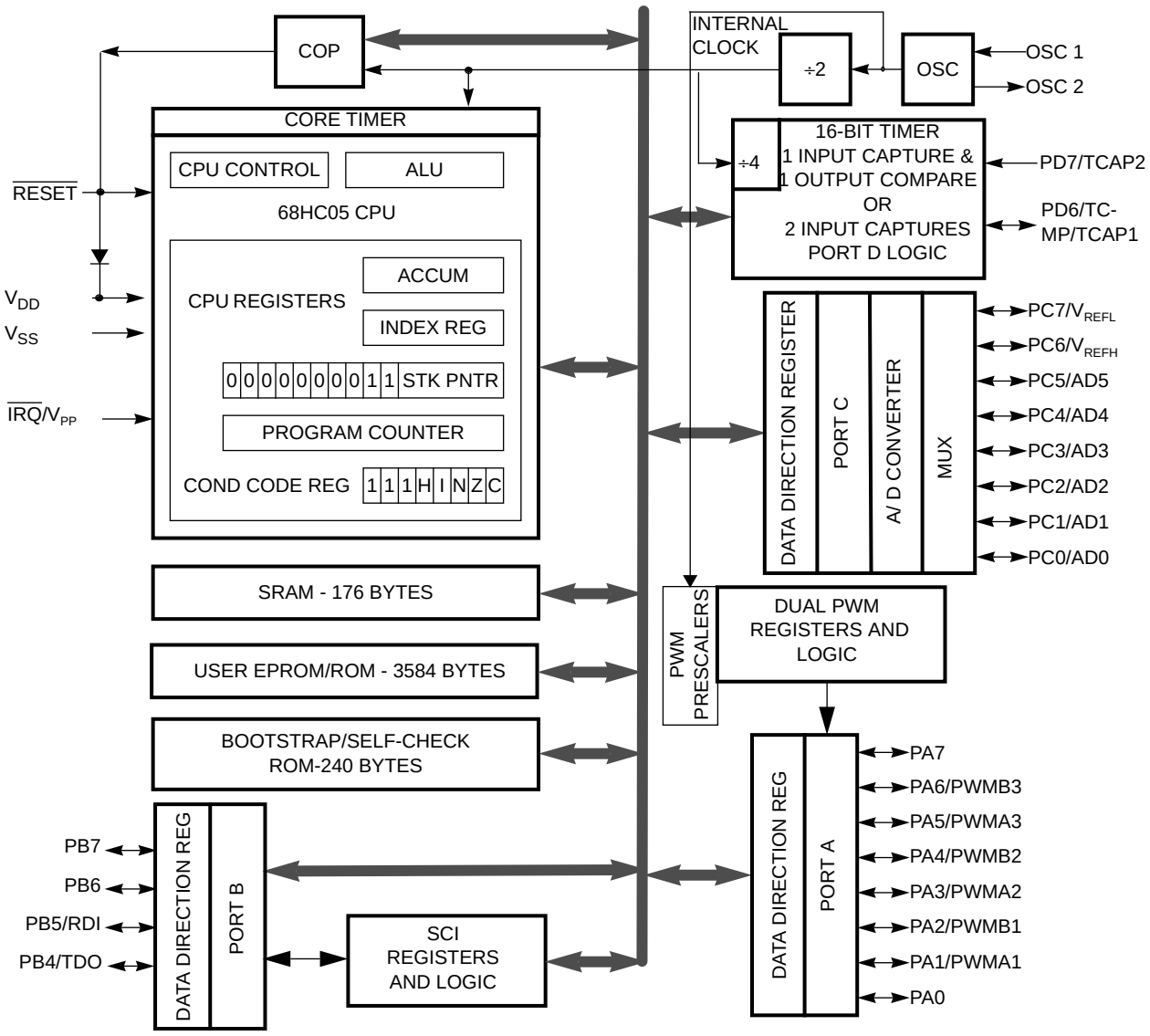


Figure 1-1: Block Diagram

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

1.2 MASK OPTIONS

There is one user selectable option on the MC68HC705. This option is provided through a bit within a Mask Option Register in the EPROM device, which is located and programmed at $0F00 as part of the EPROM array. This option will be hard wired in the ROM based device.

The normal state of this option and its alternate state are:

- COP Watchdog Timer Enabled. Option to Disable.

The ROM based device (MC68HC05MC4) will offer the following hard wired options:

- COP Watchdog Timer Enabled. Option to Disable.
- Stop Instruction Enabled. Option to Disable.

In the EPROM device, the STOP instruction will be disabled. These options are compatible with the typical application environment in which the device is expected to be used and will consequently allow the OTP devices to be used in production.

1.3 FUNCTIONAL PIN DESCRIPTION

The following paragraphs describe the functionality of each pin on the MC68HC705MC4 package. Pins connected to subsystems described in other chapters provide a reference to the chapter instead of a detailed functional description.

1.3.1 V_{DD} AND V_{SS}

Power is supplied to the MCU through V_{DD} and V_{SS}. V_{DD} is connected to a regulated +5 volt supply and V_{SS} is connected to ground. These pins are located close to each other for low Electro Magnetic Interference (EMI) emissions.

Very fast signal transitions occur on the MCU pins. The short rise and fall times place very 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. Use bypass capacitors with good high-frequency characteristics, and position them as close to the MCU as possible. Bypassing requirements vary, depending on how heavily the MCU pins are loaded.

1.3.2 OSC1, OSC2

These pins provide control input for an on-chip clock oscillator circuit. A crystal, a ceramic resonator or an external signal to these pins provides the system clock. The oscillator frequency is two times the internal bus rate.

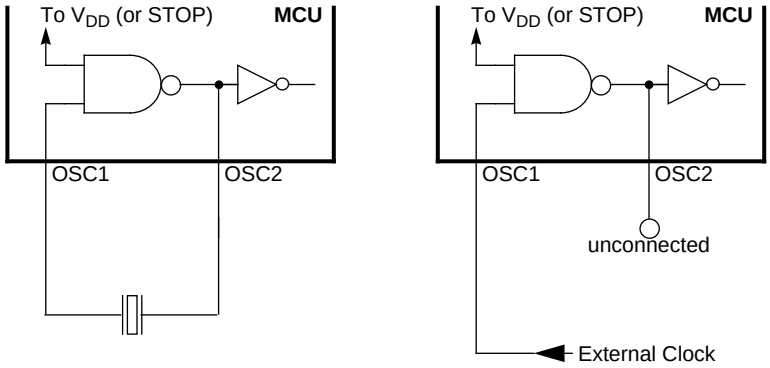
The OSC1 and OSC2 pins can accept the following:

- A crystal as shown in **Figure 1-2: Oscillator Connections(a)**
- A ceramic resonator as shown in **Figure 1-2: Oscillator Connections(a)**
- An external clock signal as shown in **Figure 1-2: Oscillator Connections(b)**

The frequency, f_{OSC}, of the oscillator or external clock source is divided by two to produce the internal bus clock operating frequency, f_{OP}. The oscillator cannot be turned off by software if the Stop disable option is enabled via Mask Option.

1.3.2.1 Crystal with Internal Components

The circuit in **Figure 1-2: Oscillator Connections(a)** shows a typical oscillator circuit for an AT-cut, parallel resonant crystal. Follow the crystal manufacturer's recommendations, as the crystal parameters determine the external component values required to provide maximum stability and reliable start-up. The load capacitance values used in the oscillator circuit design should include all stray capacitances. Mount the crystal and components as close as possible to the pins for start-up stabilization and to minimize output distortion.



(a) Crystal or Ceramic$^{\psi}$ Resonator Connections

(b) External Clock Source Connections

ψ: Additional capacitance may be required for Ceramic Resonator option. Follow the Ceramic Resonator manufacturer's recommendations.

Figure 1-2: Oscillator Connections

1.3.2.2 Ceramic Resonator

In cost-sensitive applications, use a ceramic resonator in place of a crystal. Use the circuit in **Figure 1-2: Oscillator Connections(a)** for a ceramic resonator and follow the resonator manufacturer's recommendations, as the resonator parameters determine the external component values required for maximum stability and reliable starting. The load capacitance values used in the oscillator circuit design should include all stray capacitances. Mount the resonator and components as close as possible to the pins for start-up stabilization and to minimize output distortion.

1.3.2.3 External Clock

An external clock from another CMOS-compatible device can be connected to the OSC1 input, with the OSC2 input not connected, as shown in **Figure 1-2: Oscillator Connections (b)**.

1.3.3 $\overline{\text{RESET}}$

Driving this input low will reset the MCU to a known start-up state. This pin can also be pulled low by internal resets. The $\overline{\text{RESET}}$ pin contains an internal Schmitt trigger to improve its noise immunity. Refer to **SECTION 3: RESETS**.

1.3.4 PA0, PA1/PWMA1, PA2/PWMB1, PA3/PWMA2, PA4/PWMB2, PA5/PWMA3, PA6/PWMB3, PA7

These eight I/O pins comprise Port A and are shared with the PWM subsystem. The state of any pin is software programmable and all Port A lines are configured as inputs during power-on or reset. All Port A pins have high source current capability to simplify interfacing to external devices, such as small triacs. Refer to **SECTION 6: INPUT/OUTPUT PORTS** and **SECTION 9: PULSE WIDTH MODULATOR**.

1.3.5 PB4/TDO, PB5/RDI, PB6, PB7

These four I/O pins comprise Port B. Two pins are shared with the SCI communication subsystem. The state of any pin is software programmable and all Port B lines are configured as inputs during power-on or reset. Refer to **SECTION 6: INPUT/OUTPUT PORTS** and **SECTION 10: SERIAL COMMUNICATIONS INTERFACE**.

1.3.6 PC0:5/AD0:5, PC6/V_{REFH}, PC7/V_{REFL}

These eight I/O pins comprise Port C and are shared with the A/D Converter subsystem. The state of any pin is software programmable and all Port C lines are configured as inputs during power-on or reset. Refer to **SECTION 6: INPUT/OUTPUT PORTS** and **SECTION 7: ANALOG SUBSYSTEM**.

1.3.7 PD6/TCAP1/TCMP, PD7/TCAP2

These two I/O pins comprise Port D and are shared with the 16-bit timer subsystem. PD7 is always an input. PD6 can be used as an input or output port if the TCAP1 interrupt is disabled and the TCAP1/TCMP bit is clear in the TCR. This is the state upon $\overline{\text{RESET}}$. Writes to PD7 have no effect. They may be read at any time, regardless of the mode of operation of the 16-bit timer. Refer to **SECTION 6: INPUT/OUTPUT PORTS** and **SECTION 8: 16-BIT TIMER**.

1.3.8 $\overline{\text{IRQ}}$ (MASKABLE INTERRUPT REQUEST) / V_{PP}

This pin has two different choices of interrupt triggering sensitivity through the IRQ bit in the Interrupt Status and Control Register (ISCR). The choices are:

- edge-sensitive triggering only.
- both edge-sensitive and level-sensitive triggering.

In addition, the $\overline{IRQ}$ pin may be selected to trigger and interrupt on either the rising or falling edge of the $\overline{IRQ}$ pin signal through the EDGE bit in the ISCR.

The MCU completes the current instruction before it responds to the interrupt request.

If the option is selected to include level-sensitive triggering, the IRQ input requires an external resistor to V_{DD} for “wire-OR” operation.

The $\overline{IRQ}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity. See **SECTION 4 INTERRUPTS**.

This pin is also used to supply the MC68HC705MC4 EPROM array with the programming voltage.

NOTE: If the voltage level applied to the IRQ pin exceeds V_{DD} it may affect the MCU’s mode of operation. See **SECTION 2: OPERATING MODES.**

1.4 CPU CORE

The MC68HC705MC4 uses a standard P-series CPU core. A description of the P-series instruction set can be found in *MC68HC05P4 Technical Data* (Motorola Publication MC68HC05P4/D).

SECTION 2 OPERATING MODES

The MC68HC705MC4 has two modes of operation that affect the pin-out and architecture of the MCU: User Mode and Bootloader (EPROM self-programming) Mode. The User Mode will normally be used, and the Bootloader mode is required for the special needs of EPROM programming. The MC68HC05MC4 also has two modes of operation that affect the pin-out and architecture of the MCU: User Mode and Selftest Mode. The User Mode will normally be used. The Selftest mode is required for special needs, though Selftest is specified for customer use.

The conditions required to enter each mode are shown in **Table 2-1: Operating Mode Conditions after Reset**. The mode of operation is determined by the voltages on the $\overline{\text{IRQ}}$ and PD7/TCAP2 pins on the rising edge of the external $\overline{\text{RESET}}$ pin.

The mode of operation is also determined after internal resets, which pull low the $\overline{\text{RESET}}$ pin. The mode of operation is determined at the subsequent rising edge of $\overline{\text{RESET}}$.

Table 2-1: Operating Mode Conditions after Reset

$\overline{\text{RESET}}$ Pin	IRQ/Vpp	PD7/TCAP2	MODE
	V_{SS} to V_{DD}	V_{SS} to V_{DD}	User
	V_{TST}	V_{DD}	Bootloader or Selftest

$V_{TST} = 2 \times V_{DD}$

2.1 USER MODE

The User Mode allows the MCU to function as a self-contained microcontroller, with maximum use of the pins for on-chip peripheral functions. All address and data activity occurs within the MCU and are not available externally. User Mode is entered on the rising edge of $\overline{\text{RESET}}$ if the $\overline{\text{IRQ}}$ pin is within the normal operating voltage range. The pinout for the User Mode is shown in **Figure 2-1: User Mode Pinout**.

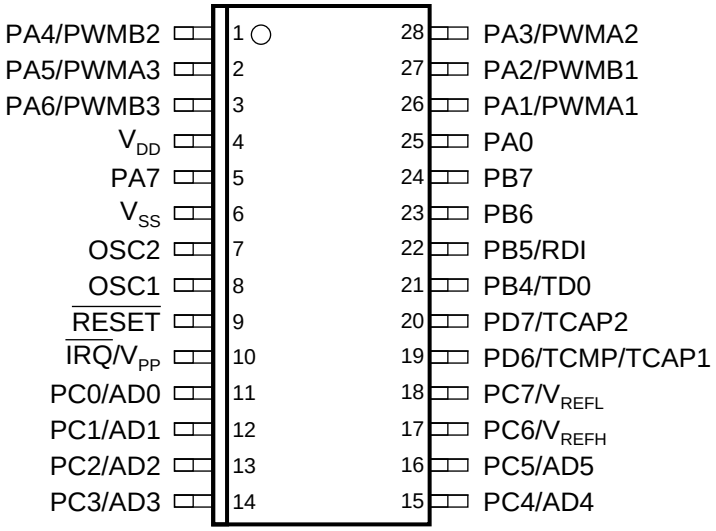


Figure 2-1: User Mode Pinout

In the User Mode, Port A shares 6 of its 8 I/O lines with the dual channel PWM subsystem. Port B shares 2 of its 4 I/O lines with the SCI subsystem. Port C shares all of its 8-bit I/O lines with the A/D subsystem. Port D shares its 2 I/O lines with the 16-bit timer subsystem.

2.2 BOOTLOADER MODE

Bootloader Mode is entered upon the rising edge of $\overline{RESET}$ if the $\overline{IRQ}/V_{PP}$ pin is at V_{PP} and the PD7/TCAP2 pin is at logic one. The Bootloader code resides in ROM between $0F01 to $0FEF. This program handles copying of user code from an external EPROM or host computer into the on-chip EPROM. **Figure 2-2: Programmer Interface to Host** shows the timing required to interface the device being programmed to a host. The bootloader performs one programming pass at t_{EPGM} per byte. When programming is complete, the bootloader code then performs a verify pass. Disable the COP hardware in bootloader mode.

The external user code addresses must correspond directly with the internal EPROM addresses.

2.2.1 Bootloader Functions

Three pins are used to select various bootloader functions. These pins are PC0, PC2, and PC3. PC0 is a SYNC pin, which is used to synchronize the MCU to an off-chip source driving EPROM data into the MCU. If an external EPROM is used, this pin must be connected to V_{SS}. PC2 and PC3 are used to select a programming mode. Two other pins, PC6 and PC7 are used to drive the PROG LED and the VERF LED respectively. The PC2, PC3 configurations required to enter the programming modes are shown in **Table 2-2: Bootloader Functions**. Alternatively, if PC1 is at a logic high after exiting reset, a jump to RAM at address $0050 will be executed instead of the bootstrap firmware.

Table 2-2: Bootloader Functions

PC0	PC2	PC3	MODE
SYNC	1	1	PROGRAM/VERIFY
SYNC	1	0	VERIFY ONLY
SYNC	0	0	DUMP EPROM

The bootloader uses an external 12-bit counter to address the external memory device containing the code to be copied. This counter requires a clock and a reset function and can address up to 4K bytes of memory.

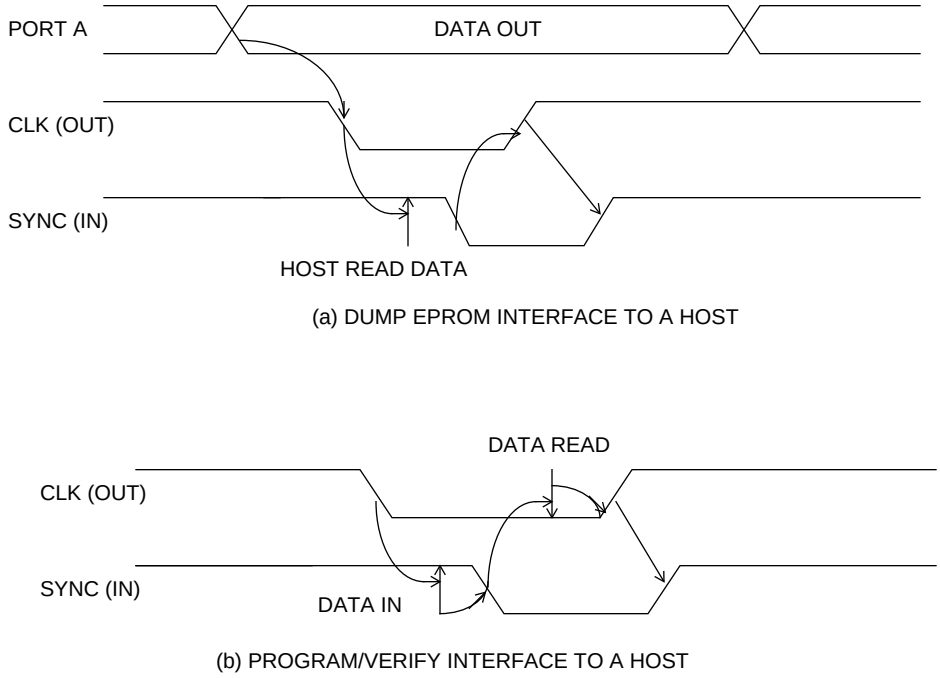


Figure 2-2: Programmer Interface to Host

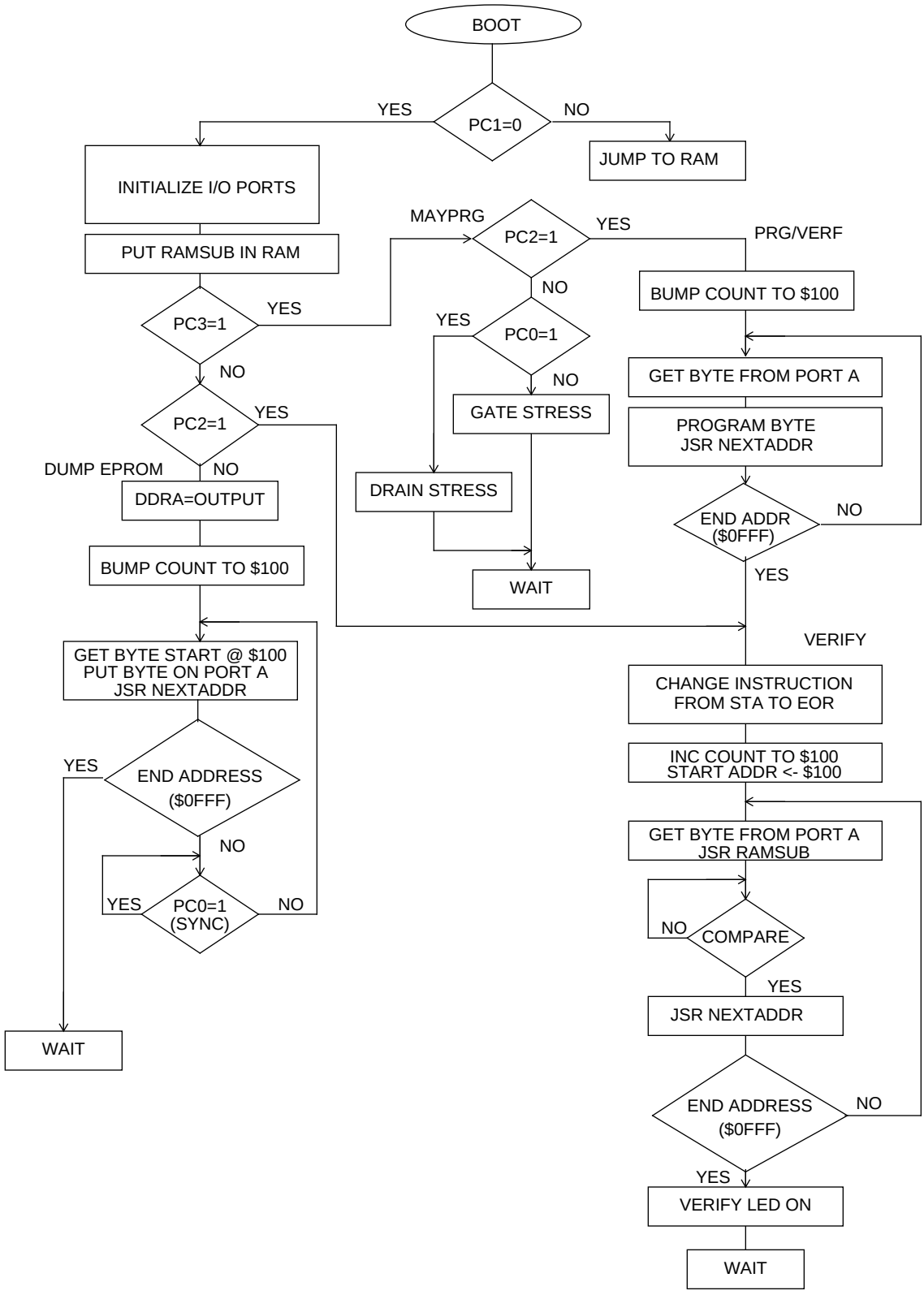


Figure 2-3: Bootloader Flowchart

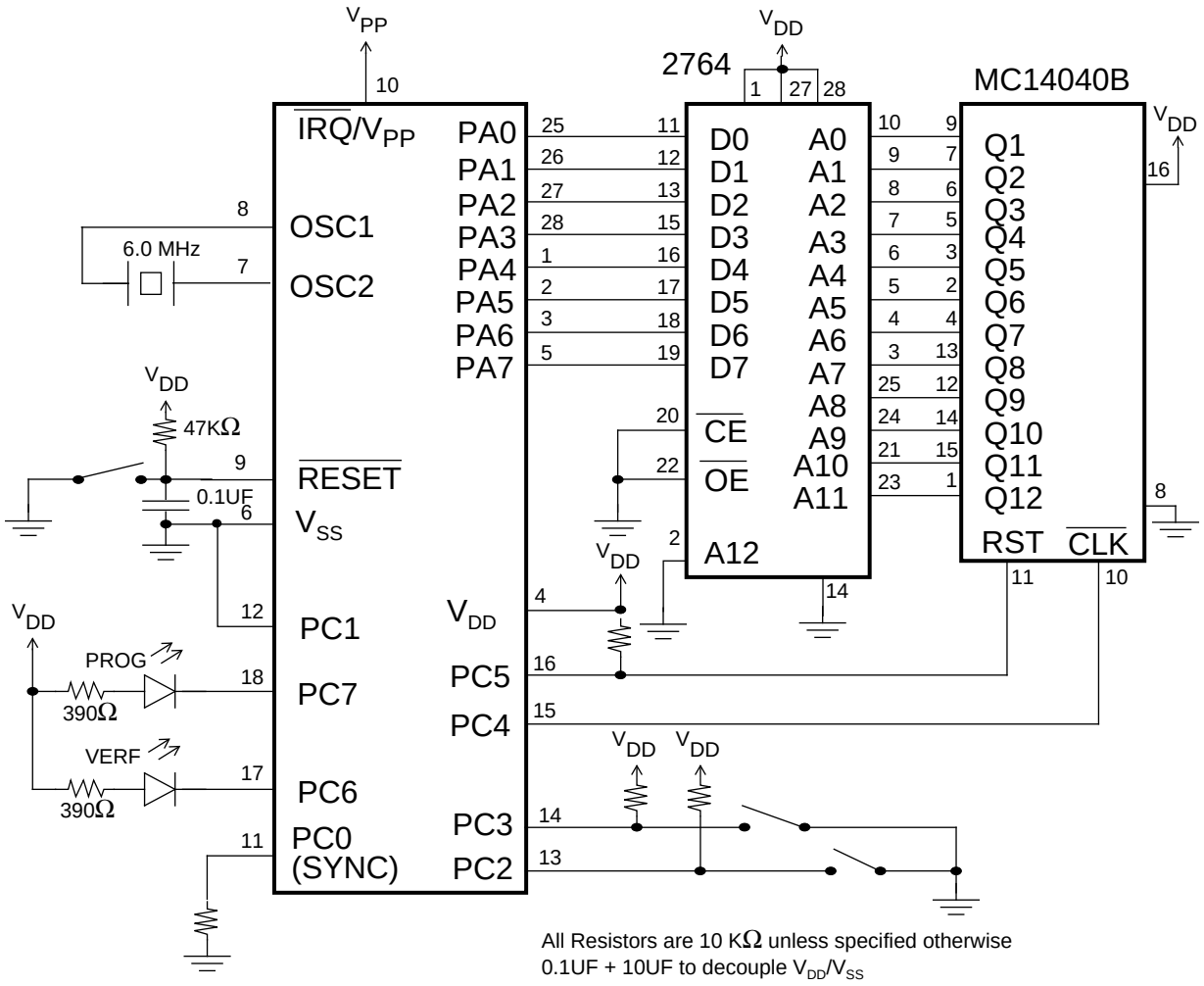


Figure 2-4: Programming Circuit

2.2.2 EPROM PROGRAMMING REGISTER

This register is used to program the EPROM array. To program a byte of EPROM, set LATCH, then write data to the desired address, then set EPGM for t_{EPGM}.

