68HC05X1

SPECIFICATION

REV 2.1

(General Release)

June 29, 1994

Microprocessor & Memory Technologies Group
Austin, Texas

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SECTION 1 INTRODUCTION

This document defines and describes the operation of the MC68HC05X1. This device is intended to be an “intelligent” node that resides on a SAEJ1850 serial data communications multiplex bus. Key features include:

- MC68HC05 CPU
- 12304 bytes user ROM
- 240 bytes Self-Check ROM
- 336 bytes RAM
- SAEJ1850 PWM Serial Multiplex Interface (SMI)
- 15-stage Multifunctional Timer with periodic interrupt
- 16-Bit Timer with input capture and two output compares
- Three 8-Bit I/O ports (one with wake-up capability)
- Synchronous Serial Interface (SSI) for further peripheral expansion
- Computer Operating Properly (COP) Watchdog
- 2.0 MHz nominal CPU Bus rate
- Crystal or Ceramic resonator oscillator
- 44-Pin plastic leaded chip carrier (PLCC) package
- -40 to +85 degrees Centigrade operation

THIS PAGE INTENTIONALLY LEFT BLANK

SECTION 2 OUTLINE

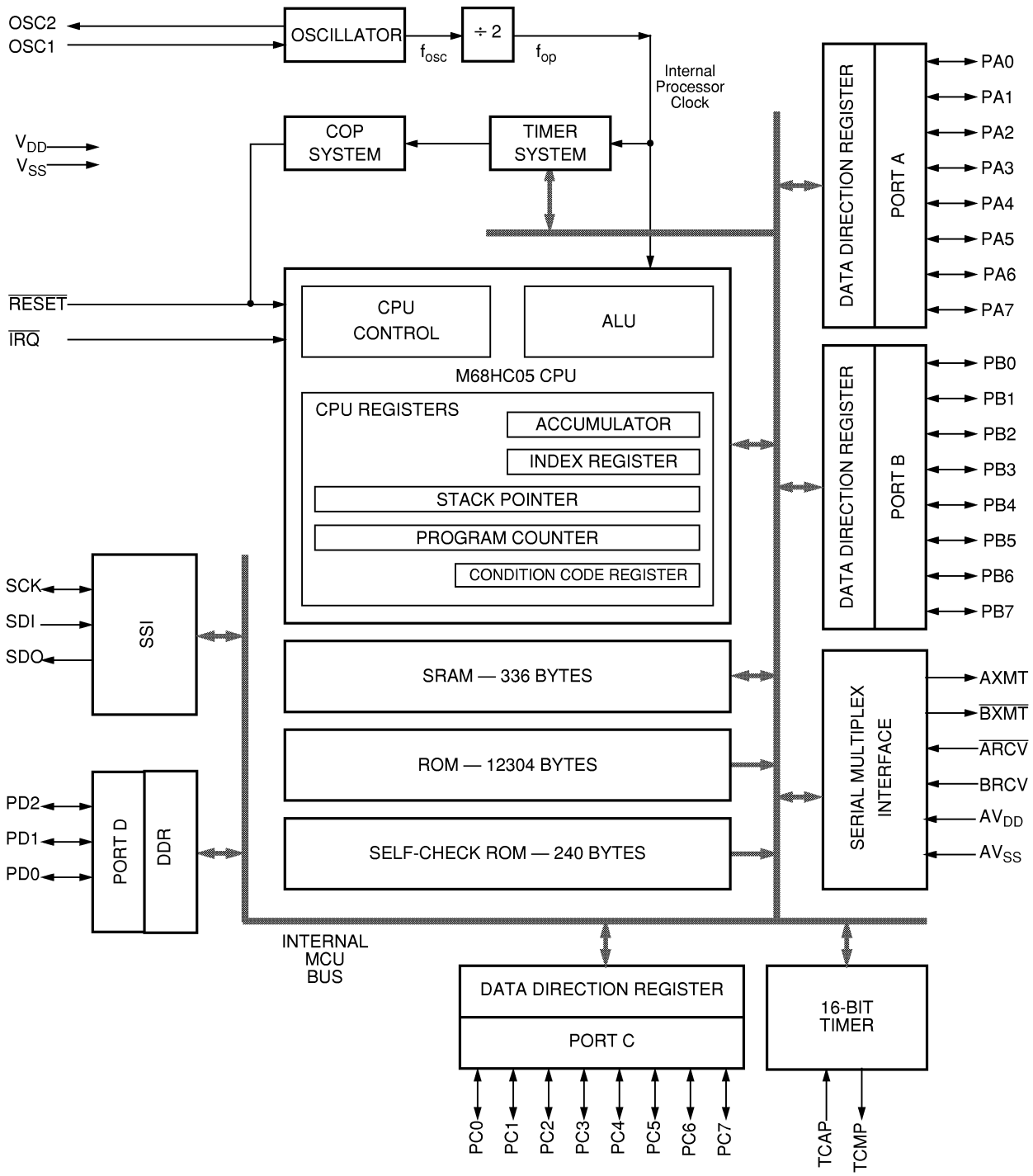


Figure 2-1: Block Diagram

2.1 FUNCTIONAL PIN DESCRIPTION

In the following descriptions, a line over a signal name indicates an active low signal. All other signals are active high. For example, $\overline{\text{RESET}}$ is active low.

2.1.1 V_{DD}, V_{SS}, AV_{DD}, AV_{SS}

Power is supplied to the microcontroller using these four pins.

V_{DD} is the positive supply to the digital sections and AV_{DD} is the positive supply to the analog sections of the device.

V_{SS} is the ground for the digital sections and AV_{SS} is the ground for the analog sections of the device.

Good supply decoupling, as close as possible to these pins, is absolutely essential to guarantee correct operation, minimize radio frequency interference (RFI) and maximize analog performance.

Figure 2-2: Recommended MC68HC05X1 Supply Scheme outlines the recommended supply scheme for this device. Note that digital and analog supply lines are kept separate once they leave the voltage regulator. Supplies to other components in the same module should also be separated to minimize interaction with the MCU.

Emphasis should be placed on keeping supply and ground line impedances as low as possible, with preference given to ground lines if a compromise is necessary. Ground planes and wide, short, printed circuit board (PCB) traces can help significantly here.

Decoupling capacitors should be of low impedance construction, with regard to high frequencies, and placed as close as is physically possible to the supply pins of the MCU and Module Voltage Regulator.

See Motorola Application Note *Designing for EMC with HCMOS Microcontrollers* (AN1050/D) for further information.

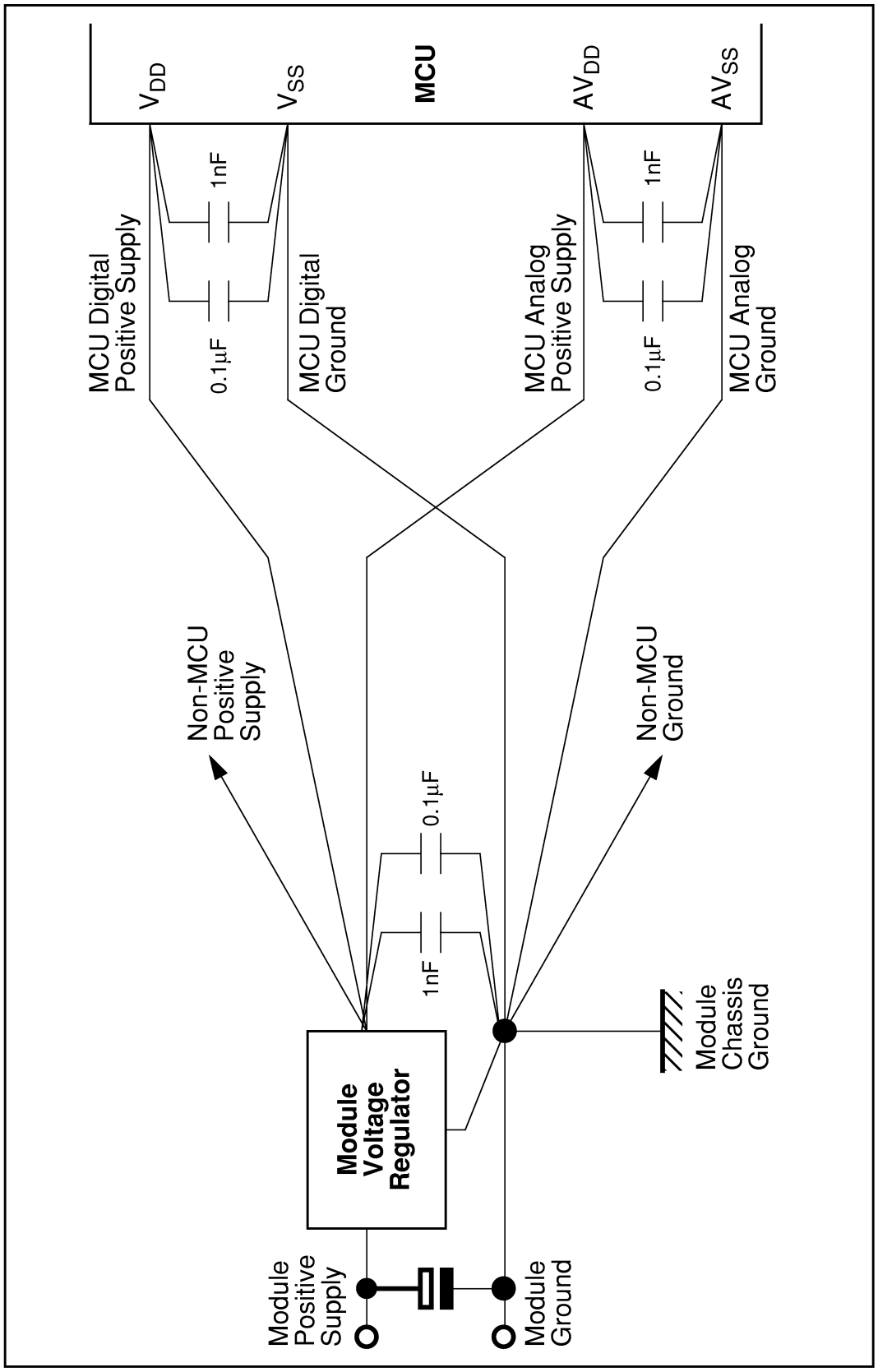


Figure 2-2: Recommended MC68HC05X1 Supply Scheme

2.1.2 $\overline{\text{IRQ}}$ (MASKABLE INTERRUPT REQUEST)

This pin has a programmable option that provides two different choices of interrupt triggering sensitivity. The options are:

1. Negative edge-sensitive triggering only.
2. Both negative edge-sensitive and level-sensitive triggering.

The MCU completes the current instruction before it responds to the interrupt request. When $\overline{\text{IRQ}}$ goes low 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 is clear, the MCU then begins the interrupt sequence.

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

The $\overline{\text{IRQ}}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity. Refer to **5.5 INTERRUPTS** for more detail.

NOTE: The voltage on this pin affects the mode of operation. See **SECTION 4 OPERATING MODES.**

2.1.3 OSC1, OSC2

These pins provide control input for an on-chip clock oscillator circuit. A crystal, a ceramic resonator, or an external signal connects to these pins providing a system clock.

2.1.3.1 CRYSTAL

The circuit shown in **Figure 2-3: Oscillator Connections**(a) is recommended when using a crystal. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and start-up stabilization time.

2.1.3.2 CERAMIC RESONATOR

A ceramic resonator may be used in place of a crystal in cost-sensitive applications. The circuit shown in **Figure 2-3: Oscillator Connections**(a) is recommended, but consult the manufacturer of the particular ceramic resonator for specific information.

The resonator and components should be mounted as close as possible to the input pins to minimize output distortion and start-up stabilization time.

2.1.3.3 RESISTOR

The value of the resistor, placed in parallel with the crystal or ceramic resonator, may be between 1 MΩ and 20 MΩ. Typically, a value of 10 MΩ is used. Consult the manufacturer of the particular ceramic resonator or crystal for their recommendations.

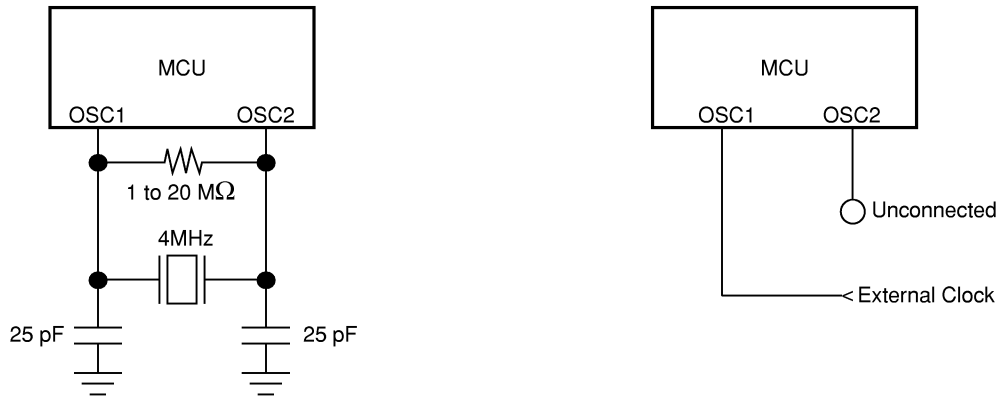
2.1.3.4 POWER-UP AND STOP MODE RECOVERY

In the STOP mode of operation, the OSC2 pin is held high, causing the OSC1 pin to be driven high via the resistor, and the oscillator is rendered inactive. The value of the resistor thus has no effect on STOP mode power consumption.

On exiting STOP mode, the OSC2 pin is immediately driven low. This rapidly “kicks” the crystal or resonator into oscillation and stable oscillations will occur much sooner than at power-up (when there is little potential across the crystal or resonator until oscillations build).

2.1.3.5 EXTERNAL CLOCK

An external clock should be applied to the OSC1 input with the OSC2 pin not connected, as shown in **Figure 2-3: Oscillator Connections**(b). This setup can be used if the user does not wish to run the CPU with a crystal or ceramic resonator.



(a) Crystal/Ceramic Resonator Oscillator Connections

(b) External Clock Source Connections

Figure 2-3: Oscillator Connections

2.1.4 $\overline{\text{RESET}}$

This active low pin is used to reset the MCU to a known start-up state by pulling $\overline{\text{RESET}}$ low. The $\overline{\text{RESET}}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity. See **5.4 RESETS** for more information.

2.1.5 $\overline{\text{ARCV}}$, BRCV, AXMT, $\overline{\text{BXMT}}$

These four pins are associated with the Serial Multiplex Interface (SMI) and provide the means of connection to the external Physical Layer Interface. Pins $\overline{\text{ARCV}}$ and BRCV are always analog inputs and should be connected, through the specified network, to the external physical layer. Pins AXMT and $\overline{\text{BXMT}}$ are always outputs and should be connected to the external physical layer driver circuit. See **7.3.1 PINS** for more details.

2.1.6 TCAP, TCMP

These two pins are associated with the 16-Bit programmable Timer. TCAP is always an input and controls the input capture feature. TCMP is always an output for the output compare feature of the timer. See **9.5 INPUT CAPTURE REGISTER (ICR)** and **9.3 OUTPUT COMPARE REGISTER 1 (OCR1)** for more details.

2.1.7 SCLK, SDI, SDO

These three pins are associated with the Synchronous Serial Interface (SSI). SCLK may be an input or an output clock signal, depending upon the programmed function. SDI is always the serial data input and SDO is always the serial data output (but will be three stated if the SSI is disabled, or the MCU is reset). See **10.1.1 SERIAL CLOCK (SCK)**, **10.1.3 SERIAL DATA IN (SDI)** and **10.1.2 SERIAL DATA OUT (SDO)** for more details.

2.1.8 PA0-PA7

These eight I/O lines comprise Port A. The state of any pin is software programmable and all Port A lines are initialized as inputs during power-on or reset. See **6.1 PORT A** for more details.

2.1.9 PB0-PB7

These eight I/O lines comprise Port B. The state of any pin is software programmable and all Port B lines are initialized as inputs during power-on or reset. See **6.2 PORT B** for more details.

2.1.10 PC0-PC7

These eight I/O lines comprise Port C. The state of any pin is software programmable and all Port C lines are initialized as inputs during power-on or reset. See **6.3 PORT C** for more details.

2.1.11 PD0-PD2

These three I/O lines comprise Port D. The state of any pin is software programmable and all Port D lines are initialized as inputs during power-on or reset. See **6.4 PORT D** for more details.

2.2 MASK OPTIONS

There are five mask options on the MC68HC05X1: Stop Recovery Time (Fast [1024 cycles] or Normal [4096 cycles]), STOP instruction (enable/disable), $\overline{\text{IRQ}}$ (Edge-sensitive only or Edge- and level-sensitive), Port C Wake-up (enable/disable) and COP Watchdog Timer (enable/disable).

NOTE: If the Fast Stop Recovery Time option is selected, a ceramic resonator **must** be used to control the frequency of operation. Furthermore, the ceramic resonator **must** assume stable operation before the mask programmed number of cycles completes, under all possible operating conditions.This will ensure correct general operation of the MC68HC05X1 when STOP mode is exited.

THIS PAGE INTENTIONALLY LEFT BLANK

SECTION 3 MEMORY

The MC68HC05X1 has a 16 Kbyte memory map, consisting of user ROM, user RAM, Self-Check ROM, Control Registers, and I/O.

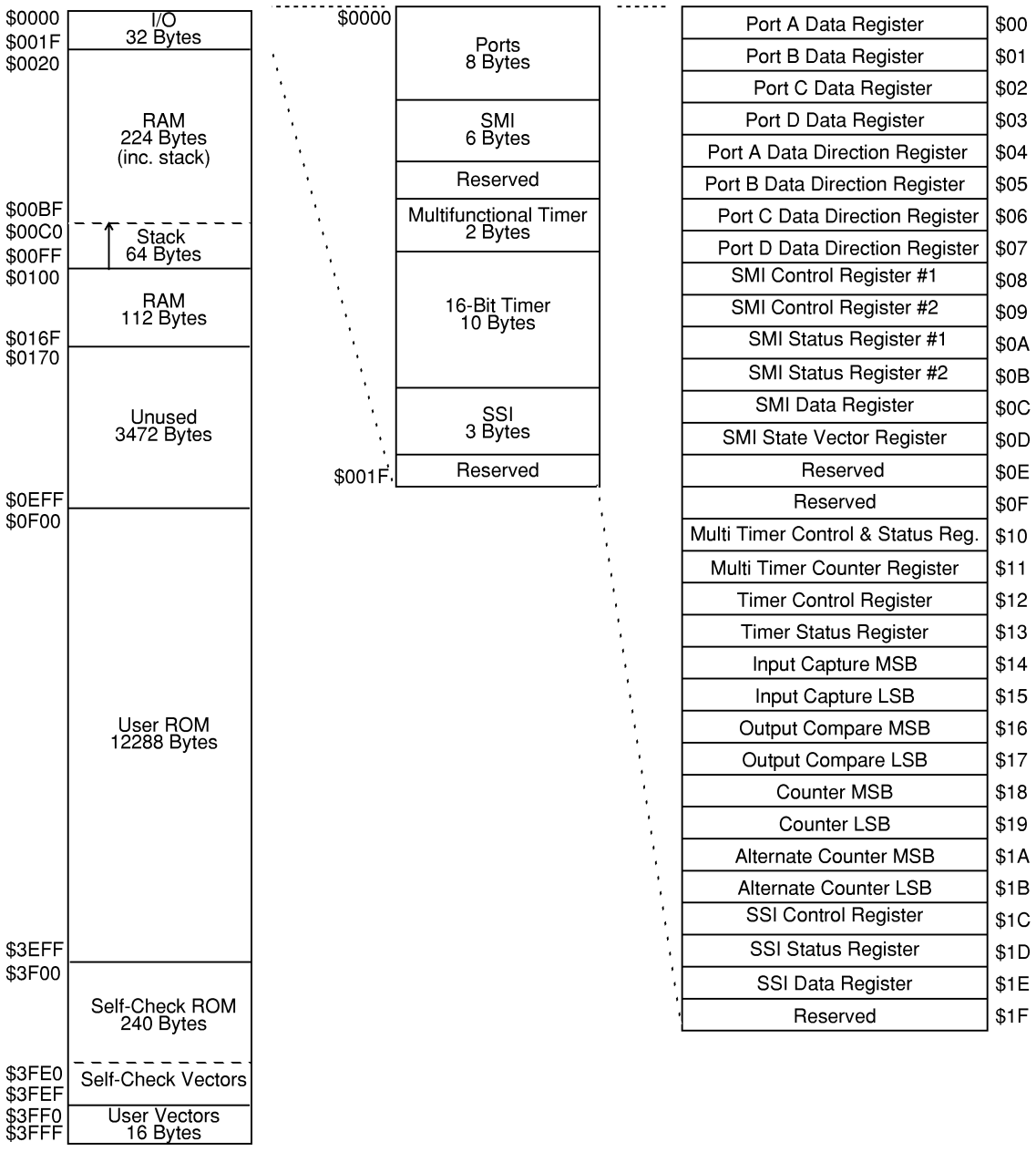


Figure 3-1: Memory Map

3.1 ROM

12288 bytes of user ROM are located from $0F00 to $3FFF, with 16 additional bytes of user vectors from $3FF0 to $3FFF. The Self-Check ROM and vectors are located from $3F00 to $3FEF.

