MC68HC05i8/D
Rev. 1

HC05

MC68HC05i8

ADVANCE
INFORMATION

MOTOROLA

MC68HC05i8

High-density Complementary Metal Oxide Semiconductor (HCMOS) Microcomputer Unit

This document contains information on a new product. Specifications and information herein are subject to change without notice.

Motorola reserves the right to make changes without further notice to any products herein to improve reliability, function or design. Motorola does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part. Motorola and Ⓜ are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

All Trade Marks recognised.

Table of Contents

Paragraph Number — Title — Page Number

INTRODUCTION 1-1

FUNCTIONAL PIN DESCRIPTION 2-1

Table of Contents (Continued)

Paragraph Number	Title	Page Number

CPU CORE AND INSTRUCTION SET 3-1

Table of Contents (Continued)

Paragraph Number — Title — Page Number

Table of Contents (Continued)

Paragraph Number | Title | Page Number

PARALLEL INPUT/OUTPUT PORTS 6-1

PROGRAMMABLE TIMER 7-1

Table of Contents (Continued)

Table of Contents (Continued)

Paragraph Number **Title** **Page Number**

Table of Contents (Concluded)

ELECTRICAL SPECIFICATIONS 11-1

MECHANICAL DATA AND ORDERING INFORMATION 12-1

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

SECTION 1
INTRODUCTION

The MC68HC05i8, with on-board M68000 interface module, is designed around the industry standard M68HC05 CPU core. The M68000 interface module allows efficient communication between an M68000 family processor (the Host) and an MC68HC05i8 (the MCU) serving as a peripheral device. The transfer of data is managed via an 8-bit data bus and a set of seven control and eight data registers. Information is held in the data registers in 4-stage FIFO buffers that can be accessed by both the M68000 processor and the MC68HC05i8. Other features of the device include a 16-bit programmable timer, a multi-purpose core timer, two independent Serial Communications Interface (SCI) modules and a Computer Operating Properly (COP) watchdog timer. The MC68HC05i8 is available in a 64-pin QFP package.

1.1 FEATURES

- Industry standard M68HC05 core and instruction set
- On-chip oscillator with divide by 2, 4 or 8 mask option
- 7936 bytes of user ROM
- 224 bytes of RAM
- Power saving STOP and WAIT modes
- M68000 interface module with 8-bit data bus, 4 register select lines and 4 control lines
- 16-bit programmable timer with two Input Captures and one Output Compare
- Multi-purpose Core Timer with Computer Operating Properly (COP) watchdog and Real Time Interrupt (RTI)
- Two independent, full duplex SCI subsystems each with its own Baud Rate Generator programmable to 32 different rates
- Receiver wake-up function for use in multi-receiver systems
- Mask option selection for interrupt sensitivity, oscillator division ratio, COP enable/disable and STOP enable/disable
- Temperature range: 0 – 70 °C
- 64-pin Quad Flat Pack (QFP) package

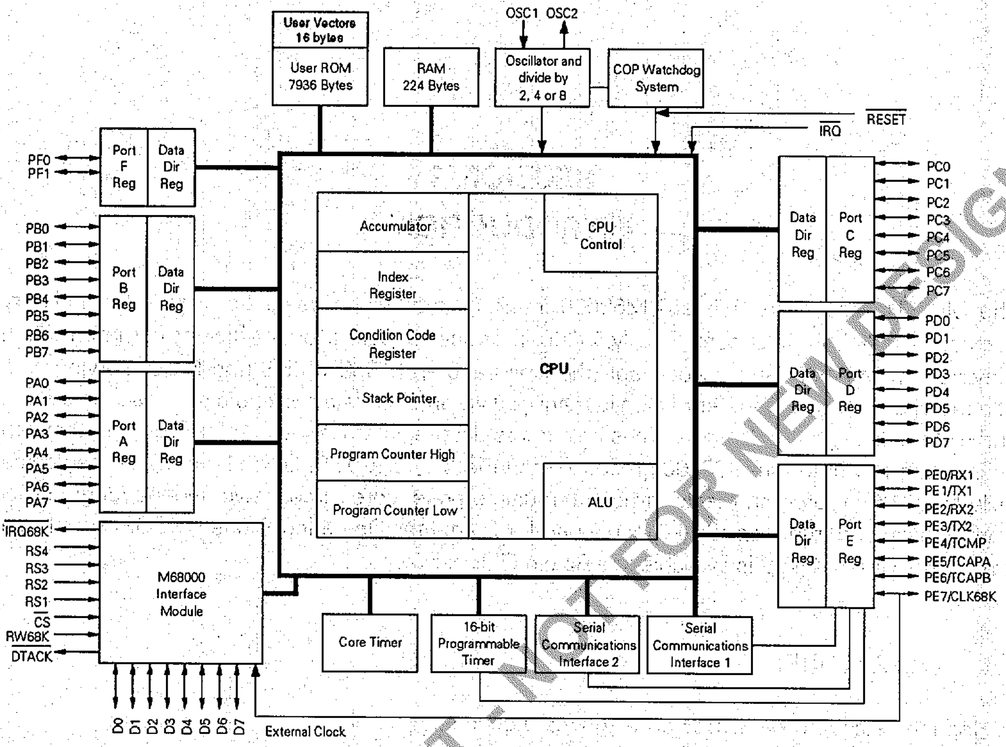


Figure 1-1 Functional Block Diagram

1.2 MASK OPTIONS

There are four mask options on the MC68HC05i8 which are programmed during the wafer fabrication process and must be specified on the order form. These options are listed below.

- Interrupt sensitivity – edge sensitive only or edge-and-level sensitive
- Oscillator division ratio selection – OSC/2, OSC/4 or OSC/8
- COP enable/disable
- STOP instruction enable/disable.

SECTION 2
FUNCTIONAL PIN DESCRIPTION

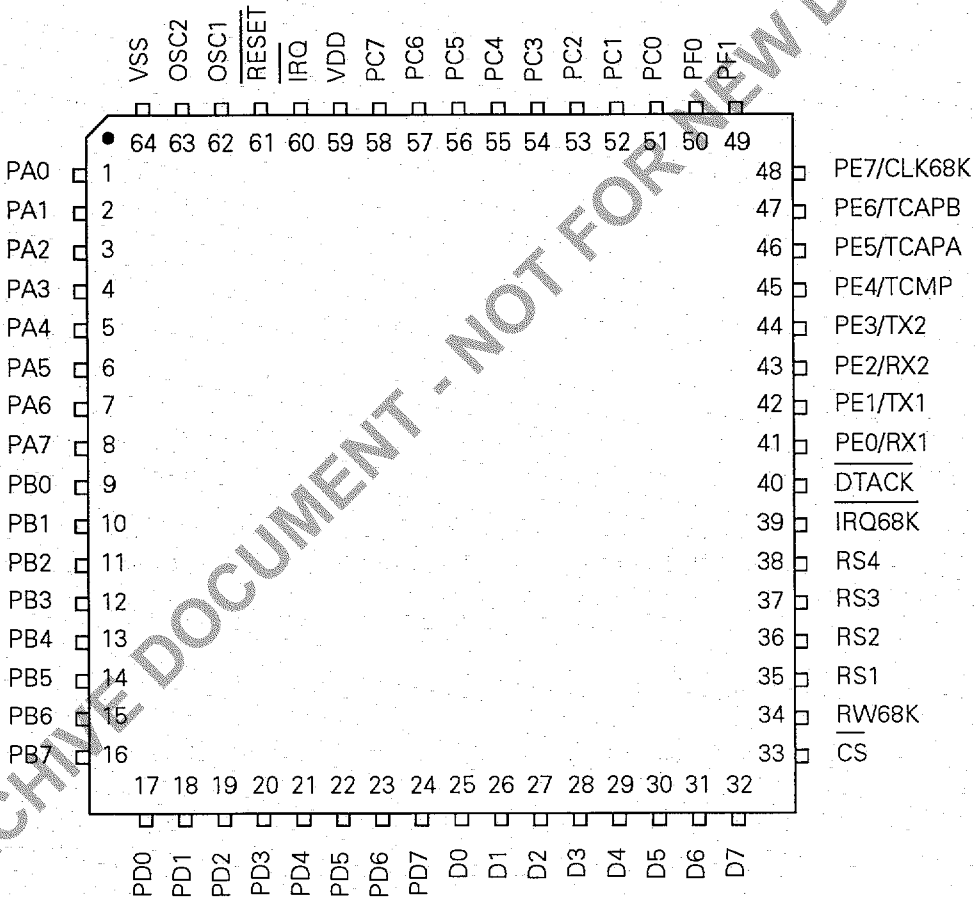


Figure 2-1 Pinout for 64-pin QFP (Quad Flat Pack)

All signal inputs on the MC68HC05i8 except $\overline{\text{RESET}}$ and OSC1 are TTL compatible.