ADDR	REGISTER	READ WRITE	7	6	5	4	3	2	1	0
$0026	PROGRAMMING REGISTER BOOTSTRAP MODE	R	0	0	0	0	0	0	LATCH	EPGM
		W								

UNIMPLEMENTED

Figure 2-5: EPROM Programming Register

2.2.2.1 LATCH - EPROM Latch Control

The LATCH bit is a read/write bit. When set, the address and data buses are latched when a write to EPROM is done. EPROM cannot be read if LATCH = 1.

1 = EPROM address and data bus configured for programming.
0 = EPROM address and data bus configured for normal reads.

2.2.2.2 EPGM - EPROM Program Control

The EPGM bit may be read or cleared at any time. It may only be set if LATCH=1. If LATCH=0, the EPGM is automatically cleared. LATCH and EPGM cannot both be set on the same write.

1 = Programming power switched on to the EPROM array.
0 = Programming power switched off the EPROM array.

2.2.3 MASK OPTION REGISTER (MOR) $0F00

This register is latched upon RESET and at regular intervals as determined by COP timeout period. It is an EPROM byte located at $0F00 and holds the option bit for COP disable/enable.

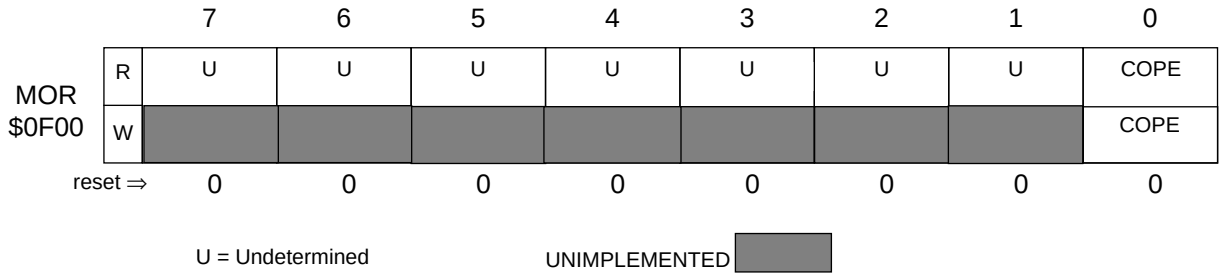


Figure 2-6: Mask Option Register

2.2.3.1 COP - COP enable/disable

1 = The COP is enabled.
0 = (erased state) The COP is disabled.

2.3 SELF-CHECK

The MC68HC05MC4 self-check operating mode will be defined in a subsequent revision of this specification.

2.4 LOW-POWER MODES

MC68HC705MC4 is capable of running in a low-power mode in each of its configurations. The WAIT and STOP instructions provide two modes that reduce the power required for the MCU by stopping various internal clocks and/or the on-chip oscillator. The STOP and WAIT instructions are not normally used if the COP Watchdog Timer is enabled. If the Stop Instruction is disabled, unintentional or otherwise execution of a STOP instruction will have no effect. The flow of the Stop and Wait modes is shown in **Figure 2-7: STOP/WAIT Flowcharts**.

NOTE: STOP is always disabled in the MC68HC705MC4.

2.4.1 STOP INSTRUCTION

The STOP instruction can result in one of two modes of operation depending on the Mask Option. If the Stop disable option is not chosen, the STOP instruction will behave like a normal STOP instruction in the MC68HC05 family and place the MCU in the Stop Mode. If the STOP disable option is chosen, the STOP instruction will be treated as a NOP instruction and will have no effect.

2.4.1.1 Stop Mode

Execution of the STOP instruction when enabled places the MCU in its lowest power consumption mode. In the Stop Mode, the internal oscillator is turned off, halting *all* internal processing, including the COP watchdog timer. Execution of the STOP instruction automatically clears the interrupt mask bit (I-bit) in the Condition Code Register so that the IRQ external interrupt is enabled. All other registers and memory remain unaltered. All input/output lines remain unchanged.

The MCU can be brought out of the Stop Mode only by an $\overline{\text{IRQ}}$ external interrupt or an externally generated $\overline{\text{RESET}}$. When exiting the Stop Mode the internal oscillator will resume after a 4064 bus clock cycle oscillator stabilization delay.

NOTE: If the IRQ mask bit (IRQM) in the Interrupt Control and Status Register is set before entering STOP mode, an active edge on IRQ will not bring the processor out of STOP.

NOTE: Execution of the STOP instruction with STOP enabled (via Mask Option) will cause the oscillator to stop, and therefore disable the COP watchdog timer. If the COP watchdog timer is to be used, the Stop mode should be disabled by selecting the appropriate Mask Option.

2.4.2 WAIT INSTRUCTION

The WAIT instruction places the MCU in a low-power mode, which consumes more power than the stop mode. In Wait mode, the bus clock is halted, suspending all processor and internal bus activity but the timer, PWM, A/D, SCI, and COP subsystems remain active. The user may optionally disable each individual subsystem through software before entering Wait mode in order to save more power. Execution of the WAIT instruction automatically clears the I-bit in the Condition Code Register enabling the $\overline{IRQ}$ external interrupt. All other registers, memory, and input/output lines remain in their previous state unless modified by an active peripheral.

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

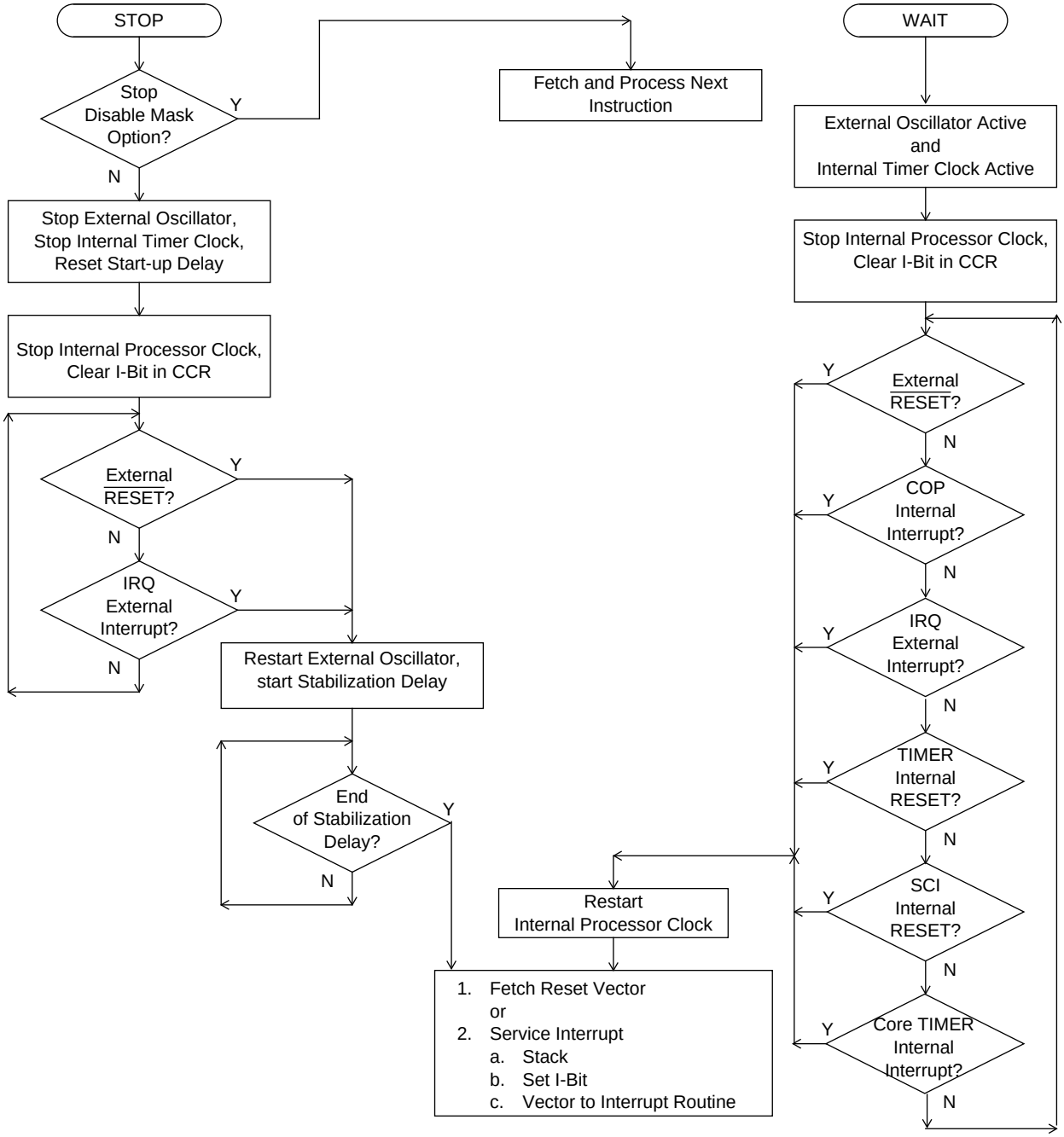


Figure 2-7: STOP/WAIT Flowcharts

2.5 COP WATCHDOG TIMER CONSIDERATIONS

The COP watchdog timer is active in single chip mode of operation when selected by Mask Option. Executing the STOP instruction when Stop mode is enabled (via Mask Option) will cause the COP to be disabled. Therefore, it is recommended that if the COP watchdog timer is to be used, the Stop mode should be disabled by selecting the appropriate Mask Option.

The COP watchdog timer should be disabled for applications that will use the WAIT mode with time periods that will exceed the COP time-out period.

COP watchdog timer interactions are summarized in **Table 2-3: COP Watchdog Timer Recommendations**.

Table 2-3: COP Watchdog Timer Recommendations

IF the following conditions exist:		THEN the COP Watchdog Timer should be:
STOP Instruction Mode	WAIT Period	
Stop Mode disabled via Mask Option	WAIT period *less than* COP time-out	Enable or disable COP via Mask Option
Stop Mode disabled via Mask Option	WAIT period *more than* COP time-out	Disable COP via Mask Option
Stop Mode enabled via Mask Option	any length WAIT period	Disable COP via Mask Option

SECTION 3 RESETS

The MCU can be reset from four sources: one external input and three internal reset conditions. The $\overline{\text{RESET}}$ pin is an input with a Schmitt trigger as shown in **Figure 3-1: Reset Block Diagram**. The CPU and all peripheral modules will be reset by the RST signal which is the logical OR of internal reset functions and is clocked by the internal bus clock. The $\overline{\text{RESET}}$ pin will also be pulled low by internal reset for 4 bus cycles.

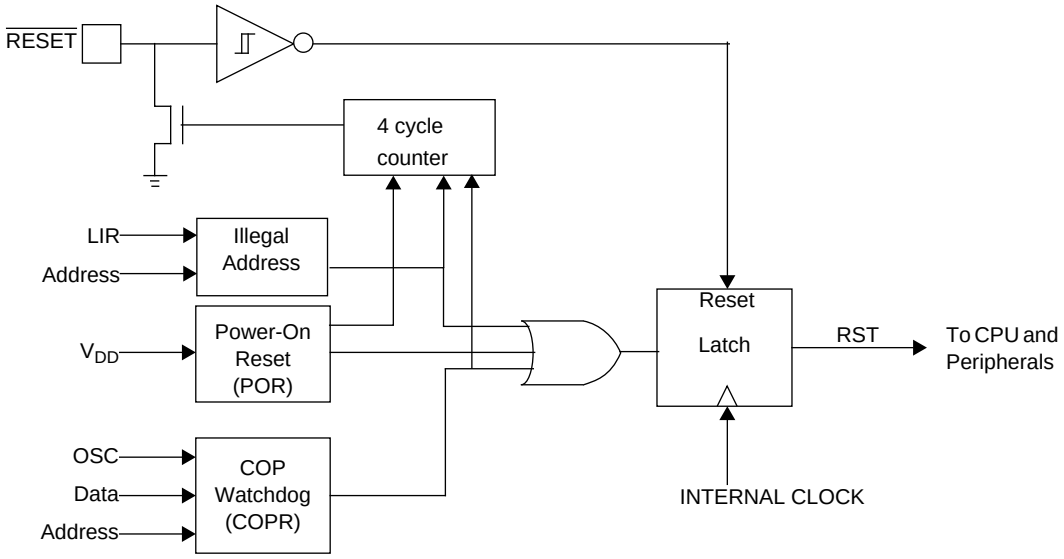


Figure 3-1: Reset Block Diagram

3.1 EXTERNAL RESET ($\overline{\text{RESET}}$)

The $\overline{\text{RESET}}$ input is the only external reset and is connected to an internal Schmitt trigger. The external reset occurs whenever the $\overline{\text{RESET}}$ input is driven below the lower threshold and remains in reset until the $\overline{\text{RESET}}$ pin rises above the upper threshold. The upper and lower thresholds are given in **SECTION 12: ELECTRICAL SPECIFICATIONS**.

3.2 INTERNAL RESETS

The three internally generated resets are the illegal address, the initial Power-On Reset (POR) function, and the COP watchdog timer function.

3.2.1 ILLEGAL ADDRESS RESET

When an opcode fetch occurs at an address that is not in the RAM or ROM/EPROM the part automatically resets. The part will also reset when an opcode fetch inadvertantly occurs at an address within the Self-Check/Bootstrap ROM while the device is in User mode.

3.2.2 POWER-ON RESET (POR)

An internal reset is generated on power-up to allow the internal clock generator to stabilize. The power-on reset is strictly for power turn-on conditions and should not be used to detect a drop in the power supply voltage. There is a 4064 internal processor clock cycle (t_{CYC}) oscillator stabilization delay after the oscillator becomes active. If the $\overline{\text{RESET}}$ pin is active at the end of this 4064 cycle delay, the MCU will remain in the reset condition until $\overline{\text{RESET}}$ goes inactive.

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

3.2.3 COMPUTER OPERATING PROPERLY (COP) RESET

When the COP watchdog timer is enabled (by Mask Option) the internal COP reset is generated automatically by a time-out of the COP watchdog timer. This timer is implemented as part of the Core Timer. See **SECTION 11: CORE TIMER**. The COP watchdog counter is cleared by writing a logical zero to bit zero at location $0FF0.

The COP watchdog timer can be disabled by Mask Option or by applying $2 \times V_{DD}$ to the $\overline{\text{IRQ}}$ pin at the rising edge of $\overline{\text{RESET}}$ (e.g. during Self-Check operation). When the $\overline{\text{IRQ}}$ pin is returned to its normal operating voltage range (between V_{SS} - V_{DD}) at the rising edge of $\overline{\text{RESET}}$, the COP watchdog timer's output will be restored if the COP Mask Option is enabled.

The COP register is shared with the MSB of the Core Timer Interrupt Vector as shown in **Figure 3-2: COP Watchdog Timer Register**. Reading this location will return the MSB of the Core Timer Interrupt Vector. Writing to this location will clear the COP watchdog timer.

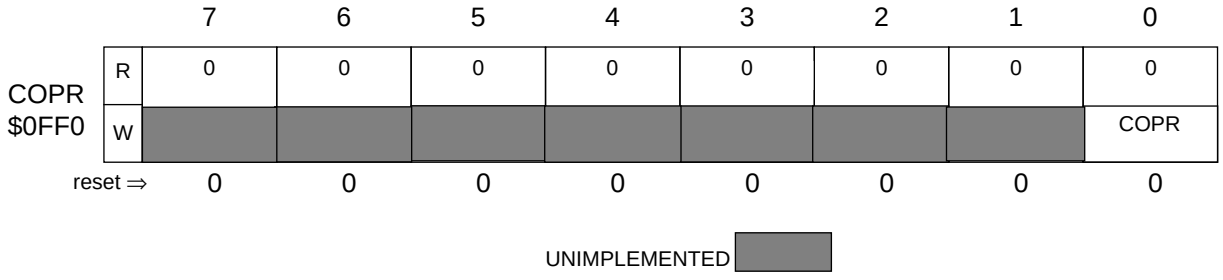


Figure 3-2: COP Watchdog Timer Register

SECTION 4 INTERRUPTS

The MCU can be interrupted eight different ways:

1. Nonmaskable Software Interrupt Instruction (SWI)
2. External Asynchronous Interrupt ($\overline{\text{IRQ}}$)
3. Input Capture Interrupt (TIMER)
4. Output Compare Interrupt (TIMER)
5. Timer Overflow Interrupt (TIMER)
6. Serial Communications Interrupt (SCI)
7. Core Timer Overflow Interrupt (CTIMER)
8. Real Time Interrupt (CTIMER)

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

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

When an interrupt is to be processed, the CPU puts the register contents on the stack, sets the I-bit in the CCR, and fetches the address of the corresponding interrupt service routine from the vector table at locations $0FF0 thru $0FFF. If more than one interrupt is pending when the interrupt vector is fetched, the interrupt with the highest vector location shown in **Table 4-1: Vector Addresses for Interrupts and Reset** will be serviced first.

An RTI instruction is used to signify when the interrupt software service routine is completed. The RTI instruction causes the CPU state to be recovered from the stack and normal processing to resume at the next instruction that was to be executed when the interrupt took place. **Figure 4-1: Interrupt Processing Flowchart** shows the sequence of events that occur during interrupt processing.

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

Table 4-1: Vector Addresses for Interrupts and Reset

Register	Flag Name	Interrupts	CPU Interrupt	Vector Address
N/A	N/A	Reset	RESET	$0FFE-$0FFF
N/A	N/A	Software	SWI	$0FFC-$0FFD
ISCR	REQ	External Interrupt	IRQ	$0FFA-$0FFB
TSR	ICF2	Timer Input Capture 2	TIMER	$0FF8-$0FF9
TSR	ICF1	Timer Input Capture 1	TIMER	$0FF6-$0FF7
TSR	OCF	Timer Output Compare *	TIMER	$0FF4-$0FF5
TSR	TOF	Timer Overflow *	TIMER	$0FF4-$0FF5
SCSR	Various	SCI	SCI	$0FF2-$0FF3
CTCSR	CTOF	Core Timer Overflow *	CTIMER	$0FF0-$0FF1
CTCSR	RTIF	Core Timer Real Time *	CTIMER	$0FF0-$0FF1

* Vector is shared

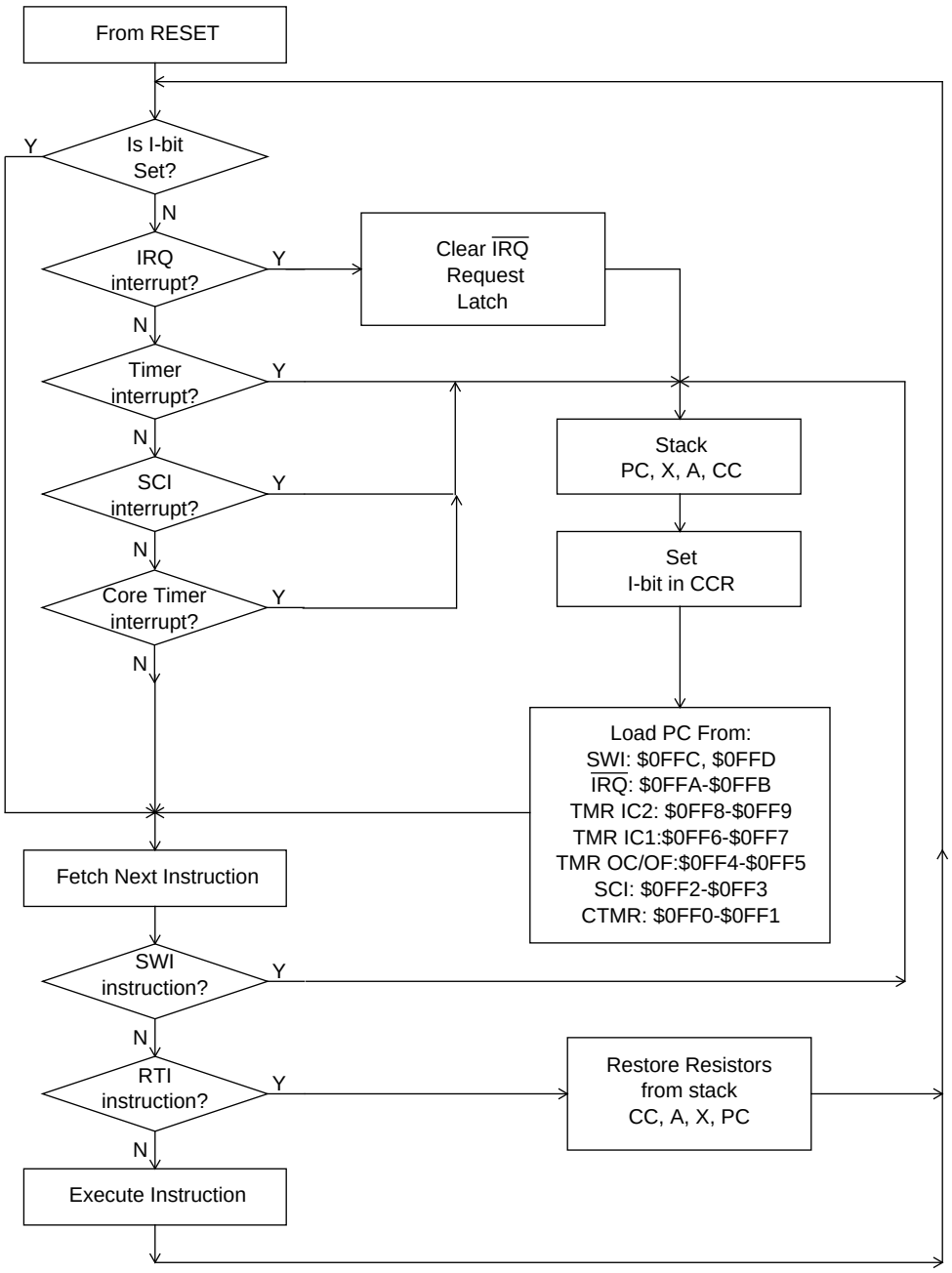


Figure 4-1: Interrupt Processing Flowchart

4.1 RESET INTERRUPT SEQUENCE

The RESET function is not in the strictest sense an interrupt; however, it is acted upon in a similar manner as shown in **Figure 4-1: Interrupt Processing Flowchart**. A low level input on the $\overline{\text{RESET}}$ pin or internally generated RST signal causes the program to vector to its starting address which is specified by the contents of memory locations $0FFE and $0FFF. The I-bit in the condition code register is also set. The MCU is configured to a known state during this type of reset, as previously described in **SECTION 3: RESETS**.

4.2 SOFTWARE INTERRUPT (SWI)

The SWI is an executable instruction. It is also a non-maskable interrupt since it is executed regardless of the state of the I-bit in the CCR. As with any instruction, interrupts pending during the previous instruction will be serviced before the SWI opcode is fetched. The interrupt service routine address for the SWI instruction is specified by the contents of memory locations $0FFC and $0FFD.

4.3 HARDWARE INTERRUPTS

All hardware interrupts are maskable by the I-bit in the CCR. If the I-bit is set, all hardware interrupts (internal and external) are disabled. Clearing the I-bit enables the hardware interrupts. There are four hardware interrupts that are explained in the following sections.

4.3.1 EXTERNAL INTERRUPT (IRQ)

If the interrupt mask bit (I-bit) of the CCR is set, all maskable interrupts (internal and external) are disabled. Clearing the I-bit enables interrupts (subject to their individual interrupt enable control flag status). IRQ now has an independent interrupt mask bit in the Interrupt Status and Control Register (ISCR) that must also be cleared to enable its corresponding interrupt.

The interrupt mask bit operates by inhibiting the interrupt signal ***after*** the appropriate interrupt request latch. This feature allows the interrupt to be recognized and latched even if the mask is set.

When the IRQ input goes to the active level for at least one t_{ILIH}, a logic one is latched internally to signify an interrupt has been requested. When the MCU completes its current instruction, the interrupt latch is tested. If the interrupt latch contains a logic one, and the interrupt mask bit (I-bit) in the condition code register and the IRQ mask bit (IRQM) in the ISCR are both clear, then the MCU can begin the interrupt sequence. The state of the interrupt latch is reflected in the interrupt request bit (REQ) in the ISCR, and is automatically cleared during interrupt processing. See **Figure 4-2: Interrupt Status and Control Register**.

IRQ interrupt requests are automatically acknowledged and cleared during interrupt processing. It may also be cleared through software by setting the acknowledge bit in the ISCR. Setting this bit is a “one-shot” operation and will not affect subsequent interrupt operation. The action of clearing the acknowledge bit will clear the request bit. This allows the programmer the option to cancel spurious interrupts that occur while the interrupt mask bits are set. This may be necessary in systems where it is desirable to prevent redundant (ghost) entries to the interrupt service routine (where the interrupt mask is eventually cleared).

Note that the IRQM is cleared (enabled) during reset, although no interrupts can occur until the interrupt mask bit of the CCR is cleared. It is set during reset. The interrupt request latches are also cleared during reset.

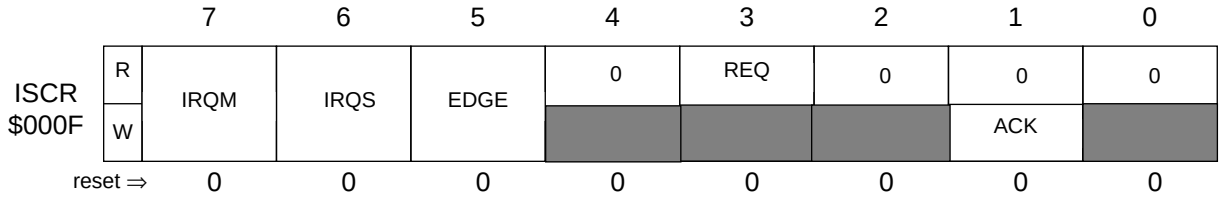


Figure 4-2: Interrupt Status and Control Register

4.3.1.1 IRQM - IRQ Enable Mask

The IRQM bit is a read/write bit that will disable the IRQ interrupt when set. IRQM is cleared by reset.

1 = IRQ interrupt request disabled
0 = IRQ interrupt request enabled

4.3.1.2 IRQS - IRQ Sensitivity

The IRQS bit is a read/write bit that will select whether the IRQ interrupt is edge-sensitive only or both edge-sensitive and level-sensitive. IRQS is cleared by reset.