3.2 RAM

The user RAM consists of 336 bytes from location $0020 to $016F including the stack area. The stack begins at address $00FF. The stack pointer can access 64 bytes of RAM from $00FF to $00C0. 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.

3.3 I/O

The I/O occupies 32 bytes from $0000 to $001F. A complete bit map for all of the registers is shown in the following figure.

All bit positions marked with a “0” will always be read as a logic zero. Writes to these bits will have no effect.

ADDRESS $0000 TO $001F	DATA							
	7	6	5	4	3	2	1	0
$00 PORT A DATA	D7	D6	D5	D4	D3	D2	D1	D0
$01 PORT B DATA	D7	D6	D5	D4	D3	D2	D1	D0
$02 PORT C DATA	D7	D6	D5	D4	D3	D2	D1	D0
$03 PORT D DATA	0	0	0	0	0	D2	D1	D0
$04 PORT A DDR	DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0
$05 PORT B DDR	DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0
$06 PORT C DDR	DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0
$07 PORT D DDR	0	0	0	0	0	DIR2	DIR1	DIR0
$08 SMI CONTROL #1	ARSOM	IMSG	HACK	SDACK	SBACK	SACK	SEOD	SSOM
$09 SMI CONTROL #2	RDRF	SMR1	SMR0	LOFF	BXEN	AXEN	IE	SME
$0A SMI STATUS #1	MORE	RPEOD	RCRC	RIB	REOM	REOD	LARB	RSOM
$0B SMI STATUS #2	TXFLT	FLTB	FLTD	FLTA	TJAB	BSOM	ROR	TUR
$0C SMI DATA	D7	D6	D5	D4	D3	D2	D1	D0
$0D SMI STATE VECTOR	SU3	SU2	SU1	SU0	I1	I0	0	0
$0E Reserved	-	-	-	-	-	-	-	-
$0F Reserved	-	-	-	-	-	-	-	-
$10 M.TIMER CTRL/STAT	TOF	RTIF	TOFE	RTIE	0	0	RT1	RT0
$11 M.TIMER COUNTER	D7	D6	D5	D4	D3	D2	D1	D0
$12 TIMER CONTROL	ICIE	OCIE1	TOIE	OCIE2	OCR2E	OLVL2	IEDG	OLVL1
$13 TIMER STATUS	ICF	OCF1	TOF	OCF2	0	0	0	0
$14 CAPTURE HIGH	D7	D6	D5	D4	D3	D2	D1	D0
$15 CAPTURE LOW	D7	D6	D5	D4	D3	D2	D1	D0
$16 COMPARE HIGH 1/2	D7	D6	D5	D4	D3	D2	D1	D0
$17 COMPARE LOW 1/2	D7	D6	D5	D4	D3	D2	D1	D0
$18 COUNTER HIGH	D7	D6	D5	D4	D3	D2	D1	D0
$19 COUNTER LOW	D7	D6	D5	D4	D3	D2	D1	D0
$1A ALT. COUNT HIGH	D7	D6	D5	D4	D3	D2	D1	D0
$1B ALT. COUNT LOW	D7	D6	D5	D4	D3	D2	D1	D0
$1C SSI CONTROL	SIE	SE	LSBF	MSTR	CPOL	0	SR1	SR0
$1D SSI STATUS	SF	DCOL	0	0	0	0	0	0
$1E SSI DATA	D7	D6	D5	D4	D3	D2	D1	D0
$1F Reserved	-	-	-	-	-	-	-	-

Figure 3-2: I/O Map

THIS PAGE INTENTIONALLY LEFT BLANK

SECTION 4 OPERATING MODES

The MCU has two modes of operation: Single-Chip (user) Mode and Self-Check Mode. **Table 4-1: Operating Mode Conditions** shows the conditions required to go into each mode.

Table 4-1: Operating Mode Conditions

V_{TST} = 12 V (Typical)

$\overline{RESET}$	$\overline{IRQ}$	PB1	MODE
	V_{SS}-V_{DD} V_{TST}	V_{SS}-V_{DD} V_{DD}	Single-chip Self-Check

4.1 SINGLE-CHIP MODE

In Single-Chip Mode, the address and data buses are not available externally, but there are three 8-bit I/O ports and one 3-bit I/O port. This 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. Single-Chip Mode is entered on the rising edge of $\overline{\text{RESET}}$ if the $\overline{\text{IRQ}}$ pin is within normal operating range.

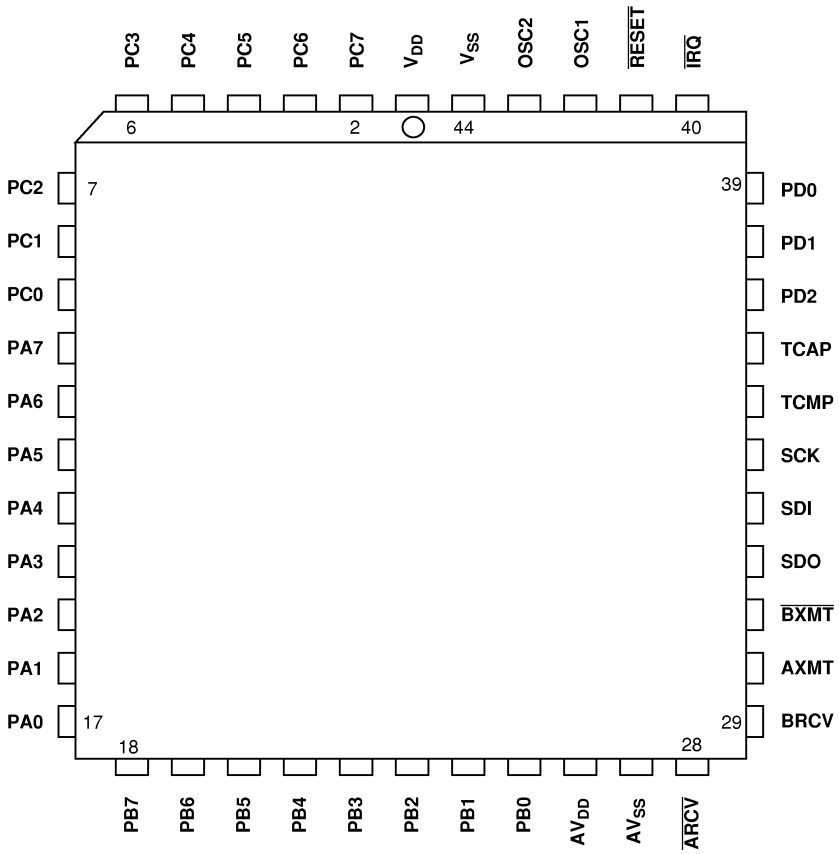


Figure 4-1: Single-Chip Mode Pinout

4.2 SELF-CHECK MODE

The Self-Check Mode provides a simple means of partially testing the device. Some internal checks are made to confirm that the device is basically functional. The tests are not exhaustive. The recommended Self-Check configuration is shown below. A ceramic resonator or crystal may be used instead of the external clock, shown below.

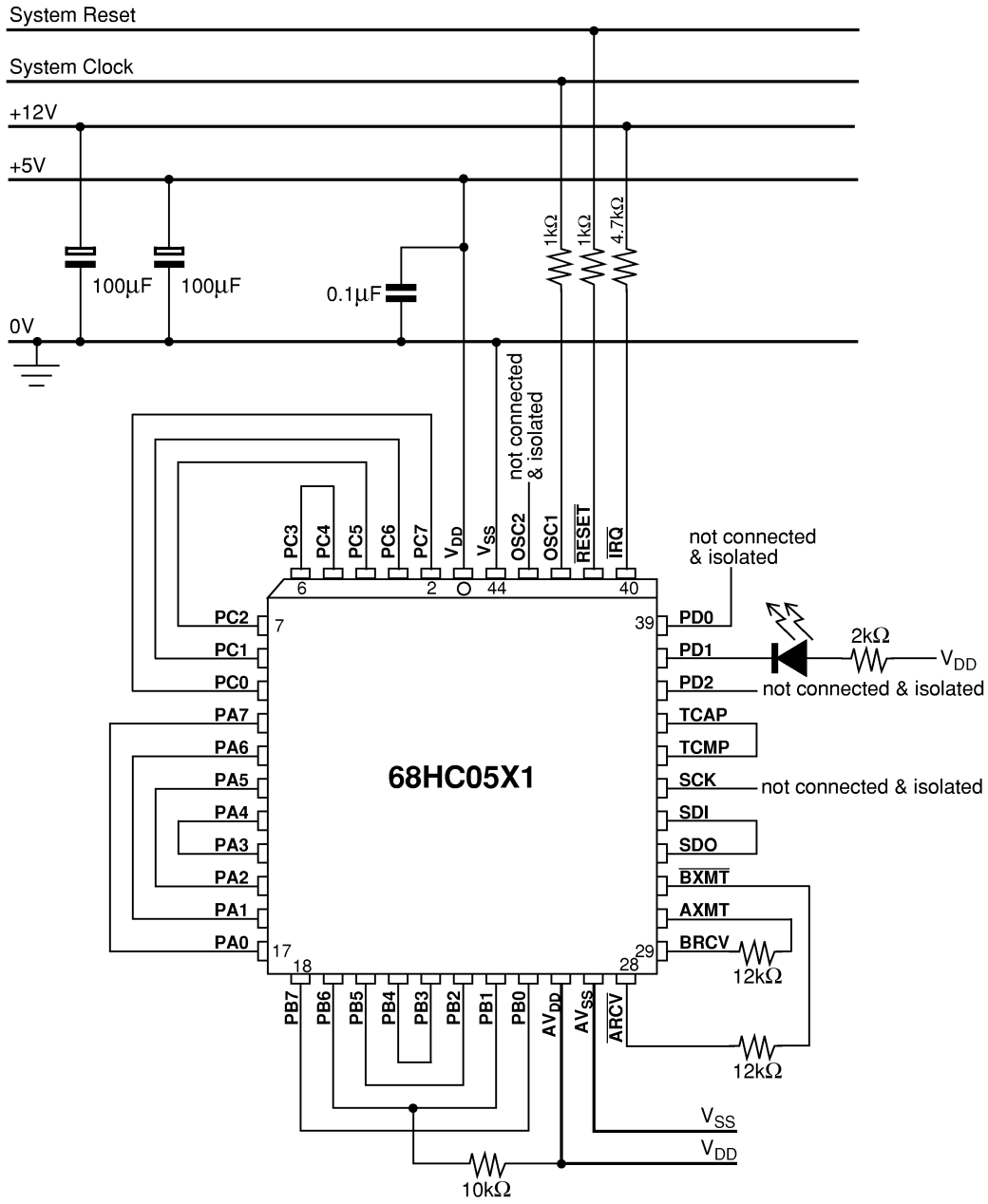


Figure 4-2: Self-Check Schematic

The Self-Check Mode is entered on the rising edge of $\overline{\text{RESET}}$ if the $\overline{\text{IRQ}}$ pin is at V_{TST} and the PB1 pin is at a logic one. $\overline{\text{RESET}}$ must be held low for 4096 cycles after POR, or for a time t_{RL} for any other reset.

After entering the Self-Check Mode, all of the MC68HC05X1 peripherals are exercised by a small program, included in all parts by Motorola, which resides in the top end of the memory map. Some of the peripheral functions are also tested by the program, though not exhaustively.

Ports A, B, and C are repeatedly configured as inputs and outputs. They are made to output and input logic ones and logic zeros. Port D is always configured as an output so as to provide some indication of the test progression.

The Multifunctional Timer continuously runs but its flags are ignored and its interrupts are disabled. The 16-Bit Timer continuously runs and is programmed to generate a squarewave out of the TCMP pin. This is fed back into the TCAP pin to exercise the output compare #1 and input capture circuits.

The SSI is configured to one mode of operation and is made to send 1-byte messages, with all possible data values. The SDO pin is connected to the SDI pin, allowing the transmitted messages to be received back and compared to the data sent.

The SMI is configured to one mode of operation and is made to send 1-byte messages, with all possible data values. The transmitter pins are connected to the receiver pins, allowing the transmitted messages to be received back and compared to the data sent.

All RAM locations are complemented and then re-complemented (to bring the data back to its original value). Checks are made that both write operations are successful. A checksum of the entire ROM is made and compared against the expected value. If a test fails, Port D will remain in a fixed state. The state of Port D gives an indication of the nature of the fault. The eight possibilities are described below.

Table 4-2: Self-Check Results

PD2	PD1	PD0	Remarks
0	0	0	Bad ROM
0	0	1	Bad RAM
0	1	0	Bad SSI
0	1	1	Bad SMI SOM/TDRE
1	0	0	Bad SMI Receiver/RDRF
1	0	1	Bad SMI Received Data
1	1	0	Bad SMI CRC
1	1	1	Bad SMI EOD/EOM
Cycling			Good Device
All Others			Bad Device

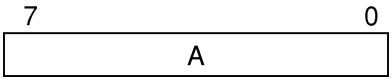
SECTION 5 CPU CORE

5.1 REGISTERS

The MCU contains the registers described in the following paragraphs.

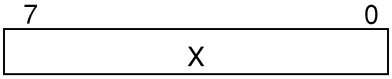
5.1.1 ACCUMULATOR (A)

The accumulator is a general purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.



5.1.2 INDEX REGISTER (X)

The index register is an 8-bit register used for the indexed addressing value to create an effective address. The index register may also be used as a temporary storage area.



5.1.3 CONDITION CODE REGISTER (CCR)

The CCR is a 5-bit register in which four bits are used to indicate the results of the instruction just executed, and the fifth bit indicates whether interrupts are masked. These bits can be individually tested by a program, and specific actions can be taken as a result of their state. Each bit is explained in the following paragraphs.

5.1.3.1 Half Carry (H)

This bit is set during ADD and ADC operations to indicate that a carry occurred between bits 3 and 4.

5.1.3.2 Interrupt (I)

When this bit is set, all further interrupts (except SWI) are masked (disabled). If an interrupt occurs while this bit is set, the interrupt is latched and processed as soon as the interrupt bit is cleared.

5.1.3.3 Negative (N)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative.

5.1.3.4 Zero (Z)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

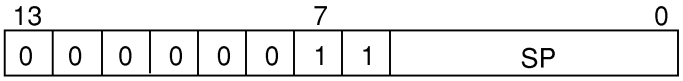
5.1.3.5 Carry/Borrow (C)

When set, this bit indicates that a carry or borrow out of the arithmetic logical unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions and during shifts and rotates.

5.1.4 STACK POINTER (SP)

The stack pointer contains the address of the next free location on the stack. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $00FF. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is pulled from the stack.

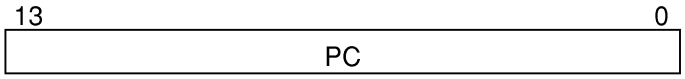
When accessing memory, the seven most significant bits are permanently set to 0000011. These seven bits are appended to the six least significant register bits to produce an address within the range of $00FF to $00C0. Subroutines and interrupts may use up to 64 (decimal) locations. If 64 locations are exceeded, the stack pointer wraps around and loses the previously stored information. A subroutine call occupies two locations on the stack; an interrupt uses five locations.



5.1.5 PROGRAM COUNTER (PC)

The program counter is a 14-bit register that contains the address of the next byte to be fetched.

NOTE: The HC05 CPU core is capable of addressing 16-bit locations. For this implementation, however, the addressing registers are limited to a 16 Kbyte memory map.



5.2 INSTRUCTION SET

The MCU has a set of 62 basic instructions. They can be divided into five different types: register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type. The instruction set and the number of cycles taken per instruction are identical to the MC68HC05C4. For more information on the instruction set and cycle times, refer to the *M6805 Family User's Manual* (M6805UM/AD3) or the *MC68HC05C4 Technical Data* (MC68HC05C4/D).

This MCU uses all the instructions available in the M146805 CMOS Family plus one more: the unsigned multiply (MUL) instruction. This instruction allows unsigned multiplication of the contents of the accumulator (A) and the index register (X). The high-order product is then stored in the index register, and the low-order product is stored in the accumulator. A detailed definition of the MUL instruction is shown below.

Operation	X:A X*A			
Description	Multiplies the eight bits in the index register by the eight bits in the accumulator to obtain a 16-bit unsigned number in the concatenated accumulator and index register.			
Condition Codes	H: Cleared I: Not affected N: Not affected Z: Not affected C: Cleared			
Source	MUL			
Form(s)	Addressing Mode Inherent	Cycles 11	Bytes 1	Opcode $42

5.2.1 REGISTER/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to the following instruction list.

Function	Mnemonic
Load A from Memory	LDA
Load X from Memory	LDX
Store A in Memory	STA
Store X in Memory	STX
Add Memory to A	ADD
Add Memory and Carry to A	ADC
Subtract Memory	SUB
Subtract Memory from A with Borrow	SBC
AND Memory to A	AND
OR Memory with A	ORA
Exclusive OR Memory with A	EOR
Arithmetic compare A with Memory	CMP
Arithmetic Compare X with Memory	CPX
Bit Test Memory with A (Logical Compare)	BIT
Jump Unconditional	JMP
Jump to Subroutine	JSR

5.2.2 READ-MODIFY-WRITE INSTRUCTIONS

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read-modify-write sequence since it does not modify the value. Do not use these read-modify-write instructions on write-only locations. Refer to the following list of instructions.

Function	Mnemonic
Increment	INC
Decrement	DEC
Clear	CLR
Complement	COM
Negate (Twos Complement)	NEG
Rotate Left Thru Carry	ROL
Rotate Right Thru Carry	ROR
Logical Shift Left	LSL
Logical Shift Right	LSR
Arithmetic Shift Right	ASR
Test for Negative or Zero	TST
Multiply	MUL

5.2.3 BRANCH INSTRUCTIONS

This set of instruction branches if a particular condition is met; otherwise, no operation is performed. Branch instructions are 2-byte instructions. Refer to the following list for branch instructions.

Function	Mnemonic
Branch Always	BRA
Branch Never	BRN
Branch if Higher	BHI
Branch if Lower or Same	BLS
Branch if Carry Clear	BCC
Branch if Higher or Same	BHS
Branch if Carry Set	BCS
Branch if Lower	BLO
Branch if Not Equal	BNE
Branch if Equal	BEQ
Branch if Half Carry Clear	BHCC
Branch if Half Carry Set	BHCS
Branch if Plus	BPL
Branch if Minus	BMI
Branch if Interrupt Mask Bit is Clear	BMC
Branch if Interrupt Mask Bit is Set	BMS
Branch if Interrupt Line is Low	BIL
Branch if Interrupt Line is High	BIH
Branch to Subroutine	BSR

5.2.4 BIT MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any writable bit which resides in the first 256 bytes of the memory space where all port registers, port DDR's, timer, timer control, ROM, and on-chip RAM reside. An additional feature allows the software to test and branch on the state of any bit within these 256 locations. The bit set, bit clear and bit test, and branch functions are all implemented with a single instruction. For test and branch instructions, the value of the bit tested is also placed in the carry bit of the condition code register. These instructions are also read-modify-write instructions. Do not bit manipulate write-only locations. Refer to the following list for bit manipulation instructions.

Function	Mnemonic
Branch if Bit n is Set	BRSET n (n = 0. . .7)
Branch if bit n is Clear	BRCLR n (n = 0. . .7)
Set Bit n	BSET n (n = 0. . .7)
Clear Bit n	BCLR n (n = 0. . .7)

5.2.5 CONTROL INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to the following list for control instructions.

Function	Mnemonic
Transfer A to X	TAX
Transfer X to A	TXA
Set Carry Bit	SEC
Clear Carry Bit	CLC
Set Interrupt Mask Bit	SEI
Clear Interrupt Mask Bit	CLI
Software Interrupt	SWI
Return from Subroutine	RTS
Return from Interrupt	RTI
Reset Stack Pointer	RSP
No-Operation	NOP
Stop	STOP
Wait	WAIT

5.3 ADDRESSING MODES

The MCU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code for all situations. The various indexed addressing modes make it possible to locate data tables, code conversion tables, and scaling tables anywhere in the memory space. Short indexed accesses are single-byte instructions; the longest instructions (3 bytes) permit accessing tables throughout memory. Short and long absolute addressing is also included. One- or 2-byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory.

The term “effective address” (EA) is used in describing the various addressing modes. Effective address is defined as the address from which the argument for an instruction is fetched or stored.

5.3.1 IMMEDIATE

In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. The immediate addressing mode is used to access constants that do not change during program execution (for example, a constant used to initialize a loop counter).