2.1 VDD AND VSS

Power is supplied to the microcomputer via these two pins. VDD is the positive supply and VSS is ground.

It is in the nature of CMOS designs that very fast signal transitions occur on the MCU pins. These short rise and fall times place very high short-duration current demands on the power supply. To prevent noise problems, special care must be taken to provide good power supply by-passing at the MCU. By-pass capacitors should have good high-frequency characteristics and be as close to the MCU as possible. By-passing requirements vary, depending on how heavily the MCU pins are loaded.

2.2 OSC1/OSC2

These pins provide control input for an on-chip clock oscillator circuit. A crystal, ceramic resonator or external clock signal connected to these pins provides the oscillator clock. The oscillator frequency (f_{OSC}) is divided by 2, 4 or 8, chosen via a mask option, to provide the internal bus frequency (f_{OP}).

2.2.1 Crystal

The circuit shown in Figure 2-2(b) is recommended when using a crystal. The internal oscillator is designed to interface with an AT-cut parallel-resonant quartz crystal resonator in the frequency range specified for f_{OSC} (refer to Section 11.4 AC Electrical Characteristics). Use of an external CMOS oscillator is recommended when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimise output distortion and start-up stabilisation time.

2.2.2 Ceramic Resonator

A ceramic resonator may be used instead of the crystal in cost-sensitive applications. The circuit in Figure 2-2(b) is recommended when using a ceramic resonator. The manufacturer of the particular ceramic resonator being considered should be consulted for specific information.

2.2.3 External Clock

An external clock should be applied to the OSC1 input with the OSC2 pin not connected, as shown in Figure 2-2(d). The t_{OXOV} specification does not apply when using an external clock input. The equivalent specification of the external clock source should be used in lieu of t_{OXOV}.

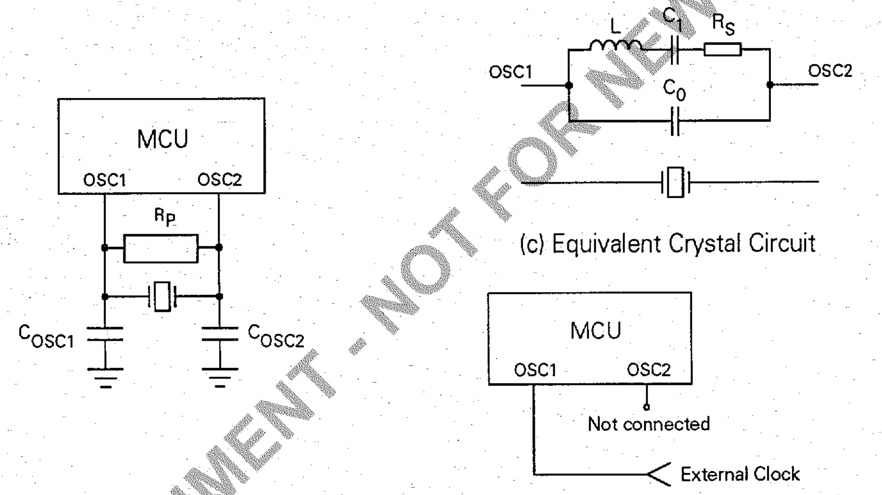
Crystal

	2 MHz	4 MHz	Units
R_S (max)	400	75	Ω
C_0	5	7	pF
C_1	0.008	0.012	µF
C_{OSC1}	15–40	15–30	pF
C_{OSC2}	15–30	15–25	pF
R_P	10	10	MΩ
Q	30k	40k	—

Ceramic Resonator

	2 – 4 MHz	Units
R_S (typical)	10	Ω
C_0	40	pF
C_1	4.3	pF
C_{OSC1}	30	pF
C_{OSC2}	30	pF
R_P	1–10	MΩ
Q	1250	—

(a) Crystal/Ceramic Resonator Parameters



(c) Equivalent Crystal Circuit

(b) Crystal/Ceramic Resonator Oscillator Connections

(d) External Clock Source Connections

Figure 2-2 Oscillator Connections

2.3 $\overline{\text{RESET}}$

This active low input pin is used to reset the MCU. Applying a logic zero to this pin forces the device to a known start-up state. An external RC-circuit can be connected to this pin to generate a power-on reset (POR) if required. In this case, the time constant must be great enough (minimum 100 ms) to allow the oscillator circuit to stabilise. This input has an internal Schmitt trigger to improve noise immunity.

2.4 $\overline{\text{IRQ}}$

$\overline{\text{IRQ}}$ is an input pin for external interrupt sources. The interrupt trigger sensitivity (edge or edge-and-level) can be selected via a mask option. The $\overline{\text{IRQ}}$ pin has an internal Schmitt trigger to improve noise immunity.

2.5 I/O PORT PINS

2.5.1 Ports A–D (Px0–Px7)

Ports A to D consist of eight bi-directional pins (Px0 to Px7). The direction and state of each pin is software programmable. All pins are configured as inputs during power-on or reset.

2.5.2 Port E (PE0–PE7)

Port E consists of eight bi-directional pins (PE0 to PE7). The direction and state of each pin is software programmable. In addition, Port E can be configured to support the SCIs, the 16-bit timer and the bus clock for the M68000 module. See Section 2.6 and following. All pins are configured as inputs during power-on or reset.

2.5.3 Port F (PF0, PF1)

Port F consists of two bi-directional pins (PF0 and PF1). The direction and state of each pin is software programmable. Both pins are configured as inputs during power-on or reset.

2.6 M68000 INTERFACE MODULE PINS

Throughout this document, the M68000 device is referred to as the Host and the MC68HC05i8 as the MCU.

2.6.1 $\overline{\text{DTACK}}$

The tri-state active-low open-drain Data Transfer Acknowledge ($\overline{\text{DTACK}}$) output signal is asserted during read and write cycles to indicate the proper transfer of data between the Host and the MCU.

2.6.2 $\overline{\text{IRQ68K}}$

This active-low open-drain output pin signals to the Host that one or more of the maskable interrupt conditions is true.

2.6.3 RS1–RS4

The register select lines select the MCU internal registers during read/write operations.

2.6.4 RW68K

The read/write input pin, RW68K is driven high by the Host when it reads and low when it writes. Read or write cycles are initiated by asserting the chip select input $\overline{CS}$.

2.6.5 $\overline{CS}$

The active low $\overline{CS}$ input enables data transfers between the Host and MCU on data lines D0 to D7. Data transfers are controlled by read/write (RW68K), chip select $\overline{CS}$ and the register select inputs RS1 to RS4. When $\overline{CS}$ is high the data lines D0 to D7 are placed in a high impedance state.

2.6.6 CLK68K

When configured to support the M68000 interface module, Port E bit 7 serves as the CLK68K input. This pin selects the external clock mode of the M68000 subsystem whereby the $\overline{DTACK}$ pin is synchronised to an external clock signal applied to this pin.

2.6.7 Data bus lines (D0–D7)

These bi-directional tri-state data lines are used to transfer commands, data and status information between the M68000 processor and MCU. D0 is the least significant bit.

2.7 16-BIT PROGRAMMABLE TIMER PINS

2.7.1 TCMP

When configured to support the 16-bit programmable timer, Port E bit 4 becomes the output pin for the output compare function.

2.7.2 TCAPA, TCAPB

When configured to support the 16-bit timer, Port E bits 5 and 6 serve as the input pins for the timer input capture functions.

2.8 SCI PINS

2.8.1 RX1, RX2

When Port E bits 0 and 2 are configured to support the SCI functions by setting bit 2 of the relevant Serial Communications Control register, they become inputs and serve as the receive data pins (RX1 and RX2) for SCI 1 and SCI 2 respectively.

2.8.2 TX1, TX2

When Port E bits 1 and 3 are configured to support the SCI functions by setting bit 3 of the relevant Serial Communications Control register, they become outputs and serve as the transmit data pins (TX1 and TX2) for SCI 1 and SCI 2 respectively.

Further details of how the RX and TX bits operate can be found in Section 9.

SECTION 3
CPU CORE AND INSTRUCTION SET

This section provides a description of the CPU core registers, the instruction set and the addressing modes of the MC68HC05i8.

3.1 REGISTERS

The MCU contains five registers, as shown in the programming model of Figure 3-1. The interrupt stacking order is shown in Figure 3-2.