1 = both edge-sensitive and level-sensitive
0 = edge-sensitive only

4.3.1.3 EDGE - IRQ Active Edge Select

The EDGE bit is a read/write bit that allows the user to select which edge, rising or falling, of the signal at the $\overline{\text{IRQ}}$ pin will generate an interrupt. Both rising and falling edge sensitivity may be achieved in software by toggling the EDGE bit from within the IRQ service routine. EDGE is cleared by reset.

1 = Rising edge IRQ interrupt
0 = Falling edge IRQ interrupt

4.3.1.4 REQ - IRQ Interrupt Request

The REQ bit is a read-only bit. The IRQ interrupt request bit and latch are cleared during IRQ exception processing. Therefore, one external IRQ interrupt pulse can be latched and subsequently serviced as soon as the I-bit is cleared. REQ will be cleared by reset.

1 = IRQ interrupt request pending
0 = No IRQ interrupt request pending

4.3.1.5 ACK - IRQ Interrupt Request Acknowledge

This bit is write only - it will always read as a zero. Writing a one to this bit will acknowledge the interrupt by clearing the corresponding interrupt request bit.

NOTE: The use of separate request and acknowledge bits allows the safe use of read-modify-write instructions (for example, BSET, BCLR) on the ISCR register.

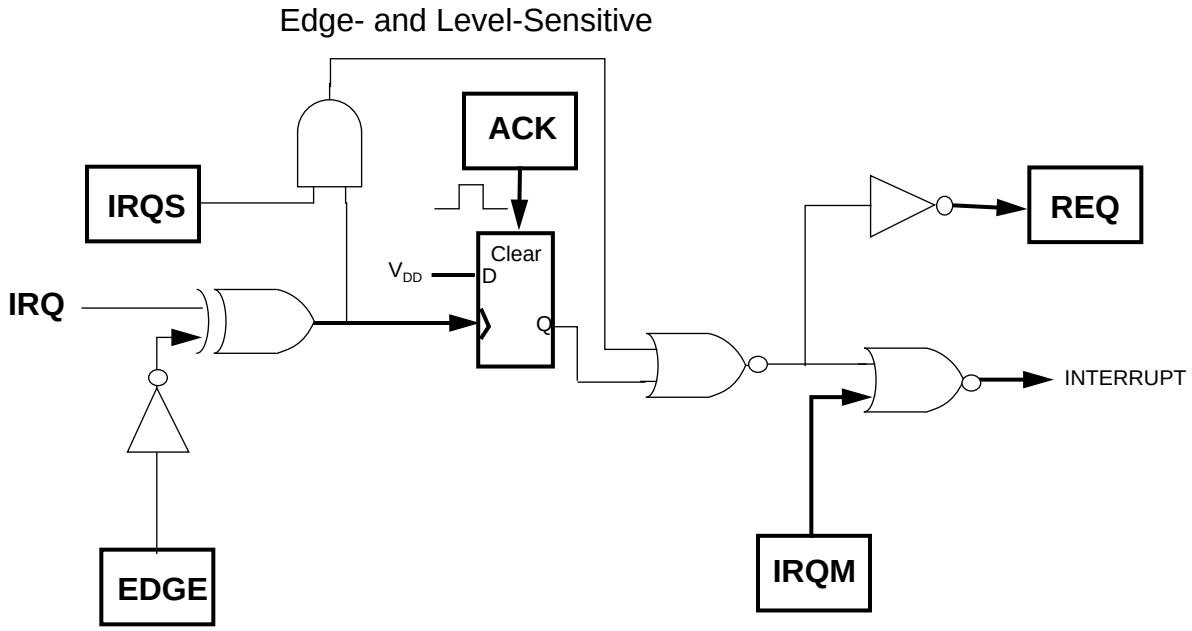


Figure 4-3: Interrupt Hardware Structure

NOTE: When the Edge- and Level-Sensitive Mask Option is selected, the voltage applied to the $\overline{IRQ}$ pin must return to the inactive state before the RTI instruction in the interrupt service routine is executed. If the $\overline{IRQ}$ pin remains in at the active level, the interrupt service routine will be re-entered after the RTI is executed. Setting the ACK bit will have no effect under these circumstances.

4.3.2 TIMER INTERRUPTS

4.3.2.1 INPUT CAPTURE INTERRUPTS

The input capture interrupts are generated by the 16-bit timer as described in **SECTION 8: 16-BIT TIMER**. The input capture interrupt flags are located in register TSR and the corresponding enable bits can be found in register TCR. The I-bit in the CCR must be clear in order for either input capture interrupt to be enabled. The interrupt service routine

addresses are specified by the contents of memory locations $0FF8 and $0FF9 for input capture 2, and by the contents of memory locations $0FF6 and $0FF7 for input capture 1.

4.3.2.2 OUTPUT COMPARE INTERRUPT

The output compare interrupt is generated by the 16-bit timer, as described in **SECTION 8: 16-BIT TIMER**. The output compare interrupt flag is located in register TSR and its corresponding enable bit can be found in register TCR. The I-bit in the CCR must be clear in order for the output compare interrupt to be enabled. The interrupt service routine address is specified by the contents of memory locations $0FF4 and $0FF5.

4.3.2.3 TIMER OVERFLOW INTERRUPT

The timer overflow interrupt is generated by the 16-bit timer as described in **SECTION 8: 16-BIT TIMER**. The timer overflow interrupt flag is located in register TSR and its corresponding enable bit can be found in register TCR. The I-bit in the CCR must be clear in order for the timer overflow interrupt to be enabled. This internal interrupt will vector to the interrupt service routine located at the address specified by the contents of memory locations $0FF4 and $0FF5. The timer overflow and the output compare function share the same interrupt vector, thus requiring the user to poll interrupt request flags.

4.3.3 SCI INTERRUPT

There are five different SCI interrupt flags that cause an SCI interrupt whenever they are set and enabled. The interrupt flags are in the SCI Status Register (SCSR), and the enable bits are in the SCI Control Register 2 (SCCR2). Any of these interrupts will vector to the same interrupt service routine, located at the address specified by the contents of memory location $0FF2 and $0FF3. See **SECTION 10: SERIAL COMMUNICATIONS INTERFACE**.

4.3.4 CORE TIMER (CTIMER) INTERRUPT

There are two different Core timer interrupt flags that cause a CTIMER interrupt whenever they are set and enabled. The interrupt flags and enable bits are located in the CTIMER Control and Status Register (CTCSR). Any of these interrupts will vector to the same interrupt service routine, located at the address specified by the contents of memory location $0FF0 and $0FF1. See **SECTION 11: CORE TIMER**.

THIS PAGE INTENTIONALLY LEFT BLANK

SECTION 5 MEMORY

The MC68HC705MC4 utilizes 12 address lines to access an internal memory space of 4K bytes. This memory space is divided into I/O, RAM, and EPROM/ROM areas.

5.1 USER MODE MEMORY MAP

When the MC68HC05MC4 is in the User Mode, the 48 bytes of I/O, 176 bytes of RAM, 3584 bytes of User ROM, 240 bytes of Self-Check ROM, and 16 bytes of User Vectors ROM are all active, as shown in **Figure 5-1: User Mode Memory Map**. When the MC68HC705MC4 is in the User Mode, the 48 bytes of I/O, 176 bytes of RAM, 3584 bytes of User EPROM, 240 bytes of Bootstrap ROM, and 16 bytes of User Vectors EPROM are all active, as shown in **Figure 5-1: User Mode Memory Map**.

The Mask Option (MOR) and EPROM Programming (EPROG) registers are unique to the MC68HC705MC4 and are therefore unimplemented in the MC68HC05MC4. The MOR resides at address $0F00 (first byte of the bootstrap code area) and the EPROG register resides at $0026.

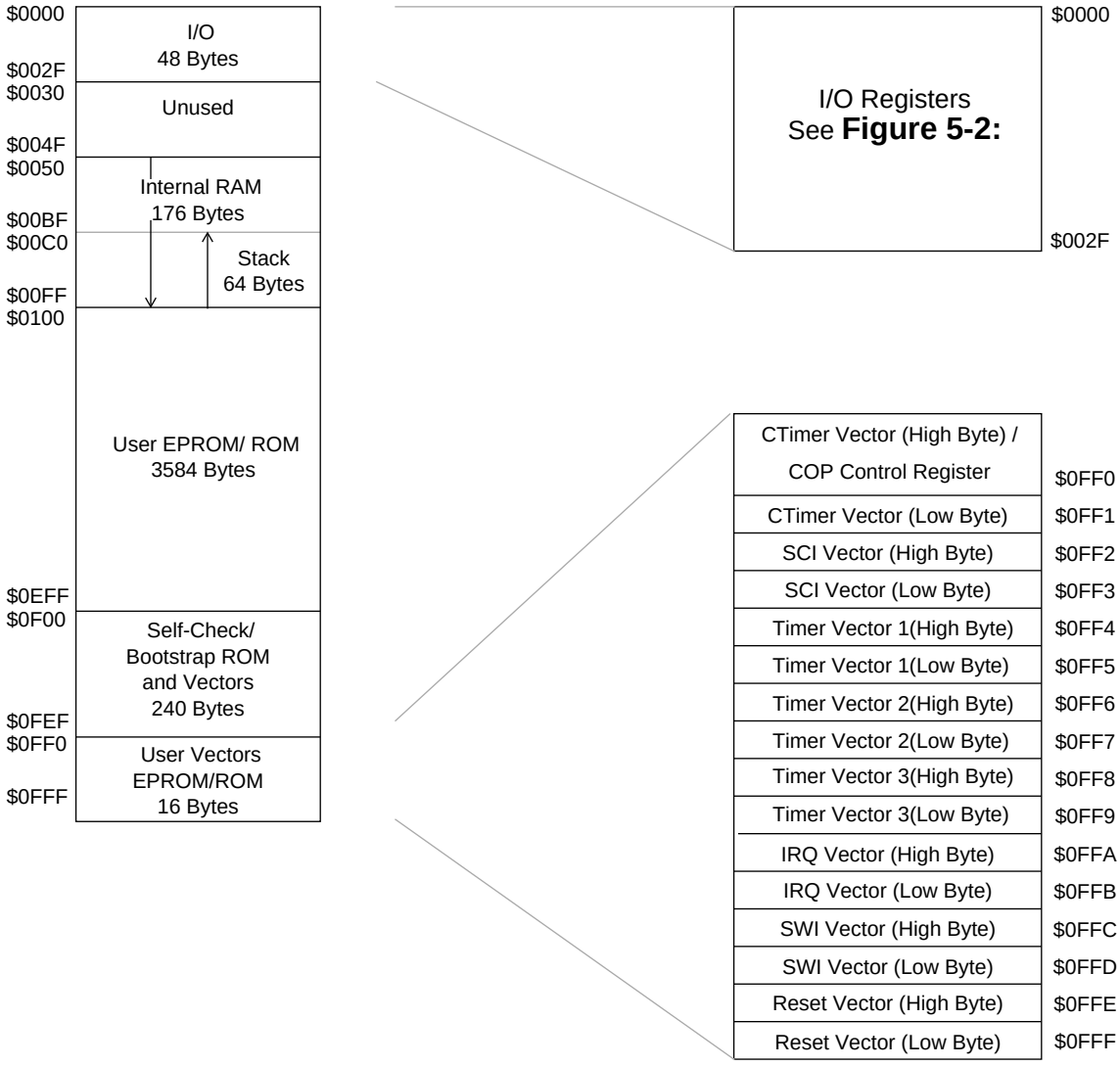


Figure 5-1: User Mode Memory Map

5.2 BOOTSTRAP/SELF-CHECK MODE MEMORY MAP

Memory space is identical to the User Mode, as shown in **Figure 5-1: User Mode Memory Map**.

5.3 I/O AND CONTROL REGISTERS

Figure 5-2: through **Figure 5-5:** briefly describe the I/O and Control Registers at locations $0000-$002F. Reading unimplemented bits will return unknown states, and writing unimplemented bits will be ignored.

Port A Data Register	$0000
Port B Data Register	$0001
Port C Data Register	$0002
Port D Data Register	$0003
Port A Data Direction Register	$0004
Port B Data Direction Register	$0005
Port C Data Direction Register	$0006
Port D Data Direction Register	$0007
Core Timer Control & Status Register	$0008
Core Timer Counter Register	$0009
SCI BAUD	$000A
SCI CONTROL 1	$000B
SCI CONTROL 2	$000C
SCI STATUS	$000D
SCI DATA	$000E
IRQ STATUS/CONTROL	$000F
PWMA Data-Direct	$0010
PWMA Data-Interlock	$0011
PWMB Data-Direct	$0012
PWMB Data-Interlock	$0013
PWM CTL-A	$0014
PWM CTL-B	$0015
PWM Rate	$0016
Timer Control Register	$0017
Timer Status Register	$0018
Input Capture2 MSB	$0019
Input Capture2 LSB	$001A
Input Capture1 MSB	$001B
Input Capture1 LSB	$001C
Output Compare MSB	$001D
Output Compare LSB	$001E
Reserved for Test	$001F
Timer MSB	$0020
Timer LSB	$0021
Alternate Counter MSB	$0022
Alternate Counter LSB	$0023
A/D Converter Data Register	$0024
A/D Control & Status Register	$0025
EPROM Program Register **	$0026
PWM Update Register	$0027
UNIMPLEMENTED	$0028
	$002F

** EPROM device only, UNIMPLEMENTED on ROM device

Figure 5-2: I/O and Control Registers Memory Map

Register	ADDR	R/W	7	6	5	4	3	2	1	0
Port A	$0000	R	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
Port A Data Register		W								
Port B	$0001	R	PB7	PB6	PB5	PB4	0	0	0	0
Port B Data Register		W								
Port C	$0002	R	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Port C Data Register		W								
Port D	$0003	R	PD7	PD6	0	0	0	0	0	0
Port D Data Register		W								
Port A Data Direction Register	$0004	R	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		W								
Port B Data Direction Register	$0005	R	DDRB7	DDRB6	DDRB5	DDRB4	0	0	0	0
		W								
Port C Data Direction Register	$0006	R	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
		W								
Port D Data Direction Register	$0007	R		DDRD6						
		W								
Core Timer Control & Status Register	$0008	R	CTOF	RTIF	CTOIE	RTIE	0	0	RT1	RT0
		W								
Core Timer Counter Register	$0009	R	CTCR7	CTCR6	CTCR5	CTCR4	CTCR3	CTCR2	CTCR1	CTCR0
		W								
SCI BAUD	$000A	R	0	0	SCP1	SCP0	0	SCR2	SCR1	SCR0
		W								
SCI CONTROL 1	$000B	R	R8	T8	0	M	WAKE	0	0	0
		W								
SCI CONTROL 2	$000C	R	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
		W								
SCI STATUS	$000D	R	TDRE	TC	RDRF	IDLE	OR	NF	FE	0
		W								
SCI DATA	$000E	R	SC7	SC6	SC5	SC4	SC3	SC2	SC1	SC0
		W								
IRQ STATUS/CONTROL	$000F	R	IRQM	IRQS	EDGE	0	REQ	0	0	0
		W							ACK	

UNIMPLEMENTED

Figure 5-3: I/O and Control Registers $0000-$000F

Register	ADDR	R/W	7	6	5	4	3	2	1	0
PWMA Data (Effective)	$0010	R	PWMA7	PWMA6	PWMA5	PWMA4	PWMA3	PWMA2	PWMA1	PWMA0
PWMA Data-Direct		W								
PWMA Data (Effective)	$0011	R								
PWMA Data-Interlock		W								
PWMB Data (Effective)	$0012	R	PWMB7	PWMB6	PWMB5	PWMB4	PWMB3	PWMB2	PWMB1	PWMB0
PWMB Data-Direct		W								
PWMB Data (Effective)	$0013	R								
PWMB Data-Interlock		W								
PWM CTL-A (Effective)	$0014	R	MEA	POLA	MSKA3	MSKA2	MSKA1	CSA3	CSA2	CSA1
PWM CTL-A (Buffer)		W								
PWM CTL-B (Effective)	$0015	R	MEB	POLB	MSKB3	MSKB2	MSKB1	CSB3	CSB2	CSB1
PWM CTL-B (Buffer)		W								
PWM Rate (Effective)	$0016	R	RA3	RA2	RA1	RA0	RB3	RB2	RB1	RB0
PWM Rate (Buffer)		W								
Timer Control Register	$0017	R	ICIE2	ICIE1	TOIE	OCIE	TCMP/ TCAP1	IEDG1	IEDG2	OLVL
		W								
Timer Status Register	$0018	R	ICF2	ICF1	TOF	OCF	0	0	0	0
		W								
Input Capture2MSB	$0019	R	IC2H7	IC2H6	IC2H5	IC2H4	IC2H3	IC2H2	IC2H1	IC2H0
		W								
Input Capture2LSB	$001A	R	IC2L7	IC2L6	IC2L5	IC2L4	IC2L3	IC2L2	IC2L1	IC2L0
		W								
Input Capture1MSB	$001B	R	IC1H7	IC1H6	IC1H5	IC1H4	IC1H3	IC1H2	IC1H1	IC1H0
		W								
Input Capture1LSB	$001C	R	IC1L7	IC1L6	IC1L5	IC1L4	IC1L3	IC1L2	IC1L1	IC1L0
		W								
Output CompareMSB	$001D	R	OCH7	OCH6	OCH5	OCH4	OCH3	OCH2	OCH1	OCH0
		W								
Output CompareLSB	$001E	R	OCL7	OCL6	OCL5	OCL4	OCL3	OCL2	OCL1	OCL0
		W								
Reserved for Test	$001F	R								
		W								

UNIMPLEMENTED RESERVED FOR TEST

Figure 5-4: I/O and Control Registers $0010-$001F

Register	ADDR	R/W	7	6	5	4	3	2	1	0
Timer MSB	$0020	R	TMRH7	TMRH6	TMRH5	TMRH4	TMRH3	TMRH2	TMRH1	TMRH0
		W								
Timer LSB	$0021	R	TMRL7	TMRL6	TMRL5	TMRL4	TMRL3	TMRL2	TMRL1	TMRL0
		W								
Alternate Counter MSB	$0022	R	ACRH7	ACRH6	ACRH5	ACRH4	ACRH3	ACRH2	ACRH1	ACRH0
		W								
Alternate Counter LSB	$0023	R	ACRL7	ACRL6	ACRL5	ACRL4	ACRL3	ACRL2	ACRL1	ACRL0
		W								
A/D Converter Data Register	$0024	R	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
		W								
A/D Control/Status Register	$0025	R	COCO	ADRC	ADON	0	CH3	CH2	CH1	CH0
		W								
EPROM Prog Register **	$0026	R	0	0	0	0	0	0	LATCH	EPGM
		W								
PWM Force Register	$0027	R								
		W	ForceA	ForceB						
UNIMPLEMENTED	$0028-$002F	R								
		W								

** EPROM device only, UNIMPLEMENTED on ROM device

Figure 5-5: I/O and Control Registers $0020-$002F

5.4 RAM

The user RAM consists of 176 bytes (including the stack) at locations $0050 thru $00FF. The stack begins at address $00FF. The stack pointer can access 64 bytes of RAM from $00FF to $00C0.

NOTE: Using the stack area for data storage or temporary work locations requires care to prevent it from being overwritten due to stacking from an interrupt or subroutine call.

5.5 ROM/EPROM

There are 3584 bytes of user ROM/EPROM at locations $0100 thru $0EFF and 16 additional bytes for user vectors at locations $0FF0 thru $0FFF. The Self-Check/Bootstrap ROM and vectors are at locations $0F00 thru $0FEF.

SECTION 6 INPUT/OUTPUT PORTS

In the User Mode, there are 20 bidirectional I/O lines arranged as two 8-bit I/O ports (Ports A and C), one 4-bit I/O port (Port B), and one 2-bit Input port (Port D). These ports are programmable as either inputs or outputs, except Port D7, under software control of the data direction registers (DDRs).

6.1 PORT A

Port A is an 8-bit bidirectional port that shares PA1-PA6 with the PWM subsystem. (See **Figure 6-1: Port A I/O Circuitry**.) The Port A data register is located at address $0000 and its Data Direction Register (DDR) is located at address $0004. Reset does not affect the data registers, but clears the DDRs, thereby setting all of the port pins to input mode. Writing a one to a DDR bit sets the corresponding port pin to output mode. PA1-PA6 may be used for general I/O applications when the PWM subsystem is disabled. PA0-7 feature larger P-channel output devices and are capable of sourcing more current than a standard port (refer to **SECTION 12: ELECTRICAL SPECIFICATIONS**).

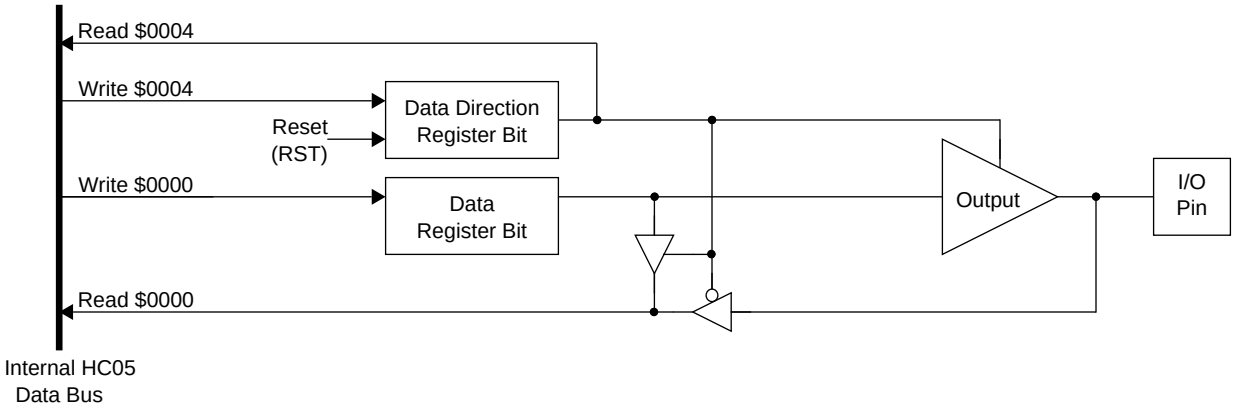


Figure 6-1: Port A I/O Circuitry

6.1.1 MCU LINE INTERFACE RECOMMENDATIONS

It is expected that some applications appropriate to the MC68HC705MC4 will be required to determine the presence of line (mains) voltage using the MCU. A low-cost MCU line interface is shown in **Figure 6-2: Line Interface Circuitry**. PA7 has been intentionally located between V_{DD} and V_{SS} to provide a lowest possible impedance path for the injected currents in particular fast transients. Although any I/O port will function in this manner, it is recommended that only PA7 be used for such an interface. The positive and negative excursions of the input voltage relative to the neutral return are clamped between V_{SS} and V_{DD} using the internal parasitic input diodes. The series resistor limits the injected currents to the specified value. (Refer to **SECTION 12 ELECTRICAL SPECIFICATIONS**.) The resistor value must be calculated based upon maximum expected transient voltage levels

(i.e., not line peak values). Care should be taken to ensure parasitic series resistor and PCB capacitance will not couple transients to the MCU input.

PA7 has been intentionally located between V_{DD} and V_{SS} to provide a lowest possible impedance path for the injected currents, in particular fast transients. Although any I/O port will function in this manner, it is recommended that only PA7 be used for such an interface.

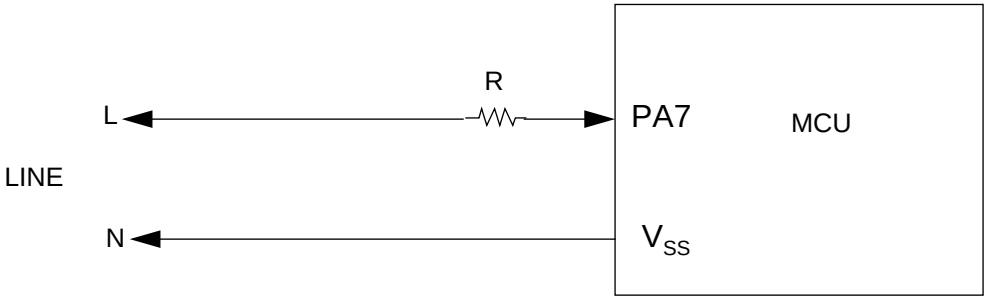


Figure 6-2: Line Interface Circuitry

6.2 PORT B

Port B is a 4-bit bidirectional port that can share pins PB4-PB5 with the SCI subsystem. The Port B data register is located at address $0001 and its Data Direction Register (DDR) is located at address $0005. Reset does not affect the data registers, but clears the DDRs, thereby setting all of the port pins to input mode. Writing a one to a DDR bit sets the corresponding port pin to output mode.

PB7 features a larger n-channel output device and can therefore sink more current than a standard port. (Refer to **SECTION 12: ELECTRICAL SPECIFICATIONS**.)

PB4-PB5 may be used for general I/O applications when the SCI subsystem is disabled. When the SCI subsystem is enabled, Port B registers are still accessible to software.

Writing to either of the Port B registers could corrupt the SCI data. See **SECTION 10: SERIAL COMMUNICATIONS INTERFACE** for a discussion of the SCI subsystem.

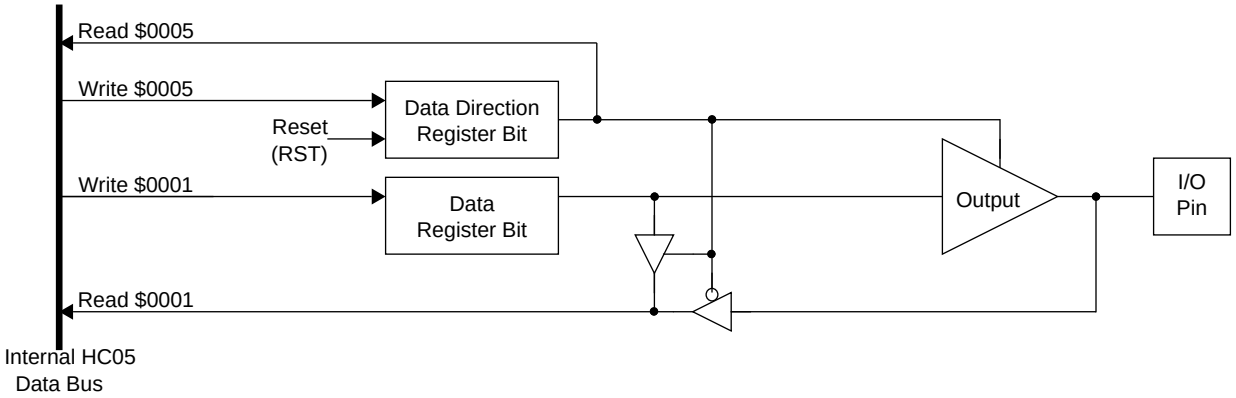


Figure 6-3: Port B I/O Circuitry

6.3 PORT C

Port C is an 8-bit bidirectional port that has shared pins with the A/D subsystem. The Port C data register is located at address $0002 and its Data Direction Register (DDR) is located at address $0006. Reset does not affect the data registers, but clears the DDRs, thereby setting all of the port pins to input mode. Writing a one to a DDR bit sets the corresponding port pin to output mode. (See **Figure 6-4: Port C I/O Circuitry**.)