5.3.2 DIRECT

In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single 2-byte instruction.

5.3.3 EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the 2 bytes following the opcode byte. Instructions with extended addressing mode are capable of referencing arguments anywhere in memory with a single 3-byte instruction. When using the Motorola assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction.

5.3.4 RELATIVE

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte (the offset) following the opcode is added to the PC if, and only if, the branch conditions are true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from -126 to +129 from the opcode address. The programmer need not calculate the offset when using the Motorola assembler, since it calculates the proper offset and checks to see that it is within the span of the branch.

5.3.5 INDEXED, NO OFFSET

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. This addressing mode can access the first 256 memory locations. These instructions are only 1 byte long. This mode is often used to

move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

5.3.6 INDEXED, 8-BIT OFFSET

In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the opcode. The addressing mode is useful for selecting the Kth element in an n element table. With this 2-byte instruction, K would typically be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510 ($1FE is the last location at which the instruction may begin).

5.3.7 INDEXED, 16-BIT OFFSET

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This address mode can be used in a manner similar to indexed, 8-bit offset except that this 3-byte instruction allows tables to be anywhere in memory. As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

5.3.8 BIT SET/CLEAR

In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode specifies the direct addressing of the byte in which the specified bit is to be set or cleared. Any read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single 2-byte instruction.

5.3.9 BIT TEST AND BRANCH

The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit that is to be tested and its condition (set or clear), is included in the opcode. The address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset in the third byte is added to the PC if the specified bit is set or cleared in the specified memory location. This single 3-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of memory. The span of branching is from -125 to +130 from the opcode address. The state of the tested bit is also transferred to the carry bit of the condition code register.

5.3.10 INHERENT

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator, as well as the control instruction with no other arguments, are included in this mode. These instructions are 1 byte long.

5.4 RESETS

The MCU can be reset four ways: by the initial power-on reset function, by an active low input to the $\overline{\text{RESET}}$ pin, by a COP watchdog-timer reset, and by the ILADR bit being set.

5.4.1 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 4096 internal processor clock cycle (t_{CYC}) oscillator stabilization delay after the oscillator becomes active. If the $\overline{\text{RESET}}$ pin is low at the end of this 4096 cycle delay, the MCU will remain in the reset condition until $\overline{\text{RESET}}$ goes high.

5.4.2 $\overline{\text{RESET}}$ PIN

The MCU is reset when a logic zero is applied to the $\overline{\text{RESET}}$ input for a period of one and one-half machine cycles (t_{CYC}).

5.4.3 COMPUTER OPERATING PROPERLY (COP) RESET

The MCU contains a watchdog timer that automatically times out if not reset (cleared) within a specific time by a program reset sequence. If the COP watchdog timer is allowed to time-out, an internal reset is generated to reset the MCU. Because the internal reset signal is used, the MCU comes out of a COP reset in the same operating mode it was in when the COP time-out was generated.

The COP reset function is enabled or disabled by a mask option.

Refer to **8.2 COMPUTER OPERATING PROPERLY (COP) WATCHDOG RESET** for more information on the COP Watchdog timer.

5.4.4 ILLEGAL ADDRESS (ILADR) RESET

The MCU monitors all opcode fetches. If an illegal address space is accessed by an opcode fetch, an internal reset is generated to reset the MCU. Illegal address spaces consist of all unused locations within the memory map and the I/O registers (see **Figure 3-1: Memory Map**.

Because the internal reset signal is used, the MCU comes out of an ILADR reset in the same operating mode it was in when the opcode was fetched.

5.5 INTERRUPTS

The MCU can be interrupted three different ways: via the external interrupt pin ($\overline{IRQ}$), a peripheral (such as the timer) and the non-maskable software interrupt instruction (SWI).

Interrupts cause the processor to save register contents on the stack and to set the interrupt mask (I-bit) to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack and normal processing to resume.

Unlike a reset, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction is complete.

NOTE: The current instruction is the one already fetched and being operated on.

When the current instruction is complete, the processor checks all pending hardware interrupts. If interrupts are not masked (CCR I-bit clear) and if 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 and is therefore non-maskable.

If more than one interrupt is pending at the end of an instruction execution, the interrupt to be recognized first is determined by a fixed priority scheme (see below).

Table 5-1: Interrupts and Vectors

Priority	Interrupt Sources	Vector Address
1 (Highest)	Reset	$3FFE-$3FFF
2	Software (SWI)	$3FFC-$3FFD
3	SMI	$3FFA-$3FFB
4	External Interrupt	$3FF8-$3FF9
5	16-Bit Timer Input Capture/Output Compare/Overflow	$3FF6-$3FF7
6	Multifunctional Timer Overflow/Real Time Interrupt	$3FF4-$3FF5
7	SSI	$3FF2-$3FF3
8 (Lowest)	Port-C Wake-Up	$3FF0-$3FF1

Once the highest priority pending interrupt has been serviced, the next highest priority pending interrupt will be recognized when the I-bit has been cleared and at least one more instruction has been executed by the CPU (these might be simultaneously accomplished by executing an RTI instruction).

5.5.1 HARDWARE CONTROLLED INTERRUPT SEQUENCE

The following three functions ($\overline{\text{RESET}}$, STOP, and WAIT) are not in the strictest sense an interrupt; however, they are acted upon in a similar manner. Flowcharts for hardware interrupts are shown in **Figure 5-1: Interrupt Flowchart** and for STOP and WAIT in **Figure 5-2: STOP/WAIT Flowcharts**. A discussion is provided below.

1. $\overline{\text{RESET}}$ - A low input on the $\overline{\text{RESET}}$ input pin causes the program to vector to its starting address, which is specified by the contents of memory locations $3FFE and $3FFF. The I-bit in the condition code register is also set. Much of the MCU is initialized to a known state during this type of reset, as previously described in **5.4 RESETS**.
2. STOP - The STOP instruction causes the oscillator to be turned off and the processor to “sleep” until an external interrupt ($\overline{\text{IRQ}}$), SMI Wake-up, Port C Wake-up or reset occurs.
3. WAIT - The WAIT instruction causes all processor clocks to stop, but leaves the timer clock running. This “rest” state of the processor can be cleared by reset, an external interrupt ($\overline{\text{IRQ}}$), or Timer interrupt. There are no special wait vectors for these individual interrupts.

5.5.2 SOFTWARE INTERRUPT (SWI)

The SWI is an executable instruction and a non-maskable interrupt: it is executed regardless of the state of the I-bit in the CCR. If the I-bit is zero (interrupts enabled), SWI executes after interrupts which were pending when the SWI was fetched, but before interrupts generated after the SWI was fetched. The interrupt service routine address is specified by the contents of memory locations $3FFC and $3FFD.

5.5.3 SMI INTERRUPT

When the CPU is in the Run (Normal) mode of operation, changes in the status of the SMI will cause an interrupt request if the programmer has set the Interrupt Enable (IE) bit within one of its control registers. The interrupt service routine address is specified by the contents of memory locations $3FFA and $3FFB.

When the CPU is in the Stop mode of operation, a passive to dominant transition on either the $\overline{\text{ARCV}}$ or BRCV Receiver pins will cause an interrupt request regardless of the state of the SMI Interrupt Enable (IE) bit. This interrupt request will cause the CPU to exit the Stop mode of operation, delay the mask programmed number of E-cycles and then execute the interrupt service routine address specified by the contents of memory locations $3FFA and $3FFB.

When the CPU is in the Wait mode of operation, a passive to dominant transition on either the $\overline{\text{ARCV}}$ or BRCV Receiver pins will cause an interrupt request regardless of the state of the SMI’s Interrupt Enable (IE) bit. This interrupt request will cause the CPU to exit the Wait mode of operation and then execute the interrupt service routine address specified by the contents of memory locations $3FFA and $3FFB.

Refer to **7.4 LOW-POWER MODES** for more information.

5.5.4 EXTERNAL INTERRUPT

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. The interrupt request is latched immediately following the falling edge of $\overline{IRQ}$. It is then synchronized internally and serviced as specified by the contents of $3FF8 and $3FF9.

Either a level-sensitive and edge-sensitive trigger, or an edge-sensitive-only trigger is available as a mask option.

NOTE: The internal interrupt latch is cleared in the first part of the interrupt service routine; therefore, one external interrupt pulse could be latched and serviced as soon as the I-bit is cleared.

5.5.5 16-BIT TIMER INTERRUPT

There are three different timer interrupt flags that cause a timer interrupt whenever they are set and enabled. The interrupt flags and enable bits are located in the Timer Control Register (TCR) and the Timer Status Register (TSR). Any of these interrupts will vector to the same interrupt service routine, located at the address specified by the contents of memory location $3FF6 and $3FF7. Refer to **9.6 TIMER CONTROL REGISTER (TCR)** for more information.

5.5.6 MULTIFUNCTIONAL TIMER INTERRUPT

There are two different timer interrupt flags that cause a timer interrupt whenever they are set and enabled. The interrupt flags and enable bits are located in the Timer Control Status Register (TCSR). Either of these interrupts will vector to the same interrupt service routine, located at the address specified by the contents of memory location $3FF4 and $3FF5. Refer to **8.1 TIMER CONTROL AND STATUS REGISTER (TCSR)** for more information.

5.5.7 SSI INTERRUPT

When the CPU is in the Run (Normal) or Wait mode of operation, only one flag can cause an SSI interrupt if it is set and enabled. The interrupt flag (SF) is located in the SSI Status Register (SSR). This interrupt will vector to the interrupt service routine located at the address specified by the contents of memory location $3FF2 and $3FF3. Refer to **Figure 10-3: SSI Control Register** for more information.

5.5.8 PORT-C WAKE-UP INTERRUPT

Refer to **6.3 PORT C** for a full description.

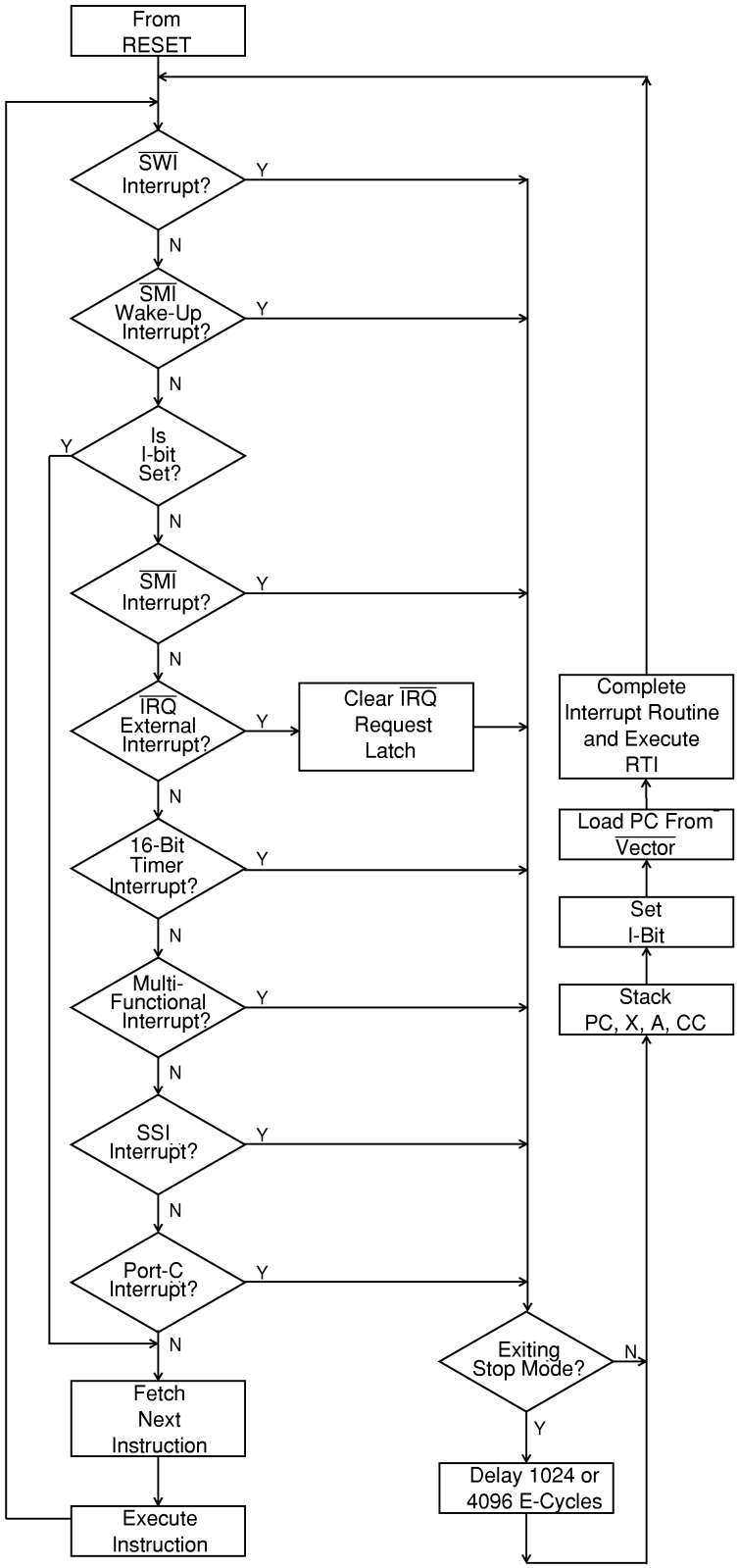


Figure 5-1: Interrupt Flowchart

5.6 LOW-POWER MODES

5.6.1 STOP

The STOP instruction places the MCU in its lowest power consumption mode. In STOP mode, the internal oscillator is turned off, halting all internal processing, including timer (and COP Watchdog timer) operation.

During the STOP mode, the interrupt flags (TOF and RTIF) and interrupt enable bits (TOFE and RTIE) in the TCSR are cleared by internal hardware to remove any pending timer interrupt requests and to disable any further timer interrupts. The timer prescaler is cleared. The I-bit in the CCR is cleared to enable external interrupts. All other registers, including the remaining bits in the TCSR, and memory remain unaltered. All input/output lines remain unchanged. The SMI Receiver comparators are placed in a low-power mode of operation, with their hysteresis disabled.

The processor can be brought out of the STOP mode by an external interrupt, an SMI Wake-Up, a level change detected on a Port C pin (if Port C wake-up mask option is selected) or $\overline{\text{RESET}}$.

The STOP instruction can be disabled by a mask option. When disabled, the STOP instruction is executed as a NOP.

Programmer's Note: If SMI network activity occurs during STOP mode recovery period due to a Port C or IRQ interrupt, an SMI interrupt may not be generated.

5.6.2 WAIT

The WAIT instruction places the MCU in a low-power consumption mode, but the WAIT mode consumes more power than the STOP mode. All CPU and peripheral action is suspended, but the Timers and SSI remain active if they were enabled while the WAIT instruction is executed. An interrupt from any of the operative peripherals can cause the MCU to exit the WAIT mode.

During the WAIT mode, the I-bit in the CCR is cleared to enable interrupts. All other registers, memory, and input/output lines remain in their previous state. The Timers may be enabled to allow a periodic exit from the WAIT mode.

5.6.3 DATA-RETENTION MODE

The contents of RAM and CPU registers are retained at supply voltages as low as 2.0 Vdc. This is called the data retention mode where the data is held, but the device is not guaranteed to operate. RESET must be held low during data-retention mode.

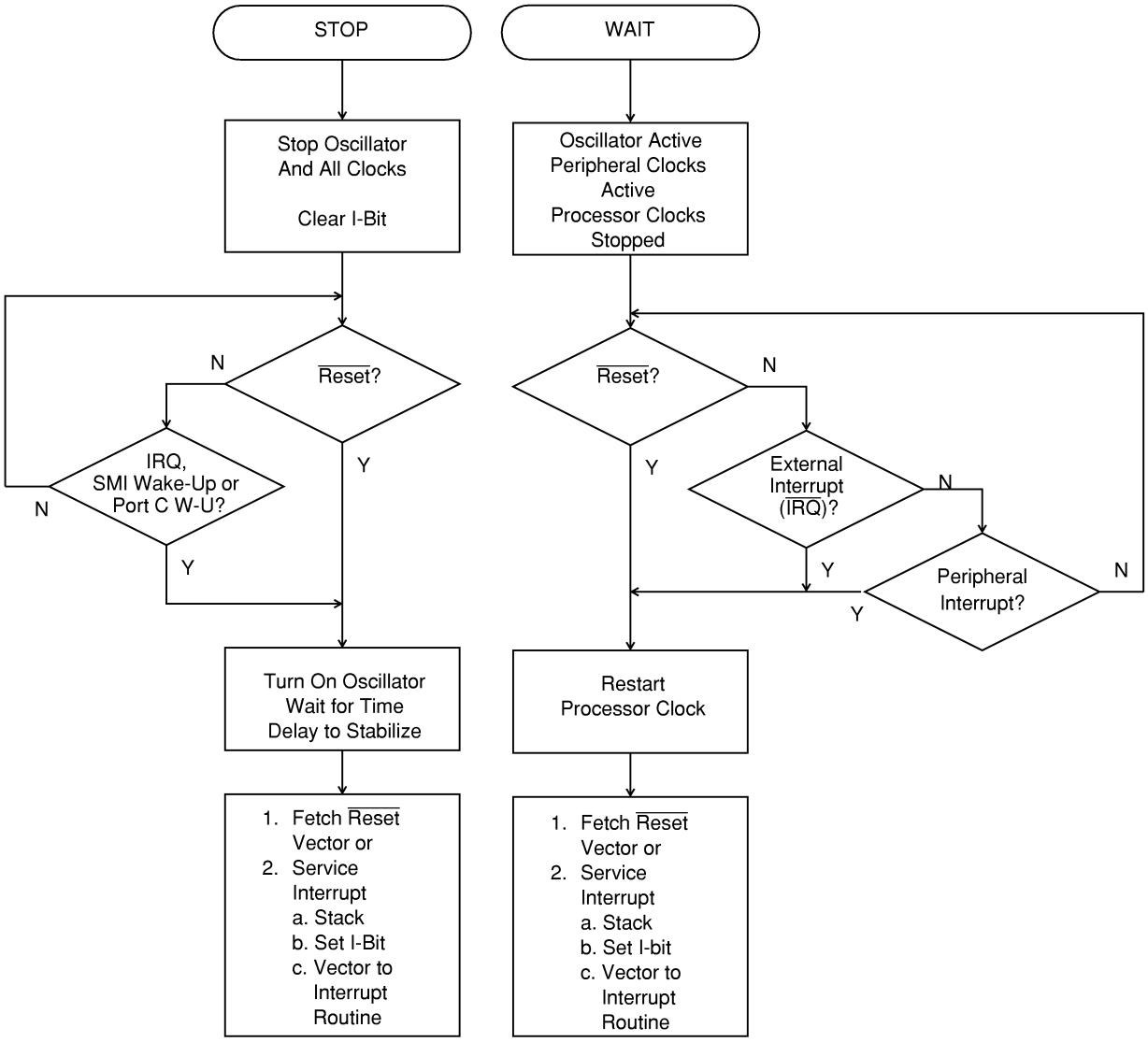


Figure 5-2: STOP/WAIT Flowcharts

SECTION 6 INPUT/OUTPUT PORTS

In Single-Chip Mode, there will be 27 lines arranged as three 8-bit I/O ports and one 3-bit I/O port. These ports are programmable as either inputs or outputs under software control of the data direction registers. In addition, Port C has a special “wake-up” capability.

To avoid a glitch on the output pins, write data to the I/O Port Data Register before writing a one to the corresponding Data Direction Register.

6.1 PORT A

Port A is an 8-bit bidirectional port that does not share any of its pins with other subsystems. The Port A data register is at $0000 and the data direction register (DDR) is at $0004. A reset does not affect the data registers, but clears the data direction registers, thereby returning the ports to inputs. Writing a one to a DDR bit sets the corresponding port bit to output mode.

6.2 PORT B

Port B is an 8-bit bidirectional port that does not share any of its pins with other subsystems. The Port B data register is at $0001 and the data direction register (DDR) is at $0005. A reset does not affect the data registers, but clears the data direction registers, thereby returning the ports to inputs. Writing a one to a DDR bit sets the corresponding port bit to output mode.

6.3 PORT C

Port C is an 8-bit bidirectional port which does not share any of its pins with other subsystems. The Port C data register is at $0002 and the data direction register (DDR) is at $0006. A reset does not affect the data registers, but clears the data direction registers, thereby returning the ports to inputs. Writing a one to a DDR bit sets the corresponding port bit to output mode.

An additional feature of Port C is its “wake-up” capability. If the MC68HC05X1 is in STOP mode, the status of each of the pins is monitored. The value last written into the **Data Output Register**, of each bit configured as an **input**, is used to do a level-sensitive comparison with the pin state. If the state of the pin matches the corresponding data register bit value, a match indicator signal will be latched. If the PTC interrupt enable mask option has been selected, an interrupt request is generated causing the CPU to exit STOP mode (once the oscillator restarts, begins generating the internal processor clock and waits the mask programmed number of cycles). Port C has its own interrupt vector at $3FF0-$3FF1.