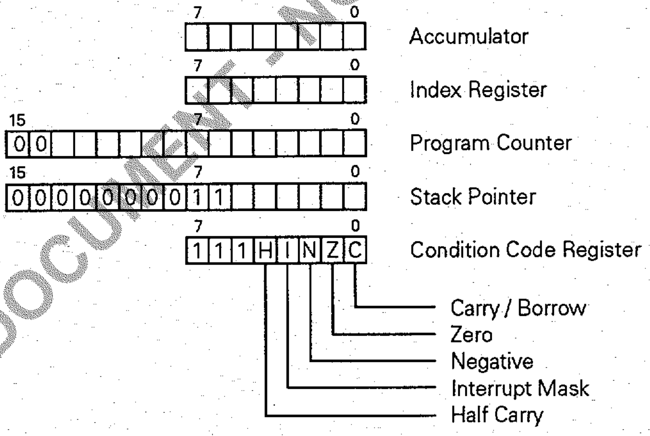


Figure 3-1 Programming Model

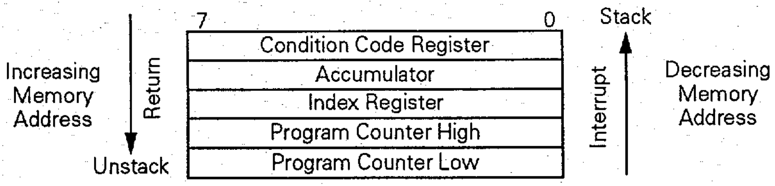


Figure 3-2 Stacking Order

3.1.1 Accumulator (A)

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

3.1.2 Index Register (X)

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

3.1.3 Program Counter (PC)

The program counter is a 16-bit register, which contains the address of the next byte to be fetched. Although the M68HC05 CPU core can address 64 kbytes of memory the actual address range of the MC68HC05i8 is limited to 16 kbytes. The two most significant bits of the Program Counter are therefore not used and are permanently set to zero.

3.1.4 Stack Pointer (SP)

The stack pointer is a 16-bit register, which 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 ten most significant bits are permanently set to 0000000011. These ten bits are appended to the six least significant register bits to produce an address within the range of $00C0 to $00FF. Subroutines and interrupts may use up to 64 (decimal) locations. If 64 locations are exceeded, the stack pointer wraps around and overwrites the previously stored information. A subroutine call occupies two locations on the stack; an interrupt uses five locations.

3.1.5 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.

3.1.5.1 Half carry (H)

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

3.1.5.2 Interrupt (I)

When this bit is set, all maskable interrupts are masked. If an interrupt occurs while this bit is set, the interrupt is latched and remains pending until the interrupt bit is cleared.

3.1.5.3 Negative (N)

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

3.1.5.4 Zero (Z)

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

3.1.5.5 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.

3.2 INSTRUCTION SET

The MCU has a set of 62 basic instructions. They can be grouped into five different types as follows:

- Register/memory
- Read-modify-write
- Branch
- Bit manipulation
- Control

The following paragraphs briefly explain each type. All the instructions within a given type are presented in individual tables.

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 in Table 3-1.

3.2.1 Register/Memory Instructions

Most of these instructions use two operands. The first operand is either the accumulator or the index register. The second 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 Table 3-2 for a complete list of register/memory instructions.

3.2.2 Branch Instructions

These instructions cause the program to branch if a particular condition is met; otherwise, no operation is performed. Branch instructions are two-byte instructions. Refer to Table 3-3.

3.2.3 Bit Manipulation Instructions

The MCU can set or clear any writable bit that resides in the first 256 bytes of the memory space (Page 0). All port data and data direction registers, timer and serial interface registers and, control/status registers and a portion of the on-chip RAM reside in Page 0. An additional feature allows the software to test and branch on the state of any bit within these locations. The bit set, bit clear, bit test, and branch functions are all implemented with single instructions. For the 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. Refer to Table 3-4.

3.2.4 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. Refer to Table 3-5 for a complete list of read/modify/write instructions.

3.2.5 Control Instructions

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to Table 3-6 for a complete list of control instructions.

3.2.6 Tables

Tables for all the instruction types listed above follow. In addition there is a complete alphabetical listing of all the instructions (see Table 3-7(a) and Table 3-7(b)), and an opcode map for the instruction set of the M68HC05 MCU family in Table 3-8.

Table 3-1 MUL Instruction

Operation	**X:A ← X*A**			
Description	Multiplies the eight bits in the index register by the eight bits in the accumulator and places the 16-bit result 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	Addressing Mode	Cycles	Bytes	Opcode
	Inherent	11	1	$42

Table 3-2 Register/Memory Instructions

		Addressing Modes																	
		Immediate			**Direct**			**Extended**			**Indexed (No Offset)**			**Indexed (8-bit Offset)**			**Indexed (16-bit Offset)**		
Function	**Mnemonic**	**Opcode**	**# Bytes**	**# Cycles**	**Opcode**	**# Bytes**	**# Cycles**	**Opcode**	**# Bytes**	**# Cycles**	**Opcode**	**# Bytes**	**# Cycles**	**Opcode**	**# Bytes**	**# Cycles**	**Opcode**	**# Bytes**	**# Cycles**
Load A from Memory	LDA	A6	2	2	B6	2	3	C6	3	4	F6	1	3	E6	2	4	D6	3	5
Load X from Memory	LDX	AE	2	2	BE	2	3	CE	3	4	FE	1	3	EE	2	4	DE	3	5
Store A in Memory	STA	▨	▨	▨	B7	2	4	C7	3	5	F7	1	4	E7	2	5	D7	3	6
Store X in Memory	STX	▨	▨	▨	BF	2	4	CF	3	5	FF	1	4	EF	2	5	DF	3	6
Add Memory to A	ADD	AB	2	2	BB	2	3	CB	3	4	FB	1	3	EB	2	4	DB	3	5
Add Memory and Carry to A	ADC	A9	2	2	B9	2	3	C9	3	4	F9	1	3	E9	2	4	D9	3	5
Subtract Memory	SUB	A0	2	2	B0	2	3	C0	3	4	F0	1	3	E0	2	4	D0	3	5
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	3	C2	3	4	F2	1	3	E2	2	4	D2	3	5
AND Memory with A	AND	A4	2	2	B4	2	3	C4	3	4	F4	1	3	E4	2	4	D4	3	5
OR Memory with A	ORA	AA	2	2	BA	2	3	CA	3	4	FA	1	3	EA	2	4	DA	3	5
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	3	C8	3	4	F8	1	3	E8	2	4	D8	3	5
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	3	C1	3	4	F1	1	3	E1	2	4	D1	3	5
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	3	C3	3	4	F3	1	3	E3	2	4	D3	3	5
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	3	C5	3	4	F5	1	3	E5	2	4	D5	3	5
Jump Unconditional	JMP	▨	▨	▨	BC	2	2	CC	3	3	FC	1	2	EC	2	3	DC	3	4
Jump to Subroutine	JSR	▨	▨	▨	BD	2	5	CD	3	6	FD	1	5	ED	2	6	DD	3	7

Table 3-3 Branch Instructions

		Relative Addressing Mode		
Function	**Mnemonic**	**Opcode**	**# Bytes**	**# Cycles**
Branch Always	BRA	20	2	3
Branch Never	BRN	21	2	3
Branch if Higher	BHI	22	2	3
Branch if Lower or Same	BLS	23	2	3
Branch if Carry Clear	BCC	24	2	3
(Branch if Higher or Same)	(BHS)	24	2	3
Branch if Carry Set	BCS	25	2	3
(Branch if Lower)	(BLO)	25	2	3
Branch if Not Equal	BNE	26	2	3
Branch if Equal	BEQ	27	2	3
Branch if Half Carry Clear	BHCC	28	2	3
Branch if Half Carry Set	BHCS	29	2	3
Branch if Plus	BPL	2A	2	3
Branch if Minus	BMI	2B	2	3
Branch if Interrupt Mask Bit is Clear	BMC	2C	2	3
Branch if Interrupt Mask Bit is Set	BMS	2D	2	3
Branch if Interrupt Line is Low	BIL	2E	2	3
Branch if Interrupt Line is High	BIH	2F	2	3
Branch to Subroutine	BSR	AD	2	6

Table 3-4 Bit Manipulation Instructions

		Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
Function	**Mnemonic**	**Opcode**	**# Bytes**	**# Cycles**	**Opcode**	**# Bytes**	**# Cycles**
Branch if Bit n is Set	BRSET n (n=0–7)	▨	▨	▨	2•n	3	5
Branch if Bit n is Clear	BRCLR n (n=0–7)	▨	▨	▨	01+2•n	3	5
Set Bit n	BSET n (n=0–7)	10+2•n	2	5	▨	▨	▨
Clear Bit n	BCLR n (n=0–7)	11+2•n	2	5	▨	▨	▨