The ADON bit in register ADSC is used to enable/disable the A/D subsystem. Port C may be used for general I/O applications when the A/D subsystem is disabled or when all A/D input channels are not required. Unselected channels revert to general purpose I/O. See **SECTION 7: ANALOG SUBSYSTEM**.

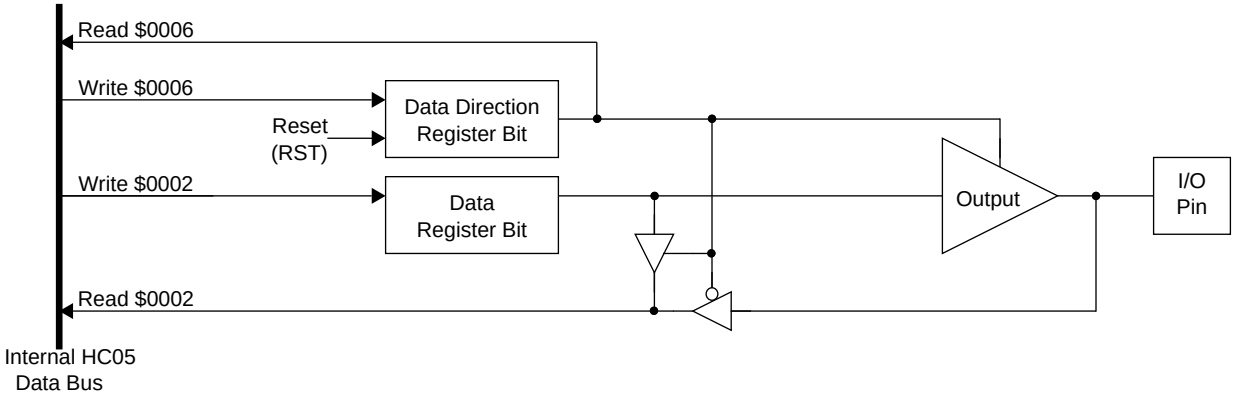


Figure 6-4: Port C I/O Circuitry

6.4 PORT D

Port D is a 2-bit port. PD7 and PD6 are shared with the 16-bit timer. PD6 is a bi-directional I/O pin but PD7 is an input only pin. The Port D data register is located at address $0003 and its Data Direction Register (DDR) is located at address $0007. Reset clears the DDR setting PD6 to an input but does not affect the data registers.

PD6 may be used for general I/O applications when the SCI subsystem is disabled. PD7 may be used as a general purpose input when the SCI subsystem is disabled. When the SCI subsystem is enabled, Port D registers are still accessible to software. Writing to either of the Port D registers with the timer enabled could interfere with timer operation. See **SECTION 8: 16-BIT TIMER** for a discussion of the timer subsystem.

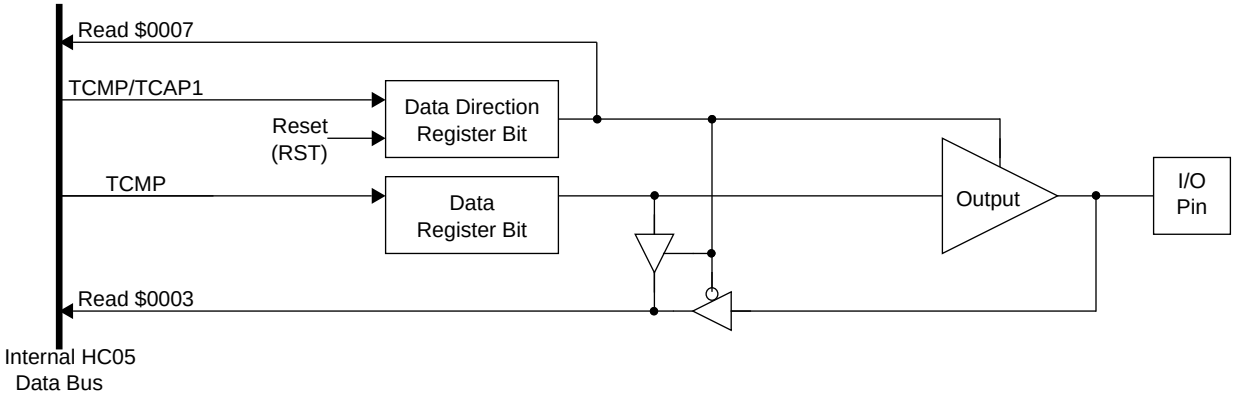


Figure 6-5: Port D Circuitry

6.5 I/O PORT PROGRAMMING

Each pin on Ports A-C may be programmed as an input or an output under software control as shown in **Table 6-1: Port A I/O Pin Functions**, **Table 6-2: Port B I/O Pin Functions**, **Table 6-3: Port C I/O Pin Functions**, and **Table 6-4: Port D I/O Pin Functions**. The direction of a pin is determined by the state of its corresponding bit in the associated port Data Direction Register (DDR). A pin is configured as an output if its corresponding DDR bit is set to a logic one. A pin is configured as an input if its corresponding DDR bit is cleared to a logic zero.

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

Table 6-1: Port A I/O Pin Functions

DDRA	I/O Pin Mode	Accesses to DDRA @ $0004	Accesses to Data Register @ $0000	
		Read/Write	Read	Write
0 1	IN, Hi-Z OUT	DDRA0-7 DDRA0-7	PA0-7 PA0-7	* PA0-7

***** Does not affect input, but stored to Data Register

Table 6-2: Port B I/O Pin Functions

DDRB	I/O Pin Mode	Accesses to DDRB @ $0005	Accesses to Data Register @ $0001	
		Read/Write	Read	Write
0 1	IN, Hi-Z OUT	DDRB4-7 DDRB4-7	PB4-7 PB4-7	* PB4-7

***** Does not affect input, but stored to Data Register

Table 6-3: Port C I/O Pin Functions

DDRC	I/O Pin Mode	Accesses to DDRC @ $0006	Accesses to Data Register @ $0002	
		Read/Write	Read	Write
0 1	IN, Hi-Z OUT	DDRC0-7 DDRC0-7	PC0-7 PC0-7	* PC0-7

***** Does not affect input, but stored to Data Register

Table 6-4: Port D I/O Pin Functions

DDRD	I/O Pin Mode	Accesses to DDRD @ $0007	Accesses to Data Register @ $0003	
		Read/Write	Read	Write
0 1	IN, Hi-Z OUT	DDRD6 DDRD6	PD6-7 PD6	* PD6

***** Does not affect input, but stored to Data Register

NOTE: To avoid generating a glitch on an I/O port pin, data should be written to the I/O port data register before writing a logical one to the corresponding Data Direction Register.

THIS PAGE INTENTIONALLY LEFT BLANK

SECTION 7 ANALOG SUBSYSTEM

The MC68HC705MC4 includes a 6-channel, 8-bit, multiplexed input, successive approximation A/D converter, with six of the inputs available on external pins and four additional internal channels.

7.1 ANALOG SECTION

7.1.1 RATIOMETRIC CONVERSION

The A/D is ratiometric, with two dedicated pins supplying the reference voltages (V_{REFH} and V_{REFL}). An input voltage equal to V_{REFH} converts to $FF (full scale) and an input voltage equal to V_{REFL} converts to $00. An input voltage greater than V_{REFH} will convert to $FF with no overflow indication. For ratiometric conversions, the source of each analog input should use V_{REFH} as the supply voltage and be referenced to V_{REFL}.

7.1.2 V_{REFH} and V_{REFL}

The reference supply for the converter uses two dedicated pins rather than being driven by the system power supply lines because the voltage drops in the bonding wires of those heavily loaded pins would degrade the accuracy of the A/D conversion. V_{REFH} and V_{REFL} are internally wired to the analog supply voltages AV_{DD} and AV_{SS}. These pins are located next to each other to permit optimal decoupling.

7.1.3 ACCURACY AND PRECISION

The 8-bit conversions shall be accurate to within $\pm 1^1/_2$ LSB including quantization.

7.2 CONVERSION PROCESS

The A/D reference inputs are applied to a precision internal digital-to-analog converter. Control logic drives this D/A and the analog output is successively compared to the selected analog input that was sampled at the beginning of the conversion time. The conversion process is monotonic and has no missing codes.

7.3 DIGITAL SECTION

7.3.1 CONVERSION TIMES

Each channel of conversion takes 32 clock cycles, which must be at a frequency equal to or greater than 1 MHz.

7.3.2 MULTI-CHANNEL OPERATION

In User Mode, a multiplexer allows the single A/D converter to select one of eight analog signals, two of which are V_{REFH} and V_{REFL}. The eight pins of Port C are input signals to the multiplexer.

7.3.3 UNUSED A/D INPUTS AS I/O

When the A/D system is enabled, two pins, V_{REFH} (PC6) and V_{REFL}(PC7), are automatically assumed to have their dedicated functions. The Channel Select Bits define which Port C pin will be used as the analog in pin and overrides any control from the Port C I/O logic by forcing that pin as the input to the analog circuitry. The Port C pins that are not selected by the Channel Select Bits, [CH3:0], are controlled by the Port C I/O logic, and thus can be used as general purpose I/O. Writes to Port C will not have any effect on the selected channel.

NOTE: The DDR bits corresponding to an A/D channel used by the application must be cleared. For example, AD2 shares a pin with PC2, so the DDRC2 bit must be cleared (unless this is the only channel the A/D ever selects). This is to ensure that the port output value held in the Port C data register is not driven out of the pin when the A/D has selected another channel for conversion.

7.4 A/D STATUS AND CONTROL REGISTER (ADSCR) $25

The following paragraphs describe the function of the A/D Status and Control Register.

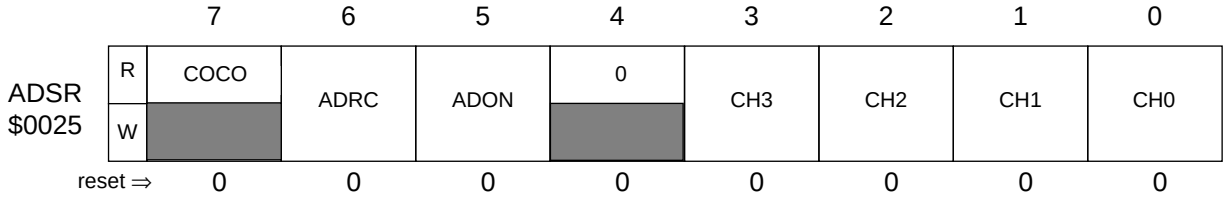


Figure 7-1: A/D Status and Control Register

7.4.1 COCO - CONVERSIONS COMPLETE

This read-only status bit is set when a conversion is completed, indicating that the A/D Data Register contains valid results. This bit is cleared whenever the A/D Status and Control Register is written and a new conversion is automatically started, or whenever the A/D Data Register is read. Once a conversion has been started by writing to the A/D Status and Control Register, conversions of the selected channel will continue every 32 cycles until the A/D Status and Control Register is written again. In this continuous conversion mode, the A/D Data Register will be filled with new data, and the COCO bit set, every 32 cycles. Data from the previous conversion will be overwritten regardless of the state of the COCO bit prior to writing.

7.4.2 ADRC - A/D RC OSCILLATOR CONTROL

When the RC oscillator is selected (ADRC = 1) to be the A/D clock source, it requires a time t_{ADRC} to stabilize. Results can be inaccurate during this time. If the CPU clock is running below 1 Mhz, the RC oscillator must be used.

When ADRC=0, the A/D uses the CPU clock.

7.4.3 ADON - A/D ON

When the A/D is turned on (ADON = 1), it requires a time t_{ADON} for the current sources to stabilize, and results can be inaccurate during this time. This bit turns on the charge pump. If the ADRC is set, clearing this bit disables the RC oscillator to save power.

7.4.4 CH3:CH0 - CHANNEL SELECT BITS

CH3, CH2, CH1, and CH0 form a 4-bit field, which is used to select one of eight A/D channels. Channels 0-5 correspond to Port C input pins on the MCU. Channels 8-a are used for internal reference points. In User Mode, channel b is reserved and converts to $00. The following table shows the signals selected by the channel select field.

Using a Port C pin as both an analog and digital input simultaneously is prohibited to prevent excess power dissipation. When the A/D is enabled (ADON = 1) and one of the channels 0-5 is selected, the corresponding Port C pin will appear as a logic zero to a digital read. The remaining Port C pins (0-5) will read normally. To digitally read all eight Port C pins simultaneously, the A/D must be disabled (ADON = 0).

Table 7-1: A/D Channel Assignments

CHANNEL	SIGNAL
0	AD0 PORT C BIT 0
1	AD1 PORT C BIT 1
2	AD2 PORT C BIT 2
3	AD3 PORT C BIT 3
4	AD4 PORT C BIT 4
5	AD5 PORT C BIT 5
6	UNUSED
7	UNUSED
8	V_{REFH}
9	V_{REFL}
a	$(V_{REFH} + V_{REFL})/2$
b-f	V_{REFL}

7.5 A/D DATA REGISTER ($24)

One 8-bit result register is provided. This register is updated each time COCO is set. Reset has no effect on this register.

ADDR $24	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

Figure 7-2: A/D Data Register

7.6 A/D DURING WAIT MODE

The A/D continues normal operation during WAIT mode. To decrease power consumption during WAIT, it is recommended that both the ADON and ADRC bits in the A/D Status and Control Register be cleared if the A/D converter is not being used. If the A/D converter is in use and the system clock rate is above 1.0 MHz, it is recommended that the ADRC bit be cleared.

7.7 A/D DURING STOP MODE

In STOP mode, the comparator and charge pump are turned off and the A/D ceases to function. Any pending conversion is aborted. When the clocks begin oscillation upon leaving the STOP mode, a finite amount of time passes before the A/D circuits stabilize enough to provide conversions to the specified accuracy. Normally, the delays built into the MC68HC705MC4 when coming out of STOP mode are sufficient for this purpose, therefore no explicit delays need to be built into the software.

SECTION 8 16-BIT TIMER

The MC68HC705MC4 MCU contains a single 16-bit programmable timer with two Input Capture functions and an Output Compare function. The 16-bit timer is driven by the output of a fixed divide-by-four prescaler operating from the internal clock. The 16-bit timer may be used for many applications including input waveform measurement while simultaneously generating an output waveform. Pulse widths can vary from microseconds to seconds depending on the oscillator frequency selected. The 16-bit timer is also capable of generating periodic interrupts. See **Figure 8-1: 16-Bit Timer Block Diagram**.

Because the timer has a 16-bit architecture, each function is represented by two registers. Each register pair contains the high and low byte of that function. Generally, accessing the low byte of a specific timer function allows full control of that function; however, an access of the high byte inhibits that specific timer function until the low byte is also accessed.

NOTE: The I-bit in the Condition Code Register (CCR) should be set while manipulating both the high and low byte registers of a specific timer function. This prevents interrupts from occurring between the time that the high and low bytes are accessed.

8.1 TIMER

The key element of the programmable timer is a 16-bit free-running counter, or Timer Registers, preceded by a prescaler which divides the internal clock by four. The prescaler gives the timer a resolution of 1.33 microseconds when a 3 MHz crystal is used. The counter is incremented to increasing values during the low portion of the internal clock cycle.

The double byte free-running counter can be read from either of two locations: the Timer Registers (TMRH, TMRL) or the Alternate Counter Registers (ACRH, ACRL). Both locations will contain identical values. A read sequence containing only a read of the LSB of the counter (TMRL / ACRL) will return the count value at the time of the read. If a read of the counter accesses the MSB first (TMRH / ACRH) it causes the LSB (TMRL / ACRL) to be transferred to a buffer. This buffer value remains fixed after the first MSB byte read even if the MSB is read several times. The buffer is accessed when reading the counter LSB (TMRL / ACRL), and thus completes a read sequence of the total counter value. When reading either the Timer or Alternate Counter registers, if the MSB is read, the LSB must also be read to complete the read sequence. See **Figure 8-2: Timer Registers (TMRH / TMRL)** and **Figure 8-3: Alternate Counter Registers (ACRH / ACRL)**.

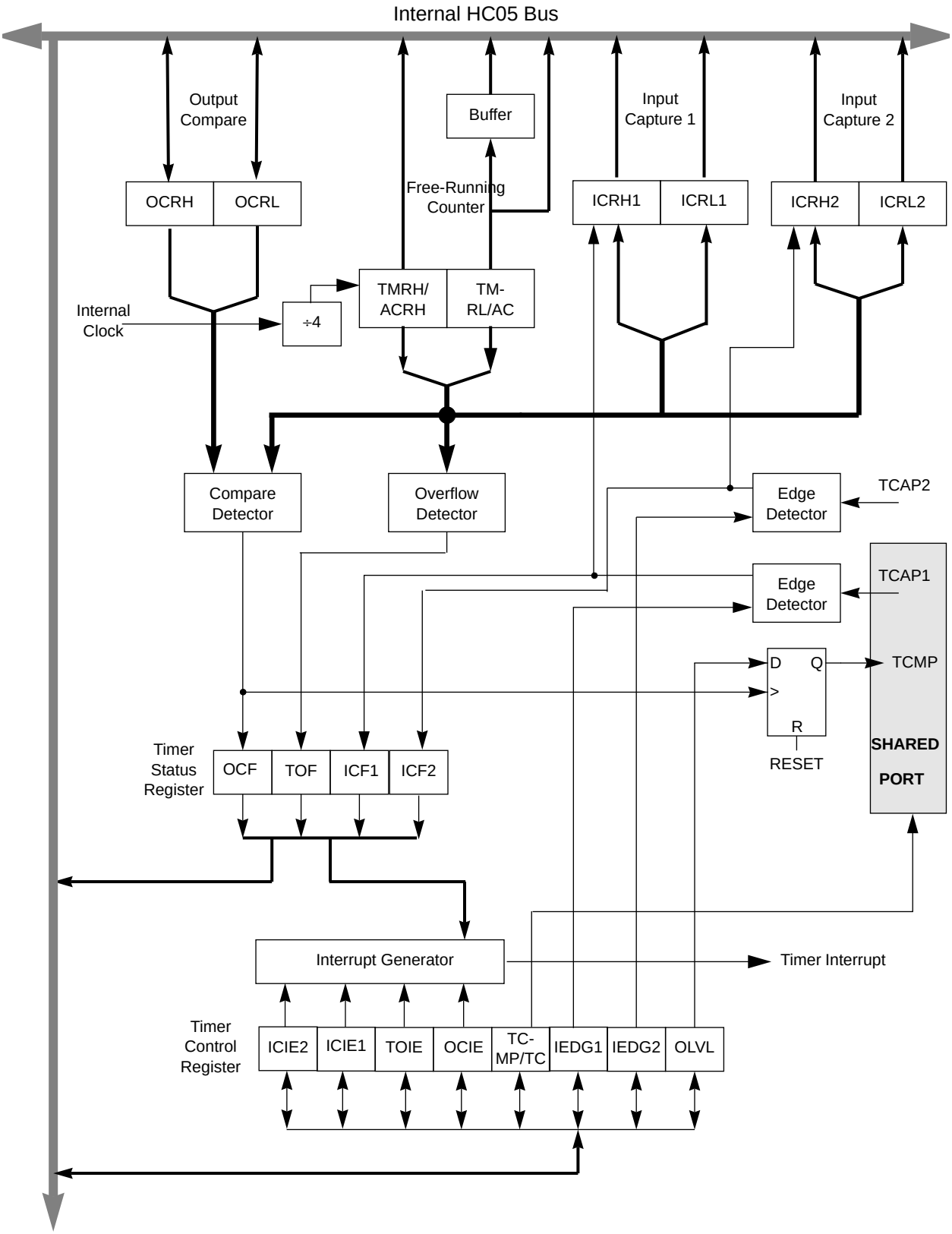


Figure 8-1: 16-Bit Timer Block Diagram

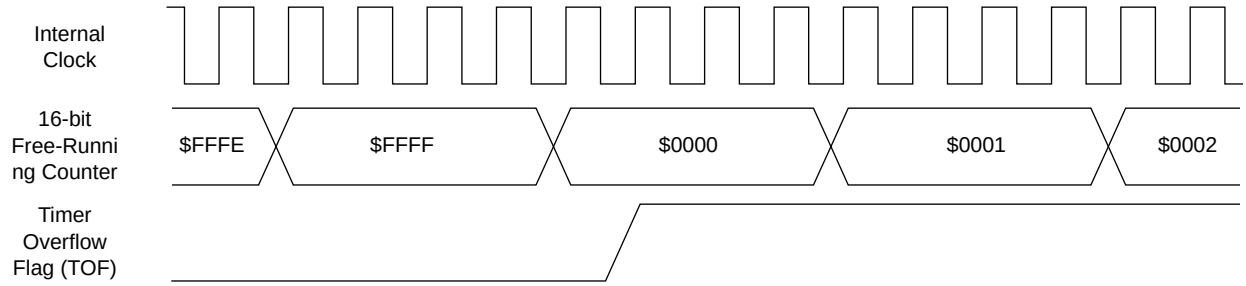
		7	6	5	4	3	2	1	0
TMRH $0020	R	TMRH7	TMRH6	TMRH5	TMRH4	TMRH3	TMRH2	TMRH1	TMRH0
	W								
reset ⇒		1	1	1	1	1	1	1	1
		7	6	5	4	3	2	1	0
TMRL $0021	R	TMRL7	TMRL6	TMRL5	TMRL4	TMRL3	TMRL2	TMRL1	TMRL0
	W								
reset ⇒		1	1	1	1	1	1	0	0

Figure 8-2: Timer Registers (TMRH / TMRL)

		7	6	5	4	3	2	1	0
ACRH $0022	R	ACRH7	ACRH6	ACRH5	ACRH4	ACRH3	ACRH2	ACRH1	ACRH0
	W								
reset ⇒		1	1	1	1	1	1	1	1
		7	6	5	4	3	2	1	0
ACRL $0023	R	ACRL7	ACRL6	ACRL5	ACRL4	ACRL3	ACRL2	ACRL1	ACRL0
	W								
reset ⇒		1	1	1	1	1	1	0	0

Figure 8-3: Alternate Counter Registers (ACRH / ACRL)

The Timer Registers and Alternate Counter Registers can be read at any time without affecting their values. However, the Alternate Counter Registers differ from the Timer Registers in one respect: a read of the Timer register LSB can clear the Timer Overflow Flag (TOF). Therefore, the Alternate Counter Registers can be read at any time without the possibility of missing timer overflow interrupts due to clearing of the TOF. See **Figure 8-4: State Timing Diagram for Timer Overflow**.



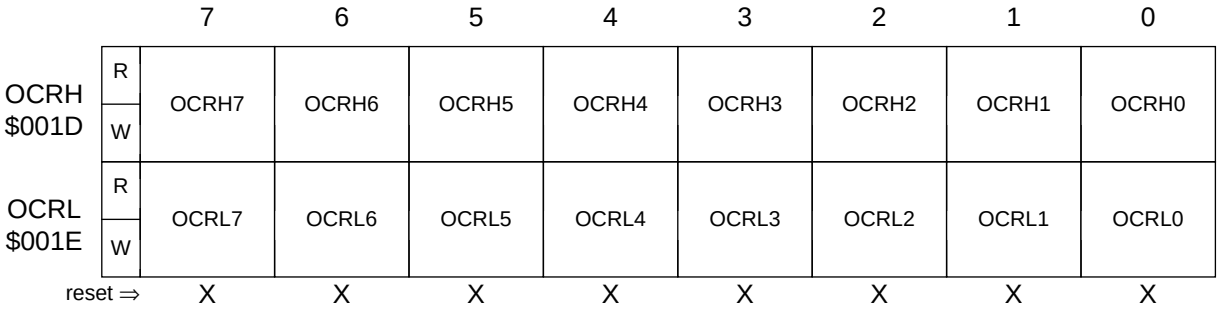
NOTE: The TOF bit is set at timer state T11 (transition of counter from $FFFF to $0000). It is cleared by reading the timer status register (TSR) during the high portion of the internal clock followed by reading the LSB of the counter register pair (TMRL).

Figure 8-4: State Timing Diagram for Timer Overflow

The free-running counter is initialized to $FFFC during reset and is a read-only register.

8.2 OUTPUT COMPARE

The Output Compare function may be used to generate an output waveform and/or as an elapsed time indicator. If the TCMP/TCAP1 bit of the TCR is set, output to the port pin is enabled. All of the bits in the Output Compare register pair OCRH / OCRL are readable and writable and are not altered by the 16-bit timer's control logic. Reset does not affect the contents of these registers. See **Figure 8-5: Output Compare Registers (OCRH / OCRL)**.



		7	6	5	4	3	2	1	0
OCRH $001D	R	OCRH7	OCRH6	OCRH5	OCRH4	OCRH3	OCRH2	OCRH1	OCRH0
	W								
OCRL $001E	R	OCRL7	OCRL6	OCRL5	OCRL4	OCRL3	OCRL2	OCRL1	OCRL0
	W								
reset ⇒		X	X	X	X	X	X	X	X

Figure 8-5: Output Compare Registers (OCRH / OCRL)

The contents of the output compare registers are compared with the contents of the free-running counter once every four internal clock cycles. If a match is found, the Output Compare Flag bit (OCF) is set and the Output Level bit (OLVL) is clocked to the output latch. The values in the output compare registers and output level bit should be changed after each successful comparison to control an output waveform, or to establish a new elapsed time-out. An interrupt can also accompany a successful output compare if the Output Compare Interrupt Enable bit (OCIE) is set.

After a CPU write cycle to the MSB of the output compare register pair (OCRH), the output compare function is inhibited until the LSB (OCRL) is written. Both bytes must be written if the MSB is written. A write made only to the LSB will not inhibit the compare function. The free-running counter increments every four internal clock cycles. The minimum time required to update the output compare registers is a function of software rather than hardware.