The interrupt request is cleared by a subsequent read of either the data register or the DDR.

The comparison is level sensitive. If STOP mode is entered while a match condition already exists, an interrupt will be immediately generated and STOP mode will be exited.

Because of the dual use of the Data Output Register, care must be taken when using read-modify-write instructions on this register. Reads of those bits programmed as inputs will always return the current pin state, but writes to those bits will be stored in the Data Output Register in preparation for comparison in STOP mode. Therefore, manipulating a bit programmed as an output may cause undesired disturbance of comparison values. The programmer should avoid using read-modify-write instructions on this register after the comparison values have been programmed if the wake-up feature is to be used.

This interrupt is maskable, for the entire port, as a mask option (Port C wake-up enable/disable). The interrupt is also maskable, on a per pin basis, by selectively configuring pins as outputs just prior to entering STOP mode. Care must be taken with this approach to ensure that the output pin conditions never exceed the device limits (See **SECTION 11 ELECTRICAL SPECIFICATIONS**).

6.4 PORT D

Port D is a 3-bit bidirectional port that does not share any of its pins with other subsystems. The port D data register is at $0003 and the data direction register (DDR) is at $0007. A reset does not affect the data registers, but clears the data direction registers, thereby returning the ports to inputs. Writing a one to a DDR bit sets the corresponding port bit to output mode.

6.5 INPUT/OUTPUT PROGRAMMING

Ports A, B, C, and D may be programmed as an input or an output under software control. The function of the pins is determined by the state of the corresponding bit in the port data direction register (DDR). Each port has an associated DDR. Any Port A, B, C, or D 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 A, B, C, and D pins as inputs. The data direction registers are capable of being written to or read by the processor. During the programmed output state, a read of the data register actually reads the value of the output data latch and not the I/O pin. This is shown overleaf.

Table 6-1: I/O Pin Functions.

R/$\overline{W}$	DDR	I/O Pin Function
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output of the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode. The output data latch is read.

R/$\overline{W}$ is an internal signal.

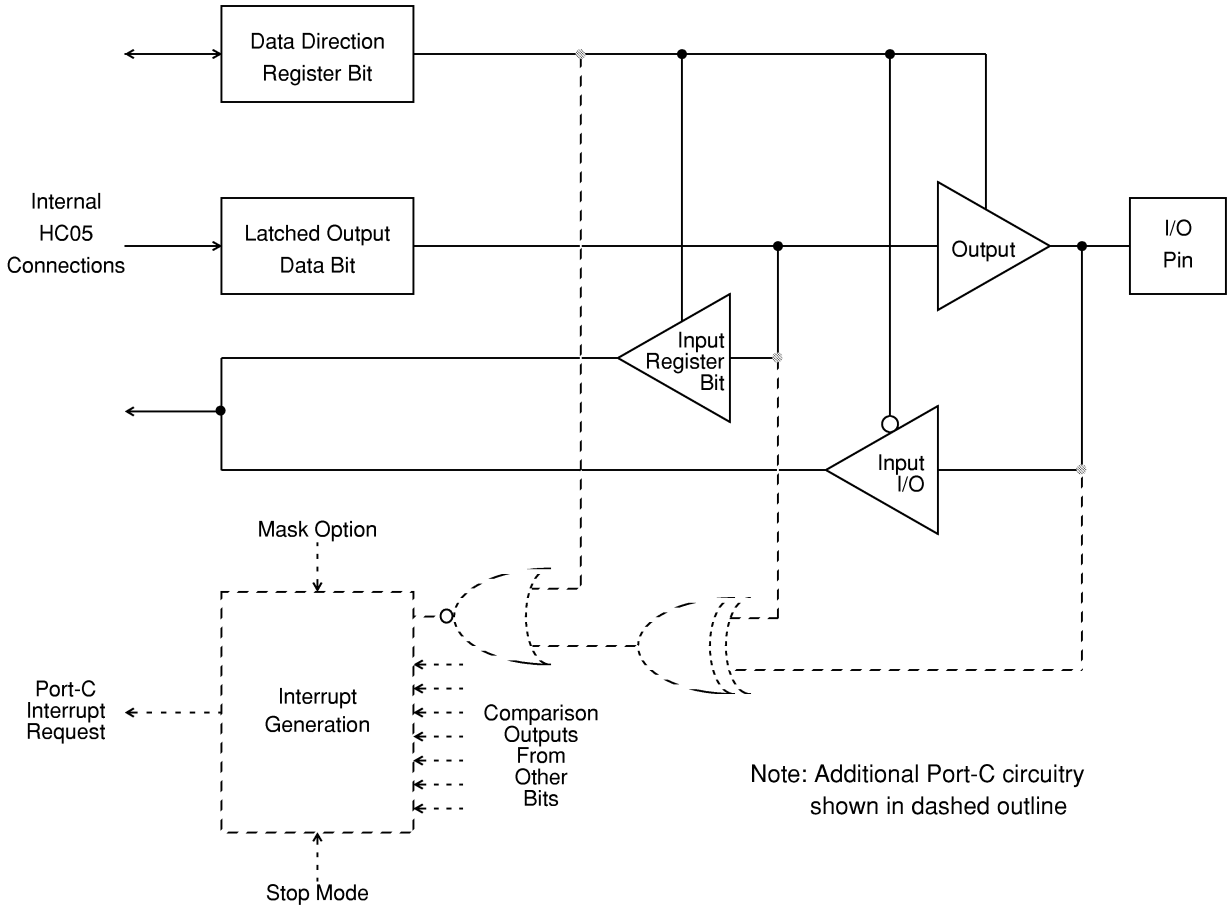


Figure 6-1: Port I/O Circuitry

THIS PAGE INTENTIONALLY LEFT BLANK

SECTION 7 SERIAL MULTIPLEX INTERFACE

The Serial Multiplex Interface (SMI) provides access to an external serial communications multiplex bus, operating according to the “Class B Data Communication Network Interface Standard SAE J1850 1994” protocol.

Key features of the SMI include;

- J1850 operation
- Fault Tolerant PWM bit decoding
- PWM operation with selectable Baud rates
- Digital Noise Filtering
- Collision Detection
- CRC Generation & Checking
- Transmitter Jabber Watchdog
- Double Byte Buffered Transmitter and Receiver
- Wake-up mode
- Message Acknowledgment supported
- Interrupts for key events
- Local Loopback mode

7.1 OUTLINE

The SMI contains three main blocks, namely the CPU Interface, the Serial Machine and the MUX Interface, as shown **Figure 7-1: SMI Outline**.

The CPU Interface contains the software addressable registers and provides the link between the CPU and the Serial Machine. The Serial Machine is responsible for the encoding and decoding of data bits and special message symbols during transmission and reception. The MUX Interface provides the link between the MUX bus and the Serial Machine.

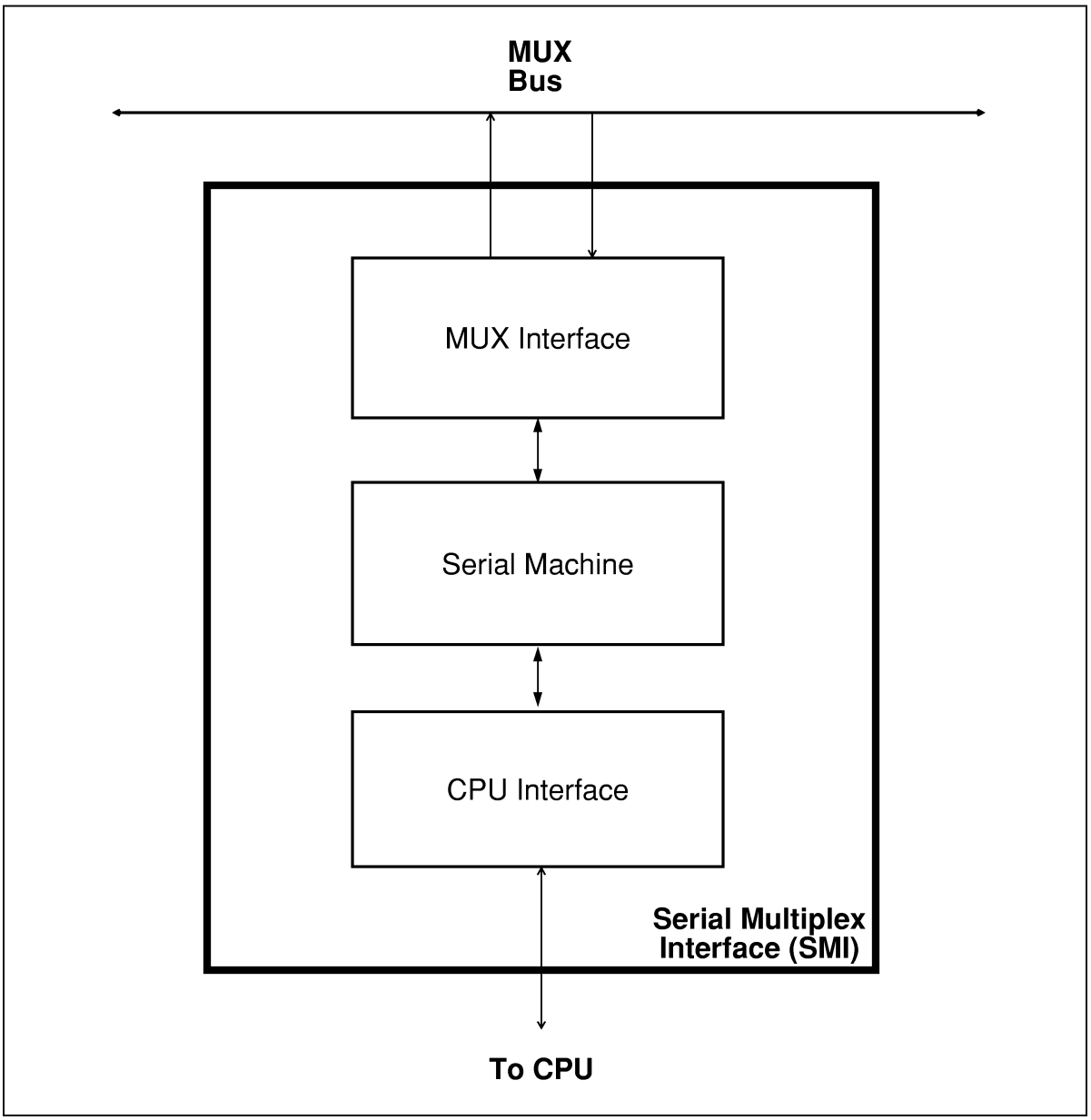


Figure 7-1: SMI Outline

7.2 CPU INTERFACE

This block provides the interface between the CPU and the Serial Machine. It consists mainly of the user registers.

7.2.1 INTERRUPTS

The type of interrupt request that the SMI makes to the CPU depends upon the operating mode of the processor. All interrupt requests utilize the same interrupt vector.

When the SMI is enabled, its Interrupt Enable (IE) bit has been set, and the processor is in the normal mode of operation, changes in the status of the SMI will cause an interrupt request to be generated. The condition(s) that caused the interrupt request may be determined by examining the SMI State Vector Register (SMSVR).

When the SMI is enabled and the processor is in the STOP or WAIT mode of operation, a passive to dominant transition on the $\overline{\text{ARCV}}$ or BRCV pins will cause the SMI to generate a “MUX bus wake-up” interrupt request to the CPU, regardless of the state of the SMI’s interrupt enable (IE) bit. This will cause the CPU to exit the STOP or WAIT mode of operation. The SMI provides no explicit indication, other than the interrupt request itself, that this situation has occurred. This requires software to take special precautions before entering STOP/WAIT modes. A “MUX bus wake-up” interrupt request is cleared by any access (read or write) of any SMI register.

Note that the host processor must also have the appropriate interrupt(s) enabled for an interrupt to occur.

7.2.2 SMI REGISTERS

These software addressable registers provide control information to the SMI, report status to the CPU and allow the exchange of data between the SMI and the CPU. The registers occupy a total of 6 bytes of the memory map, as shown below.

Addr	Reg.	Bit Names							
		7	6	5	4	3	2	1	0
$0008	SMCR1	ARSOM	IMSG	HACK	SDACK	SBACK	SACK	SEOD	SSOM
$0009	SMCR2	RDRF	SMR1	SMR0	LOFF	BXEN	AXEN	IE	SME
$000A	SMSR1	MORE	RPEOD	RCRC	RIB	REOM	REOD	LARB	RSOM
$000B	SMSR2	TXFLT	FLTB	FLTD	FLTA	TJAB	BSOM	ROR	TUR
$000C	SMDR	D7	D6	D5	D4	D3	D2	D1	D0
$000D	SMSVR	SU7	SU6	SU5	SU4	I1	I0	0	0

Figure 7-2: SMI Registers

7.2.2.1 SMCR1 - SMI Control Register #1

This register is one of two that control the operation of the SMI. The functions controlled by this register are oriented towards the support of operation under the SAE J1850 protocol.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$08	ARSOM	IMSG	HACK	SDACK	SBACK	SACK	SEOD	SSOM
	R	RS	RS	RS	RS	RS	RS	RW
RESET	0	1	0	0	0	0	0	0

R=Read, S=Set-only, W=Write

Figure 7-3: SMCR1

This register is initialized to $40 at reset.

All of the bits in this register are readable. The RSOM bit is read-only, and the SSOM bit may be set or cleared directly. All other bits are set only, and will be cleared automatically, or indirectly by the IMSG or SME bits.

Note that the IMSG bit is set by reset.

IMPORTANT: The user should avoid the use of read-modify-write instructions on this register to avoid unintended side effects due to asynchronous bit changes.

IMPORTANT: The user should avoid setting the SDACK, SBACK, and SACK bits during a transmission to avoid unexpected results.

Each of the bits will now be described in more detail.

ARSOM - Alternate RSOM

When set, this bit indicates that a SOM symbol has been detected by the receiver. It is a duplicate image of the RSOM bit in SMSR1.

This bit is provided to allow the state of the RSOM bit in SMSR1 to be observed without altering the state of the RSOM bit in SMSR1. This is useful for non-interrupt routines attempting to initiate transmissions using the SSOM bit.

Reading this register has no effect on the RSOM bit in SMSR1.

This bit will be cleared when the RSOM bit in SMSR1 is cleared.

IMSG - Ignore Message

This bit is used to disable the Receiver until a new SOM is detected. This is useful when it is decided that the rest of the incoming message is of no interest to the user. The IMSG bit also resets all requests for acknowledgments by clearing the SACK, SBACK, and SDACK control bits.

The IMSG bit is prevented from being set when an RSOM is pending to prevent new incoming messages from being ignored.

When the IMSG bit is set, it has the following effect:

1. The following bits are held cleared until IMSG is automatically cleared: HACK, SDACK, SBACK, SACK, RPEOD, RCRC, RIB, REOM, REOD, LARB, TUR, ROR.
2. SMI interrupt requests, except for the “MUX bus wake-up” request, are masked until IMSG is automatically cleared.

IMSG is automatically cleared when:

1. A SOM is received (interrupt request due to RSOM will be pending).
2. An error occurs during an SMI SOM transmission (interrupt request due to BSOM will be pending).

When cleared, the SMI is allowed to generate interrupt requests and will allow changes of the Status Registers to occur. However, these interrupt requests may still be masked by the Interrupt Enable (IE) bit of the SMCR2 register.

Note that if IMSG is set while the SMI is transmitting, the following conditions apply:

1. If IMSG is set after RSOM has been handled and before SEOD has been set, the following will occur:
   a. The transmission of the current byte will be completed.
   b. If the SMDR has been written since the transmission of the current byte began, the contents of the SMDR will be transmitted.
   c. An additional bit will be transmitted beyond the last byte boundary.
   d. Transmission will halt.
2. If IMSG is set after RSOM has been handled and after SEOD has been set, the transmission will proceed normally through the CRC, but no additional SMI interrupt requests will be generated until IMSG is cleared.

The IMSG bit should never be set while the SMI is in the process of transmitting an acknowledgment byte.

HACK - Halt Acknowledgment

This bit is used to clear the SACK bit to prevent the SMI from making any further attempts to acknowledge the current message if it is not successful by the next byte boundary.

The HACK bit cannot be set unless the SACK bit is already set, and the HACK bit is cleared when the SACK bit is cleared.

SDACK - Send Data Acknowledgment

This bit is used to command the SMI to acknowledge the received message with a multiple byte data acknowledgment.

When SDACK is set, the following conditions apply:

1. If a premature EOD, CRC error, invalid bit, or loss of arbitration occurs before or at EOD, the SDACK bit will be cleared and no acknowledgment will be sent.
2. If no errors occur before or at EOD, the transmitter will be enabled and an attempt will be made to send the data that was previously loaded into the SMDR register. A Transmit Data Register Empty (TDRE) interrupt will be generated at this point. Once transmitting, if an invalid bit is detected, arbitration is lost, or a transmitter underrun occurs, the SMI will halt transmission and the SDACK bit will be automatically cleared.

At the TDRE interrupt after all acknowledgment data has been loaded, the user must set the SEOD bit to generate a CRC and an EOD symbol, The SDACK bit will be automatically cleared during the first rising edge of the first CRC bit that is sent.

The SDACK bit will also be cleared if the SME bit is cleared or the IMSG bit is set by software.

SBACK - Send Broadcast Acknowledgment

This bit is used to command the SMI to make one attempt to acknowledge the received message with the last byte written to the SMDR.

When SBACK is set, the following conditions apply:

1. If a premature EOD, CRC error, invalid bit, or loss of arbitration occurs before or at EOD, the SBACK bit will be cleared and no acknowledgment will be sent.
2. If no errors occur before or at EOD, the transmitter will be enabled and a single attempt will be made to send the data that was previously loaded into the SMDR register. A TDRE interrupt will **not** be generated at this point. Once transmitting, if an invalid bit is detected, arbitration is lost, or the byte transmission completes successfully, the SMI will halt transmission and the SBACK bit will be automatically cleared.
3. If a loss of arbitration does occur, once the byte boundary is reached, the LARB bit will be set.

The SBACK bit will also be cleared if the SME bit is cleared or the IMSG bit is set by software.

SACK - Send Acknowledgment

This bit is used to command the SMI to acknowledge the received message with the last byte written to the SMDR, and to retry the acknowledgment if arbitration is lost until the acknowledgment is successful, the HACK bit is set, or an error occurs.

When SACK is set, the following conditions apply:

1. If a premature EOD, CRC error, invalid bit, or loss of arbitration occurs before or at EOD, the SACK bit will be cleared and no acknowledgment will be sent.
2. If the HACK bit is set before the start of EOD, the SACK bit will be cleared and no acknowledgment will be sent.
3. If no errors occur before or at EOD, the transmitter will be enabled and a single attempt will be made to send the data that was previously loaded into the SMDR register. A TDRE interrupt will **not** be generated at this point. Once transmitting, the following conditions apply:
   a. If an invalid bit is detected or the byte transmission completes successfully, the SMI will halt transmission and the SACK bit will be cleared.
   b. If arbitration is lost and the HACK bit is not set before the next byte boundary, the SMI will suspend transmission and retry at the next byte boundary. The retry process will **not** cause any TDRE interrupts to be generated. Note that the LARB bit will not be set until the byte boundary.
   c. If arbitration is lost and the HACK bit is set before the next byte boundary, the SMI will halt transmission and the SACK bit will be cleared. Note that the LARB bit will not be set until the byte boundary.

The SACK bit will also be cleared if the SME bit is cleared or the IMSG bit is set by software.

SEOD - Send End of Data

This bit is used to end a message being sent by the SMI.

When SEOD is set, the following conditions apply:

1. Any pending SMI TDRE interrupt is cleared.
2. If a loss of arbitration was pending while SEOD was set, no CRC will be transmitted, and SEOD will be cleared at the byte boundary following the loss of arbitration, or in conjunction with RPEOD if an end of data occurs before the byte boundary is reached.
3. If a premature EOD, invalid bit, or loss of arbitration occurs before the CRC transmission begins, the SEOD bit will be cleared and no CRC will be sent.
4. If no errors occur before the transmission of the current byte completes, transmission of an 8-bit CRC will begin following the current byte. A TDRE interrupt will NOT be generated at the start of CRC transmission, The SEOD bit will be cleared during the rising edge of the first CRC bit that is sent. Once CRC transmission has begun, if an invalid bit is detected, a loss of arbitration occurs, or the CRC transmission completes successfully, the SMI will halt transmission. A TDRE interrupt will **not** be generated at the completion of the CRC transmission.

SSOM - Send Start of Message

This bit is used to command the SMI to begin the transmission of a message.

The effect of the SSOM bit on the SMI depends upon the state of the network when the SSOM bit is set.

1. If there is non-SOM or no network activity present when the SSOM bit is set, the SMI will wait until it has detected:

   a. The start of a probable SOM (two consecutive idle bit periods followed by a passive to dominant transition on the network).