Table 3-5 Read/Modify/Write Instructions

		Addressing Modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (No Offset)			Indexed (8-bit Offset)		
Function	Mnemonic	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles
Increment	INC	4C	1	3	5C	1	3	3C	2	5	7C	1	5	6C	2	6
Decrement	DEC	4A	1	3	5A	1	3	3A	2	5	7A	1	5	6A	2	6
Clear	CLR	4F	1	3	5F	1	3	3F	2	5	7F	1	5	6F	2	6
Complement	COM	43	1	3	53	1	3	33	2	5	73	1	5	63	2	6
Negate (Two's Complement)	NEG	40	1	3	50	1	3	30	2	5	70	1	5	60	2	6
Rotate Left Through Carry	ROL	49	1	3	59	1	3	39	2	5	79	1	5	69	2	6
Rotate Right Through Carry	ROR	46	1	3	56	1	3	36	2	5	76	1	5	66	2	6
Logical Shift Left	LSL	48	1	3	58	1	3	38	2	5	78	1	5	68	2	6
Logical Shift Right	LSR	44	1	3	54	1	3	34	2	5	74	1	5	64	2	6
Arithmetic Shift Right	ASR	47	1	3	57	1	3	37	2	5	77	1	5	67	2	6
Test for Negative or Zero	TST	4D	1	3	5D	1	3	3D	2	4	7D	1	4	6D	2	5
Multiply	MUL	42	1	11												

Table 3-6 Control Instructions

		Inherent Addressing Mode		
Function	Mnemonic	Opcode	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	10
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2
Stop	STOP	8E	1	2
Wait	WAIT	8F	1	2

Table 3-7 (a) Instruction Set (1 of 2)

Mnemonic	Addressing Modes										Condition Codes				
	INH	IMM	DIR	EXT	REL	IX	IX1	IX2	BSC	BTB	H	I	N	Z	C
ADC	▨				▨				▨	▨	◊	•	◊	◊	◊
ADD	▨				▨				▨	▨	◊	•	◊	◊	◊
AND	▨				▨				▨	▨	•	•	◊	◊	•
ASL		▨		▨	▨			▨	▨	▨	•	•	◊	◊	◊
ASR		▨		▨	▨			▨	▨	▨	•	•	◊	◊	◊
BCC	▨	▨	▨	▨		▨	▨	▨	▨	▨	•	•	•	•	•
BCLR	▨	▨	▨	▨	▨	▨	▨	▨		▨	•	•	•	•	•
BCS	▨	▨	▨	▨		▨	▨	▨	▨	▨	•	•	•	•	•
BEQ	▨	▨	▨	▨		▨	▨	▨	▨	▨	•	•	•	•	•
BHCC	▨	▨	▨	▨		▨	▨	▨	▨	▨	•	•	•	•	•
BHCS	▨	▨	▨	▨		▨	▨	▨	▨	▨	•	•	•	•	•
BHI	▨	▨	▨	▨		▨	▨	▨	▨	▨	•	•	•	•	•
BHS	▨	▨	▨	▨		▨	▨	▨	▨	▨	•	•	•	•	•
BIH	▨	▨	▨	▨		▨	▨	▨	▨	▨	•	•	•	•	•
BIL	▨	▨	▨	▨		▨	▨	▨	▨	▨	•	•	•	•	•
BIT	▨				▨				▨	▨	•	•	◊	◊	•
BLO	▨	▨	▨	▨		▨	▨	▨	▨	▨	•	•	•	•	•
BLS	▨	▨	▨	▨		▨	▨	▨	▨	▨	•	•	•	•	•
BMC	▨	▨	▨	▨		▨	▨	▨	▨	▨	•	•	•	•	•
BMI	▨	▨	▨	▨		▨	▨	▨	▨	▨	•	•	•	•	•
BMS	▨	▨	▨	▨		▨	▨	▨	▨	▨	•	•	•	•	•
BNE	▨	▨	▨	▨		▨	▨	▨	▨	▨	•	•	•	•	•
BPL	▨	▨	▨	▨		▨	▨	▨	▨	▨	•	•	•	•	•
BRA	▨	▨	▨	▨		▨	▨	▨	▨	▨	•	•	•	•	•
BRN	▨	▨	▨	▨		▨	▨	▨	▨	▨	•	•	•	•	•
BRCLR	▨	▨	▨	▨	▨	▨	▨	▨	▨		•	•	•	•	◊
BRSET	▨	▨	▨	▨	▨	▨	▨	▨	▨		•	•	•	•	◊
BSET	▨	▨	▨	▨	▨	▨	▨	▨		▨	•	•	•	•	•
BSR	▨	▨	▨	▨		▨	▨	▨	▨	▨	•	•	•	•	•
CLC		▨	▨	▨	▨	▨	▨	▨	▨	▨	•	•	•	•	0
CLI		▨	▨	▨	▨	▨	▨	▨	▨	▨	•	0	•	•	•
CLR		▨		▨	▨			▨	▨	▨	•	•	0	1	•
CMP	▨				▨				▨	▨	•	•	◊	◊	◊

Address Mode Abbreviations

BSC	Bit Set/Clear	IMM	Immediate
BTB	Bit Test & Branch	IX	Indexed (No Offset)
DIR	Direct	IX1	Indexed, 1 byte Offset
EXT	Extended	IX2	Indexed, 2 byte Offset
INH	Inherent	REL	Relative

▨ Not Implemented

Condition Code Symbols

H	Half Carry (from Bit 3)	◊	Tested and Set if True, Cleared otherwise
I	Interrupt Mask	•	Not Affected
N	Negate (Sign Bit)	?	Load CCR from Stack
Z	Zero	0	Cleared
C	Carry/Borrow	1	Set

Table 3-7 (b) Instruction Set (2 of 2)

Mnemonic	Addressing Modes										Condition Codes				
	INH	IMM	DIR	EXT	REL	IX	IX1	IX2	BSC	BTB	H	I	N	Z	C
COM		▨		▨	▨			▨	▨	▨	•	•	◊	◊	1
CPX	▨				▨				▨	▨	•	•	◊	◊	◊
DEC		▨		▨	▨			▨	▨	▨	•	•	◊	◊	•
EOR	▨				▨				▨	▨	•	•	◊	◊	•
INC		▨		▨	▨			▨	▨	▨	•	•	◊	◊	•
JMP	▨	▨			▨				▨	▨	•	•	•	•	•
JSR	▨	▨			▨				▨	▨	•	•	•	•	•
LDA	▨				▨				▨	▨	•	•	◊	◊	•
LDX	▨				▨				▨	▨	•	•	◊	◊	•
LSL		▨		▨	▨			▨	▨	▨	•	•	◊	◊	◊
LSR		▨		▨	▨			▨	▨	▨	•	•	0	◊	◊
MUL		▨	▨	▨	▨	▨	▨	▨	▨	▨	0	•	•	•	0
NEG		▨		▨	▨			▨	▨	▨	•	•	◊	◊	◊
NOP		▨	▨	▨	▨	▨	▨	▨	▨	▨	•	•	•	•	•
ORA	▨				▨				▨	▨	•	•	◊	◊	•
ROL		▨		▨	▨			▨	▨	▨	•	•	◊	◊	◊
ROR		▨		▨	▨			▨	▨	▨	•	•	◊	◊	◊
RSP		▨	▨	▨	▨	▨	▨	▨	▨	▨	•	•	•	•	•
RTI		▨	▨	▨	▨	▨	▨	▨	▨	▨	?	?	?	?	?
RTS		▨	▨	▨	▨	▨	▨	▨	▨	▨	•	•	•	•	•
SBC	▨				▨				▨	▨	•	•	◊	◊	◊
SEC		▨	▨	▨	▨	▨	▨	▨	▨	▨	•	•	•	•	1
SEI		▨	▨	▨	▨	▨	▨	▨	▨	▨	•	1	•	•	•
STA	▨	▨			▨				▨	▨	•	•	◊	◊	•
STOP		▨	▨	▨	▨	▨	▨	▨	▨	▨	•	0	•	•	•
STX	▨	▨			▨				▨	▨	•	•	◊	◊	•
SUB	▨				▨				▨	▨	•	•	◊	◊	◊
SWI		▨	▨	▨	▨	▨	▨	▨	▨	▨	•	1	•	•	•
TAX		▨	▨	▨	▨	▨	▨	▨	▨	▨	•	•	•	•	•
TST		▨		▨	▨			▨	▨	▨	•	•	◊	◊	•
TXA		▨	▨	▨	▨	▨	▨	▨	▨	▨	•	•	•	•	•
WAIT		▨	▨	▨	▨	▨	▨	▨	▨	▨	•	0	•	•	•

Address Mode Abbreviations

BSC	Bit Set/Clear	IMM	Immediate
BTB	Bit Test & Branch	IX	Indexed (No Offset)
DIR	Direct	IX1	Indexed, 1 byte Offset
EXT	Extended	IX2	Indexed, 2 byte Offset
INH	Inherent	REL	Relative