The output compare Output Level bit (OLVL) will be clocked to its output latch regardless of the state of the Output Compare Flag bit (OCF). A valid output compare must occur before the OLVL bit is clocked to its output latch (TCMP).

NOTE: The TCMP/TCAP1 bit of the TCR only affects the output of the latch to the port pin and has no effect on other parts of the Output Compare function.

Since neither the Output Compare Flag (OCF) nor the output compare registers are affected by reset, care must be exercised when initializing the output compare function. The following procedure is recommended:

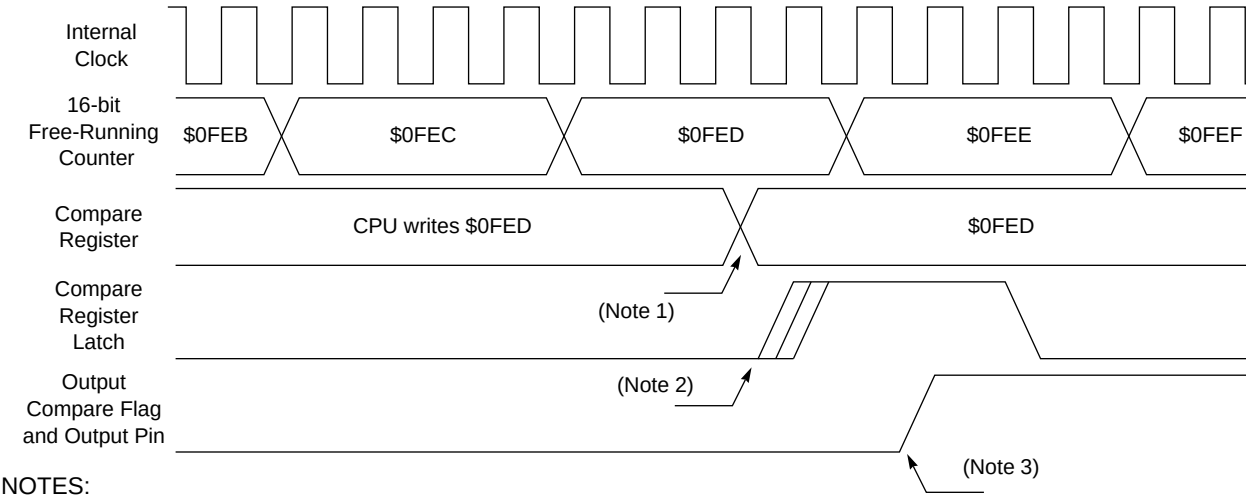
1. Block interrupts by setting the I-bit in the Condition Code Register (CCR).

2. Write the MSB of the output compare register pair (OCRH) to inhibit further compares until the LSB is written.
3. Read the Timer Status Register (TSR) to arm the Output Compare Flag (OCF).
4. Write the LSB of the output compare register pair (OCRL) to enable the output compare function and to clear its flag OCF (and interrupt).
5. Unblock interrupts by clearing the I-bit in the CCR.

This procedure prevents the Output Compare Flag bit (OCF) from being set between the time it is read and the time the output compare registers are updated. A software example is shown in **Figure 8-6: Output Compare Software Initialization Example**.

```
9B      SEI         BLOCK INTERRUPTS
.   .   .   .       .
.   .   .   .       .
B6  XX  LDA DATAH   HI BYTE FOR COMPARE
BE  XX  LDX DATAL   LO BYTE FOR COMPARE
B7  16  STA OCRH    INHIBIT OUTPUT COMPARE
B6  13  LDA TSR     ARM OCF BIT TO CLEAR
BF  17  STX OCRL    READY FOR NEXT COMPARE
.   .   .   .       .
.   .   .   .       .
9A      CLI         UNBLOCK INTERRUPTS
```

Figure 8-6: Output Compare Software Initialization Example



NOTES:
1. The CPU write to the compare register may take place at any time, but a compare occurs only at timer state T01. Thus, up to a four cycle difference may exist between the write to the compare register and the actual compare.
2. Internal compare takes place during timer state T01.
3. The Output Compare flag bit (OCF) is set at timer state T11, which follows the comparison match ($0FED in this example).

Figure 8-7: State Timing Diagram for Output Compare

8.3 INPUT CAPTURES

Registers are used to latch the value of the free-running counter after a defined transition is sensed by the input capture edge detector (*Note:* The input capture edge detector contains a Schmitt trigger to improve noise immunity.) The edge that triggers the counter transfer is defined by each input edge bit (IEDG1, IEDG2) in register TCR. Dynamically changing from Capture to Compare function will not affect the contents of the registers. All of the bits in the Input Capture register pair ICRH / ICRL are readable and are not altered by the 16-bit timer's control logic. Writes have no effect. Reset does not affect the contents of these registers. See **Figure 8-8: Input Capture Registers (ICRH1 / ICRL1)** and **Figure 8-9: Input Capture Registers (ICRH2 / ICRL2)**.

		7	6	5	4	3	2	1	0
ICRH1 $001B	R	ICRH7	ICRH6	ICRH5	ICRH4	ICRH3	ICRH2	ICRH1	ICRH0
	W								
ICRL1 $001C	R	ICRL7	ICRL6	ICRL5	ICRL4	ICRL3	ICRL2	ICRL1	ICRL0
	W								
reset ⇒		X	X	X	X	X	X	X	X

Figure 8-8: Input Capture Registers (ICRH1 / ICRL1)

		7	6	5	4	3	2	1	0
ICRH2 $0019	R	ICRH7	ICRH6	ICRH5	ICRH4	ICRH3	ICRH2	ICRH1	ICRH0
	W								
ICRL2 $001A	R	ICRL7	ICRL6	ICRL5	ICRL4	ICRL3	ICRL2	ICRL1	ICRL0
	W								
reset ⇒		X	X	X	X	X	X	X	X

Figure 8-9: Input Capture Registers (ICRH2 / ICRL2)

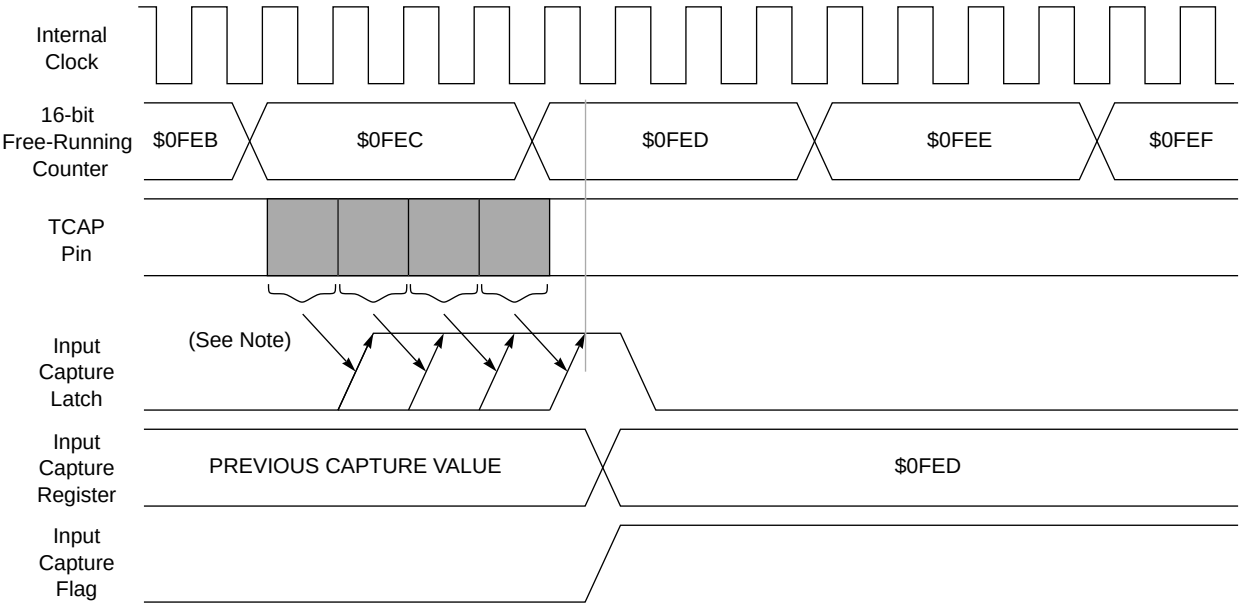
The result obtained by an input capture will be one more than the value of the free-running counter on the rising edge of the internal clock preceding the external transition (see **Figure 8-10: State Timing Diagram for Input Capture**). This delay is required for internal synchronization. Resolution is affected by the prescaler, allowing the free-running counter to increment once every four internal clock cycles.

The contents of the free-running counter are transferred to the input capture registers on each proper signal transition regardless of the state of the respective Input Capture Flag bit (ICF1, ICF2) in register TSR, the respective Flag will be set. The input capture registers always contain the free-running counter value, which corresponds to the most recent input capture. An interrupt can also accompany a successful input capture if the respective Input Capture Interrupt Enable bit (ICIE) is set.

When the TCMP/TCAP1 bit of TCR is set, input capture function for TCAP1 is inhibited.

After a read of the MSB of the input capture register pair (ICRH1, ICRH2), counter transfers are inhibited until the respective LSB of the register pair (ICRL1, ICRL2) is also read. This characteristic forces the minimum pulse period attainable to be determined by the time required to execute an input capture software routine in an application.

Reading the LSB of the input capture register pair (ICRL1, ICRL2) does not inhibit transfer of the free-running counter. Again, minimum pulse periods are ones that allow software to read the LSB of the register pair (ICRL1, ICRL2) and perform needed operations. There is no conflict between reading the LSB (ICRL1, ICRL2) and the free-running counter transfer since they occur on opposite edges of the internal clock.



NOTE: If the input edge occurs in the shaded area from one T10 timer state to the other T10 timer state, the Input Capture flag is set during the next T11 timer state.

Figure 8-10: State Timing Diagram for Input Capture

8.4 TIMER CONTROL REGISTER (TCR)

The timer control (TCR) and free-running counter (TMRH, TMRL, ACRH, ACRL) registers are the only registers of the 16-bit timer affected by reset. The Output Compare port (TCMP) is forced low after reset and remains low until OLVL is set and a valid Output Compare occurs.

		7	6	5	4	3	2	1	0
TCR $0017	R	ICIE2	ICIE1	TOIE	OCIE	TCMP /TCAP1	IEDG1	IEDG2	OLVL
	W								
reset ⇒		0	0	0	0	0	X	X	0

Figure 8-11: Timer Control Register (TCR)

8.4.1 ICIE2 - INPUT CAPTURE INTERRUPT ENABLE 2

Bit 7, when set, enables input capture 2 interrupts to the CPU. The interrupt will occur at the same time bit 7 (ICF2) in the TSR register is set.

8.4.2 ICIE1 - INPUT CAPTURE INTERRUPT ENABLE 1

Bit 6, when set, enables input capture 1 interrupts to the CPU. The interrupt will occur at the same time bit 6 (ICF1) in the TSR register is set.

8.4.3 TOIE - TIMER OVERFLOW INTERRUPT ENABLE

Bit 5, when set, enables timer overflow (rollover) interrupts to the CPU. The interrupt will occur at the same time bit 5 (TOF) in the TSR register is set.

8.4.4 OCIE - OUTPUT COMPARE INTERRUPT ENABLE

Bit 4, when set, enables output compare interrupts to the CPU. The interrupt will occur at the same time bit 4 (OCF) in the TSR register is set.

8.4.5 TCMP/TCAP1

Bit 3, when set, enables the TCMP function, when clear, the TCAP1 function. Reset clears this bit. When set it enables the TCMP output latch value to be output to the port pin and disables the edge detect of TCAP1. When clear, it disables the TCMP output latch from the port pin and enables the edge detect of TCAP1. Note that this bit has no effect on the setting of OCF and ICF1.

8.4.6 IEDG1 - INPUT CAPTURE EDGE SELECT 1

Bit 2 selects which edge of the input capture signal will trigger a transfer of the contents of the free-running counter registers to the input capture registers (ICRH1, ICRL1). Clearing this bit will select the falling edge, setting it selects the rising edge.

8.4.7 IEDG2 - INPUT CAPTURE EDGE SELECT 2

Bit 1 selects which edge of the input capture signal will trigger a transfer of the contents of the free-running counter registers to the input capture registers (ICRH2, ICRL2). Clearing this bit will select the falling edge, setting it selects the rising edge.

8.4.8 OLVL - OUTPUT COMPARE OUTPUT LEVEL SELECT

Bit 0 selects the output level (high or low) that is clocked into the output compare output latch at the next successful output compare.

8.5 TIMER STATUS REGISTER (TSR)

Reading the Timer Status Register (TSR) satisfies the first condition required to clear status flags and interrupts. The only remaining step is to read (or write) the register associated with the active status flag (and/or interrupt). This method does not present any problems for input capture or output compare functions.

However, a problem can occur when using a timer interrupt function and reading the free-running counter at random times to, for example, measure an elapsed time. If the proper precautions are not designed into the application software, a timer interrupt flag (TOF) could unintentionally be cleared if:

1. The TSR is read when bit 5 (TOF) is set, and
2. The LSB of the free-running counter is read, but not for the purpose of servicing the flag or interrupt.

The alternate counter registers (ACRH, ACRL) contain the same values as the timer registers (TMRH, TMRL). Registers ACRH and ACRL can be read at any time without affecting the Timer Overflow Flag (TOF) or interrupt.

		7	6	5	4	3	2	1	0
TSR $0018	R	ICF2	ICF1	TOF	OCF	0	0	0	0
	W								
reset ⇒		X	X	X	X	0	0	0	0

Figure 8-12: Timer Status Register (TSR)

8.5.1 ICF2 - INPUT CAPTURE 2 FLAG

Bit 7 is set when the edge specified by IEDG2 in register TCR has been sensed by the input capture edge detector fed by pin TCAP2. This flag, and the input capture interrupt, can be cleared by reading register TSR followed by reading the LSB of the input capture register pair (ICRL2).

8.5.2 ICF1 - INPUT CAPTURE 1 FLAG

Bit 6 is set when the edge specified by IEDG1 in register TCR has been sensed by the input capture edge detector fed by pin TCAP1. This flag, and the input capture interrupt, can be cleared by reading register TSR followed by reading the LSB of the input capture register pair (ICRL1).

8.5.3 TOF - TIMER OVERFLOW FLAG

Bit 5 is set by a rollover of the free-running counter from $FFFF to $0000. This flag, and the timer overflow interrupt, can be cleared by reading register TSR followed by reading the LSB of the timer register pair (TMRL).

8.5.4 OCF - OUTPUT COMPARE FLAG

Bit 4 is set when the contents of the output compare registers match the contents of the free-running counter. This flag, and the output compare interrupt, can be cleared by reading register TSR followed by writing the LSB of the output compare register pair (OCRL).

8.6 TIMER OPERATION DURING WAIT/HALT MODES

During WAIT modes, the 16-bit timer continues to operate normally and may generate an interrupt to trigger the MCU out of the WAIT mode.

8.7 TIMER OPERATION DURING STOP MODE

When the MCU enters the Stop Mode, the free-running counter stops counting (the internal clock is stopped). It remains at that particular count value until the Stop Mode is exited by applying an active signal to the IRQ pin, at which time the counter resumes from its stopped value as if nothing had happened. If Stop Mode is exited via an external RESET (logic low applied to the RESET pin) the counter is forced to $FFFC.

If a valid input capture edge occurs at either of the TCAP pins during Stop Mode, the input capture detect circuitry will be armed. This action does not set any flags or “wake up” the MCU, but when the MCU does “wake up” there will be an active input capture flag (and data) from the first valid edge. If the Stop Mode is exited by an external RESET, no input capture flag or data will be present even if a valid input capture edge was detected during the Stop Mode.

SECTION 9 PULSE WIDTH MODULATOR

The pulse width modulator (PWM) subsystem has two 8-bit channels (PWMA and PWMB). The PWM has a programmable prescaler, divide by 1.5 added to the initial prescaler, polarity, and mux enable with channel masking for motor control applications. The PWM is capable of generating signals from 0% to 100% duty cycle. A $00 in either PWM Data Register yields an “off” output (0%) with the polarity control bit set to one for that channel (i.e. PWMA or PWMB), but a $FF yields a duty of 255/256. To achieve the 100% duty (“ON” output), the polarity control bit is set to zero for that channel (i.e. PWMA or PWMB) while the data register has $00.

.

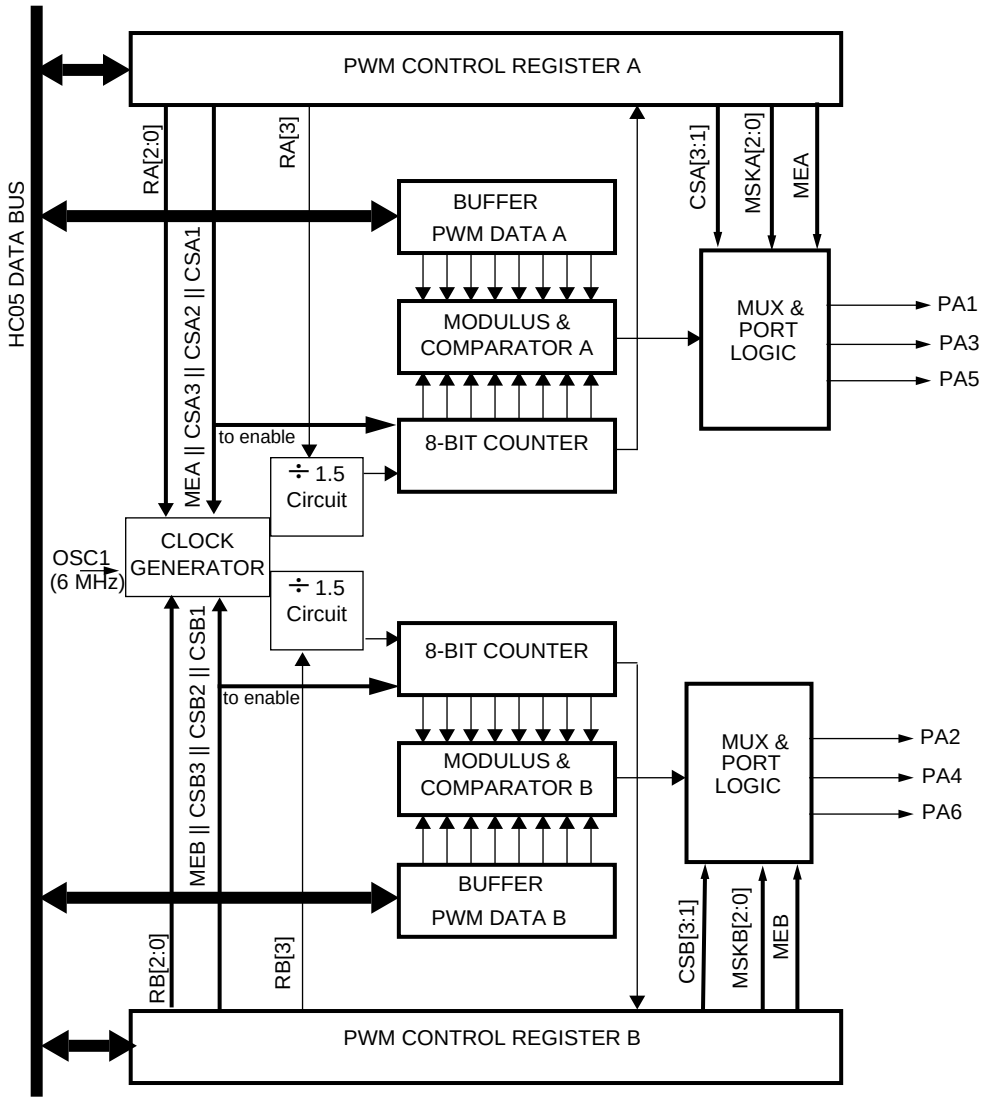


Figure 9-1: PWM Block Diagram

9.1 PWM REGISTERS

CTL-A $0014	R	MEA	POLA	MSKA3	MSKA2	MSKA1	CSA3	CSA2	CSA1
	W								
CTL-B $0015	R	MEB	POLB	MSKB3	MSKB2	MSKB1	CSB3	CSB2	CSB1
	W								
RATE $0016	R	RA3	RA2	RA1	RA0	RB3	RB2	RB1	RB0
	W								
UPDATE $0027	R	UPDATEA	UPDATEB						
	W								
reset ⇒		0	0	0	0	0	0	0	0

Figure 9-2: PWM CTL Registers

PWMA-D $0010	R	PWMA7	PWMA6	PWMA5	PWMA4	PWMA3	PWMA2	PWMA1	PWMA0
	W								
PWMA-I $0011	R	PWMA7	PWMA6	PWMA5	PWMA4	PWMA3	PWMA2	PWMA1	PWMA0
	W								
PWMB-D $0012	R	PWMB7	PWMB6	PWMB5	PWMB4	PWMB3	PWMB2	PWMB1	PWMB0
	W								
PWMB-I $0013	R	PWMB7	PWMB6	PWMB5	PWMB4	PWMB3	PWMB2	PWMB1	PWMB0
	W								
reset ⇒		X	X	X	X	X	X	X	X

Figure 9-3: PWMA & PWMB Data Registers

The PWM subsystem control and data registers are all buffered, as shown in **Figure 9-10: PWM Register Structure**. Each register consists of an active register, which contains the data used by the PWM subsystem, and a buffer register, which contains the data most recently written to the register address. Writes to the buffer registers are transferred to the active registers at the end of the PWM period if the respective bit in the UPDATE register is set to zero. If it is set to one, the transfer will occur immediately. In addition, a predefined sequence of register accesses may also need to be completed before the new contents

of these registers are transferred. This sequence of accesses is referred to as a register interlock mechanism, and is intended to allow more than one register to be modified before effecting the PWM operation. The data register interlock mechanism is managed by mapped each data register to an Interlock address (PWMx-I) and a Direct address (PWMx-D). Writes to the Interlock address will engage the interlock mechanism. Writes to the Direct address will not engage the interlock mechanism unless they are already engaged prior to the write.

9.2 PWM CONTROL BITS

The PWM subsystem is controlled through three control registers: CTL-A, CTL-B, RATE, and UPDATE. CTL-A, CTL-B, and the Data registers feature a write interlock and buffer mechanism to permit their contents to be updated simultaneously, preventing undesirable glitching of the associated Port A I/O.

The RATE register selects the PWM counter input clock rate, defining the PWM period. This register is buffered but not interlocked with other registers, so writes to this register will become effective at the end of the current PWM period (if the update bit has been cleared), irrespective of the state of any interlock mechanism.

9.2.1 POLx - POLARITY

Initializes PWM output. Toggles at Match point to Data Register. See **Figure 9-4: PWM Waveforms (POLx = 1)** and **Figure 9-5: PWM Waveforms (POLx = 0)**.

1 = Initialize output to one. Toggles to zero at data match.

0 = Initialize output to zero. Toggles to one at data match.

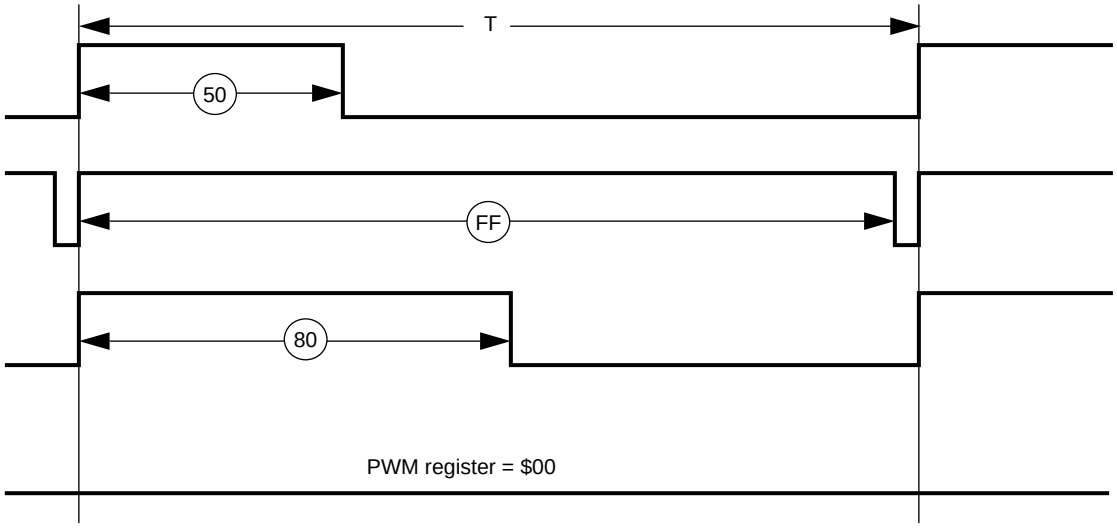


Figure 9-4: PWM Waveforms (POLx = 1)

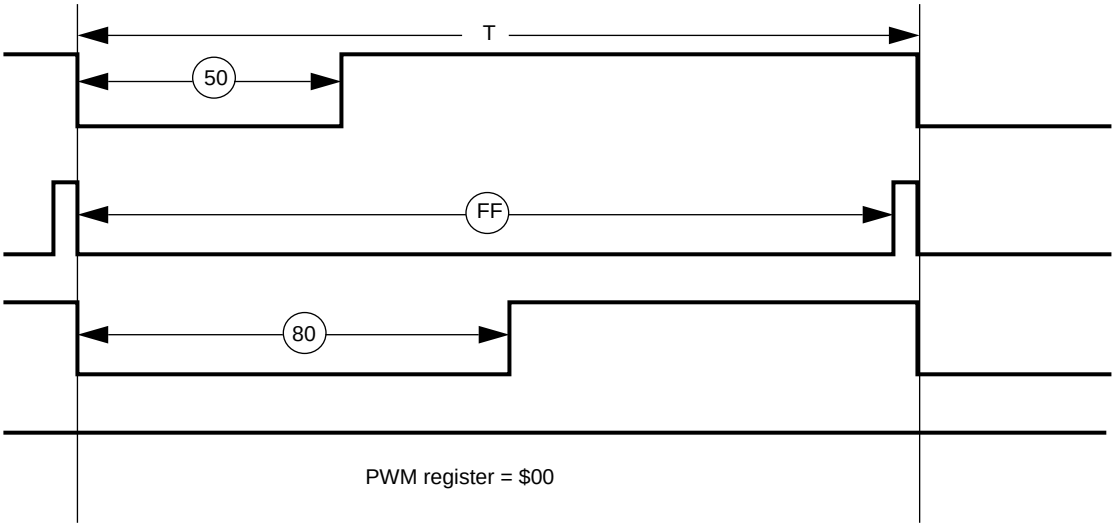


Figure 9-5: PWM Waveforms (POLx = 0)

9.2.2 Rx[3:0] - PWM OUTPUT RATE SELECT

Selects input clock for the 8-bit PWM counter.