   -- or --

   b. Three consecutive idle bit periods.

2. If a SOM is in-process on the network, but not yet complete, when the SSOM bit is set, the following condition applies.

If 1a or 2 occurs, the SMI will not initiate its own SOM transmission, but will attempt to transmit its data following the completion of the network SOM.

If the network activity results in a valid SOM, the following actions apply:

- The SMI will initiate transmission of the contents of the SMDR register.
- The SSOM bit will be cleared.

- The RSOM bit will be set, generating a Status Register Bit Set (SRBS) interrupt request.
- A TDRE interrupt request will be generated, which will be hidden behind the SRBS interrupt request.

If the network activity does not result in a valid SOM, the following actions apply:

- The SSOM bit will be cleared.
- The BSOM bit in SMSR2 will be set, generating an SRBS interrupt request.
- A TDRE interrupt request will not be generated.

If 1b occurs, the SMI will initiate a SOM transmission, and will attempt to transmit the contents of the SMDR following the completion of the SOM.

If the SOM transmission completes successfully, the following actions apply:

- The SMI will initiate transmission of the contents of the SMDR register.
- The SSOM bit will be cleared.
- The RSOM bit will be set, generating an SRBS interrupt request.
- A TDRE interrupt request will be generated, which will be hidden behind the SRBS interrupt request.

If the SOM transmission does not complete successfully, the following actions apply:

- The AXMT and BXMT pins will be immediately driven to their passive states.
- The transmission will be immediately terminated.
- The SSOM bit will be cleared.
- The BSOM bit in SMSR2 will be set, generating an SRBS interrupt request.
- A TDRE interrupt request will not be generated.

The send SOM operation (SSOM) is accepted by the SMI up to point where RSOM is detected and declared in SMSR1. If SSOM is issued after RSOM is declared, the SSOM bit remains set, and condition 1 applies.

Note that the SSOM bit may be cleared almost immediately after being set if a SOM was in-process (condition 2) at the time it was set.

The SSOM bit is the only bit in SMCR1 that may be cleared by writing a “0” to it. This should only be done after it has been determined that another message is in progress, in which case no SOM should be sent until that message has finished transmission. Bit manipulation instructions, since they are of the read/modify/write type, must not be used to write the “0” to this bit.

Note that when the SSOM bit is set it takes a finite time for the RSOM bit to be set, as shown in the following example:

```
        sei                     Disable interrupts

        lda     #$01            Load byte to issue SSOM command
        brset   7,SMCR1,TOREC   If RSOM=1 go to Receive routine
        sta     SMCR1           Issue SSOM command
        brclr   7,SMCR1,MLOOP   If RSOM still clear begin
                                transmission
        brset   0,SMCR1,TOREC   If RSOM=1 and SSOM=1 then receive

MLOOP   (start of send routine)
                .
                .
                .
```

7.2.2.2 SMCR2 - SMI Control Register #2

This register is one of two that control the operation of the SMI. The functions controlled by this register are oriented towards the configuration of the SMI.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$09	RDRF	SMR1	SMR0	LOFF	BXEN	AXEN	IE	SME
	R	RW	RW	RW	RW	RW	RW	RW
RESET	0	0	0	0	0	0	0	0

R=Read, S=Set-only, W=Write

Figure 7-4: SMCR2

This register is initialized to $00 at reset. The RDRF bit is read only. All other bits are readable and writable.

RDRF - Receiver Data Register Full

This bit is set when a Receiver Data Register Full (RDRF) interrupt is pending. This bit will be cleared when the pending RDRF is cleared. Reading this register has no effect on the pending interrupt.

SMR1, SMR0 - Rate Select

These bits determine the amount by which the system clock is divided to form the bit clock period for the SMI, according to the following table:

f_{osc} (MHz)	Period (E-cycles)	SMR1	SMR0	PWM Rate (kbps)	Comment
2	24	0	0	41.7	May not be used in Ford applications
4	**48**	**0**	**1**	**41.7**	**Normal selection**
6	72	1	0	41.7	Not yet available
8	96	1	1	41.7	Not yet available

Note that these bits should be set to their desired values before the SMI is enabled, and should not be modified while the SMI is enabled.

LOFF - Loopback Off

This bit enables and disables the loopback circuit.

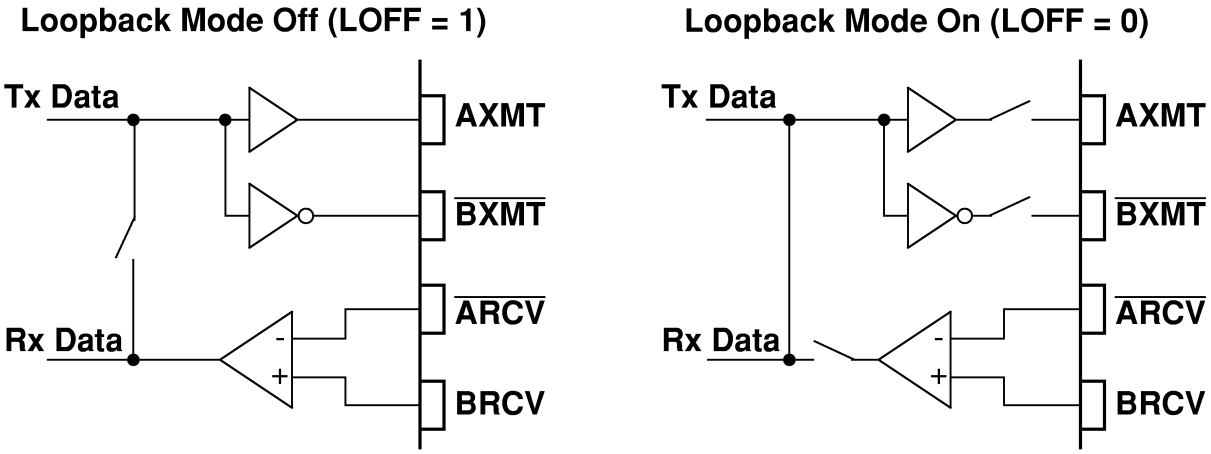


Figure 7-5: Loopback Configuration Outline

When set, all transmitted data will be passed in an inverted condition the $\overline{\text{BXMT}}$ pin and in an non-inverted condition the AXMT pin.

When cleared, all transmitted serial data will be passed directly to the Receiver. Additionally, the AXMT pin will be pulled low and the $\overline{\text{BXMT}}$ pin will be pulled high (corresponding to a passive network state), regardless of the state of the AXEN and BXEN control bits.

The programmer must set this bit before any external communications can take place.

IMPORTANT: The SDACK, SBACK, or SACK bits must never be set while the SMI is in loopback mode, or unpredictable SMI operation will result.

BXEN - $\overline{\text{BXMT}}$ Driver Enable

This bit enables and disables transmissions from the $\overline{\text{BXMT}}$ pin.

When set, all inverted serial data will be sent out of the $\overline{\text{BXMT}}$ pin, if LOFF is also set.

When cleared, the $\overline{\text{BXMT}}$ pin will be pulled high.

This bit will be cleared by the transmit jabber watchdog if it determines that the transmitter is jabbering

AXEN - AXMT Driver Enable

This bit enables and disables transmissions from the AXMT pin.

When set, all non-inverted serial data will be sent out of the AXMT pin, if LOFF is also set.

When cleared, the AXMT pin will be pulled low.

This bit will be cleared by the transmit jabber watchdog if it determines that the transmitter is jabbering

IE - Interrupt Enable

This bit enables and disables SMI interrupt requests while the SMI is enabled and the CPU is operating in normal mode. It **does not** affect SMI "MUX bus wake-up" interrupt requests, which may only occur while the SMI is enabled and the CPU is in STOP or WAIT mode.

When set, SMI interrupt requests are enabled during normal CPU operation.

When cleared, SMI interrupt requests, except for “MUX bus wake-up” interrupt requests, are disabled.

SME - SMI Enable

This bit enables and disables operation of the SMI.

When set, the SMI is enabled and all of the status bits associated with the SMI become valid. SMI “MUX bus wake-up” interrupts will be enabled if the CPU enters STOP or WAIT mode.

When cleared:

1. The SMI is disabled, and the SMCR1, SMSR1, SMSR2, SMSVR, and SMTCR registers are forced to their reset state.
2. Any reception or transmission will immediately be terminated.
3. The AXMT pin will be pulled low, and the $\overline{\text{BXMT}}$ pin will be pulled high, corresponding to a passive network state.
4. The SMI will not generate interrupt requests of any kind.

7.2.2.3 SMSR1 - SMI Status Register #1

This register is one of two that indicate the status of the SMI. The status conditions indicated by this register are oriented towards the condition of the Receiver under the SAE J1850 protocol.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$0A	MORE	RPEOD	RCRC	RIB	REOM	REOD	LARB	RSOM
	R	R	R	R	R	R	R	R
RESET	0	0	0	0	0	0	0	0

R=Read, S=Set-only, W=Write

Figure 7-6: SMSR1

This register is initialized to $00 at reset. All bits except the MORE bit are automatically cleared following a read of this register. The SMSR2 register must be read to clear the MORE bit. Writes to this register have no effect.

When any of these bits is set, and if the Interrupt Enable (IE) bit of SMCR1 is also set, then a CPU interrupt request will be made. The interrupt request will remain in place until SMSR1 is read.

Each of the bits will now be described in more detail.

MORE - More Status Information in SMSR2

This bit is set when one or more of the Status bits in SMI Status Register #2 (SMSR2) have been set. The programmer must read SMSR2 to obtain the additional status information and to clear the MORE bit.

RPEOD - Received Premature End of Data

This bit is set when the receiver detects the first idle bit period following an active bus, and the idle bit period did not begin at a byte boundary.

RCRC - Received CRC Error

This bit is set when the receiver detects an EOD, and its CRC calculation for the bytes received prior to the EOD does not match the expected result of C4 hex.

Note that during the course of a normal message reception, there are two points at which a CRC evaluation will be performed: Once after the first EOD of the message, and once after the first idle bit period (EOD) of EOM, following the acknowledgment(s).

The programmer should be aware that it is likely, but not certain, that this bit will be set following acknowledgment types that do not contain an embedded CRC. This is because the last acknowledgment byte will probably not have the same value as a legitimate CRC

for the acknowledgments preceding it. It is also likely that this bit will be set at the end of a successful SMI transmission due to the manner in which the CRC logic is shared between the transmitter and receiver.

RIB - Received Invalid Bit

This bit is set after an invalid bit has been detected. An invalid bit will be detected when any of the following conditions occurs:

1. A passive bit phase is detected while the transmitter is transmitting a dominant bit phase.
2. A dominant bus state is detected in the third phase of a (non SOM) bit.
3. A passive-to-dominant transition is detected outside of a legal window.

When an invalid bit is detected, the SMI takes the following actions:

1. If the invalid bit is detected while the SMI is transmitting, the following two actions apply:
   a. The AXMT pin is immediately pulled low, and the $\overline{\text{BXMT}}$ pin is immediately pulled high, corresponding to a passive network state.
   b. The transmission is immediately terminated.
2. The RIB bit will be set at the byte boundary following the invalid bit, or in conjunction with RPEOD if an end of data occurs before the byte boundary is reached.

NOTE: A loss of arbitration will occur if an invalid bit is received while the SMI is transmitting.

REOM - Received End of Message

This bit is set when the receiver detects the first two consecutive idle bit periods following an active bus.

REOD - Received End of Data

This bit is set when the receiver detects the first idle bit period following an active bus, and the idle bit period began at a byte boundary.

Note that this bit will be set after the first idle bit period of an EOM.

LARB - Loss of Arbitration

This bit is set after a loss of arbitration has been detected. A loss of arbitration occurs anytime transmitted data does not match received data.

When a loss of arbitration is detected, the SMI takes the following actions:

1. The AXMT pin is immediately pulled low and the $\overline{\text{BXMT}}$ pin is immediately pulled high, corresponding to a passive network state.
2. The transmission is immediately terminated.
3. The LARB bit will be set at the byte boundary following the loss of arbitration, or in conjunction with RPEOD if an end of data occurs before the byte boundary is reached.
4. If the SACK and HACK bits are set, the SACK bit is cleared.

NOTE: If a loss of arbitration occurs while the SMI is transmitting the last bit of a byte after which a TDRE interrupt request would normally be generated, a TDRE interrupt request will still be generated. Otherwise, no TDRE interrupt request will be generated.

Also note that if a loss of arbitration occurs while the SMI is transmitting a functional (SACK) acknowledgment, another transmission attempt will begin at the next byte boundary unless the HACK bit is set before the byte boundary.

RSOM - Received Start of Message

This bit is set when the receiver detects a Start of Message (SOM) symbol.

Note that any pending Receiver Data Register Full (RDRF) interrupt request is cleared when the RSOM bit is set.

7.2.2.4 SMSR2 - SMI Status Register #2

This register is one of two that indicate the status of the SMI. The status conditions indicated by this register are oriented towards more serious conditions and faults. Under normal operating conditions, these bits should remain cleared.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$0B	TXFLT	FLTB	FLTD	FLTA	TJAB	BSOM	ROR	TUR
	R	R	R	R	R	R	R	R
RESET	0	0	0	0	0	0	0	0

R=Read, S=Set-only, W=Write

Figure 7-7: SMSR2

This register is initialized to $00 at reset. All bits are readable but not writable. All bits are set to zero following a read of this register. Writes to this register have no effect.

When any of the bits in this register are set, the MORE bit in SMSR1 will be set, and an interrupt request will be generated if the IE bit in SMCR1 is set.

A read of this register will cause any set bits to be cleared, causing the MORE bit in SMSR1 to be cleared.

Each of the bits will now be described in more detail.

TXFLT - Network Fault detected while Transmitting

This bit is set when one or more of the bits FLTB, FLTD, or FLTA were set while the SMI was transmitting a SOM.

FLTB,D,A - Channel B,D or A Fault

These bits are set when their corresponding receiver channel (Single-ended “B”, Differential “D”, or Single-ended “A”) did not detect a valid SOM Precursor symbol when at least one other channel detected a valid SOM Precursor symbol. These bits are updated coincidentally with the RSOM bit within SMSR1.

NOTE: It is possible for the FLTD bit to be set without the RSOM bit being set, following a “Mux bus wake-up” interrupt request.

TJAB - Transmitter Jabber Watchdog

This bit is set when the SMI determines that its transmitter has sent more than 98 dominant to passive transitions between SOMs. When this occurs, the AXEN and BXEN bits within SMCR2 will be automatically cleared, causing the AXMT pin to be pulled low and the $\overline{\text{BXMT}}$ pin to be pulled high. This immediately terminates any SMI transmission, and helps prevent violations of the SAE J1850 protocol due to a “jabbering” transmitter.

Once set, the TJAB bit will remain set until the SMSR2 register is read or the SME bit is cleared. Note that the AXEN and BXEN bits are held cleared while the TJAB bit is set, and must be set by software after the TJAB bit has been cleared before any further transmissions may be attempted.

The reception or transmission of a valid SOM resets the Jabber Watchdog counter, but does not clear the TJAB bit if it has been set.

It is up to the programmer to prevent a node from continuously sending legitimate frames that always win arbitration but withhold all other nodes from the bus.

BSOM - Bad SOM Detected

This bit is set if the SMI attempts to transmit a SOM or sync up to a network SOM, but no SOM is detected by the receiver. When this condition occurs, the transmission is terminated and the IMSG bit in SMCR1 is cleared.

ROR - Receiver Overrun

This bit is set if the SMI receives another byte before the previous byte has been read from the SMDR register. The previous byte remains in the SMDR register, and the new byte is lost.

TUR - Transmitter Underrun

This bit can only be set when the SMI is transmitting the main body of a message or a data (SDACK) acknowledgment. It cannot be set while the SMI is transmitting a functional (SACK) or broadcast (SBACK) acknowledgment.

This bit is set when the transmitter completes the transmission of the last byte written to the SMDR, the SEOD bit has not been set, and no new data has been written to the SMDR in response to the last Transmit Data Register Empty (TDRE) interrupt request.

When a transmitter underrun condition occurs, the transmitter will re-transmit the most significant bit of the SMDR after the byte boundary and halt. This will guarantee that the network sees a premature EOD, unless another node is also transmitting.

Note that a TUR will not be generated if the SMDR has not been loaded either before a SOM transmission completes, or before an acknowledgment, and whatever happens to be in the SMDR will be transmitted after the SOM or for the acknowledgment. In each of these cases, when the SMI attempts to send a second byte, if the SMDR is not loaded, a TUR will be generated.

7.2.2.5 SMDR - SMI Data Register

This register is used to pass the data to be transmitted from the CPU to the SMI. It is also used to pass the data received from the SMI to the CPU.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$0C	D7	D6	D5	D4	D3	D2	D1	D0
RESET	X	X	X	X	X	X	X	X

x = Unknown

Figure 7-8: SMDR

The contents of this register are indeterminate following a reset.

The following sections provide detailed descriptions of each bit.

D7 - D0

The function of these bits depends upon whether the register is being written or read.

Data written to these bits will be transmitted serially, after a SOM symbol has been sent. The first data byte to be sent should be written to this register prior to setting the SSOM

bit (in the SMCR1 register). Each subsequent data byte should be written only after a "Transmit Data Register (empty)" interrupt has occurred.

Data read from these bits will be the last serially received data byte. This received data should only be read after a "Receive Data Register (full)" interrupt has occurred.

Acknowledgment bytes are also sent and received via this register.

NOTE: The 68HC05 CPU always performs a dummy read cycle at the beginning of a write operation. This dummy read **will not** be sufficient to prevent a Receiver Overrun condition from occurring during transmissions (when each byte sent will normally be received back).The programmer must explicitly read the SMDR as each byte is received back.

The programmer should not attempt to perform Read-Modify-Write instructions on this register since the data represented by reads and writes differs.

7.2.2.6 SMSVR - SMI State Vector Register

This register is provided to substantially decrease the CPU overhead associated with servicing interrupts while under operation of a MUX protocol. It provides an index offset, that is directly related to the current state of the SMI, which can be used with a user-supplied jump table to rapidly enter an interrupt service routine. This eliminates the need for the user to maintain a duplicate state machine in software.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$0D	SU3	SU2	SU1	SU0	I1	I0	0	0
$\overline{\text{RESET}}$	0	0	0	0	0	0	0	0

Figure 7-9: SMSVR

This register is initialized to $00 at reset. The two least significant bits will always read as zeros. Bits 4, 5, 6, and 7 are always readable and writable. Bits 2 and 3 are readable only.

The following sections provide detailed descriptions of each bit.

SU0, 1, 2, 3 - State Number

In this implementation, these bits are provided for the convenience of the programmer. They are not affected by the SMI, except during a reset (when they are set to zero).

These bits may be used by the programmer to hold the current state of the software that is servicing the SMI. The current state and the interrupt source, together, will normally provide sufficient information to determine which service routine should be used.

After servicing the SMI, the current state is then updated by the programmer, and no more actions are taken until the next interrupt occurs.

I0, 1 - Interrupt Source

These bits indicate the source of the highest priority interrupt request that is currently pending.

These bits are updated, and the corresponding interrupts generated or cleared, when one or more of the following conditions occur:

- The SMDR register contains newly received data (BDRF interrupt generated)
- The received data is read (BDRF interrupt cleared)
- The SMDR register requires new transmit data (TDRE interrupt generated)
- The transmit data is written (TDRE interrupt cleared)
- Any status bit (or bits) within SMSR1 or SMSR2 change state (SRBS interrupt generated)
- The SMSR1 or SMSR2 registers are read (SRBS interrupt cleared)

When more than one of these events occurs, I0 and I1 reflect the event with the highest priority. The encoding of these bits, together with the priority level of each event follows:

I1	I0	Interrupt Source	Priority
0	0	No Interrupts Pending	0 (lowest)
0	1	Receiver Data Register (full	1
1	0	Transmitter Data Register (empty)	2
1	1	Status Register (Bit(s) set)	3 (highest)

Upon receiving an SMI interrupt, the user may read the value within the SMSVR, transferring it to the CPU Index Register. The value may then be used to index into a “jump table”, with entries 4 bytes apart, to quickly enter the appropriate service routine. For example:

```
SERVICE    LDX    SMSVR          Fetch State Vector Number
           JMP    JMPTAB,X       Enter service routine,
*                                (must end in an 'RTI')
*
*
JMPTAB     JMP    SERVE0         Service condition #0
           NOP
           JMP    SERVE1         Service condition #1
           NOP
           JMP    SERVE2         Service condition #2
           NOP
           .      .              ..
.          .      .              .
.          .      .              .
           JMP    SERVE63        Service condition #63
           END
```

Note that the NOPs are just used to align the JMPs onto 4-byte boundaries so that the value in the SMSVR may be used intact. Each of the service routines **must** end with an “RTI” instruction to guarantee correct continued operation of the device. Note also that the first entry can be omitted since it corresponds to no interrupt occurring.