▨ Not Implemented

Condition Code Symbols

H	Half Carry (from Bit 3)	◊	Tested and Set if True, Cleared otherwise
I	Interrupt Mask	•	Not Affected
N	Negate (Sign Bit)	?	Load CCR from Stack
Z	Zero	0	Cleared
C	Carry/Borrow	1	Set

Table 3-8 Opcode Map

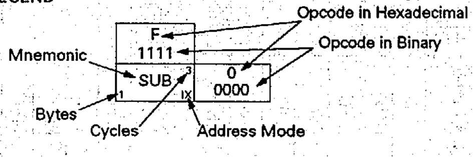
	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
High Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	High Low
0 0000	BRSET0 5 3 BTB	BSET0 5 2 BSC	BRA 3 2 REL	NEG 5 2 DIR	NEGA 3 1 INH	NEGX 3 1 INH	NEG 6 2 IX1	NEG 5 1 IX	RTI 9 1 INH	▨	SUB 2 2 IMM	SUB 3 2 DIR	SUB 4 3 EXT	SUB 5 3 IX2	SUB 4 2 IX1	SUB 3 1 IX	0 0000
1 0001	BRCLR0 5 3 BTB	BCLR0 5 2 BSC	BRN 3 2 REL	▨	▨	▨	▨	▨	RTS 6 1 INH	▨	CMP 2 2 IMM	CMP 3 2 DIR	CMP 4 3 EXT	CMP 5 3 IX2	CMP 4 2 IX1	CMP 3 1 IX	1 0001
2 0010	BRSET1 5 3 BTB	BSET1 5 2 BSC	BHI 3 2 REL	▨	MUL 11 1 INH	▨	▨	▨	▨	▨	SBC 2 2 IMM	SBC 3 2 DIR	SBC 4 3 EXT	SBC 5 3 IX2	SBC 4 2 IX1	SBC 3 1 IX	2 0010
3 0011	BRCLR1 5 3 BTB	BCLR1 5 2 BSC	BLS 3 2 REL	COM 5 2 DIR	COMA 3 1 INH	COMX 3 1 INH	COM 6 2 IX1	COM 5 1 IX	SWI 10 1 INH	▨	CPX 2 2 IMM	CPX 3 2 DIR	CPX 4 3 EXT	CPX 5 3 IX2	CPX 4 2 IX1	CPX 3 1 IX	3 0011
4 0100	BRSET2 5 3 BTB	BSET2 5 2 BSC	BCC 3 2 REL	LSR 5 2 DIR	LSRA 3 1 INH	LSRX 3 1 INH	LSR 6 2 IX1	LSR 5 1 IX	▨	▨	AND 2 2 IMM	AND 3 2 DIR	AND 4 3 EXT	AND 5 3 IX2	AND 4 2 IX1	AND 3 1 IX	4 0100
5 0101	BRCLR2 5 3 BTB	BCLR2 5 2 BSC	BCS 3 2 REL	▨	▨	▨	▨	▨	▨	▨	BIT 2 2 IMM	BIT 3 2 DIR	BIT 4 3 EXT	BIT 5 3 IX2	BIT 4 2 IX1	BIT 3 1 IX	5 0101
6 0110	BRSET3 5 3 BTB	BSET3 5 2 BSC	BNE 3 2 REL	ROR 5 2 DIR	RORA 3 1 INH	RORX 3 1 INH	ROR 6 2 IX1	ROR 5 1 IX	▨	▨	LDA 2 2 IMM	LDA 3 2 DIR	LDA 4 3 EXT	LDA 5 3 IX2	LDA 4 2 IX1	LDA 3 1 IX	6 0110
7 0111	BRCLR3 5 3 BTB	BCLR3 5 2 BSC	BEQ 3 2 REL	ASR 5 2 DIR	ASRA 3 1 INH	ASRX 3 1 INH	ASR 6 2 IX1	ASR 5 1 IX	▨	TAX 2 1 INH	▨	STA 4 2 DIR	STA 5 3 EXT	STA 6 3 IX2	STA 5 2 IX1	STA 4 1 IX	7 0111
8 1000	BRSET4 5 3 BTB	BSET4 5 2 BSC	BHCC 3 2 REL	LSL 5 2 DIR	LSLA 3 1 INH	LSLX 3 1 INH	LSL 6 2 IX1	LSL 5 1 IX	▨	CLC 2 1 INH	EOR 2 2 IMM	EOR 3 2 DIR	EOR 4 3 EXT	EOR 5 3 IX2	EOR 4 2 IX1	EOR 3 1 IX	8 1000
9 1001	BRCLR4 5 3 BTB	BCLR4 5 2 BSC	BHCS 3 2 REL	ROL 5 2 DIR	ROLA 3 1 INH	ROLX 3 1 INH	ROL 6 2 IX1	ROL 5 1 IX	▨	SEC 2 1 INH	ADC 2 2 IMM	ADC 3 2 DIR	ADC 4 3 EXT	ADC 5 3 IX2	ADC 4 2 IX1	ADC 3 1 IX	9 1001
A 1010	BRSET5 5 3 BTB	BSET5 5 2 BSC	BPL 3 2 REL	DEC 5 2 DIR	DECA 3 1 INH	DECX 3 1 INH	DEC 6 2 IX1	DEC 5 1 IX	▨	CLI 2 1 INH	ORA 2 2 IMM	ORA 3 2 DIR	ORA 4 3 EXT	ORA 5 3 IX2	ORA 4 2 IX1	ORA 3 1 IX	A 1010
B 1011	BRCLR5 5 3 BTB	BCLR5 5 2 BSC	BMI 3 2 REL	▨	▨	▨	▨	▨	▨	SEI 2 1 INH	ADD 2 2 IMM	ADD 3 2 DIR	ADD 4 3 EXT	ADD 5 3 IX2	ADD 4 2 IX1	ADD 3 1 IX	B 1011
C 1100	BRSET6 5 3 BTB	BSET6 5 2 BSC	BMC 3 2 REL	INC 5 2 DIR	INCA 3 1 INH	INCX 3 1 INH	INC 6 2 IX1	INC 5 1 IX	▨	RSP 2 1 INH	▨	JMP 2 2 DIR	JMP 3 3 EXT	JMP 4 3 IX2	JMP 3 2 IX1	JMP 2 1 IX	C 1100
D 1101	BRCLR6 5 3 BTB	BCLR6 5 2 BSC	BMS 3 2 REL	TST 4 2 DIR	TSTA 3 1 INH	TSTX 3 1 INH	TST 5 2 IX1	TST 4 1 IX	▨	NOP 2 1 INH	BSR 6 2 REL	JSR 5 2 DIR	JSR 6 3 EXT	JSR 7 3 IX2	JSR 6 2 IX1	JSR 5 1 IX	D 1101
E 1110	BRSET7 5 3 BTB	BSET7 5 2 BSC	BIL 3 2 REL	▨	▨	▨	▨	▨	STOP 2 1 INH	▨	LDX 2 2 IMM	LDX 3 2 DIR	LDX 4 3 EXT	LDX 5 3 IX2	LDX 4 2 IX1	LDX 3 1 IX	E 1110
F 1111	BRCLR7 5 3 BTB	BCLR7 5 2 BSC	BIH 3 2 REL	CLR 5 2 DIR	CLRA 3 1 INH	CLRX 3 1 INH	CLR 6 2 IX1	CLR 5 1 IX	WAIT 2 1 INH	TXA 2 1 INH	▨	STX 4 2 DIR	STX 5 3 EXT	STX 6 3 IX2	STX 5 2 IX1	STX 4 1 IX	F 1111

Abbreviations for Address Modes and Registers

BSC	Bit Set/Clear	IX	Indexed (No Offset)
BTB	Bit Test and Branch	IX1	Indexed, 1 byte (8 bit) Offset
DIR	Direct	IX2	Indexed, 2 byte (16 bit) Offset
EXT	Extended	REL	Relative
INH	Inherent	A	Accumulator
IMM	Immediate	X	Index Register

▨ Not Implemented

LEGEND



3.3 ADDRESSING MODES

Ten different addressing modes provide the programmer with the flexibility to optimise his 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 (three bytes) permit accessing tables throughout memory. Short absolute (direct) and long absolute (extended) addressing is also included. One or two byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory locations.

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. The ten addressing modes of the processor are described below. Parentheses are used to indicate "contents of" the location or register referred to; e.g., (PC) indicates the contents of the location pointed to by the PC (Program Counter). An arrow indicates "is replaced by", and a colon indicates concatenation of two bytes. For additional details and graphical illustrations, refer to the ***M6805 HMOS/M146805 CMOS Family Microcomputer/Microprocessor User's Manual.***

3.3.1 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 one byte long.