Table 9-1: PWM Rate Select Table

Rx[3:0]	PWM Output CYCLE
0	23.4 KHz
1	11.7 KHz
2	5.86 KHz
3	2.93 KHz
4	1.46 KHz
5	732 Hz
6	366 Hz
7	183 Hz
8	15.6 KHz
9	7.8 KHz
A	3.9 KHz
B	1.95 KHz
C	975 Hz
D	488 Hz
E	244 Hz
F	122 Hz

9.2.3 CSx[3:1] - CHANNEL SELECT

Selects which channel/s will receive the PWM waveform output. When CSx[y] is set, the channel is selected and the PWMx Output waveform will be sent to the Port Logic. When CSx[y] is clear, the output will depend on the MEx bit and the corresponding PORT register bit. See description for MEx.

9.2.4 MEx - MASK ENABLE

When set, MEx enables the output mask feature of the PWMx subsystem. When enabled, all three channels will be used for subsystem output, that is the PORT registers will have no effect. All unselected channels (channels that have their corresponding CSx bit clear) will output the corresponding value of the MSKx bit. All selected channels (channels that have their corresponding CSx bit set) will output the PWM waveform.

When MEx is clear, output mask feature is disabled and the port will function as a normal I/O port if the CSx bit is cleared. This feature allows some of the ports to be freed up for normal I/O when not used for PWM output. If the corresponding CSx bit is set, the

selected channel will output the PWM waveform. See **Figure 9-6: PWM Output MUX Logic** and **Figure 9-7: PWM Control Example**.

If MEx is clear and all CSx bits are clear, it also disables the corresponding 8-bit counter to save power. If both counters are disabled, the clock generator will also be disabled.

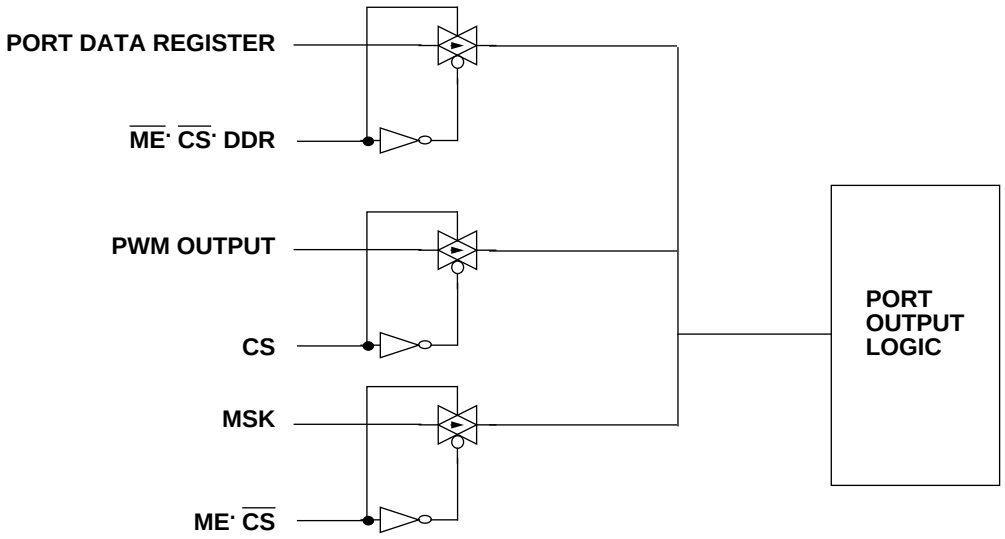


Figure 9-6: PWM Output MUX Logic

9.2.5 MSKx[3:1] - MASK

Set mask values for channels that are not selected by CSx when MEx is set. A Mask value of 1 drives unselected mux-pins high. A Mask value of 0 drives unselected mux-pins low. When MEx is clear, the Mask bits have no effect.

If the mask feature is enabled (MEx =1), the mask bits also provide an alternative method of generating 0% or 100% values. The conventional method generates 100% duty cycle by inverting the output polarity and simultaneously clearing the PWM Data bits.

See **Figure 9-6: PWM Output MUX Logic** and **Figure 9-7: PWM Control Example**.

9.2.6 UPDATEx

The UPDATEA and UPDATEB bits control when the registers for PWM A and PWM B will be updated. If the corresponding bit is clear, the registers will be updated at the end of the current cycle. If the corresponding bit is set, the registers will be updated immediately. The normal interlock restrictions still apply.

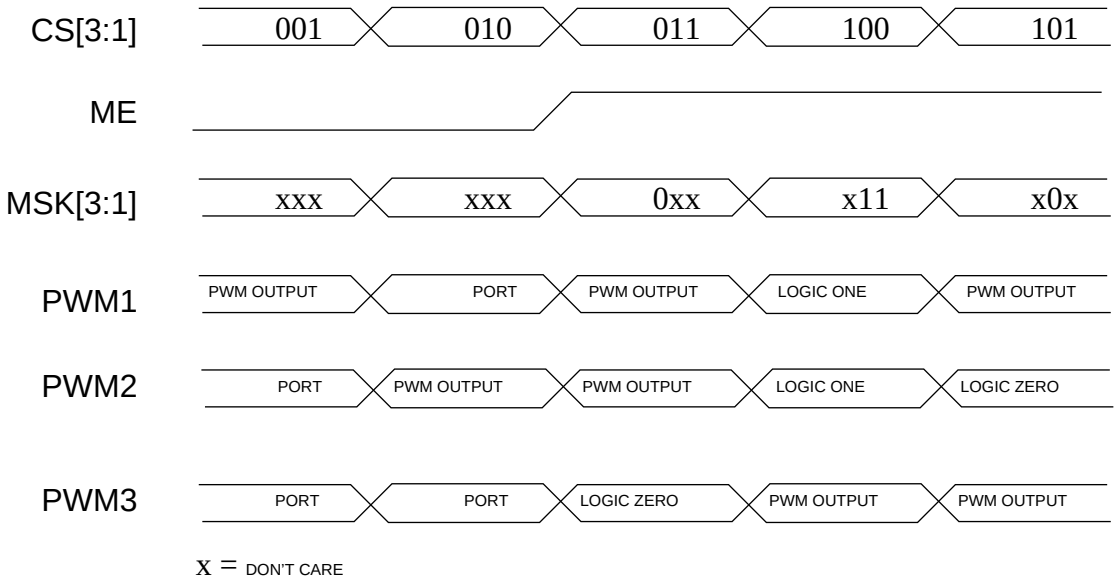


Figure 9-7: PWM Control Example

9.3 PWM DATA BITS

The pulse width of the PWM waveform is controlled by the two data registers PWMA and PWMB. Each data register can be accessed from one of two locations, PWMx-D (Direct) and PWMx-I (Interlock). A write interlock and buffer mechanism is used to permit their contents to be updated simultaneously, preventing undesirable glitching of the associated Port A I/O. See **9.5.1: PWM DATA REGISTER**.

9.4 PWM DURING RESETS

The PWM subsystem has two types of resets. One is a hardware reset denoted by $\overline{\text{RESET}}$. The other is a PWM reset denoted by PRESET.

After a $\overline{\text{RESET}}$, the user should write to the data registers, RATE, CTL-B, then CTL-A. This will avoid an erroneous duty cycle from being driven on any of the selected PWM Port pins.

A PRESET condition is reached by clearing the MEB bit and the CSB[3:1] bits in CTL-B, then the MEA bit and the CSA[3:1] bits in CTL-A. This disables the PWM subsystem, resets the 8-bit counters, resets the clock generator, and sets the port pins to the state defined by the respective Port Data Registers and Data Direction Registers.

The data registers are unaffected by $\overline{\text{RESET}}$. The CTL Registers are cleared by $\overline{\text{RESET}}$. A PRESET does not affect any of the PWM registers.

9.5 PWM OPERATION IN USER MODE

All PWM registers are buffered. The register buffering prevents data written to either the control or data registers from affecting the PWM cycle underway at the time of the data

write. The interlock mechanism extends this principle to multiple registers by preventing data written to groups of data and/or control registers from affecting the PWM configuration currently active until all writes are complete. There are several interlock options from which the user can pick depending upon the change of function desired. The register interlock mechanism operation is shown diagrammatically in **Figure 9-8: PWM Interlock Mechanisms**.

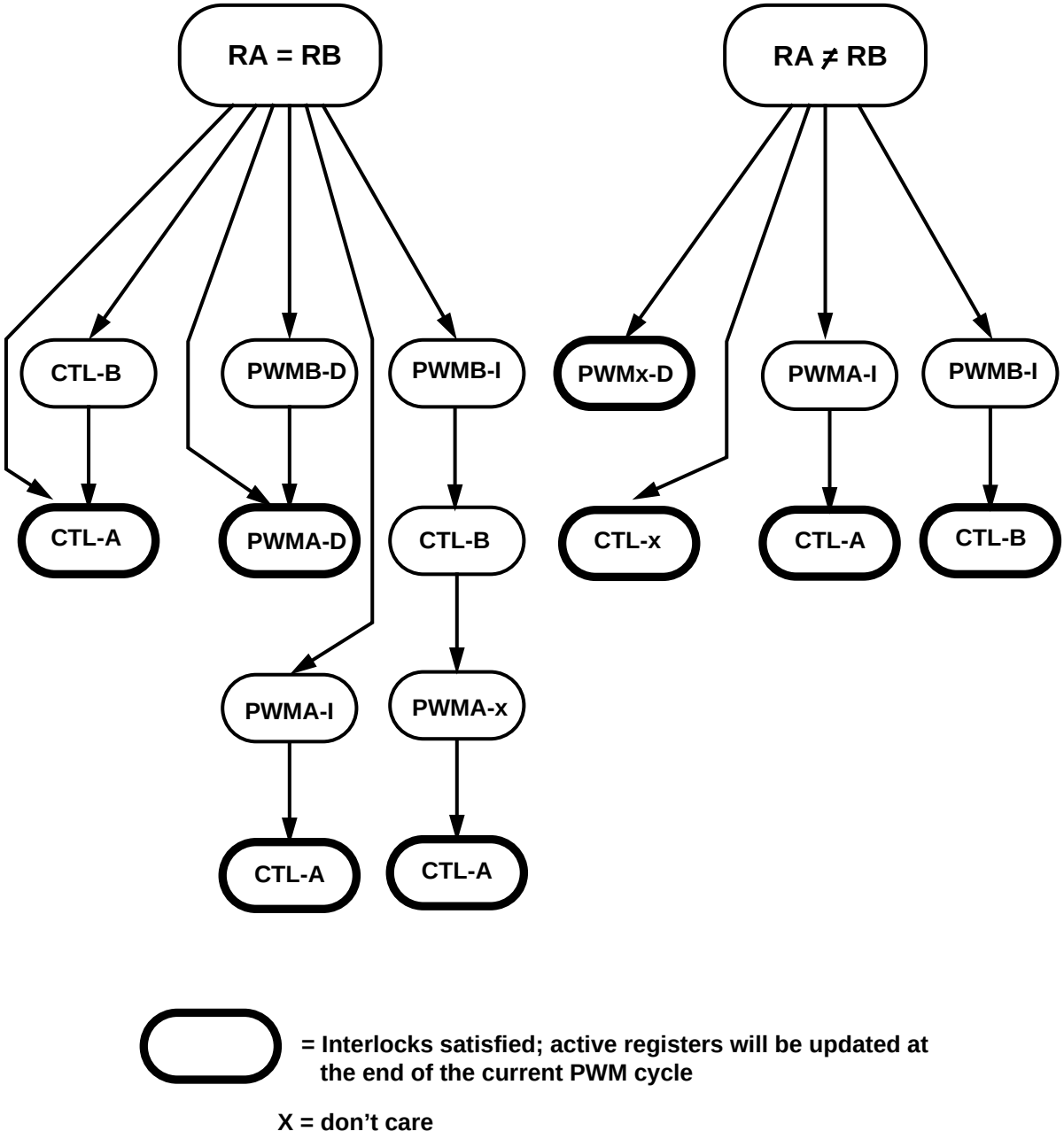


Figure 9-8: PWM Interlock Mechanisms

9.5.1 PWM DATA REGISTER

The PWMA and PWMB data registers have been mapped to two different addresses: the Direct address and the Interlock address. The PWMA-D Direct address is $10 and the PWMA-I Interlock address is $11. The PWMB-D Direct address is $12 and the PWMB-I Interlock address is $13. A read from either the Direct or the Interlock address will read the PWM active register. A write to either address will write to the PWM buffer register.

Writing to a PWMx Interlock address will activate a Data - Control interlock mechanism with the corresponding CTL-x register. Under such a condition, the new value written to the PWM Interlock data register will not be effective until the end of the current PWM cycle during which a read or write of the corresponding Control register was executed.

A typical application for such a mechanism is to generate 100% duty cycle when not using the PWM output mask feature (MEx =0). Synchronized changes to both Data and Control registers are therefore necessary to avoid PWM glitches. 100% duty cycle can be generated by clearing PWMx-I, then toggling the state of the POLx bit in the corresponding CTL-x register. Any new data written to either register will become effective at the end of the current PWM cycle during which the write or read to CTL-x took place.

Writing to the Direct address will not activate the interlock mechanism with the Control register. The new value will be updated at the end of the current PWM cycle if the update bit is clear. The new value will be updated immediately if the update bit is set..

NOTE: In either case above (write to Interlock or Direct addresses), the interlock mechanism that interlocks Channel A and Channel B together may preempt the transfer of the new data to the active registers. See **9.5.3: OPERATION WITH THE SAME PWM RATES** for more detail.

9.5.2 OPERATION WITH DIFFERENT PWM RATES

If RA does not equal RB, channel A and channel B are assumed to be operating independently of each other and are not interlocked. New data values written to either PWM channel will occur as discussed in **9.5.1: PWM DATA REGISTER**.

Interlocking between the channels only applies when both channels have the same period (RA = RB). The RATE register is not interlocked with any other registers but it is buffered. Changes to this register will affect the PWM cycle subsequent to the write. Consequently, changing the PWM period while generating a PWM signal will not cause erroneous PWM operation (i.e., glitches). Note that the RATE register is treated as two separate 4-bit registers, each buffered with the corresponding PWM channel cycle.

NOTE: Changing the channels from having different periods to having the same period may cause a phase difference between the channels due to accumulated clock phase difference. If synchronization is needed between channel A and channel B, a PRESET cycle must be executed to provide correct operation of the channel A/B interlock mechanism.

9.5.3 OPERATION WITH THE SAME PWM RATES

If RA equals RB, channel A and channel B are assumed to be operating together in a synchronous fashion and are interlocked. This interlock mechanism described below is in addition to the buffering and the PWM Data - Control interlock described in **9.5.1: PWM DATA REGISTER**.

A write to either PWMB data register must be followed by a write or a read to either PWMA data register. Any new data written will become effective at the end of the current PWM cycle during which the write or read to PWMA took place as shown in **Figure 9-8: PWM Interlock Mechanisms**. The interlocking between the data registers is disabled when the channels have different periods.

A write to CTLB control register must be followed by a write to the CTLA control register. Any new data written to either register will become effective at the end of the current PWM cycle during which the write to CTLA took place. The interlocking between the control registers is disabled when the channels have different periods.

Writing to the PWMx Interlock address will also activate the interlock mechanism with the CTL-x register as described in **9.5.1: PWM DATA REGISTER**. The two interlocking mechanisms, channel A - channel B and Data - Control, may be in effect at the same time.

Example 1: Writing to PWMB-I ($13) will require a write to CTL-B ($15) to satisfy the Data - Control interlock. In addition, if RA=RB, the write to either PWMB data register will require a read/write to either PWMA data register, and the write to CTL-B control register, will require a write to CTL-A register in order to satisfy the channel A - channel B interlock. The new contents of all these register will be transferred into their respective active registers at the end of the current PWM cycle during which all invoked interlock mechanisms become satisfied if the update bit is clear. If the corresponding update bit is set, the transfer will occur immediately after the interlock mechanism is satisfied.

Example 2: A write to PWMB-D ($12) will not invoke the Data - Control interlock. However, if RA=RB, the write to either PWMB data register, will require a read/write to either PWMA data register to satisfy the channel A - channel B interlock. Note that if a read/write was made to the PWMA interlock data register, the channel A - channel B interlock would still be satisfied but the Data - Control interlock will now be invoked for channel A. A write to CTL-A control register is now necessary to satisfy the channel A Data - Control interlock. Assuming UPDATEA and UPDATEB are clear in the UPDATE register, the new contents of all these registers will be transferred into their respective active registers at the end of the current PWM cycle during which all invoked interlock mechanisms become satisfied.

9.6 PWM DURING WAIT MODE

The PWM continues normal operation during WAIT mode. To decrease power consumption during WAIT, it is recommended that the PWM subsystem be put into PRESET state.

9.7 PWM DURING STOP MODE

In STOP mode the oscillator is stopped causing the PWM to cease function. Any signal in process is suspended in whatever phase the signal happens to be in. In the EPROM device, the STOP instruction will be disabled.

9.8 APPLICATION EXAMPLES

The following examples demonstrate PWM configuration options to drive DC brush and permanent magnet brushless motors. The schematic diagrams are simplified for clarity.

9.8.1 DC BRUSH MOTOR INTERFACE

The basic interface for a single DC brush motor is shown below.

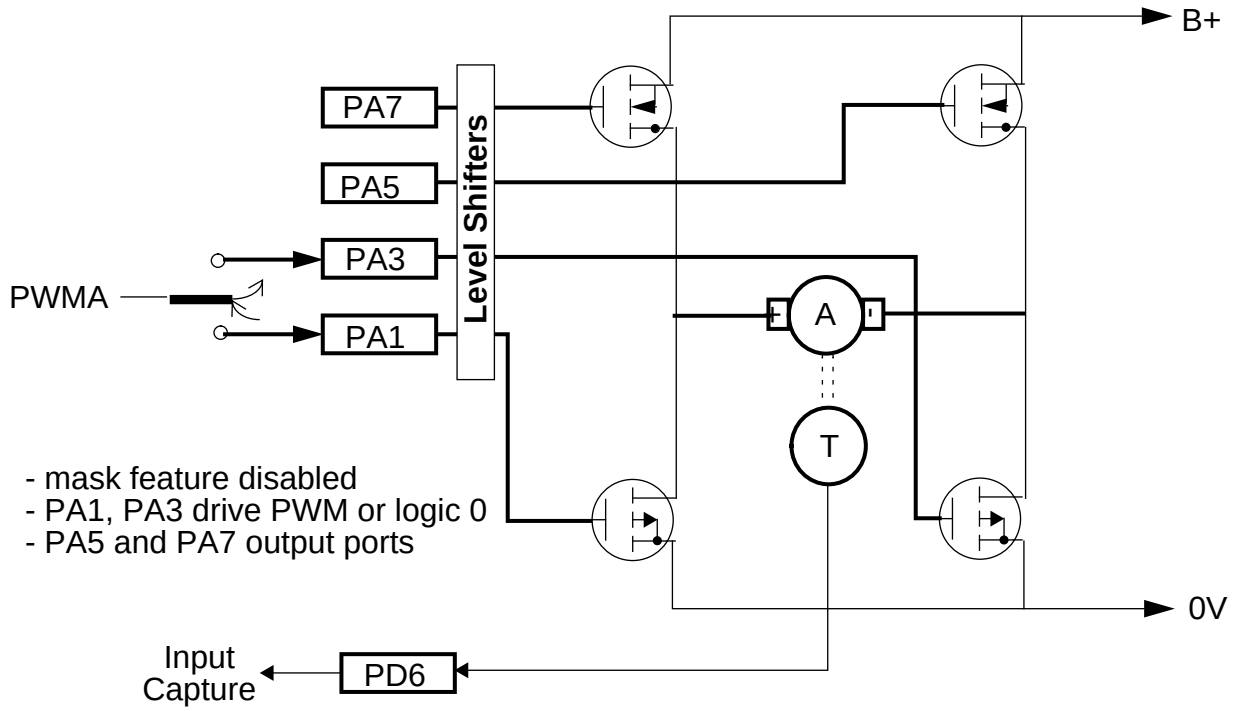


Figure 9-9: DC Brush Motor Interface

Table 9-2: DC Brush Motor Truth Table

Function	PA1	PA3	PA5	PA7
Forward	0	PWM	1	0
Reverse	PWM	0	0	1
Stop	0	0	1	1

PWM channel A is configured such that the PWM signal may be directed to either PA1 (CSA1=1) or PA3 (CSA2=1). When not driving the PWM signal, these ports should drive a logic 0 (MEA=0, PA1=0, DDRA1=1, PA3=0, DDRA3=1). PA5 is configured as an output

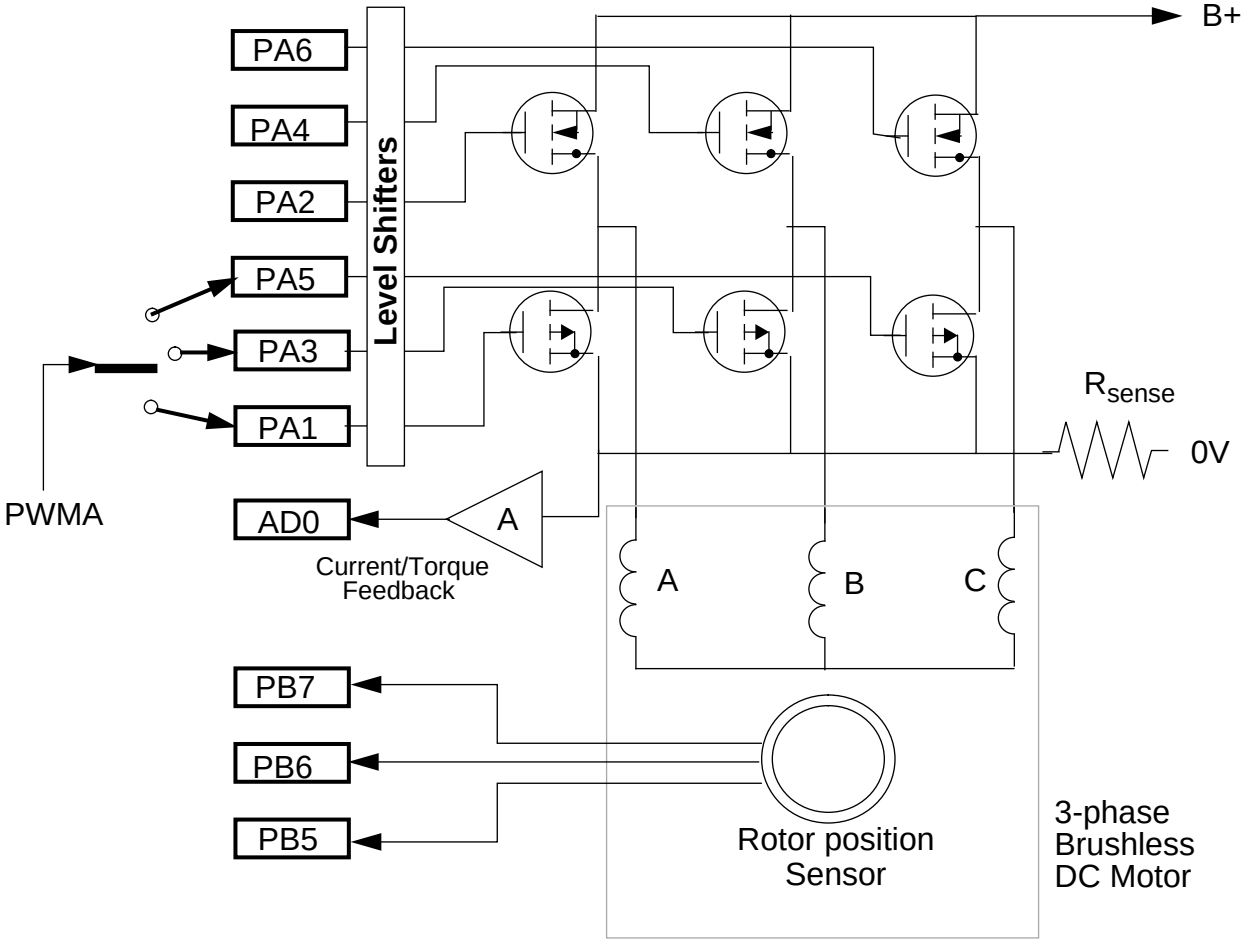
port (MEA=0, DDRA5=1). PA7 is configured as an output port (DDRA7=1). The software moderates the PWM output pulse width based on speed feedback data obtained from a tachometer or other such device. The tachometer would typically drive an input capture, providing data to the MCU from which velocity and acceleration information can be derived. The system could be modified to provide positional data for servo applications.

The truth table shown in **Table 9-2: DC Brush Motor Truth Table** defines the H-bridge drive requirements for the motor from which the software would select based upon other system inputs.

The MC68HC705MC4 can be configured to drive two motors using both PWM channels.

9.8.2 DC BRUSHLESS MOTOR INTERFACE

A typical interface for a single DC brushless motor is shown below although many other configurations are possible. The rotor sensor usually consists of a Hall effect sensor, optical encoder, or back-EMF detector. The coil current feedback is shown to be linear in



this example, using the on-chip A/D to provide torque data to the MCU. Other systems may only require a current limit, which could be achieved with an interrupt pin (which will offer some hysteresis) and an external amplifier. With some reorganization of the I/O, it is also possible to configure the device to drive two brushless motor simultaneously

although this will double the load on the processor. Depending upon the complexity of the control algorithms adopted, and considering that the commutation must be performed with software, care must be taken to maintain commutation delays to within acceptable limits. The commutation would follow the sequence shown in **Table 9-3: Brushless DC Motor Commutation Sequence**.