The service routines should clear all of the sources that are causing the pending interrupts. Note that the clearing of a high priority interrupt may still leave a lower priority interrupt pending, in which case bits I0 and I1 of the SMSVR will then reflect the source of the remaining interrupt request.

If fewer states are used, or if a different software approach is taken, the jump table may be made smaller or omitted altogether.

7.3 MUX INTERFACE

This block provides the interface between the Serial Machine and the MUX bus.

7.3.1 PINS

Four pins are utilized by the SMI. Their functions are described below. A typical connection to a physical interface circuit is shown overleaf in a conceptual diagram.

7.3.1.1 AXMT

This pin is the non-inverted serial data output from the Transmitter. When sending a **passive** signal, this pin will be **driven low**. When sending a **dominant** signal, this pin will be **driven high**.

The resulting data should be further inverted and buffered before being driven onto the “A-Bus” line of the MUX bus.

7.3.1.2 $\overline{\text{BXMT}}$

This pin is the inverted serial data output from the Transmitter. When sending a **passive** signal, this pin will be **driven high**. When sending a **dominant** signal, this pin will be **driven low**.

The resulting data should be further inverted and buffered before being driven onto the “B-Bus” line of the MUX bus.

7.3.1.3 $\overline{\text{ARCV}}$

This pin should be connected to the “A-Bus” line of the MUX bus through the network, as shown in **Figure 7-10: Physical Interface Outline**.

7.3.1.4 BRCV

This pin should be connected to the “B-Bus” line of the MUX bus through the network, as shown in **Figure 7-10: Physical Interface Outline**.

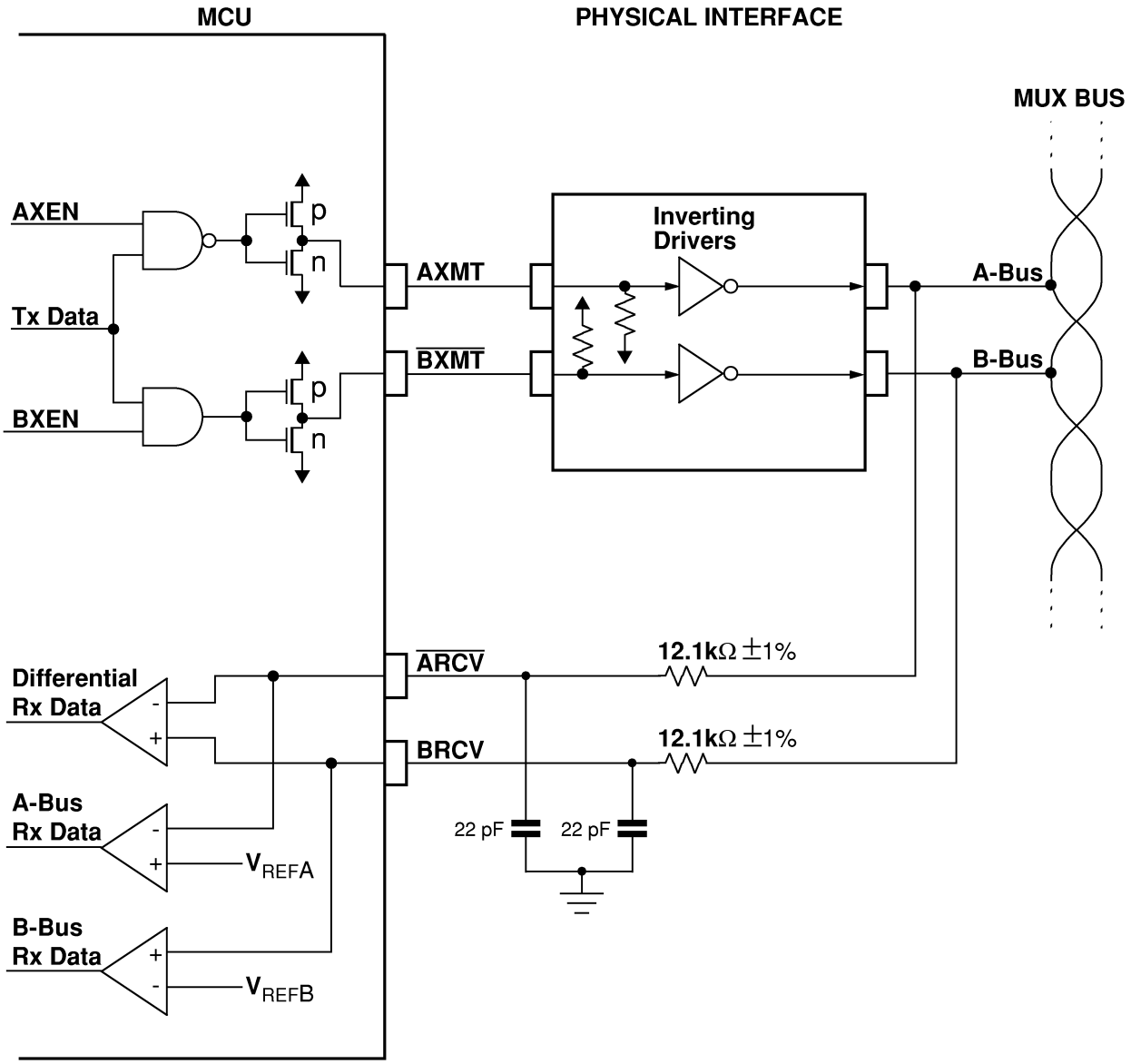


Figure 7-10: Physical Interface Outline

7.4 LOW-POWER MODES

If the CPU executes the “WAIT” or “STOP” instruction, the SMI will immediately suspend operation. Any subsequent passive to dominant transition detected on pins $\overline{\text{ARCV}}$ and BRCV will cause an interrupt request to be made to the CPU and the SMI will resume operation as soon as possible. The CPU will wait 1024 or 4096 (mask programmable) “E” cycles before beginning operation.

When the WAIT or STOP mode is entered, reception and transmission of messages is immediately terminated. Therefore, the user should ensure that a clean termination of communications has taken place and that the network is completely inactive before a WAIT or STOP instruction is executed.

Since the clocks to the SMI are disabled in WAIT and STOP mode, power consumption is minimized. To further reduce power consumption, the analog circuitry enters a special mode whereby the hysteresis is completely eliminated and the comparator trip points become nominal.

This is done to still allow the comparators to detect general network activity. Any passive to dominant transition that is detected will cause the SMI to generate an interrupt request to the CPU, further causing the CPU to awake and clocks to the SMI to be enabled. The interrupt request will be cleared by a subsequent read or write to **any** SMI register.

NOTE: If SMI network activity occurs during STOP mode recovery period due to a Port C or IRQ interrupt, an SMI interrupt may not be generated.

7.5 FAULT TOLERANCE

The SMI receives information from other J1850 devices on the serial bus through three analog comparators. One of these comparators observes both lines differentially. Its output is referred to as the D channel output. The other two comparators observe a single line in a single-ended mode and their outputs are known as the A and B channel outputs.

The switchover select logic for the SMI determines which of the three comparators will be used by the SMI to listen for received data or acknowledgment responses. Channel selection is made upon either detection of a valid SOM precursor or by detecting that one of the lines is stuck in the active state (“stuck-at-one”). This provides a considerable amount of fault tolerance.

Thus, the switchover select logic samples six SMI internal signals to determine which channel output should be used as the Receiver’s input:

1) SOM precursor detected on channel D (SOMD)
2) SOM precursor detected on channel A (SOMA)
3) SOM precursor detected on channel B (SOMB)
4) Channel D stuck-at-one (ST1D)
5) Channel A stuck-at-one (ST1A)

6) Channel B stuck-at-one (ST1B)

In making this decision, the differential channel is considered the preferred channel because it has a much greater noise immunity. Some of the combinations of the preceding signals appear to be conflicting and can only exist, in an otherwise healthy system, in the presence of considerable system noise. The switchover logic still takes these combinations into account.

The truth table for the switchover logic is shown below.

			ST1A	0	1	0	1	0	1	0	1
			ST1B	0	0	1	1	0	0	1	1
			ST1D	0	0	0	0	1	1	1	1
SOMD	SOMB	SOMA									
0	0	0	Q_{n-1}	D	D	D	A	B	A	D	
0	0	1	A	D	A	D	A	B	A	D	
0	1	0	B	B	D	D	B	B	A	D	
0	1	1	A	B	A	D	A	B	A	D	
1	0	0	D	D	D	D	A	B	A	D	
1	0	1	A	D	A	D	A	B	A	D	
1	1	0	B	B	D	D	B	B	A	D	
1	1	1	D	D	D	D	B	B	A	D	

A - Single-ended channel A selected
B - Single-ended channel B selected
D - Differential channel D selected
Q_{n-1} - Previous selection maintained

Figure 7-11: Switchover Logic Truth Table

In general, the differential channel is chosen if it appears “healthy” or if none of the three channels look healthy (as in the case that all three channels are in a stuck-at-one condition).

In the unlikely event that a valid SOM precursor is detected by one of the channels, but that it goes away before it can be sampled, the first row of the truth table applies. If a stuck-at-one condition is also detected, the channel selection is updated appropriately, or the previous channel selection is maintained.

At reset, the SMI assumes that all channels are healthy and as a result, the last row of the truth table will initially apply.

In obviously, conflicting cases, such as when a valid SOM precursor is only detected on channel A but channel A is stuck-at-one, the differential channel D is selected. Some of these conflicting cases cannot be readily tested. All testable cases are tested.

7.6 GLOSSARY OF TERMS

A fuller description of many SMI terms can be found in “Class B Data Communication Network Interface Standard SAE J1850 1994” and the “Hosted Bus Controller Chip User’s Guide”. Other terms used in this document are explained below.

7.6.1 Bit Phase

One third of the duration of 1 bit period.

7.6.2 Channel A

Circuitry associated with single-ended data on the network’s A-Bus line.

7.6.3 Channel B

Circuitry associated with single-ended data on the network’s B-Bus line.

7.6.4 Channel D

Circuitry associated with differential data on the network’s A-Bus and B-Bus lines.

7.6.5 Idle Bit

A passive network condition lasting for the same duration as 1 bit period.

7.6.6 Valid SOM Precursor

A dominant network condition lasting for 4 bit phases, following a passive network condition lasting for 6 bit phases.

7.6.7 SOM

A Start of Message (SOM) is identical to the SAE defined term, Start of Frame (SOF).

7.6.8 EOM

An End of Message (EOM) is identical to the SAE defined term, End of Frame (EOF).

7.6.9 ACK

An Acknowledgment (ACK) is identical to the SAE defined term, In-Frame Response (IFR).

7.7 SMI ELECTRICAL CHARACTERISTICS

(AV_{DD} = 5 Vdc ±5%, V_{SS} = AV_{SS} = 0 Vdc, T_A = -40°C to +85°C, unless otherwise noted)

Table 7-1: SMI Electrical Characteristics

Characteristic	Symbol	Min.	Max.	Unit
Total Receiver Comparator Supply Current (see Note 1) Run Stop/Wait	I_{DDA}	 — —	 8.0 120	 mA µA
Input Capacitance BRCV, $\overline{ARCV}$	C_{IN}	—	10	pF
Positive-going Input Threshold Voltage, Comparator A and B	V_{TBP} V_{TAP}	$AV_{DD}/2 + 0.05$ $AV_{DD}/2 + 0.05$	$AV_{DD}/2 + 0.10$ $AV_{DD}/2 + 0.10$	V V
Negative-going Input Threshold Voltage, Comparator A and B	V_{TBN} V_{TAN}	$AV_{DD}/2 - 0.10$ $AV_{DD}/2 - 0.10$	$AV_{DD}/2 - 0.05$ $AV_{DD}/2 - 0.05$	V V
Awake Comparator Hysteresis, All Comparators	V_{HYS}	100	200	mV
Comparator Clamp Voltage BRCV (I_{IN} = 250µA) $\overline{ARCV}$ (I_{IN} = -250µA)	 V_{CB} V_{CA}	 $2AV_{DD}/3 - 0.16$ $AV_{DD}/3 - 0.16$	 $2AV_{DD}/3 + 0.16$ $AV_{DD}/3 + 0.16$	 V V
Output High Voltage (I_{LOAD} = -4.2mA) AXMT, $\overline{BXMT}$)	V_{OH}	$V_{DD} - 0.5$	—	V
Output Low Voltage (I_{LOAD} = 4.2mA) AXMT, $\overline{BXMT}$)	V_{OL}	—	0.4	V
Awake Differential Comparator Common Mode Range	V_{CMR}	1.8	$V_{DD} - 2$	V
V_{DD} Decoupling Capacitance	Cdcp	0.1	—	µF
Clock Frequency Tolerance (see Note 3)	ftol	-1.5	+1.5	%
Transmitter $\overline{BXMT}$ - AXMT Skew (rise & fall) (see Note 4)	tskew	-20	+20	ns
Receiver pin leakage, powered and unpowered ARCV (V_{IN} = AV_{DD}) BRCV (V_{IN} = 0 Vdc)	 I_{ARCV} I_{BRCV}	 -1 -1	 1 1	 µA µA

NOTES:
1. Run I_{DDA}: Entire SMI active.
2. Wake-up Time: Time from bus passive-dominant transition 50% point until all other specs are valid.
3. When SMI is programmed for 41.7 kbps operation, SMI will still function when f_{osc} deviates from nominal value by specified amount.
4. Measured between 50% transition points.

SECTION 8 MULTIFUNCTIONAL TIMER

This timer is a 15-stage multi-functional ripple counter. The features include Timer Over Flow, Power-On Reset (POR), Real Time Interrupt, and COP Watchdog Timer.

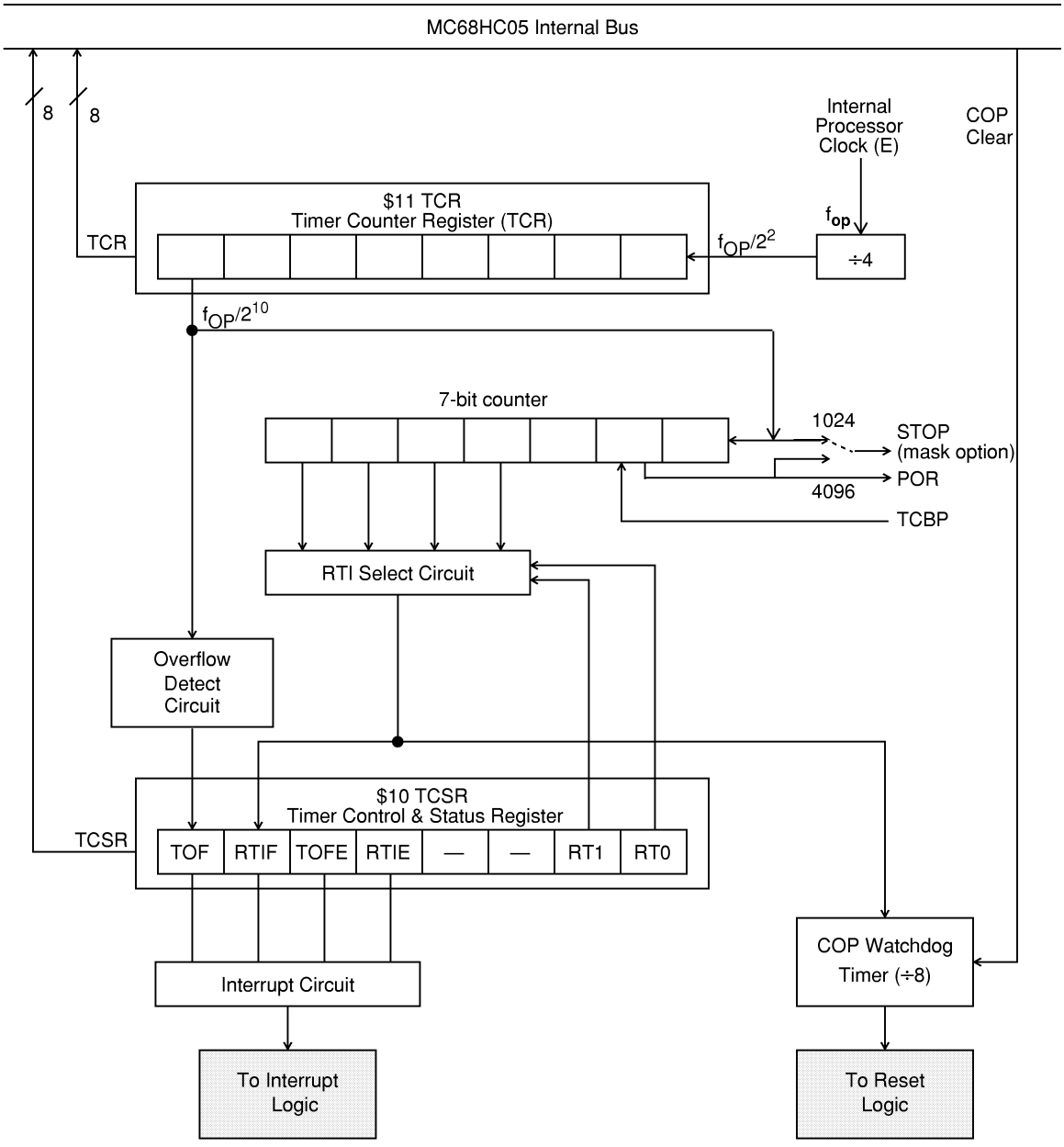


Figure 8-1: Multifunctional Timer Block Diagram

As seen in **Figure 8-1: Multifunctional Timer Block Diagram**, the Timer begins with a fixed divide by four 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 Timer Counter Register (TCR) at address $11.

A timer overflow function is implemented on the last stage of this counter, giving a possible periodic interrupt at the rate of $f_{OP}/1024$.

This circuit is followed by two more stages, with the resulting clock ($f_{OP}/16384$) driving the Real Time Interrupt circuit. The RTI circuit consists of three 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 TOF enable bits and flags are located in the Timer Control and Status Register at location $10.

The POR function is provided by the $f_{OP}/4096$ signal.

The STOP mode recovery timing is provided either by the $f_{OP}/4096$ signal or the $f_{OP}/1024$ signal, depending upon the mask option chosen.

8.1 TIMER CONTROL AND STATUS REGISTER (TCSR)

The TCSR contains the timer interrupt flag, the timer interrupt enable bits, and the real time interrupt rate select bits. **Figure 8-2: Timer Control and Status Register (TCSR)** shows the value of each bit in the TCSR when coming out of reset.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$10	TOF	RTIF	TOFE	RTIE	0	0	RT1	RT0
RESET	0	0	0	0	0	0	1	1

Figure 8-2: Timer Control and Status Register (TCSR)

8.1.1 TOF - Timer Over Flow

TOF 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 TOFE is set. Clearing the TOF is done by writing a “0” to it. Writing a “1” to TOF has no effect on the bit value. A reset clears TOF.

8.1.2 RTIF - Real Time Interrupt Flag

The Real Time Interrupt circuit consists of a three-stage divider and a 1 of 4 selector. The clock frequency that drives the RTI circuit is E/2**14 (or E/16384) with three additional divider stages giving a maximum interrupt period of 65.5 milliseconds at a bus rate of 2 MHz. 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.

8.1.3 TOFE - Timer Over Flow Enable

When this bit is set, a CPU interrupt request is generated when the TOF bit is set. A reset clears this bit.

8.1.4 RTIE - Real Time Interrupt Enable

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

8.1.5 RT1:RT0 - Real Time Interrupt Rate Select

These two bits select one of four taps from the Real Time Interrupt circuit. **Table 8-1: RTI Rates** shows the available interrupt rates with a variety of oscillator frequencies. A reset sets these two bits, which selects the lowest periodic rate and gives 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.

The encoding for these two bits is shown overleaf.

Table 8-1: RTI Rates

	RTI RATES AT f_{osc} FREQUENCY SPECIFIED	
RT1:RT0	**2.0MHz**	**4.0 MHz**
00	16.384 ms	8.192 ms
01	32.768 ms	16.384 ms
10	65.536 ms	32.768 ms
11	131.072 ms	65.535 ms

8.2 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.

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 $3FF0. When the COP is cleared, only the final divide by eight stage (output of the RTI) is cleared.

The minimum COP reset rates are listed in **Table 8-2: Minimum COP Reset Times**. Because it is not readily possible to determine the state of the divider chain ahead of the COP circuit, the COP should be reset within a period equivalent to **seven** real-time interrupts, rather than the eight that one might expect.

This function is a mask option.