3.3.2 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 (e.g., a constant used to initialise a loop counter).

EA = PC+1; PC ← PC+2

3.3.3 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 two-byte instruction.

EA = (PC+1); PC ← PC+2
Address Bus High ← 0; Address Bus Low ← (PC+1)

3.3.4 Extended

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode byte. Instructions with extended addressing mode are capable of referencing arguments anywhere in memory with a single three-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 short form of the instruction.

$$EA = (PC+1):(PC+2);\ PC \leftarrow PC+3$$
$$\text{Address Bus High} \leftarrow (PC+1);\ \text{Address Bus Low} \leftarrow (PC+2)$$

3.3.5 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 one 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.

$$EA = X;\ PC \leftarrow PC+3$$
$$\text{Address Bus High} \leftarrow 0;\ \text{Address Bus Low} \leftarrow X$$

3.3.6 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. Therefore the operand can be located anywhere within the lowest 511 memory locations. This addressing mode is useful for selecting the mth element in an n element table.

$$EA = X+(PC+1);\ PC \leftarrow PC+2$$
$$\text{Address Bus High} \leftarrow K;\ \text{Address Bus Low} \leftarrow X+(PC+1)$$
$$\text{where } K = \text{the carry from the addition of } X \text{ and } (PC+1)$$

3.3.7 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 three-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.

$$EA = X+[(PC+1):(PC+2)];\ PC \leftarrow PC+3$$
$$\text{Address Bus High} \leftarrow (PC+1)+K;\ \text{Address Bus Low} \leftarrow X+(PC+2)$$
$$\text{where } K = \text{the carry from the addition of } X \text{ and } (PC+2)$$

3.3.8 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.

$$EA = PC+2+(PC+1);\ PC \leftarrow EA \text{ if branch taken;}$$
$$\text{otherwise } EA = PC \leftarrow PC+2$$

3.3.9 Bit Set/Clear

In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode. The byte following the opcode specifies the address 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 two-byte instruction.

$$EA = (PC+1);\ PC \leftarrow PC+2$$
$$\text{Address Bus High} \leftarrow 0;\ \text{Address Bus Low} \leftarrow (PC+1)$$

3.3.10 Bit Test and Branch

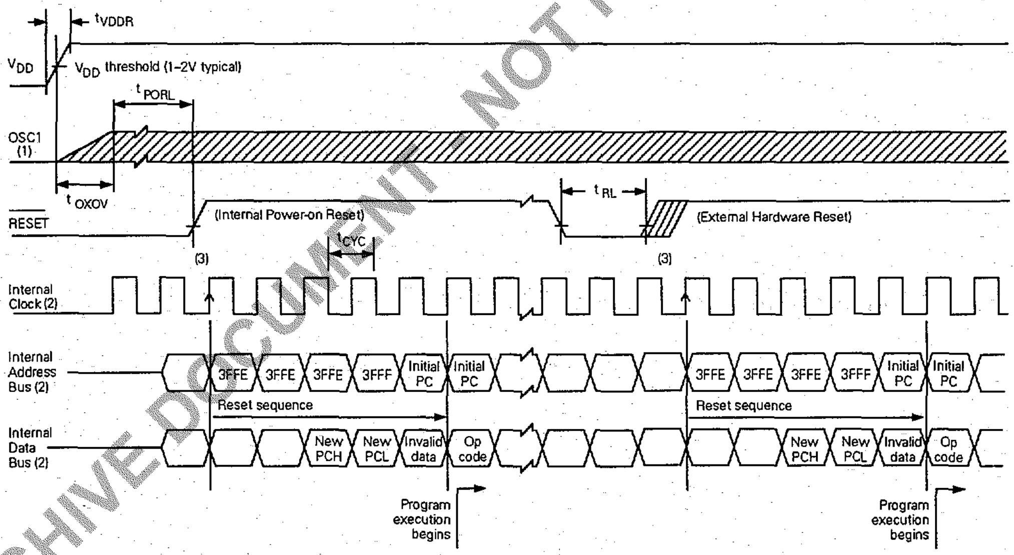
The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit to be tested and its condition (set or clear) is included in the opcode. The address of the byte to be tested is in the single byte immediately following the opcode byte (EA1). The signed relative 8-bit offset in the third byte (EA2) is added to the PC if the specified bit is set or cleared in the specified memory location. This single three-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 branch 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.

$$EA1 = (PC+1);\ PC \leftarrow PC+2$$
$$\text{Address Bus High} \leftarrow 0;\ \text{Address Bus Low} \leftarrow (PC+1)$$
$$EA2 = PC+3+(PC+2);\ PC \leftarrow EA2 \text{ if branch taken;}$$
$$\text{otherwise } PC \leftarrow PC+3$$

SECTION 4
RESETS, INTERRUPTS AND LOW POWER MODES

4.1 RESETS

The MCU can be reset in four ways: by the initial power-on reset function, by an active low input to the $\overline{\text{RESET}}$ pin, by an opcode fetch from an illegal address, and by a COP watchdog timer reset. See Figure 4-1.



Notes:
1. The OSC1 line is not meant to represent frequency; it is used to represent time only.
2. Internal clock and bus signals are not available externally.
3. The first rising edge of the internal clock after the rising edge of $\overline{\text{RESET}}$ initiates the reset sequence.
4. Refer to Electrical Specifications section for specific values of timing parameters.

Figure 4-1 Power-on Reset and $\overline{\text{RESET}}$

4.1.1 Power-on Reset

A power-on reset occurs when a positive transition is detected on VDD. The power-on reset function is strictly for power turn-on conditions and should not be used to detect drops in the power supply voltage. The power-on circuitry provides a stabilisation delay (t_{PORL}) from when the oscillator becomes active. If the external $\overline{\text{RESET}}$ pin is low at the end of this delay then the processor remains in the reset state until $\overline{\text{RESET}}$ goes high. The user must ensure that the voltage on VDD has risen to a point where the MCU can operate properly by the time t_{PORL} has elapsed. If there is doubt, the external $\overline{\text{RESET}}$ pin should remain low until the voltage on VDD has reached the specified minimum operating voltage. This may be accomplished by connecting an external RC-circuit to this pin to generate a power-on reset (POR). In this case, the time constant must be great enough (at least 100 ms) to allow the oscillator circuit to stabilise.

4.1.2 $\overline{\text{RESET}}$ Pin

When the oscillator is running in a stable state, the MCU is reset when a logic zero is applied to the $\overline{\text{RESET}}$ input for a minimum period of 1.5 machine cycles (t_{CYC}). This pin contains an internal Schmitt Trigger as part of its input to improve noise immunity.

4.1.3 Illegal Address Reset

When an opcode fetch occurs from an address which is not part of the RAM ($0050–$012F) or of the ROM ($2000–$3FFF) then the device is automatically reset.

4.1.4 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 Section 8.2 for more information on the COP Watchdog timer.

4.2 INTERRUPTS

The MCU can be interrupted by seven different sources, six maskable hardware interrupts and one non-maskable software interrupt:

- External signal on the $\overline{\text{IRQ}}$ pin
- 15-stage ripple counter (Core Timer)
- 16-bit Programmable Timer
- SCI1
- SCI2
- $\overline{\text{IRQ68K}}$
- Software Interrupt instruction (SWI)

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

Unlike reset, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction is complete. 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 the corresponding interrupt enable bit is set, the processor proceeds with interrupt processing; otherwise, the next instruction is fetched and executed.

If both an external interrupt and a timer interrupt are pending after an instruction execution, the external interrupt is serviced first.

Table 4-1 shows the relative priority of all the possible interrupt sources. Figure 4-2 shows the Interrupt Processing flow.

Table 4-1 Interrupt Priorities and Vector Addresses

Source	Register	Flags	Vector Address	Priority
Reset	—	—	\$3FFE, \$3FFF	highest
Software Interrupt (SWI)	—	—	\$3FFC, \$3FFD	
External Interrupt ($\overline{\text{IRQ}}$)	—	—	\$3FFA, \$3FFB	↑
Core Timer (CTIMER)	CTCSR	CTOF, RTIF	\$3FF8, \$3FF9	
M68000 Interface (M68K)	IMR05	RMx, TMx	\$3FF6, \$3FF7	
16-bit Timer	TSR	ICAF, ICBF, OCAF, TOF	\$3FF4, \$3FF5	
SCI1	SCSR	TDRE, TC, RDRF, IDLE, OR, NF, FE	\$3FF2, \$3FF3	
SCI2	SCSR		\$3FF0,\$3FF1	lowest

4.2.1 Non-Maskable Software Interrupt (SWI)

The software interrupt (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 is executed after interrupts that were pending when the SWI was fetched, but before interrupts generated after the SWI was fetched. The SWI interrupt service routine address is specified by the contents of memory locations $3FFC and $3FFD.