Table 9-3: Brushless DC Motor Commutation Sequence

Motor Phase	0	1	2	3	4	5
Phase A top Phase A bottom	PWM off	PWM off	off off	off on	off on	off off
Phase B top Phase B bottom	off on	off off	PWM off	PWM off	off off	off on
Phase C top Phase C bottom	off off	off on	off on	off off	PWM off	PWM off

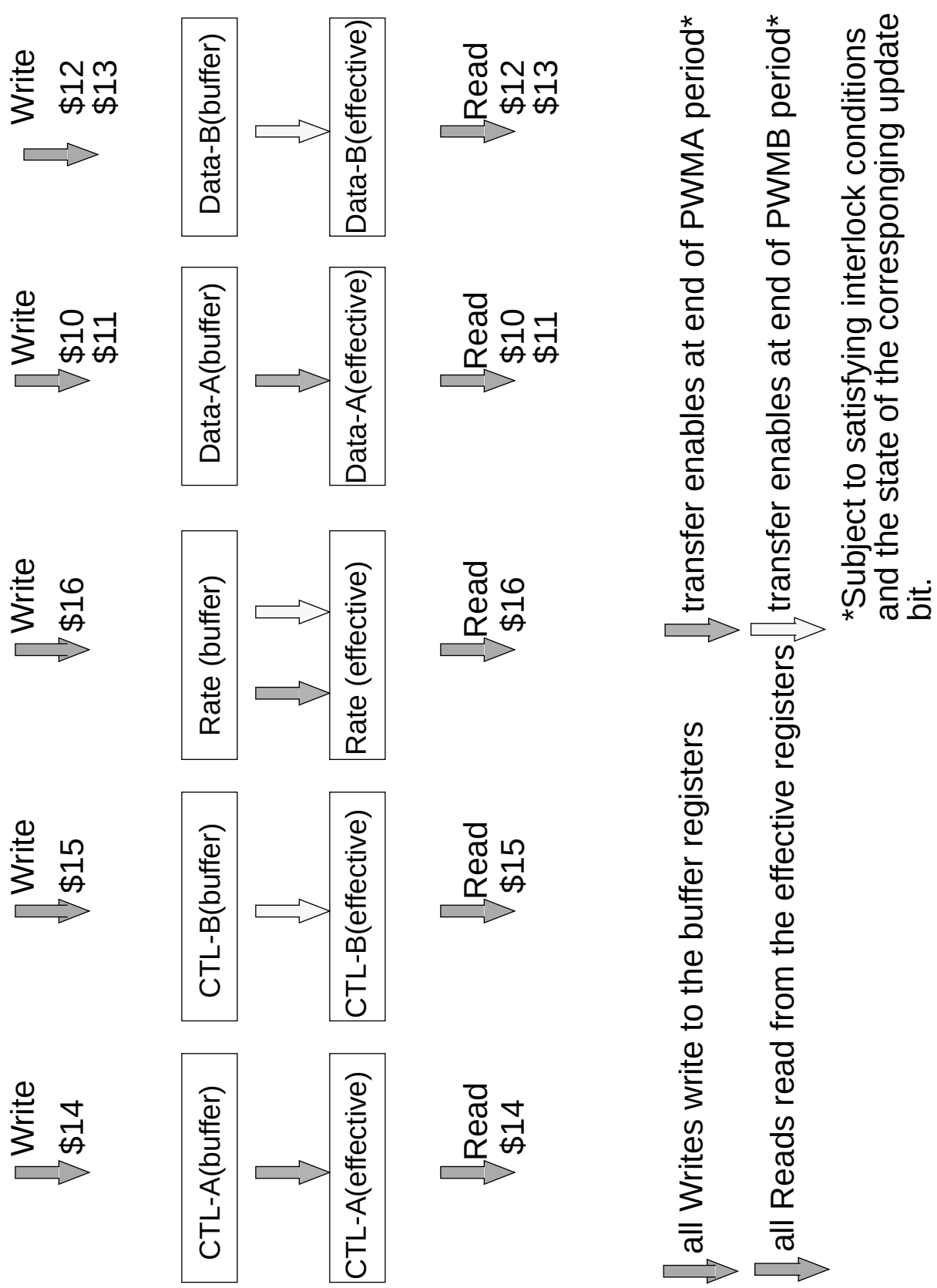


Figure 9-10: PWM Register Structure

SECTION 10 SERIAL COMMUNICATIONS INTERFACE

10.1 INTRODUCTION

The Serial Communications Interface (SCI) module allows high-speed asynchronous communication with peripheral devices and other MCUs.

10.2 FEATURES

Features of the SCI module include the following:

- Standard Mark/Space Non-Return-to-Zero Format
- Full Duplex Operation
- 32 Programmable Baud Rates
- Programmable 8-Bit or 9-Bit Character Length
- Separately Enabled Transmitter and Receiver
- Two Receiver Wakeup Methods:
  - Idle Line Wakeup
  - Address Mark Wakeup
- Interrupt-Driven Operation Capability with Five Interrupt Flags:
  - Transmitter Data Register Empty
  - Transmission Complete
  - Receiver Data Register Full
  - Receiver Overrun
  - Idle Receiver Input
- Receiver Framing Error Detection
- 1/16 Bit-Time Noise Detection

10.3 SCI DATA FORMAT

The SCI uses the standard non-return-to-zero mark/space data format illustrated in **Figure 10-1: SCI Data Format**.

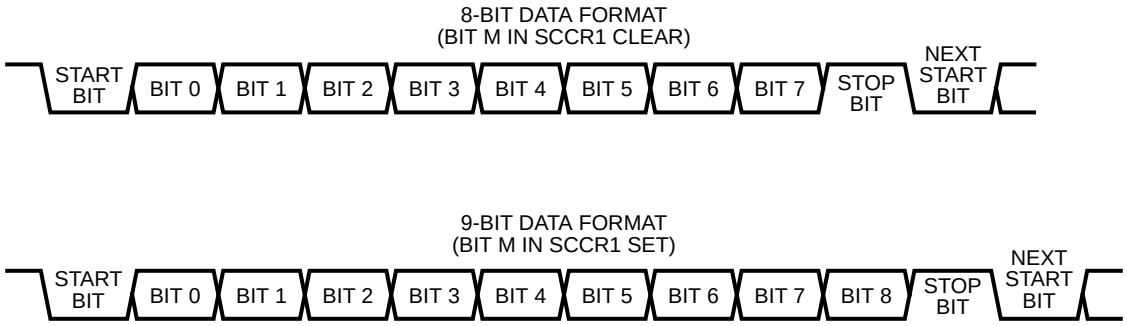


Figure 10-1: SCI Data Format

10.4 SCI OPERATION

The SCI allows full-duplex, asynchronous, RS232 or RS422 serial communication between the MCU and remote devices, including other MCUs. The transmitter and receiver of the SCI operate independently, although they use the same baud-rate generator. The following paragraphs describe the operation of the SCI transmitter and receiver.

10.4.1 TRANSMITTER

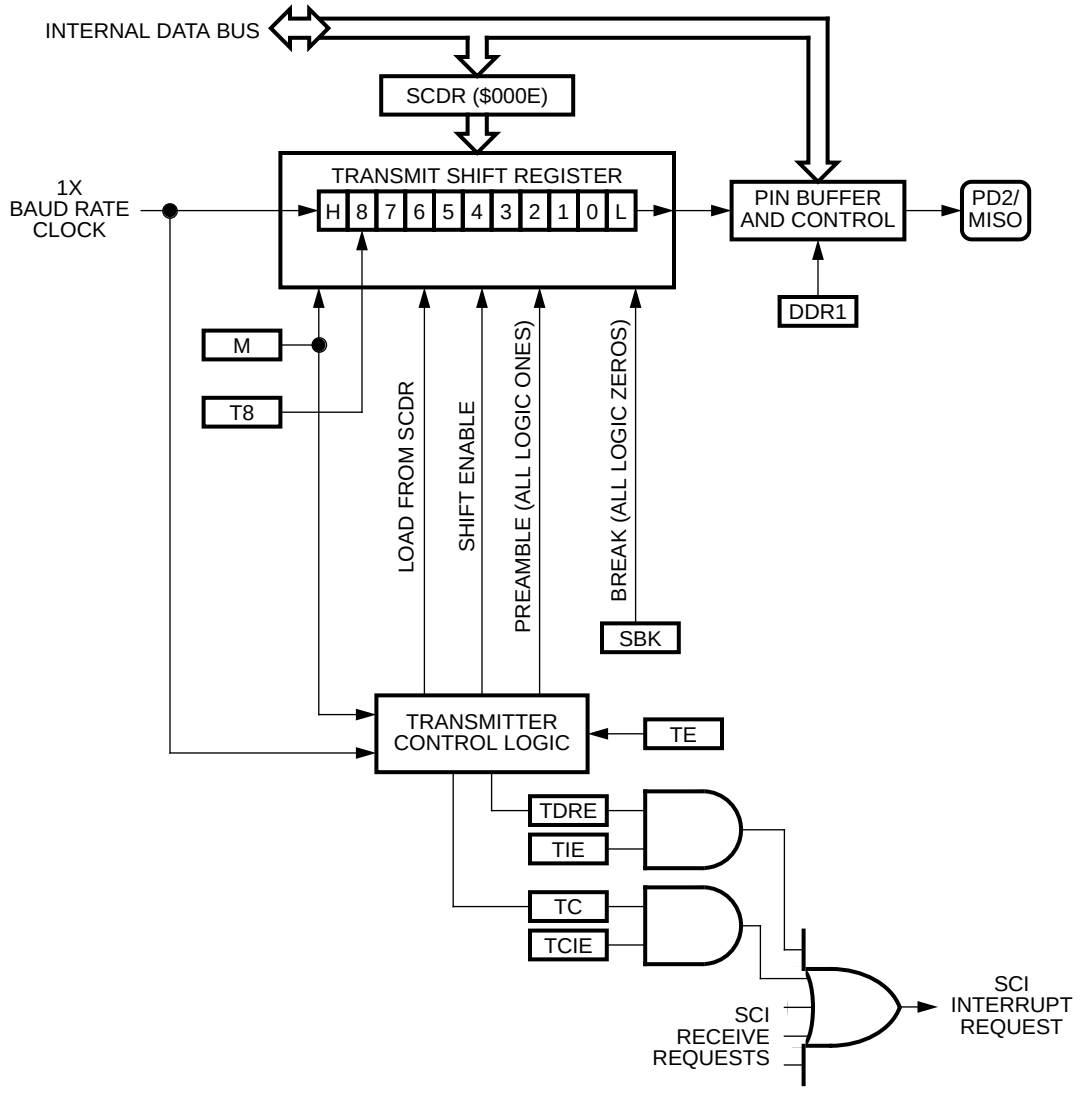
Figure 10-2: SCI Transmitter shows the structure of the SCI transmitter.

10.4.1.1 Character Length

The transmitter can accommodate either 8-bit or 9-bit data. The state of the M bit in SCI control register 1 (SCCR1) determines character length. When transmitting 9-bit data, bit T8 in SCCR1 is the ninth bit (bit 8).

10.4.1.2 Character Transmission

During transmission, the transmit shift register shifts a character out to the PB4/TDO pin. The SCI data register (SCDR) is the write-only buffer between the internal data bus and the transmit shift register.



	7	6	5	4	3	2	1	0	
BAUD RATE REGISTER (BAUD)			SCP1	SCP0		SCR2	SCR1	SCR0	$000A
SCI CONTROL REGISTER 1 (SCCR1)	R8	T8		M	WAKE				$000B
SCI CONTROL REGISTER 2 (SCCR2)	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	$000C
SCI STATUS REGISTER (SCSR)	TDRE	TC	RDRF	IDLR	OR	NF	FE	0	$000D
SCI DATA REGISTER (SCDR)	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	$000E

Figure 10-2: SCI Transmitter

Writing a logic one to the TE bit in SCI control register 2 (SCCR2), and then writing data to the SCDR, begins the transmission. At the start of a transmission, transmitter control logic automatically loads the transmit shift register with a preamble of logic ones. After the preamble shifts out, the control logic transfers the SCDR data into the shift register. A logic zero start bit automatically goes into the least significant bit position of the shift register, and a logic one stop bit goes into the most significant bit position.

When the data in the SCDR transfers to the transmit shift register, the transmit data register empty (TDRE) flag in the SCI status register (SCSR) becomes set. The TDRE flag indicates that the SCDR can accept new data from the internal data bus.

When the shift register is not transmitting a character, the PB4/TDO pin goes to the idle condition, logic one. If software clears the TE bit during the idle condition, and while TDRE is set, the transmitter relinquishes control of the PB4/TDO pin.

10.4.1.3 Break Characters

Writing a logic one to the SBK bit in SCCR2 loads the shift register with a break character. A break character contains all logic zeros and has no start and stop bits. Break character length depends on the M bit in SCCR1. As long as SBK is at logic one, transmitter logic continuously loads break characters into the shift register. After software clears the SBK bit, the shift register finishes transmitting the last break character and then transmits at least one logic one. The automatic logic one at the end of a break character is to guarantee the recognition of the start bit of the next character.

10.4.1.4 Idle Characters

An idle character contains all logic ones and has no start or stop bits. Idle character length depends on the M bit in SCCR1. The preamble is a synchronizing idle character that begins every transmission.

Clearing the TE bit during a transmission relinquishes the PB4/TDO pin after the last character to be transmitted is shifted out. The last character may already be in the shift register, or waiting in the SCDR, or a break character generated by writing to the SBK bit. Toggling TE from logic zero to logic one while the last character is in transmission generates an idle character (a preamble) that allows the receiver to maintain control of the PB4/TDO pin.

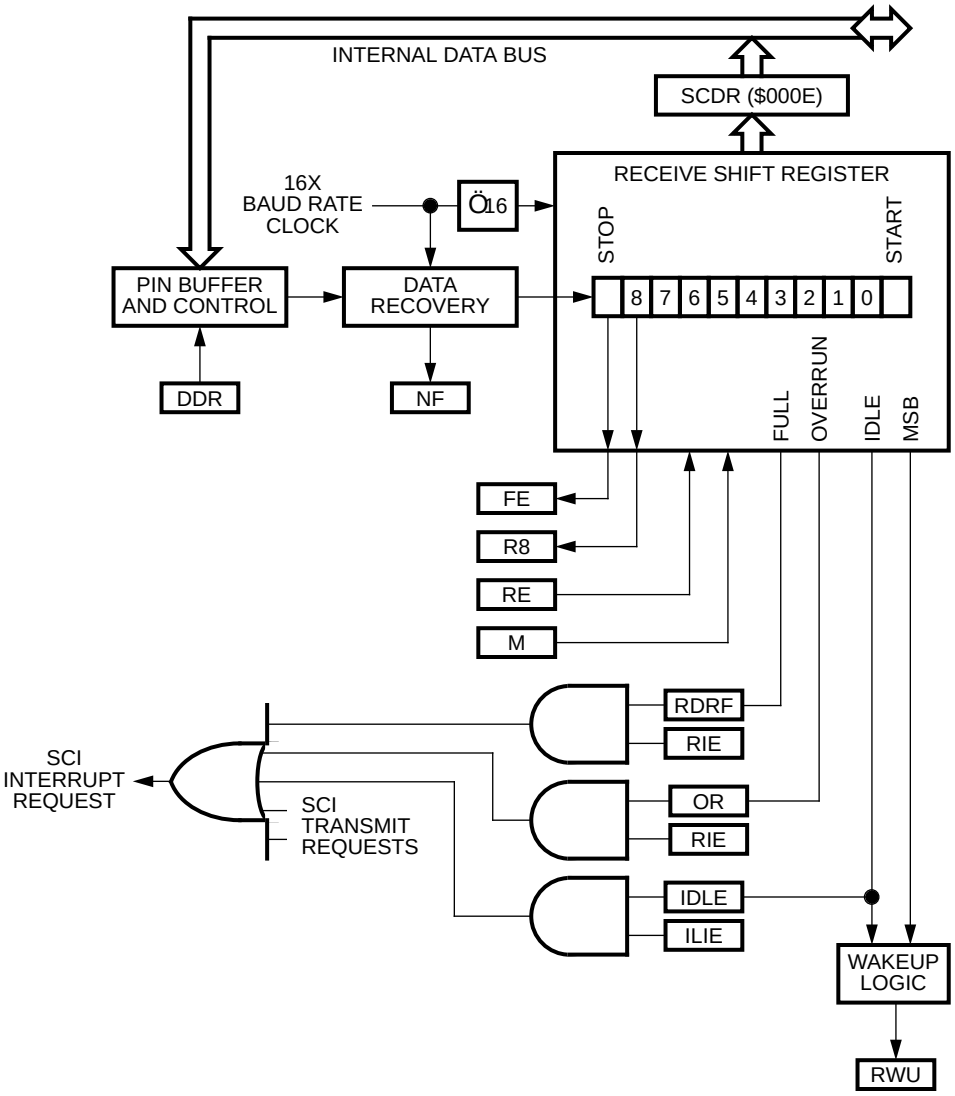
10.4.1.5 Transmitter Interrupts

The following sources can generate SCI transmitter interrupt requests:

- Transmit Data Register Empty (TDRE) — The TDRE bit in the SCSR indicates that the SCDR has transferred a character to the transmit shift register. TDRE is a source of SCI interrupt requests. The transmission complete interrupt enable bit (TCIE) in SCCR2 is the local mask for TDRE interrupts.
- Transmission Complete (TC) — The TC bit in the SCSR indicates that both the transmit shift register and the SCDR are empty and that no break or idle character has been generated. TC is a source of SCI interrupt requests. The transmission complete interrupt enable bit (TCIE) in SCCR2 is the local mask for TC interrupts.

10.4.2 RECEIVER

Figure 10-3: SCI Receiver shows the structure of the SCI receiver.



	7	6	5	4	3	2	1	0	
BAUD RATE REGISTER (BAUD)			SCP1	SCP0		SCR2	SCR1	SCR0	$000A
SCI CONTROL REGISTER 1 (SCCR1)	R8	T8		M	WAKE				$000B
SCI CONTROL REGISTER 2 (SCCR2)	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	$000C
SCI STATUS REGISTER (SCSR)	TDRE	TC	RDRF	IDLR	OR	NF	FE	0	$000D
SCI DATA REGISTER (SCDR)	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	$000E

Figure 10-3: SCI Receiver

10.4.2.1 Character Length

The receiver can accommodate either 8-bit or 9-bit data. The state of the M bit in SCI control register 1 (SCCR1) determines character length. When receiving 9-bit date, bit R8 in SCCR1 is the ninth bit (bit 8).

10.4.2.2 Character Reception

During reception, the receive shift register shifts characters in from the PB5/RDI pin. The SCI data register (SCDR) is the read-only buffer between the internal data bus and the receive shift register.

After a complete character shifts into the receive shift register, the data portion of the character is transferred to the SCDR, setting the receive data register full (RDRF) flag. The RDRF flag can be used to generate an interrupt.

10.4.2.3 Receiver Wakeup

So that the MCU can ignore transmissions intended only for other receivers in multiple-receiver systems, the MCU can be put into a standby state. Setting the receiver wakeup enable (RWU) bit in SCI control register 2 (SCCR2) puts the MCU into a standby state during which receiver interrupts are disabled.

Either of two conditions on the PB5/RDI pin can bring the MCU out of the standby state:

- Idle input line condition — If the PD5/RDI pin is at logic one long enough for 10 or 11 logic ones to shift into the receive shift register, receiver interrupts are again enabled.
- Address mark — If a logic one occurs in the most significant bit position of a received character, receiver interrupts are again enabled.

The state of the WAKE bit in SCCR1 determines which of the two conditions wakes up the MCU.

10.4.2.4 Receiver Noise Immunity

The data recovery logic samples each bit 16 times to identify and verify the start bit and to detect noise. Any conflict between noise-detection samples sets the noise flag (NF) in the SCSR. The NF bit is set at the same time that the RDRF bit is set.

10.4.2.5 Framing Errors

If the data recovery logic does not detect a logic one where the stop bit should be in an incoming character, it sets the framing error (FE) bit in the SCSR. The FE bit is set at the same time that the RDRF bit is set.

10.4.2.6 Receiver Interrupts

The following sources can generate SCI receiver interrupt requests:

- Receive Data Register Full (RDRF) — The RDRF bit in the SCSR indicates that the receive shift register has transferred a character to the SCDR.
- Receiver Overrun (OR) — The OR bit in the SCSR indicates that the receive shift register shifted in a new character before the previous character was read from the SCDR.
- Idle Input (IDLE) — The IDLE bit in the SCSR indicates that 10 or 11 consecutive logic ones shifted in from the PD5/RDI pin.

10.5 SCI I/O REGISTERS

The following I/O registers control and monitor SCI operation:

- SCI Data Register (SCDR)
- SCI Control Register 1 (SCCR1)
- SCI Control Register 2 (SCCR2)
- SCI Status Register (SCSR)

10.5.1 SCI Data Register (SCDR)

The SCI data register is the buffer for characters received and for characters transmitted.

		7	6	5	4	3	2	1	0
SCDR $000E	READ:	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	WRITE:								
RESET:		UNAFFECTED BY RESET							

Figure 10-4: SCI Data Register (SCDR)

10.5.2 SCI Control Register 1 (SCCR1)

SCI control register 1 has the following functions:

- Stores ninth SCI data bit received and ninth SCI data bit transmitted
- Controls SCI character length
- Controls SCI wakeup method

		7	6	5	4	3	2	1	0
SCCR1 $000B	READ	R8	T8		M	WAKE			
	WRITE								
RESET:		UNAFFECTED BY RESET							

Figure 10-5: SCI Control Register 1 (SCCR1)

R8 — Bit 8 (Received)

When the SCI is receiving 9-bit characters, R8 is the ninth bit of the received character. R8 receives the ninth bit at the same time that the SCDR receives the other eight bits. Reset has no effect on the R8 bit.

T8 — Bit 8 (Transmitted)

When the SCI is transmitting 9-bit characters, T8 is the ninth bit of the

transmitted character. T8 is loaded into the transmit shift register at the same time that SCDR is loaded into the transmit shift register. Reset has no effect on the T8 bit.

M — Character Length

This read/write bit determines whether SCI characters are 8 bits long or 9 bits long. The ninth bit can be used as an extra stop bit, as a receiver wakeup signal, or as a mark or space parity bit. Reset has no effect on the M bit.

1 = 9-bit SCI characters
0 = 8-bit SCI characters

WAKE — Wakeup Bit

This read/write bit determines which condition wakes up the SCI: a logic one (address mark) in the most significant bit position of a received character or an idle condition of the PD5/RDI pin. Reset has no effect on the WAKE bit.

1 = Address mark wakeup
0 = Idle line wakeup

10.5.3 SCI Control Register 2 (SCCR2)

SCI control register 2 has the following functions:

- Enables the SCI receiver and SCI receiver interrupts
- Enables the SCI transmitter and SCI transmitter interrupts
- Enables SCI receiver idle interrupts
- Enables SCI transmission complete interrupts
- Enables SCI wakeup
- Transmits SCI break characters

		7	6	5	4	3	2	1	0
SCCR2 $000C	READ / WRITE	TIE	TCIE	RIE	ILIE	TE	RE	TWU	SBK
	RESET:	0	0	0	0	0	0	0	0

Figure 10-6: SCI Control Register 2 (SCCR2)

TIE — Transmit Interrupt Enable

This read/write bit enables SCI interrupt requests when the TDRE bit becomes set. Reset clears the TIE bit.

1 = TDRE interrupt requests enabled
0 = TDRE interrupt requests disabled

TCIE — Transmission Complete Interrupt Enable

This read/write bit enables SCI interrupt requests when the TC bit becomes set. Reset clears the TCIE bit

1 = TC interrupt requests enabled
0 = TC interrupt requests disabled

RIE — Receive Interrupt Enable

This read/write bit enables SCI interrupt requests when the RDRF bit or the OR bit becomes set. Reset clears the RIE bit.

1 = RDRF interrup requests enabled
0 = RDRF interrupt requests disabled

ILIE — Idle Line Interrupt Enable

This read/write bit enables SCI interrpt requests when the IDLE bit becomes set. Reset clears the ILIE bit.

1 = IDLE interrupt requests enabled
0 = IDLE interrupt requests disabled

TE — Transmit Enable

Setting this read/write bit begins the transmission by sending a preamble of 10 or 11 logic ones from the transmit shift register to the PB4/TDO pin. Reset clears the TE bit.

1 = Transmission enabled
0 = Transmission disabled

RE — Receive Enable

Setting this read/write bit enables the receiver. Clearing the RE bit disables the receiver and receiver interrupts but does not affect the receiver interrupt flags. Reset clears the RE bit.

1 = Receiver enabled
0 = Receiver disabled

RWU — Receiver Wakeup Enable

This read/write bit puts the receiver in a standby state. Typically, data transmitted to the receiver clears the RWU bit and returns the receiver to normal operation. The WAKE bit in SCCR1 determines whether an idle input or an address mark brings the receiver out of the standby state. Reset clears the RWU bit.

1 = Standby state
0 = Normal operation

SBK — Send Break

Setting this read/write bit continuously transmits break codes in the form of 10-bit or 11-bit groups of logic zeros. Clearing the SBK bit stops the break codes and transmits a logic one as a start bit. Reset clears the SBK bit.

1 = Break codes being transmitted
0 = No break codes being transmitted

10.5.4 SCI Status Register (SCSR)

The SCI status register contains flags to signal the following conditions:

- Transfer of SCDR data to transmit shift register complete
- Transmission complete
- Transfer of receive shift register data to SCDR complete
- Receiver input idle
- Receiver overrun
- Noisy data
- Framing error

		7	6	5	4	3	2	1	0
SCSR $000D	READ:	TDRE	TC	RDRF	IDLE	OR	NF	FE	
	WRITE:								
	RESET:	1	1	0	0	0	0	0	—

Figure 10-7: SCI Status Register (SCSR)

TDRE — Transmit Data Register Empty

This clearable, read-only bit is set when the data in the SCDR transfers to the transmit shift register. TDRE generates an interrupt request if the TIE bit in SCCR2 is also set. Clear the TDRE bit by reading the SCSR with TDRE set, and then writing to the SCDR. Reset sets the TDRE bit. Software must initialize the TDRE bit to logic zero to avoid an instant interrupt request when turning on the transmitter.

1 = SCDR data transferred to transmit shift register
0 = SCDR data not transferred to transmit shift register

TC — Transmission Complete

This clearable, read-only bit is set when the TDRE bit is set, and no data, preamble, or break character is being transmitted. TC generates an interrupt request if the TCIE bit in SCCR2 is also set. Clear the TC bit by reading the SCSR with TC set, and then writing to the SCDR. Reset sets the TC bit. Software must initialize the TC bit to logic zero to avoid an instant interrupt request when turning on the transmitter.

1 = No transmission in progress
0 = Transmission in progress

RDRF — Receive Data Register Full

This clearable, read-only bit is set when the data in the receive shift register transfers to the SCI data register. RDRF generates an interrupt request if the

RIE bit in SCCR2 is also set. Clear the RDRF bit by reading the SCSR with RDRF set, and then reading the SCDR. Reset clears the RDRF bit.