Table 8-2: Minimum COP Reset Times

	MINIMUM COP RESET AT f_{osc} FREQUENCY SPECIFIED	
RT1:RT0	**2.0 MHz**	**4.0 MHz**
00	114.688 ms	57.344 ms
01	229.376 ms	114.688 ms
10	458.752 ms	229.376 ms
11	917.504 ms	458.752 ms

8.3 TIMER COUNTER REGISTER (TCR)

The Timer Counter Register is a read-only register that contains the current value of the 8-bit ripple up-counter at the beginning of the timer chain. This counter is clocked at E divided by 4 and can be used for various functions including a software input capture. Extended time periods can be attained using the TOF function to increment a temporary RAM storage location thereby simulating a 16-bit (or more) counter.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$11	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

Figure 8-3: Timer Counter Register

The power-on cycle clears the entire counter chain and begins clocking the counter. After 4096 E-cycles, the power-on reset circuit is released which 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.

8.4 TIMER DURING WAIT MODE

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

8.5 TIMER DURING STOP MODE

The timer is cleared when going into STOP mode. When STOP is exited by an external interrupt or an external $\overline{\text{RESET}}$, the internal oscillator will resume, followed by an internal processor oscillator stabilization delay. The timer is then cleared and operation resumes.

THIS PAGE INTENTIONALLY LEFT BLANK

SECTION 9 16-BIT TIMER

9.1 INTRODUCTION

The timer consists of a 16-bit, software-programmable counter driven by a fixed divide-by-four prescaler. This timer can be used for many purposes, including input waveform measurements while simultaneously generating an output waveform. Pulse widths can vary from several microseconds to many seconds. **Figure 9-1: Timer Block Diagram** below shows a timer block diagram.

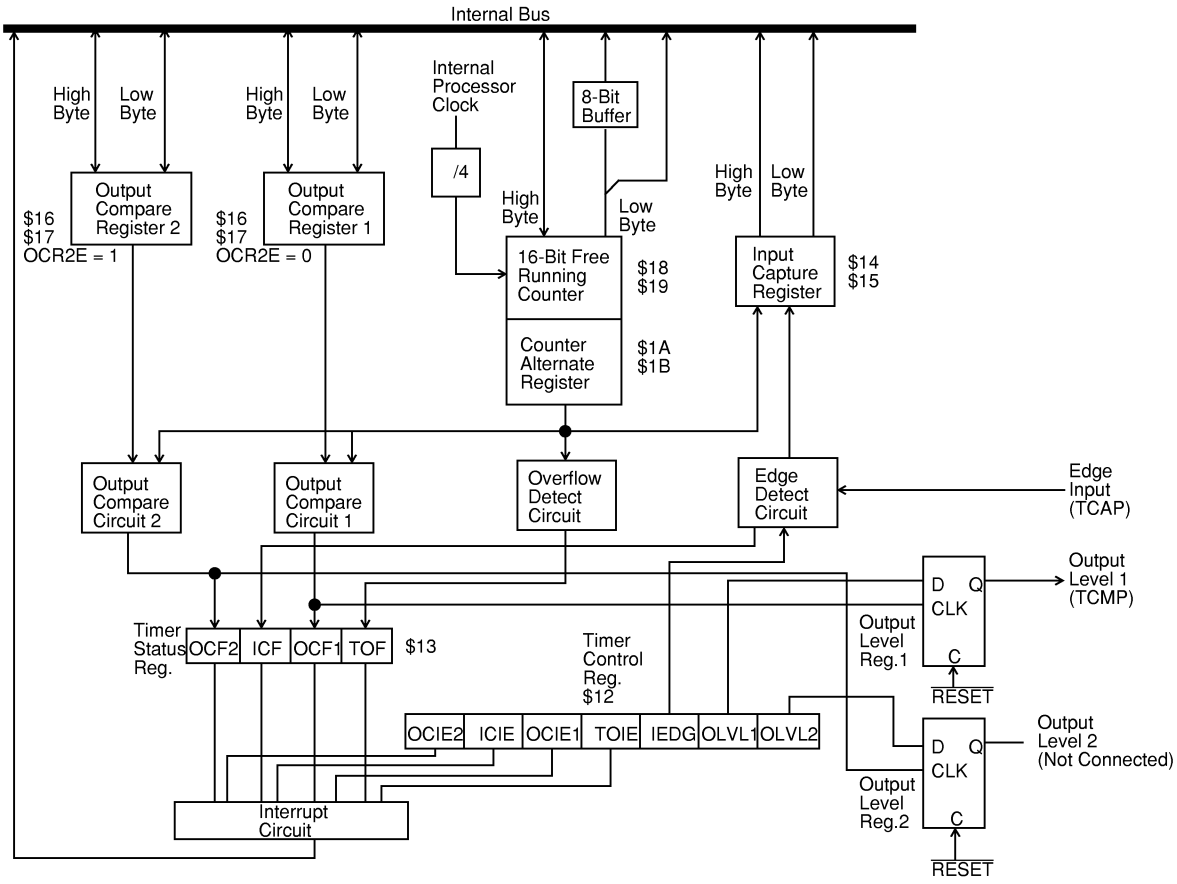


Figure 9-1: Timer Block Diagram

Because the timer has a 16-bit architecture, each specific functional segment (capability) is represented by two registers. These registers contain the high and low byte of that functional segment. 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.

9.2 COUNTER

The key element in the programmable timer is a 16-bit, free-running counter or counter register, preceded by a prescaler that divides the internal processor clock by four. The prescaler gives the timer a resolution of 2.0 microseconds if the internal bus clock is 2.0 MHz. The counter is incremented during the low portion of the internal bus clock. Software can read the counter at any time without affecting its value.

The double-byte, free-running counter can be read from either of two locations, $18-$19 (counter register) or $1A-$1B (counter alternate register). A read from only the least significant byte (LSB) of the free-running counter ($19, $1B) receives the count value at the time of the read. If a read of the free-running counter or counter alternate register first addresses the most significant byte (MSB) ($18, $1A), the LSB ($19, $1B) is transferred to a buffer. This buffer value remains fixed after the first MSB read, even if the user reads the MSB several times. This buffer is accessed when reading the free-running counter or counter alternate register LSB ($19 or $1B) and, thus, completes a read sequence of the total counter value. In reading either the free-running counter or counter alternate register, if the MSB is read, the LSB must also be read to complete the sequence.

The counter alternate register differs from the counter register in one respect: a read of the counter register LSB can clear the timer overflow flag (TOF). Therefore, the counter alternate register can be read at any time without the possibility of missing timer overflow interrupts due to clearing of the TOF.

The free-running counter is initialized to $FFFC during reset and is always a read-only register. During a power-on reset, the counter is also preset to $FFFC and begins running after the oscillator start-up delay. Because the free-running counter is 16 bits preceded by a fixed divided-by-four prescaler, the value in the free-running counter repeats every 262,144 internal bus clock cycles. When the counter rolls over from $FFFF to $0000, the TOF bit is set. An interrupt can also be enabled when counter rollover occurs by setting its interrupt enable bit (TOIE).

NOTE: The I-bit in the CCR should be set while manipulating both the high and low byte register of a specific timer function to ensure that an interrupt does not occur.

9.3 OUTPUT COMPARE REGISTER 1 (OCR1)

The 16-bit Output Compare Register 1 is made up of two 8-bit registers at locations $16 (MSB) and $17 (LSB). This register is accessed when the Output Compare Register 2 Enable bit (OCR2E) is set to zero. This is the default condition on exiting reset.

The output compare register is used for several purposes, such as indicating when a period of time has elapsed. All bits are readable and writable and are not altered by the timer hardware or reset. If the compare function is not needed, the 2 bytes of the output compare register can be used as storage locations.

The output compare register contents are compared with the contents of the free-running counter continually, and if a match is found, the corresponding output compare flag (OCF1) bit is set and the corresponding output level (OLVL1) bit is clocked to an output level register. The output compare register values and the output level bit should be changed after each successful comparison to establish a new elapsed time-out. An interrupt can also accompany a successful output compare provided the corresponding interrupt enable bit (OCIE1) is set.

After a processor write cycle to the output compare register containing the MSB ($16), the output compare function is inhibited until the LSB ($17) is also written. The user must write both bytes (locations) if the MSB is written first. A write made only to the LSB ($17) will not inhibit the compare function. The free-running counter is updated every four internal bus clock cycles. The minimum time required to update the output compare register is a function of the program rather than the internal hardware. When updating OCR1, the other output compare function (OCR2) is unaffected and will not be inhibited.

The processor can write to either byte of the output compare register without affecting the other byte. The output level (OLVL1) bit is clocked to the output level register regardless of whether the output compare flag (OCF1) is set or clear. This output level register is connected to the Timer Compare (TCMP) pin.

9.4 OUTPUT COMPARE REGISTER 2 (OCR2)

The 16-bit Output Compare Register 2 is made up of two 8-bit registers at locations $16 (MSB) and $17 (LSB). This register is accessed when the Output Compare Register 2 Enable bit (OCR2E) is set to one.

The output compare register is used for several purposes, such as indicating when a period of time has elapsed. All bits are readable and writable and are not altered by the timer hardware or reset. If the compare function is not needed, the 2 bytes of the output compare register can be used as storage locations.

The output compare register contents are compared with the contents of the free-running counter continually, and if a match is found, the corresponding output compare flag (OCF2) bit is set and the corresponding output level (OLVL2) bit is clocked to an output level register. The output compare register values and the output level bit should be changed after each successful comparison to establish a new elapsed timeout. An interrupt can also accompany a successful output compare provided the corresponding interrupt enable bit (OCIE2) is set.

After a processor write cycle to the output compare register containing the MSB ($16), the output compare function is inhibited until the LSB ($17) is also written. The user must write both bytes (locations) if the MSB is written first. A write made only to the LSB ($17) will not inhibit the compare function. The free-running counter is updated every four internal bus clock cycles. The minimum time required to update the output compare register is a function of the program rather than the internal hardware.When updating OCR2, the other output compare function (OCR1) is unaffected and will not be inhibited.

The processor can write to either byte of the output compare register without affecting the other byte. The output level (OLVL2) bit is clocked to the output level register regardless of whether the output compare flag (OCF2) is set or clear. This output level register is not connected to any pins in this implementation. However the interrupt capability is still fully operative and may be used to generate timed interrupts.

9.5 INPUT CAPTURE REGISTER (ICR)

Two 8-bit registers, which make up the 16-bit input capture register, are read-only and are used to latch the value of the free-running counter after the corresponding input capture edge detector senses a defined transition. The level transition which triggers the counter transfer is defined by the corresponding input edge bit (IEDG). Reset does not affect the contents of the input capture register except when exiting stop mode.

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 bus clock preceding the external transition. This delay is required for internal synchronization. Resolution is one count of the free running counter, which is four internal bus clock cycles.

The free-running counter contents are transferred to the input capture register on each proper signal transition regardless of whether the input capture flag (ICF) is set or clear. The input capture register always contains the free-running counter value that corresponds to the most recent input capture.

After a read of the input capture register ($14) MSB, the counter transfer is inhibited until the LSB ($15) is also read. This characteristic causes the time used in the input capture software routine and its interaction with the main program to determine the minimum pulse period. A read of the input capture register LSB ($15) does not inhibit the free-running counter transfer since they occur on opposite edges of the internal bus clock.

9.6 TIMER CONTROL REGISTER (TCR)

The TCR is a read/write register containing eight control bits.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$12	ICIE	OCIE1	TOIE	OCIE2	OCR2E	OLVL2	IEDG	OLVL1
RESET	0	0	0	0	0	0	0	0

Figure 9-2: Timer Control Register

ICIE - Input Capture Interrupt Enable
1 = Interrupt enabled
0 = Interrupt disabled

OCIE1 - Output Compare Interrupt Enable 1
1 = Interrupt enabled
0 = Interrupt disabled

TOIE - Timer Overflow Interrupt Enable
1 = Interrupt enabled
0 = Interrupt disabled

OCIE2 - Output Compare Interrupt Enable 2
1 = Interrupt enabled
0 = Interrupt disabled

OCR2E - Output Compare Register 2 Enable
1 = Output Compare Register 2 enabled at locations $16,$17
0 = Output Compare Register 1 enabled at locations $16,$17

OLVL2 - Output Level 2
Value of output level is clocked into Output Level Register 2 by the next successful output compare, but will not appear on any pin.
1 = High output
0 = Low output

IEDG - Input Edge
Value of input edge determines which level transition on TCAP pin will trigger free-running counter transfer to the input capture register.
1 = Positive edge
0 = Negative edge

OLVL1 - Output Level 1
Value of output level is clocked into output level register by the next successful output compare and will appear on the TCMP pin.
1 = High output
0 = Low output
The TCMP output pin will always be at a low level following a reset.

9.7 TIMER STATUS REGISTER (TSR)

The TSR is a read-only register containing three status flag bits.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$13	ICF	OCF1	TOF	OCF2	0	0	0	0
RESET	0	0	0	0	0	0	0	0

Figure 9-3: Timer Status Register

ICF - Input Capture Flag
1 = Flag set when selected polarity edge is sensed by input capture edge detector
0 = Flag cleared when TSR and input capture low register ($15) are accessed

OCF1 - Output Compare Flag 1
1 = Flag set when Output Compare Register 1 contents match the free-running counter contents
0 = Flag cleared when TSR and Output Compare Low Register 1 ($17) are accessed

TOF - Timer Overflow Flag
1 = Flag set when free-running counter transition from $FFFF to $0000 occurs
0 = Flag cleared when TSR and counter low register ($19) are accessed

OCF2 - Output Compare Flag 2
1 = Flag set when Output Compare Register 2 contents match the free-running counter contents
0 = Flag cleared when TSR and Output Compare Low Register 1 ($17) are accessed

Bits 0-3 - Not used
Always read zero

Accessing the timer status register satisfies the first condition required to clear status bits. The remaining step is to access the register corresponding to the status bit.

A problem can occur when using the timer overflow function and reading the free-running counter at random times to measure an elapsed time. Without incorporating the proper precautions into software, the timer overflow flag could unintentionally be cleared if:

1. The timer status register is read or written when TOF is set.
2. The LSB of the free-running counter is read but not for the purpose of servicing the flag.

The counter alternate register at address $1A and $1B contains the same value as the free-running counter (at address $18 and $19); therefore, this alternate register can be read at any time without affecting the timer overflow flag in the timer status register.

9.8 TIMER DURING WAIT MODE

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

9.9 TIMER DURING STOP MODE

In the STOP mode, the timer stops counting and holds the last count value. If an interrupt is used to exit STOP mode, the timer will resume counting once the STOP recovery period has completed. If $\overline{\text{RESET}}$ is used to exit STOP mode, the counter is forced to $FFFC. During STOP, if at least one valid input capture edge occurs at the TCAP pin, the input capture detect circuit is armed. This does not set any timer flags nor wake up the MCU, but when the timer resumes activity, there is an active input capture flag and data from the first valid edge that occurred during the STOP mode. If RESET is used to exit STOP mode, then no input capture flag or data remains, even if a valid input capture edge occurred.

THIS PAGE INTENTIONALLY LEFT BLANK

SECTION 10 SYNCHRONOUS SERIAL INTERFACE

The Synchronous Serial Interface (SSI) is a three-wire Master/Slave system including Serial Clock (SCK), Serial Data Input (SDI), and Serial Data Output (SDO). Data is transferred 8 bits at a time. A software programmable option determines whether the SSI transfers data MSB or LSB first, and an interrupt may be generated at the completion of each transfer. When operating as a master device, the serial clock speed is selectable between four rates; as a slave device, the clock speed may be chosen over a wide range.

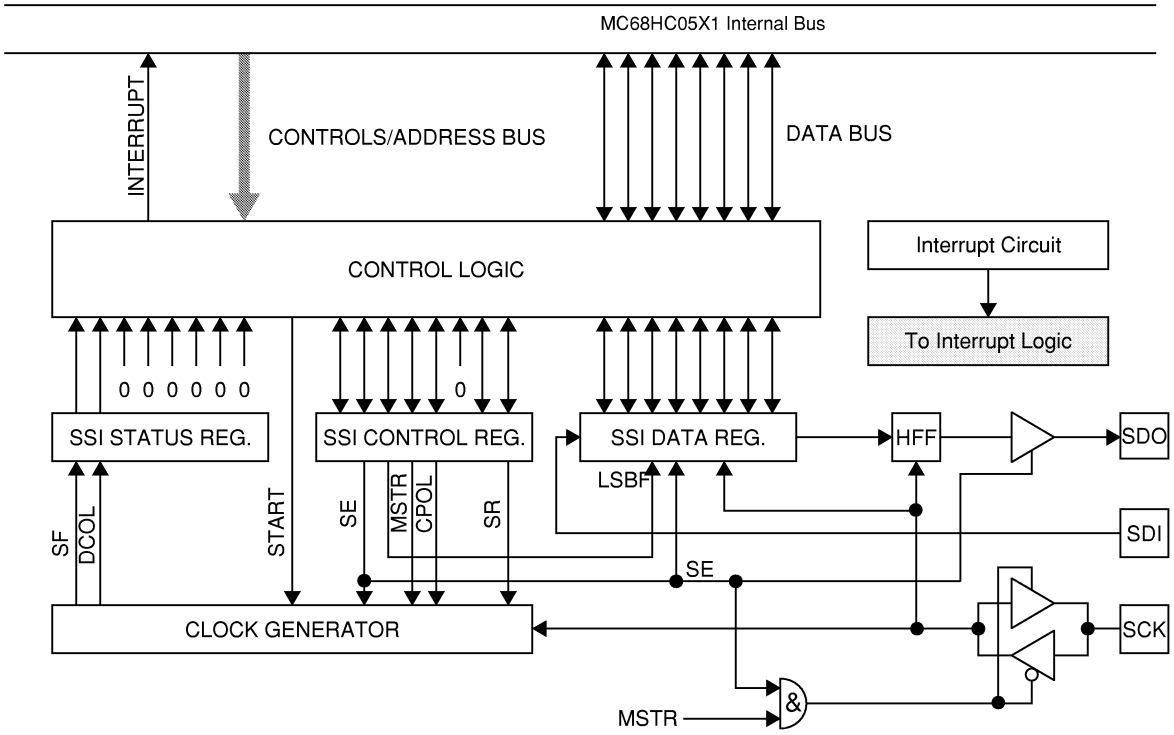


Figure 10-1: SSI Block Diagram

Transmission, in Master mode is initiated by a write to the SSI Data Register (SDR). A transfer cannot be initiated in Slave mode, the external master will initiate the transfer. The programmer must choose between Master or Slave mode before the SSI is enabled. It is up to the programmer to ensure that only one Master exists in the system at any one time. All devices in the system must operate with the same clock polarity and data rates. Slaves should always be disabled before the Master is disabled.

10.1 SIGNAL FORMAT

10.1.1 SERIAL CLOCK (SCK)

In Master mode (MSTR = 1), the Serial Clock (SCK) pin is an output with four selectable frequencies. This pin will be high (CPOL = 1) or low (CPOL = 0) between transmissions.

In Slave Mode (MSTR = 0), the SCK pin is an input and the clock must be supplied by an external master with a maximum frequency of $f_{OP}/2$. There is no minimum SCK frequency. This pin should be driven high (CPOL = 1) or low (CPOL = 0) between transmissions by the external master and must be stable before the SSI is first enabled (SE = 1).

Data is always captured at the Serial Data In (**SDI**) pin on the **rising** edge of SCK.

Data is always shifted out and presented at the Serial Data Out (**SDO**) pin on the **falling** edge of SCK.

10.1.2 SERIAL DATA OUT (SDO)

Prior to enabling the SSI (SE = 0), the Serial Data Out (SDO) pin will be three-stated. When the SSI is enabled (SE = 1) the SDO pin will be initially driven to either an unknown value (CPOL = 1) or the value of the first data bit to be sent (CPOL = 0).

The data can be sent or received in either MSB first format (LSBF = 0) or LSB first format (LSBF = 1).

If (CPOL = 1), the first falling edge of SCK will shift the first data bit out to the SDO pin. Subsequent falling edges of SCK will shift the remaining data bits out.

If (CPOL = 0), the first data bit will be driven out to the SDO pin before the first rising edge of SCK. Subsequent falling edges of SCK will shift the remaining data bits out.

10.1.3 SERIAL DATA IN (SDI)

The Serial Data In (SDI) input pin will accept data once the SSI is enabled. Valid data must be present at least 100 ns before the rising edge of the clock and remain valid for 100 ns after the edge. New data may be presented to the SDI pin on the falling edge of SCK.