4.2.2 Maskable Hardware Interrupts

If the interrupt mask bit (I-bit) of the CCR is set, all maskable interrupts (internal and external) are masked. Clearing the I-bit allows interrupt processing to occur.

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.

4.2.2.1 External Interrupt ($\overline{\text{IRQ}}$)

The interrupt request is latched immediately following the selected edge on the $\overline{\text{IRQ}}$ pin. It is then synchronized internally and serviced by the routine whose start address is contained in memory locations $3FFA and $3FFB. Edge sensitive only or edge-and-level sensitive triggering is selected via a mask option.

4.2.2.2 Real Time and Core Timer (CTIMER) Interrupts

There are two different core timer interrupt flags that cause a CTIMER interrupt whenever an interrupt is enabled and its flag becomes set, namely RTIF and CTOF. The interrupt flags and enable bits are located in the CTIMER Control and Status Register (CTCSR). These interrupts will vector to the same interrupt service routine, whose start address is contained in memory locations $3FF8 and $3FF9 (see 8.3.1 Core Timer Control and Status Register, and Figure 8-1 Core Timer Block Diagram).

To make use of the Real Time interrupt the RTIE bit must first be set. The RTIF bit will then be set after the specified number of counts.

To make use of the Core Timer Overflow Interrupt the CTOFE bit must first be set. The CTOF bit will then be set when the Core Timer Counter register overflows from $FF to $00.

4.2.2.3 M68000 Interrupt

There are eight sources that can cause an M68000 interface module interrupt to be latched. Once latched, the interrupt is synchronized internally and serviced by the routine whose start address is contained in memory locations $3FF6 and $3FF7. The M68000 interface module interrupt mask bits are held in the Interrupt Mask register at location $3D. See Section 10 for more details.

4.2.2.4 Programmable 16-bit Timer Interrupt

There are four sources of interrupt on the 16-bit timer; two Input Captures, one Output Compare and a Timer Overflow. The interrupt enable bits for these functions are held in the Timer Control register ($1C) and the interrupt flags in the Timer Status register ($1E). These interrupts will be serviced by the routine whose start address is contained in memory locations $3FF4 and $3FF5.

4.2.2.5 SCI1 Interrupt

Serial Communications Interface 1 can generate four types of interrupt; Transmit Interrupt (TI), Transmit Complete Interrupt (TCI), Receiver Interrupt (RI) and Idle Line Interrupt (ILI). The interrupt enable bits for each of these sources are held in the SCI1 Control register ($25) and the status bits in the SCI1 Status register ($26). All the interrupts are serviced by the routine whose start address is contained in memory locations $3FF2 and $FFF3.

4.2.2.6 SCI2 Interrupt

Serial Communications Interface 2 can generate four types of interrupt; Transmit Interrupt (TI), Transmit Complete Interrupt (TCI), Receiver Interrupt (RI) and Idle Line Interrupt (ILI). The interrupt enable bits for each of these sources are held in the SCI2 Control register ($2D) and the status bits in the SCI2 Status register ($2E). All the interrupts are serviced by the routine whose start address is contained in memory locations $3FF0 and $3FF1.

4.2.3 Hardware Controlled Interrupt Sequence

The following three functions (RESET, STOP, and WAIT) are not in the strictest sense interrupts. However, they are acted upon in a similar manner. Flowcharts for STOP and WAIT are shown in Figure 4-3.

RESET: A reset condition causes the program to vector to its starting address, which is contained in memory locations $3FFE (MSB) and $3FFF (LSB). The I-bit in the condition code register is also set, to disable interrupts.

STOP: The STOP instruction causes the oscillator to be turned off and the processor to "sleep" until an external interrupt ($\overline{IRQ}$) occurs or the device is reset.

WAIT: The WAIT instruction causes all processor clocks to stop, but leaves the core timer running. This "rest" state of the processor can be cleared by reset, an external interrupt ($\overline{IRQ}$), or a core timer interrupt. There are no special WAIT vectors for these interrupts. Note that the 16-bit timer, SCI1, SCI2 and the MC68000 module are not active during WAIT mode.

4.3 LOW POWER MODES

4.3.1 STOP

The STOP instruction places the MCU in its lowest power consumption mode. The processor can only be started again by an external interrupt on the $\overline{\text{IRQ}}$ pin or a reset. The oscillator is stopped, all CPU and timer functions are stopped, and an oscillator stabilization delay of t_{PORL} is required to start the processor again.

During the STOP mode, the core timer interrupt flags and interrupt enable bits and the 16-bit Timer interrupt and enable bits are cleared by internal hardware to remove any pending interrupt requests. The timer prescaler is also cleared. The I-bit in the CCR is cleared to enable external interrupts. All other bits, registers and memory remain unaltered. All input/output lines remain unchanged. The STOP instruction can be disabled by a mask option; when disabled, STOP is executed as a NOP (no operation).

Note: Pending interrupts from the M68000 interface, SCI1 and SCI2 are not cleared by the STOP instruction.

4.3.2 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 action is suspended, but the Core Timer remains active. An interrupt from the Core Timer can cause the MCU to exit the WAIT mode.

During the WAIT mode, the I-bit in the CCR is cleared to allow interrupt processing. All other registers, memory and input/output lines remain in their previous state.

4.3.3 Data Retention Mode

The contents of the RAM are retained at supply voltages as low as 2.0 Vdc. This is called the Data Retention Mode, in which data is maintained but the device is not guaranteed to operate.

For lowest power consumption in data retention mode the device should be put into STOP mode before reducing the supply voltage, to ensure that all the clocks are stopped. If the device is not in STOP mode then it is recommended that $\overline{\text{RESET}}$ be held low whilst the power supply is outwith the normal operating range, to ensure that processing is suspended in an orderly manner.

Recovery from Data Retention Mode, after the power supply has been restored, is by an external interrupt, or by pulling the $\overline{\text{RESET}}$ line high.