1 = Received data available in SCDR
0 = Received data not available in SCDR

IDLE — Receiver Idle

This clearable, read-only bit is set when 10 or 11 consecutive logic ones appear on the receiver input. IDLE generates an interrupt request if the ILIE bit in SCCR2 is also set. Clear the IDLE bit by reading the SCSR with IDLE set, and then reading the SCDR. Reset clears the IDLE bit.

1 = Receiver input idle
0 = Receiver input not idle

OR — Receiver Overrun

This clearable, read-only bit is set if the SCDR is not read before the receive shift register receives the next word. OR generates an interrupt request if the RIE bit in SCCR2 is also set. The data in the shift register is lost, but the data already in the SCDR is not affected. Clear the OR bit by reading the SCSR with OR set, and then reading the SCRD. Reset clears the OR bit.

1 = Receiver shift register full and RDRF = 1
0 = No receiver overrun

NF — Receiver Noise Flag

This clearable, read-only bit is set when noise is detected in data received in the SCI data register. Clear the NF bit by reading the SCSR, and then reading the SCDR. Reset clears the NF bit.

1 = Noise detected in SCDR
0 = No noise detected in SCDR

FE — Receiver Framing Error

This clearable, read-only flag is set when there is a logic zero where a stop bit should be in the character shifted into the receive shift register. If the received word causes both a framing error and an overrun error, the OR bit is set and the FE bit is not set. Clear the FE bit by reading the SCSR, and then reading the SCDR. Reset clears the FE bit.

1 = Framing error
0 = No framing error

10.5.5 Baud Rate Register (BAUD)

The baud rate register selectes the baud rate for both the receiver and the transmitter.

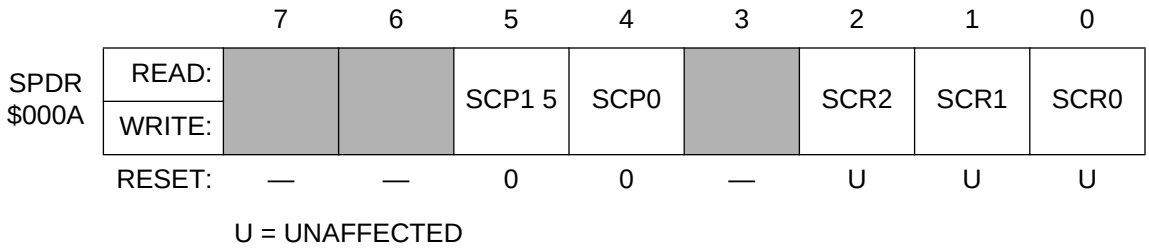


Figure 10-8: Baud Rate Register (BAUD)

SCP1 and SCP0 — SCI Prescaler Select Bits

These read/write bits control prescaling of the baud rate generator clock, as shown in Table 10-1:. Resets clear both SCP1 and SCP0.

Table 10-1: Baud Rate Generator Clock Prescaling

SCP[0:1]	Baud Rate Generator Clock
00	Internal Clock Ö1
01	Internal Clock Ö3
10	Internal Clock Ö4
11	Internal Clock Ö13

SCR2–SCR0 — SCI Baud Rate Select Bits

These read/write bits select the SCI baud rate, as shown in **Table 10-2: Baud Rate Selection**. Reset has no effect on the SCR2–SCR0 bits.

Table 10-2: Baud Rate Selection

SCR[2:1:0]	SCI Baud Rate (Baud)
000	Prescaled Clock Ö1
001	Prescaled Clock Ö2
010	Prescaled Clock Ö4
011	Prescaled Clock Ö8
100	Prescaled Clock Ö16
101	Prescaled Clock Ö32
110	Prescaled Clock Ö64
111	Prescaled Clock Ö128

Table 10-3: Baud Rate Selection Examples shows all possible SCI baud rates derived from crystal frequencies of 2 MHz, 4 MHz, 4.194304 MHz, and 6 MHz.

Table 10-3: Baud Rate Selection Examples

SCP[1:0]	SCR[2:1:0]	SCI Baud Rate		
		f_{OSC} = 4 MHz f_{OP} = 2 MHz	f_{OSC} = 4.194304 MHz f_{OP} = 2.097152 MHz	f_{OSC} = 6 MHz f_{OP} = 3 MHz
00	000	125 kbaud	131.1 kbaud	187.5 kbaud
00	001	62.50 kbaud	65.54 kbaud	93.75 kbaud
00	010	31.25 kbaud	32.77 kbaud	46.89 kbaud
00	011	15.63 kbaud	16.38 kbaud	23.44 kbaud
00	100	7813 Baud	8192 Baud	11.72 kbaud
00	101	3906 Baud	4096 Baud	5859 Baud
00	110	1953 Baud	2048 Baud	2930 Baud
00	111	976.6 Baud	1024 Baud	1465 Baud
01	000	41.67 kbaud	43.69 kbaud	62.49 kbaud
01	001	20.83 kbaud	21.85 kbaud	31.26 kbaud
01	010	10.42 kbaud	10.92 kbaud	15.62 kbaud
01	011	5208 Baud	5461 Baud	7812 Baud
01	100	2604 Baud	2731 Baud	3906 Baud
01	101	1302 Baud	1365 Baud	1953 Baud
01	110	651.0 Baud	682.7 Baud	976.5 Baud
01	111	325.5 Baud	341.3 Baud	488.4 Baud
10	000	31.25 kbaud	32.77 kbaud	46.89 kbaud
10	001	15.63 kbaud	16.38 kbaud	23.44 kbaud
10	010	7813 Baud	8192 Baud	11.72 kbaud
10	011	3906 Baud	4906 Baud	5859 Baud
10	100	1953 Baud	2048 Baud	2930 Baud
10	101	976.6 Baud	1024 Baud	1465 Baud
10	110	488.3 Baud	512.0 Baud	732.3 Baud
10	111	244.1 Baud	256.0 Baud	366.3 Baud
11	000	9615 Baud	10.08 kbaud	14.42 kbaud
11	001	4808 Baud	5041 Baud	7212 Baud
11	010	2404 Baud	2521 Baud	3606 Baud
11	011	1202 Baud	1260 Baud	1803 Baud
11	100	601.0 Baud	630.2 Baud	901.5 Baud
11	101	300.5 Baud	315.1 Baud	450.6 Baud
11	110	150.2 Baud	157.5 Baud	225.4 Baud
11	111	75.12 Baud	78.77 Baud	112.7 Baud

SECTION 11 CORE TIMER

The Core Timer (Ctimer) for this device is a 15-stage multi-functional ripple counter. The features include Timer Overflow, Power-On Reset (POR), Real Time Interrupt, and COP Watchdog Timer

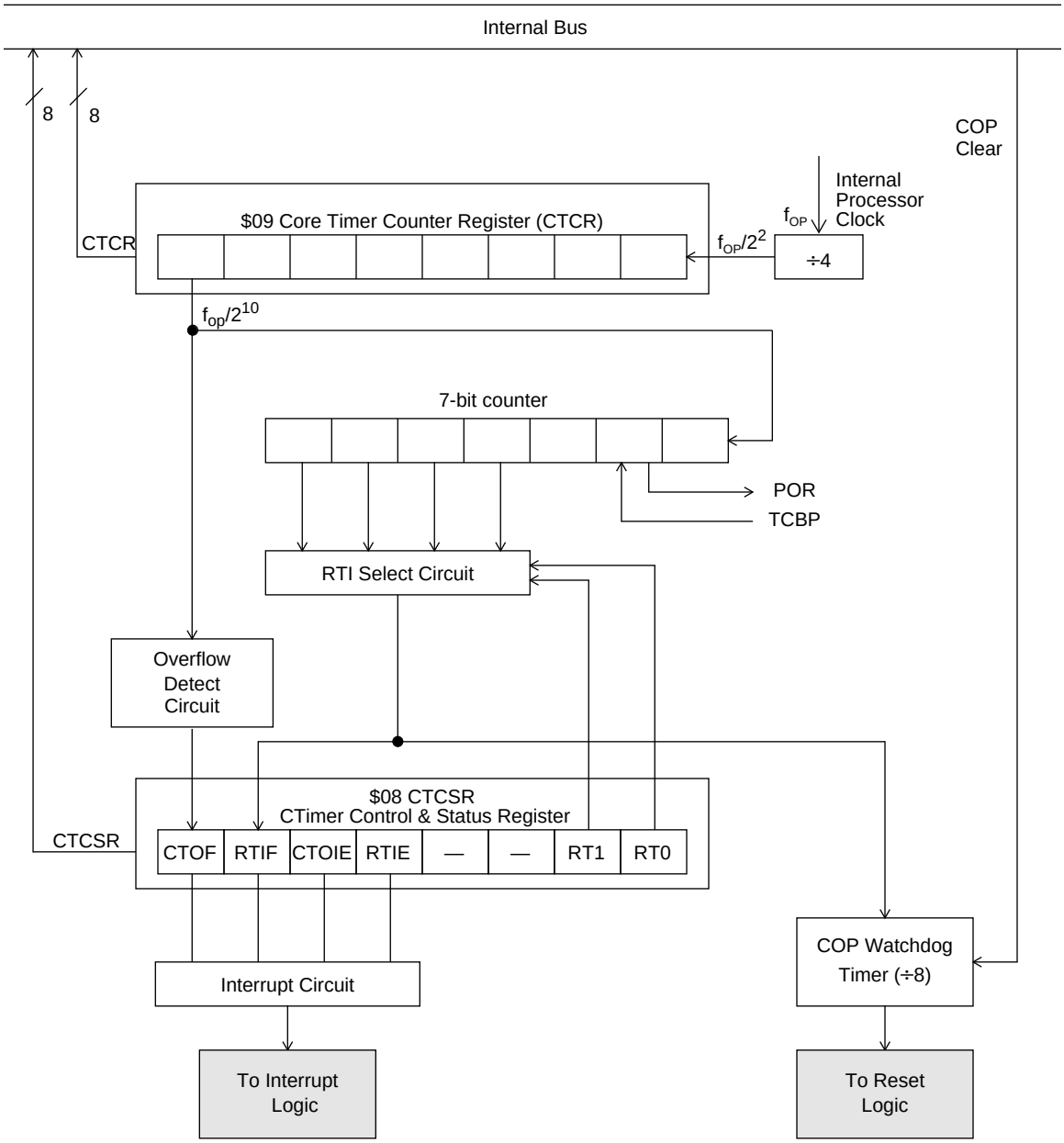


Figure 11-1: Core Timer Block Diagram

As seen in **Figure 11-1: Core Timer Block Diagram**, the Core Timer is driven by the internal bus clock divided by four as a fixed prescaler. This signal drives an 8-bit ripple counter. The value of this 8-bit ripple counter can be read by the CPU at any time by accessing the Ctimer Counter Register (CTCR) at address $09. A timer overflow function is implemented on the last stage of this counter, giving a possible interrupt at the rate of E/1024. Two additional stages produce the POR function at E/4064. The Timer Counter Bypass circuitry (available only in Test Mode) is at this point in the timer chain. This circuit is followed by one more stage, with the resulting clock (E/8192) driving the Real Time Interrupt circuit. The RTI circuit consists of four divider stages with a 1 of 4 selector. The output of the RTI circuit is further divided by eight to drive the mask optional COP Watchdog Timer circuit. The RTI rate selector bits, and the RTI and CTOF enable bits and flags are located in the Ctimer Control and Status Register(CTCSR) at location $08.

11.1 CTIMER CONTROL AND STATUS REGISTER (CTCSR) $08

The CTCSR contains the timer interrupt flag, the timer interrupt enable bits, and the real time interrupt rate select bits. **Figure 11-2: Core Timer Control and Status Register (CTCSR)** shows the value of each bit in the CTCSR when coming out of reset.

$08	CTOF	RTIF	CTOIE	RTIE	--	--	RT1	RT0
RESET:	0	0	0	0	0	0	1	1

Figure 11-2: Core Timer Control and Status Register (CTCSR)

11.1.1 CTOF - Core Timer Overflow Flag

CTOF is a clearable, read-only status bit and is set when the 8-bit ripple counter rolls over from $FF to $00. A CPU interrupt request will be generated if CTOIE is set. Clearing the CTOF is done by writing a ‘0’ to it. Writing a ‘1’ to CTOF has no effect on the bit’s value. Reset clears CTOF.

11.1.2 RTIF - Real Time Interrupt Flag

The Real Time Interrupt circuit consists of a four stage divider and a 1 of 4 selector. The clock frequency that drives the RTI circuit is $E/2^{13}$ (or E/8192) with four additional divider stages giving a maximum interrupt period of 4 seconds at a crystal frequency of 32.768 kHz. RTIF is a clearable, read-only status bit and is set when the output of the chosen (1 of 4 selection) stage goes active. A CPU interrupt request will be generated if RTIE is set. Clearing the RTIF is done by writing a ‘0’ to it. Writing a ’1’ to RTIF has no effect on this bit. Reset clears RTIF.

11.1.3 CTOIE - Core Timer Overflow Interrupt Enable

When this bit is set, a CPU interrupt request is generated when the CTOF bit is set. Reset clears this bit.

11.1.4 RTIE - Real Time Interrupt Enable

When this bit is set, a CPU interrupt request is generated when the RTIF bit is set. Reset clears this bit.

11.1.5 RT1:RT0 - Real Time Interrupt Rate Select

These two bits select one of four taps from the Real Time Interrupt circuit.**Table 11-1: RTI Rates** shows the available interrupt rates with several f_{OP} values. Reset sets these RT0 and RT1, selecting the lowest periodic rate and therefore the maximum time in which to alter these bits if necessary. Care should be taken when altering RT0 and RT1 if the time-out period is imminent or uncertain. If the selected tap is modified during a cycle in which the counter is switching, an RTIF could be missed or an additional one could be generated. To avoid problems, the COP should be cleared before changing RTI taps.

Table 11-1: RTI Rates

	RTI RATES AT BUS FREQUENCY OF:		
RT1:RT0	**16.384 kHz**	**3.0 MHz**	**Div. Ratio**
00	1 s	5.5 ms	2^{14}
01	2 s	10.9 ms	2^{15}
10	4 s	21.8 ms	2^{16}
11	8s	43.75 ms	2^{17}

11.2 COMPUTER OPERATING PROPERLY (COP) WATCHDOG RESET

The COP watchdog timer function is implemented on this device by using the output of the RTI circuit and further dividing it by eight. The minimum COP reset rates are listed in **11.3: CTIMER COUNTER REGISTER (CTCR) $09**. If the COP circuit times out, an internal reset is generated and the normal reset vector is fetched. Preventing a COP time-out is done by writing a ‘0’ to bit 0 of address $0FF0. When the COP is cleared, only the final divide by eight stage (output of the RTI) is cleared. This function is a mask option.

Table 11-2: Minimum COP Reset Times

	MINIMUM COP RESET AT Bus FREQUENCY:		
RT1:RT0	**16.384 kHz**	**3.0 MHz**	**f_{op}**
00	7 s	38.2 ms	7 × (RTI RATE)
01	14 s	76.5 ms	7 × (RTI RATE)
10	28 s	153.0 ms	7 × (RTI RATE)
11	56 s	305.9 ms	7 × (RTI RATE)

11.3 CTIMER COUNTER REGISTER (CTCR) $09

The Core Timer Counter Register is a read-only register that contains the current value of the 8-bit ripple counter at the beginning of the timer chain. This counter is clocked at f_{op}

divided by 4 and can be used for various functions including a software input capture. Extended time periods can be attained using the CTOF function to increment a temporary RAM storage location thereby simulating a 16-bit (or more) counter.

$09	CTCR7	CTCR6	CTCR5	CTCR4	CTCR3	CTCR2	CTCR1	CTCR0

Figure 11-3: Core Timer Counter Register

The power-on cycle clears the entire counter chain and begins clocking the counter. After 4064 cycles, the power-on reset circuit is released that again clears the counter chain and allows the device to come out of reset. At this point, if $\overline{\text{RESET}}$ is not asserted, the timer will start counting up from zero and normal device operation will begin. When $\overline{\text{RESET}}$ is asserted anytime during operation (other than POR), the counter chain will be cleared.

11.4 CORE TIMER DURING WAIT MODE

The CPU clock halts during the WAIT mode, but the timer remains active. If the interrupts are enabled, the timer interrupt will cause the processor to exit the WAIT mode.

11.5 CORE TIMER DURING STOP MODE

The core timer is cleared when going into STOP mode. When STOP is exited by an external interrupt or an external RESET, the internal oscillator will resume, followed by 4064 cycles internal processor stabilization delay. The timer is then cleared and operation resumes.

SECTION 12 ELECTRICAL SPECIFICATIONS

12.1 MAXIMUM RATINGS

(Voltages referenced to V_{SS})

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	-0.3 to + 7.0	V
Input Voltage	V_{IN}	V_{SS} - 0.3 to V_{DD} + 0.3	V
Self-Check Mode ($\overline{IRQ}$ Pin Only)	V_{IN}	V_{SS} - 0.3 to $2 \times V_{DD}$ + 0.3	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	25	mA
Operating Temperature Range MC68HC705MC4 (Standard) MC68HC705MC4 (Extended) MC68HC705MC4 (Automotive)	T_A	T_L to T_H 0 to +70 -40 to +85 -40 to +105	°C
Storage Temperature Range	T_{STG}	-65 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation, it is recommended that V_{IN} and V_{OUT} be constrained to the range $V_{SS} \le (V_{IN}$ or $V_{OUT}) \le V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (i.e., either V_{SS} or V_{DD}).

12.2 THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic SOIC	θ_{JA}	 60 60	°C/W

12.3 DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc, T_A = -40°C to +TBD°C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output voltage I_{LOAD} = 10.0 µA	V_{OL} V_{OH}	— V_{DD} - 0.1	— —	0.1 —	V
Output High Voltage (I_{LOAD} = -0.8 mA) PB4-7,PC0-7, PD6/TCMP, OSC2 (I_{LOAD} = -10.0 mA) PA0-7 (Max Total I_{LOAD} = 20mA)	V_{OH} V_{OH}	V_{DD} - 0.8 V_{DD} -2.0	— —	— —	V V
Output Low Voltage (I_{Load} = 1.6 mA) PA0-7, PB4-6, PC0-7, PD6, TCMP, OSC2 (I_{Load} = 10.0 mA) PB7	V_{OL} V_{OL}	— —	— —	0.4 1.0	V V
Input High Voltage PA0-7, PB4-7, PC0-7, PD6, TCAP/PD7, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	0.7 × V_{DD}	—	V_{DD}	V
Input Low Voltage PA0-7, PB4-7, PC0-7, PD6, TCAP/ PD7, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	0.2 × V_{DD}	V
Supply Current (see Notes) Run Wait Stop 25°C 0°C to +70°C (Standard) -40°C to +85°C (Extended) -40°C to +125°C (Automotive)	 I_{DD} I_{DD} I_{DD} I_{DD} I_{DD} I_{DD}	 — — — — — —	 5 (Note 8) TBD TBD TBD TBD TBD	 TBD TBD TBD TBD TBD TBD	 mA mA µA µA µA µA
I/O Ports Hi-Z Leakage Current PA0-7, PB4-7, PC0-7, PD6, TCAP/PD7	I_{IL}	—	—	±10	µA
A/D Ports Hi-Z Leakage Current PC0-5	I_{IL}	—	—	±1	µA
Input Current $\overline{RESET}$, $\overline{IRQ}$, OSC1	I_{in}	—	—	±1	µA
Capacitance Ports (as Input or Output) $\overline{RESET}$, $\overline{IRQ}$	 C_{OUT} C_{IN}	 — —	 — —	 12 8	 pF pF
Input Injection Current PA7	I_{INJ}	—	—	±100	µA

NOTES:

1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only.
3. Wait I_{DD}: Only timer system active.
4. Run (Operating) I_{DD}, Wait 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 50pF on all outputs, C_L = 20 pF on OSC2.
5. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 V, V_{IH} = V_{DD} -0.2 V.
6. Stop I_{DD} measured with OSC1 = V_{SS}.
7. Wait I_{DD} is affected linearly by the OSC2 capacitance.
8. Pre-silicon estimate.

12.4 A/D CONVERTER CHARACTERISTICS

(V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc, T_A = -40°C to +TBD°C, unless otherwise noted)

Characteristic	Min	Max	Unit	Comments
Resolution	8	8	Bits	
Absolute Accuracy ($V_{DD} \geq V_{REFH} > 4.0$)	—	±1 1/2	LSB	Including quantization
Conversion Range V_{REFH}	V_{SS} V_{SS}	V_{REFH} V_{DD}	V V	A/D accuracy may decrease proportionately as V_{REFH} is reduced below 4.0
Input Leakage AD0, AD1, AD2, AD3, AD4, AD5 V_{REFH}	— —	±1 ±1	µA µA	
Conversion Time (Includes Sampling Time)	32	32	t_{AD}*	
Monotonicity	Inherent (Within Total Error)			
Zero Input Reading	00	01	Hex	V_{IN} = 0V
Full-scale Reading	FE	FF	Hex	$V_{IN} = V_{REFH}$
Sample Time	12	12	t_{AD}*	
Input Capacitance	—	12	pF	
Analog Input Voltage	V_{SS}	V_{REFH}	V	

*$t_{AD} = t_{CYC}$ if clock source equals MCU.

12.5 CONTROL TIMING

(V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc, T_A = -40°C to +TBD°C, unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Crystal Option External Clock Option	 f_{OSC} f_{OSC}	 — dc	 6 6	 MHz MHz
Internal Operating Frequency Crystal (f_{osc} ÷ 2) External Clock (f_{osc} ÷ 2)	 f_{OP} f_{OP}	 — dc	 3 3	 MHz MHz
Cycle Time	t_{CYC}	333	—	ns
Crystal Oscillator Start-up Time	t_{OXOV}	—	100	ms
Stop Recovery Start-up Time (Crystal Oscillator)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ Pulse Width	t_{RL}	1.5	—	t_{CYC}
Interrupt Pulse Width Low (Edge-Triggered)	t_{ILIH}	TBD	—	ns
Interrupt Pulse Period	t_{ILIL}	*	—	t_{CYC}
OSC1 Pulse Width	t_{OH}, t_{OL}	200	—	ns
A/D On Current Stabilization Time	t_{ADON}	—	100	µs
RC Oscillator Stabilization Time (A/D)	t_{RCON}	—	5.0	µs

* The minimum period T_{ILIL} should not be less than the number of cycles it takes to execute the interrupt service routine plus 19 t_{CYC}.

CONTROL TIMING (CONTINUED)

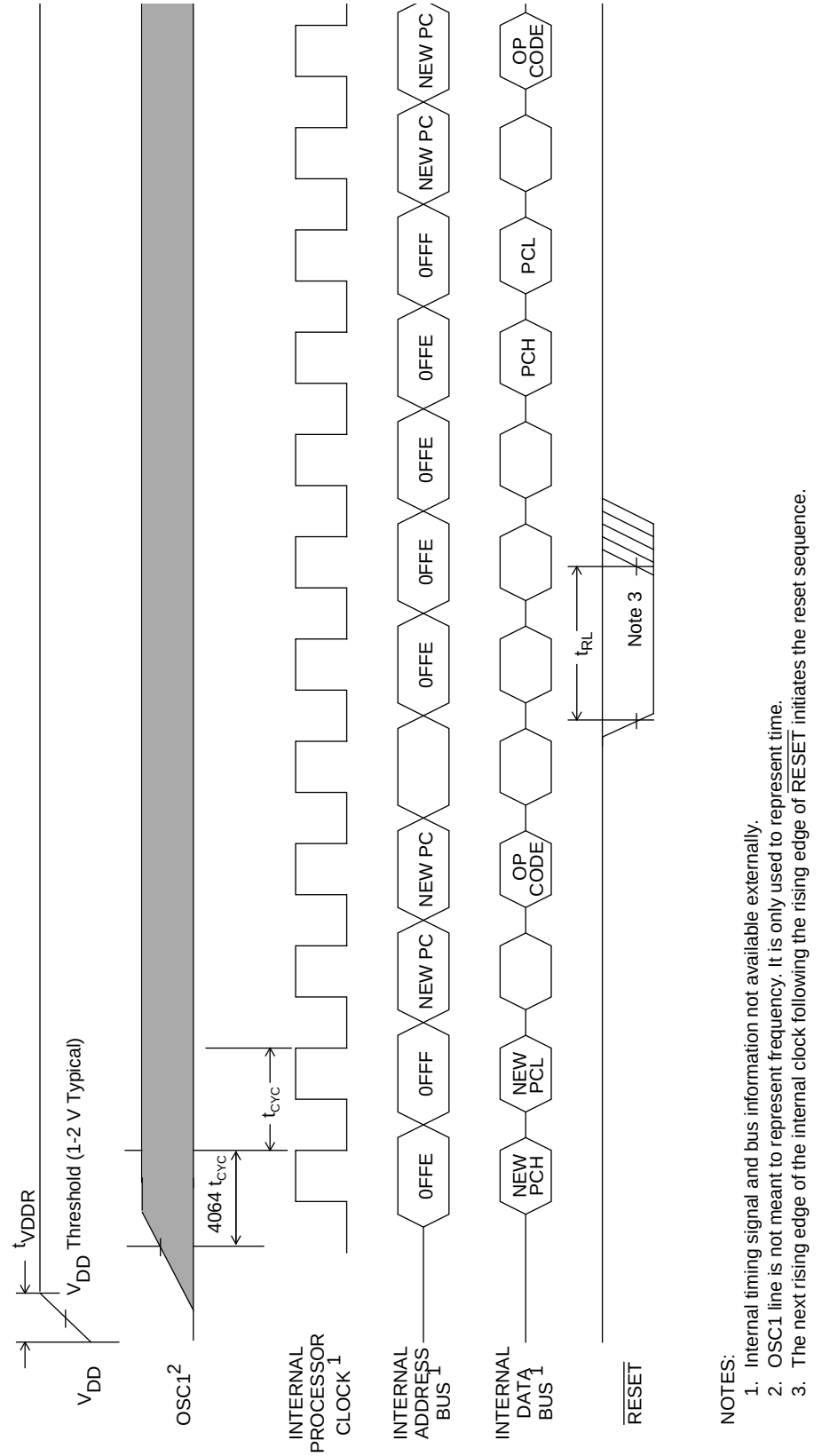


Figure 12-1: Power-On Reset and External Reset Timing Diagram

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

MOTOROLA

HC705MC4GRS/D