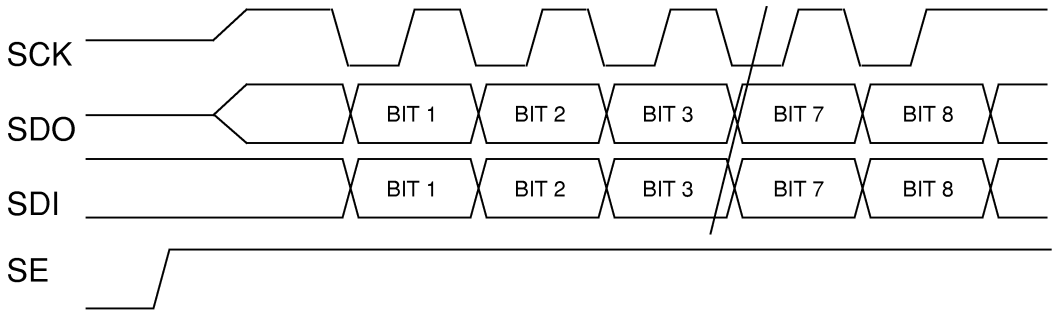


Figure 10-2: Serial I/O Port Timing (CPOL=1)

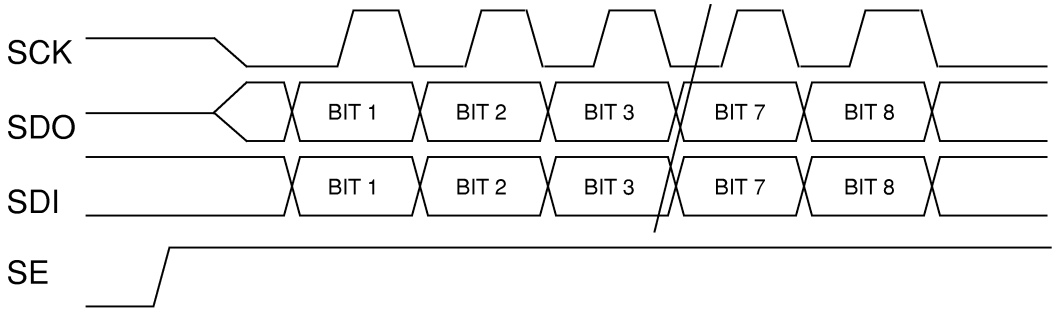


Figure 10-3: SSI Control Register

10.2 SSI CONTROL REGISTER (SCR)

This register is located at address $001C and contains 6 bits. A reset clears all of these bits, except Bit 3 which is set. Writes to this register during a transfer should be avoided, with the exception of clearing the SE bit to disable the SSI.

In addition, the clock polarity, rate, data format and master/slave selection should not be changed while the SSI is enabled (SE = 1) or being enabled. Always disable the SSI first, by clearing the SE bit, before altering these control bits within the SSI Control Register (SCR).

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$1C	SIE	SE	LSBF	MSTR	CPOL	0	SR1	SR0
RESET	0	0	0	0	1	0	0	0

Figure 10-4: SSI Control Register

10.2.1 SIE - SSI INTERRUPT ENABLE

This bit determines whether an interrupt request should be generated when a transfer is complete.

When set, an interrupt request will be made if the CPU is in the Run or Wait mode of operation and status flag bit SF is set.

When cleared, no interrupt requests will be made by the SSI.

10.2.2 SE - SSI ENABLE

When set, this bit enables the SSI, begins driving SDO as an output, pin SDI as an input and enables pin SCK.

When SE is cleared, any transmission in progress is aborted, the bit counter is reset, pins SCK and SDO are three-stated and pin SDI remains an input but all data is ignored.

10.2.3 LSBF - LEAST SIGNIFICANT BIT (LSB)FIRST

When set, data will be sent and received in least significant bit (LSB) first format.

When cleared, data will be sent and received in a most significant bit (MSB) first format.

10.2.4 MSTR - MASTER MODE

When set, this bit configures the SSI to the Master mode. This means that the transmission is initiated by a write to the data register and the SCK pin becomes an output providing a synchronous data clock at a rate determined by the SR bit.

When cleared, this bit configures the SSI to the Slave Mode and aborts any transmission in progress. Transfers are initiated by an external master which should supply the clock to the SCK pin.

Pins SDO and SDI retain their function in either Master or Slave Mode.

10.2.5 CPOL - CLOCK POLARITY

The Clock Polarity bit controls the state of the SCK pin between transmissions.

When this bit is set, pin SCK will be high between transmissions.

When this bit is cleared, pin SCK will be low between transmissions.

In both cases the data is latched on the rising edge of SCK for serial input and is valid on the rising edge of SCK for serial output. A reset sets this bit.

10.2.6 SR0,1 - SSI RATE

These bits determine the frequency of SCK when in Master mode (MSTR = 1). They have no effect in Slave mode (MSTR = 0).

Table 10-1: SSI Rates

	SCK RATES (Hz) AT f_{osc} FREQUENCY SPECIFIED:	
SR1:SR0	**2.0MHz**	**4.0 MHz**
00	62.5k	125k
01	125k	250k
10	250k	500k
11	500k	1M

10.3 SSI STATUS REGISTER (SSR)

This register is located at address $001D and contains 2 bits. Reset clears both of these bits.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$1D	SF	DCOL	0	0	0	0	0	0
RESET	0	0	0	0	0	0	0	0

Figure 10-5: SSI Status Register

10.3.1 SF - SSI FLAG

This bit is set upon occurrence of the last rising clock edge and indicates that a data transfer has taken place.

If the SSI is configured as a slave and the interrupts are disabled (MSTR=0, SIE=0), SF has no effect on any further data transfers and can be ignored without a problem.

If the SSI is configured as a master (MSTR=1), or if interrupts are enabled as a slave (MSTR=0, SIE=1), SF must be cleared between transfers.

SF is cleared by reading the SSR with SF set followed by a read or write of the serial data register. If it is cleared before the last edge of the next byte it will be set again. A reset or disabling of the SSI (SE=0) also clears this bit.

10.3.2 DCOL - DATA COLLISION

This is a read only status bit, which indicates that an invalid access to the data register has been made. An invalid access is:

- An access of the SDR register in the middle of a transfer (after the first falling edge of SCK and before SF is set).

- An access of the SDR register made before an access of the SSR register (after SF is set).

DCOL is cleared by reading the status register with SF set followed by a read or write of the data register. A reset or disabling of the SSI (SE=0) also clears this bit.

10.4 SSI DATA REGISTER (SDR)

This register is located at address $001E and is both the transmit and receive data register.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$1E	D7	D6	D5	D4	D3	D2	D1	D0
RESET	X	X	X	X	X	X	X	X

X = Unknown

Figure 10-6: SSI Data Register

This system is not double buffered but writes to this register during transfers are masked and will not destroy the previous contents.

Writes to this register while the SSI is enabled (SE=1) will initiate a data transfer, within a time equivalent to one cycle of the SCK signal, if the SSI is configured as a master (MSTR=1). If the SSI is configured as a slave (MSTR=0) the data will just be held in preparation for the master initiated transfer.

Writes to this register while the SSI is disabled (SE=0) will just cause the data to be held. Subsequent enabling of the SSI will not cause a data transfer to automatically occur until the conditions given in the previous paragraph are met.

The SDR can be read at any time but if a transfer is in progress the results may be ambiguous. The contents of this register might be altered whenever the CPOL bit is altered. This register should only be written to when the SSI is enabled (SE=1).

10.5 OPERATION DURING WAIT MODE

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

10.6 OPERATION DURING STOP MODE

In the STOP mode, the SSI halts operation. The SDO and SCK pins will maintain their states.

If the SSI was nearing completion of a transfer when the STOP mode is entered, it might be possible for the SSI to generate an interrupt request and thus cause the processor to immediately exit the STOP mode. To prevent this occurrence, the programmer should ensure that all transfers are complete before entering the STOP mode.

If the SSI is configured to Slave mode then further care should be taken in entering STOP mode. The SCK pin will still accept a clock from an external master, allowing potentially unwanted transfers to take place and power consumption to be increased. The SSI will not generate interrupt requests in this situation but, on exiting STOP mode through some other means, the SF flag may be found to be set and an interrupt request will be generated if SIE is also set at this point.

To avoid these potential problems, it is safer to disable the SSI completely (SE = 0) before entering STOP mode.

10.7 USING SSI AS A DIGITAL PORT

If the SSI is unused in a system then the programmer may use the SDO pin as a digital output port and the SDI pin as a digital input port. This may be accomplished by configuring the SSI to Master mode (MSTR = 1) and enabling it (SE =1). Writing $FF to the SDR will cause the SDO pin to be driven high indefinitely. Writing $00 to the SDR will cause the SDO pin to be driven low indefinitely. Each write will also cause eight samples of the state of the SDI pin to be shifted into the SDR where they may be examined after the SF flag has been set.

THIS PAGE INTENTIONALLY LEFT BLANK

SECTION 11 ELECTRICAL SPECIFICATIONS

11.1 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 x V_{DD} + 0.3	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	25	mA
Operating Temperature Range MC68HC05X1CP (Extended)	T_A	T_L to T_H -40 to +85	°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 (for example, either V_{SS} or V_{DD}).

11.2 THERMAL CHARACTERISTICS

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

11.3 DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc, T_A = - 40°C to +85°C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage I_{Load} = 10.0 µA I_{Load} = -10.0 µA	 V_{OL} V_{OH}	 — V_{DD}-0.1	 — —	 0.1 —	 V
Output High Voltage (I_{Load} = -0.8 mA) Ports A,B,C,D, TCMP, SDO, SCK	V_{OH}	V_{DD}-0.8	—	—	V
Output Low Voltage (I_{Load} = 1.6 mA) Ports A,B,C D, TCMP, SDO, SCK	V_{OL}	—	—	0.40	V
Input High Voltage Ports A, B, C, D, SCK, SDI, TCAP, $\overline{IRQ}$,$\overline{RESET}$, OSC1	V_{IH}	0.7. x	—	V_{DD}	V
Input Low Voltage Ports A, B, C, D, SCK, SDI, TCAP, $\overline{IRQ}$,$\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	0.3 x V_{DD}	V
Supply Current (see Notes) Run (4.0 MHz) Wait Stop 25°C -40°C to +85°C	 I_{DD} I_{DD} I_{DD} I_{DD}	 — — — —	 TBD TBD 100 150	 TBD TBD TBD TBD	 mA mA µA µA
I/O Ports Hi-Z Leakage Current PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD2, SCK	I_{OZ}	—	—	10	µA
Input Current SDI, TCAP, $\overline{IRQ}$, $\overline{RESET}$,OSC1	I_{IN}	—	—	1	µA
Capacitance Ports (as Input or Output) SCK, SDI, TCAP, $\overline{IRQ}$, $\overline{RESET}$	 C_{OUT} C_{IN}	 — —	 — —	 12 8	 pF pF

NOTES:

1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only.
3. Wait I_{DD}: Multifunctional Timer and SMI systems active only.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external squarewave clock source, all inputs 0.2 V from rail; no dc loads, less than 50 pF 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. Wait I_{DD} is affected linearly by the OSC2 capacitance.
7. The digital V_{DD} supply tolerance is completely independent of that for the analog AV_{DD} supply.

11.4 SSI TIMING

(V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc, T_A = -40°C to +85°C, unless otherwise noted)

Num.	Characteristic	Symbol	Min	Max	Unit
1	Cycle Time Master Slave	 t_{CYCM} t_{CYCS}	 2 note 1 2	 16 note 2 —	 t_{CYC} t_{CYC}
2	Clock (SCK) Low Time	t_{SCKL}	467	—	ns
3	Clock (SCK) High Time	t_{SCKH}	467	—	ns
4	SDO Data Valid Time	t_V	—	200	ns
5	SDO Hold Time	t_{HO}	0	—	ns
6	SDI Setup Time	t_S	100	—	ns
7	SDI Hold Time	t_H	100	—	ns
8	SDO, SCK (Master) Enable/Disable Time	t_{SEN}	—	1	t_{CYC}

NOTES:
1. SR1 = 1, SR0 = 1
2. SR1 = 0, SR0 = 0

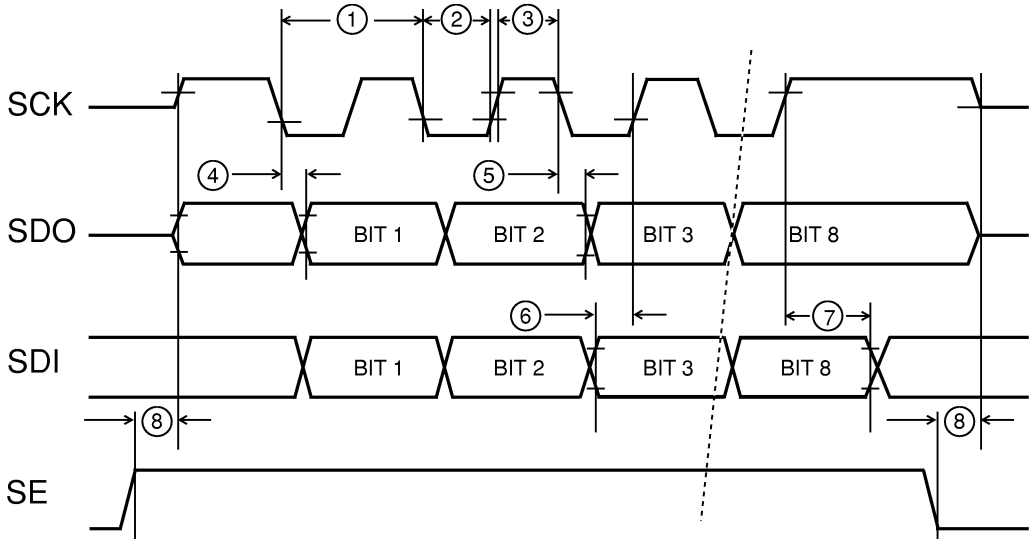


Figure 11-1: SSI I/O Port Timing (CPOL = 1)

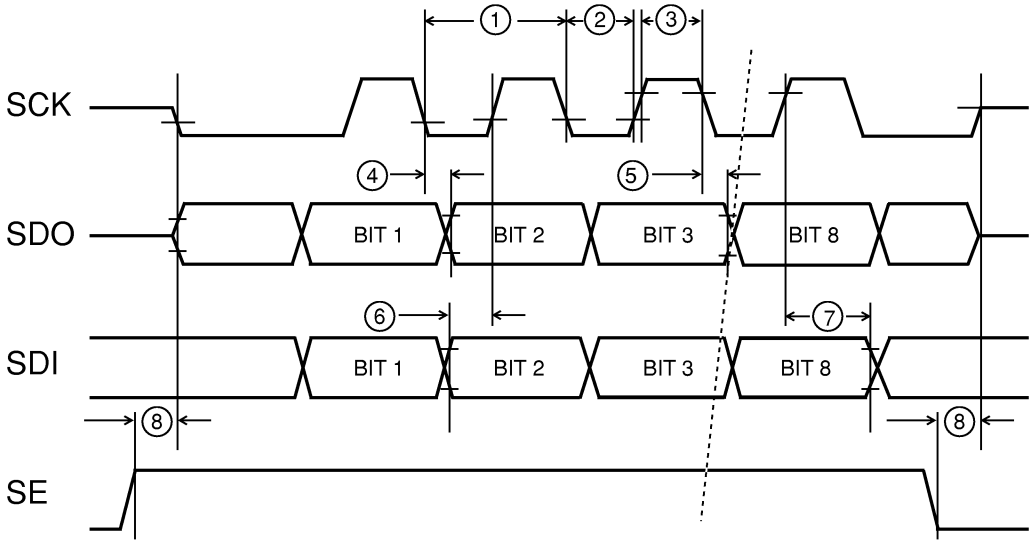


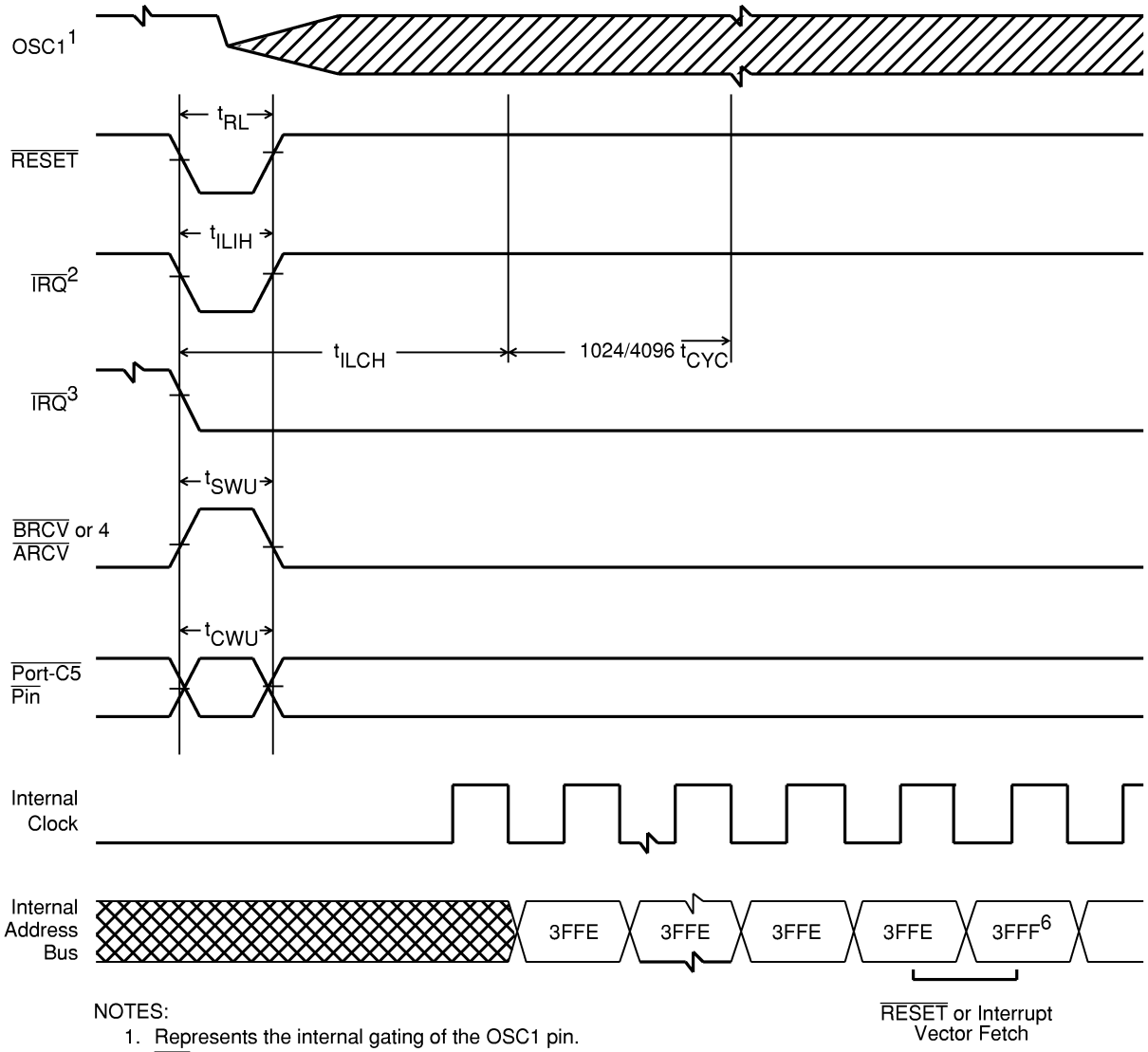
Figure 11-2: SSI I/O Port Timing (CPOL = 0)

11.5 CONTROL TIMING

(V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc, T_A = -40°C to +85°C, unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Crystal or Ceramic Resonator External Clock	 f_{OSC} f_{OSC}	 — dc	 4.06 4.06	 MHz MHz
Internal Operating Frequency Crystal or Ceramic Resonator (f_{osc} ÷ 2) External Clock (f_{osc} ÷ 2)	 f_{OP} f_{OP}	 — dc	 2.03 2.03	 MHz MHz
Internal Cycle Time	t_{CYC}	487	—	ns
Crystal Oscillator Start-up Time and Stop Recovery Time	t_{OXOV}	—	100	ms
Ceramic Resonator Start-up Time and Stop Recovery Time	t_{CER}	—	10	ms
$\overline{\text{RESET}}$ Pulse Width	t_{RL}	1.5	—	t_{CYC}
Interrupt Pulse Width Low (Edge-Triggered)	t_{ILIH}	125	—	ns
Interrupt Pulse Period	t_{ILIL}	*	—	t_{CYC}
SMI Wake-up Pulse Width	t_{SWU}	TBD	—	ns
Port-C Wake-up Pulse Width	t_{CWU}	100	—	ns
OSC1 Pulse Width	t_{OH},t_{OL}	100	—	ns

NOTES:
1. The minimum period T_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{CYC}.



NOTES:
1. Represents the internal gating of the OSC1 pin.
2. $\overline{IRQ}$ pin edge-sensitive mask option.
3. $\overline{IRQ}$ pin level and edge-sensitive mask option.
4. Passive to Dominant transition on either SMI pin.
5. Port-C wake-up mask option selected.
6. $\overline{RESET}$ vector address shown for timing example.

Figure 11-3: Stop Recovery Timing Diagram

Home Page:
www.freescale.com
email:
support@freescale.com
USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
(800) 521-6274
480-768-2130
support@freescale.com
Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com
Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku
Tokyo 153-0064, Japan
0120 191014
+81 2666 8080
support.japan@freescale.com
Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate,
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com
For Literature Requests Only:
Freescale Semiconductor
Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
(800) 441-2447
303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor
@hibbertgroup.com

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

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