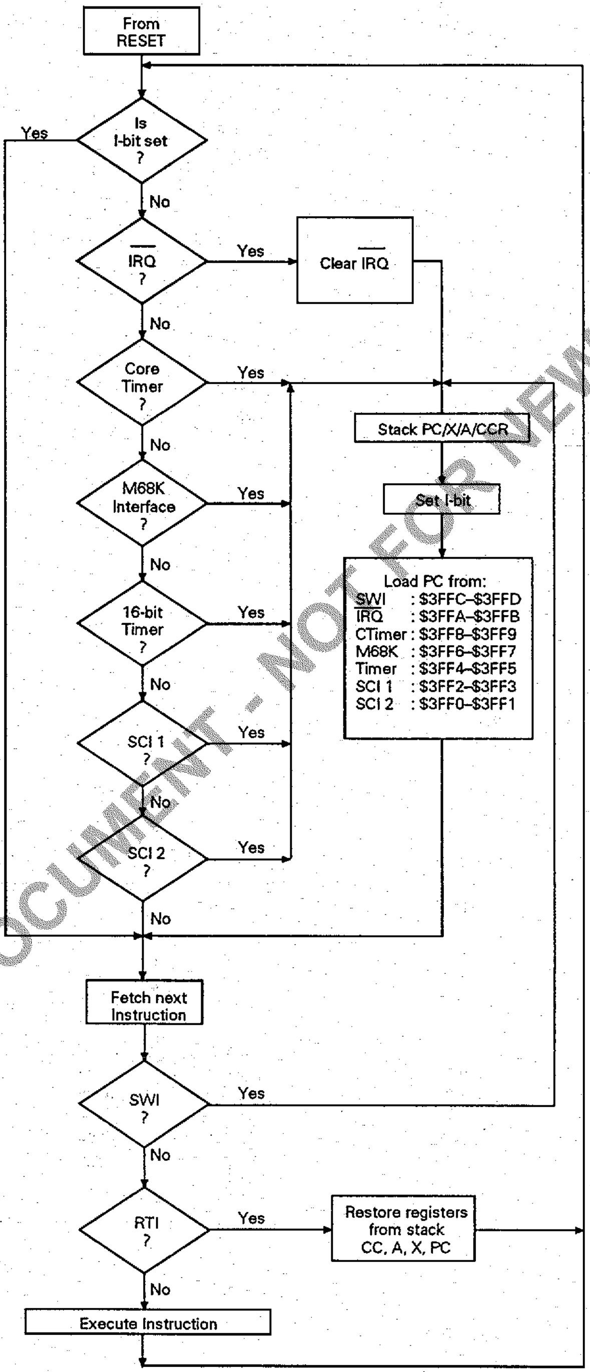


Figure 4-2 Hardware Interrupt Flow Chart

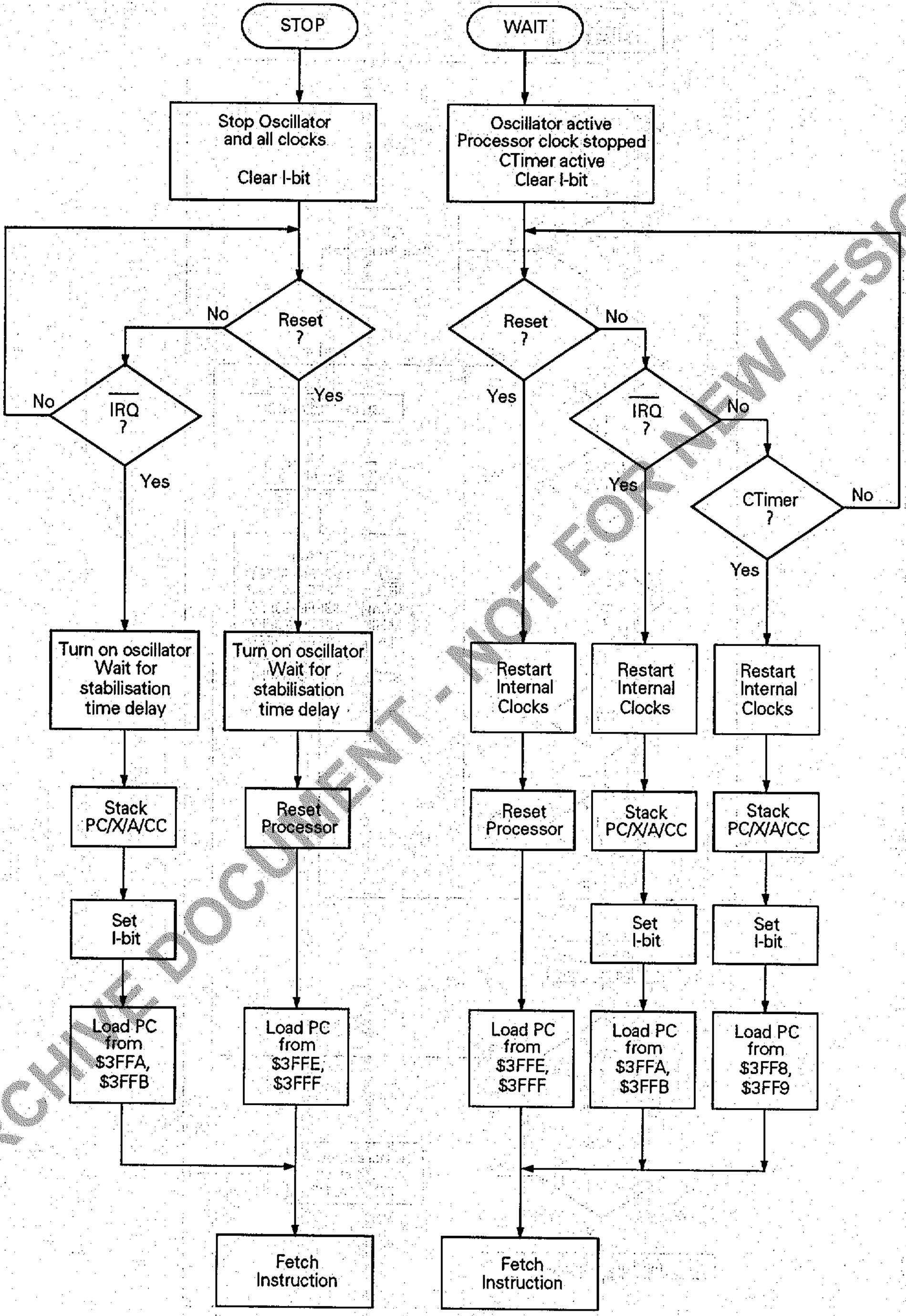


Figure 4-3 STOP/WAIT Flow Chart

SECTION 5
MEMORY AND REGISTERS

5.1 MEMORY MAP

The MC68HC05i8 has a 16 kbyte memory map comprising 7936 bytes of user ROM, 224 bytes of RAM, 240 bytes of self-check ROM, 16 bytes of user vectors and 64 bytes containing the I/O registers.

5.2 RAM

224 bytes of on-chip static RAM are located from $0050 to $012F. The processor stack starts at $00FF and is limited to 64 bytes ($00C0 to $00FF). When the stack overflows it wraps round from $00C0 to $00FF, overwriting any existing data.

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

5.3 ROM

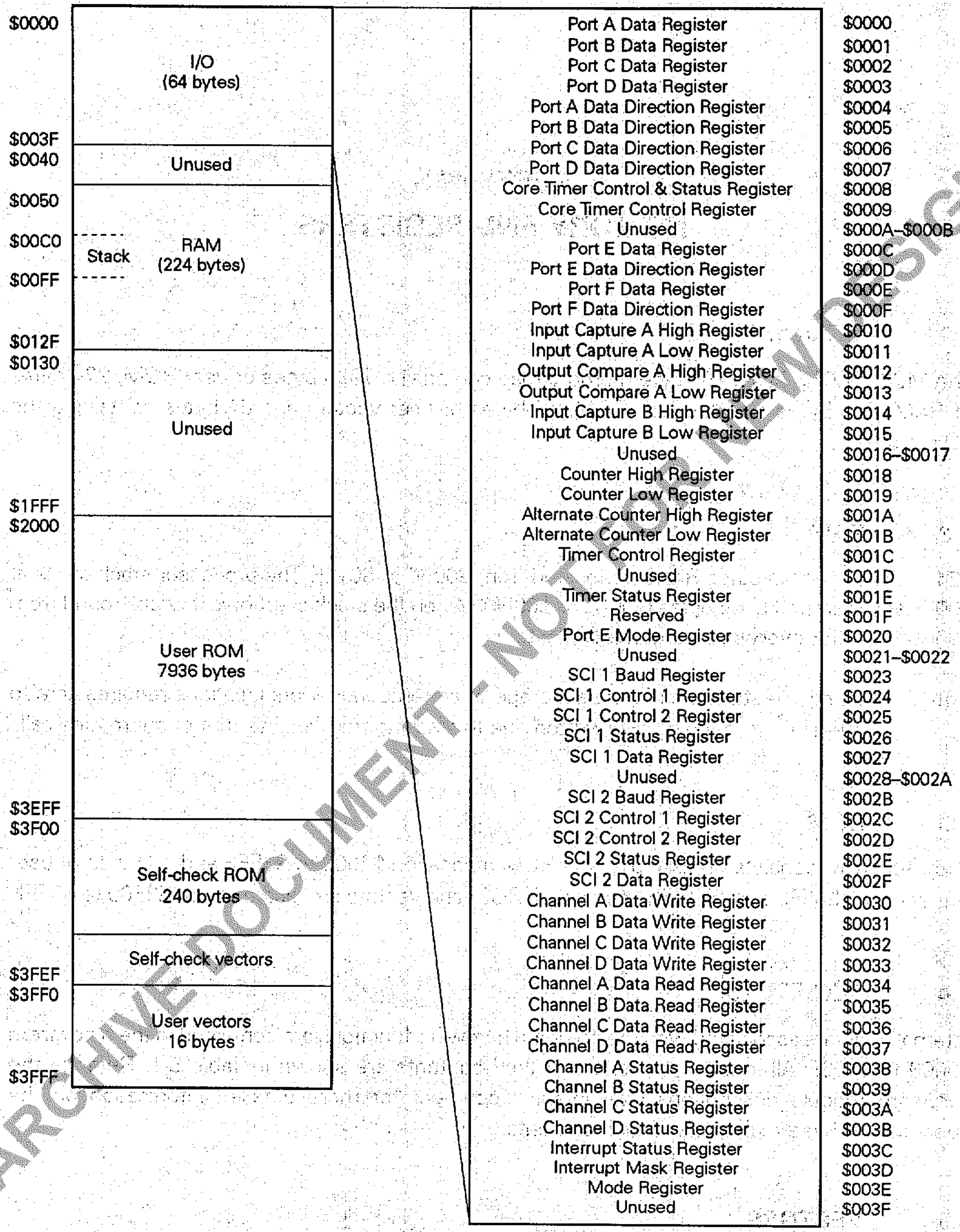
The user ROM consists of 7936 bytes of ROM at location $2000 to $3EFF and 16 bytes of user vectors from $3FF0 to $3FFF. The self-check ROM and vectors are located from $3F00 to $3FEF.

5.4 REGISTERS

Internal registers associated with the on-board hardware functions are contained in the block from $0000 to $003F. All internal registers and their contents are shown in Table 5-1. Refer to the hardware module descriptions later in this document for more detailed information on the operation of these registers and their contents.

5.5 VECTORS

All vectors for reset, hardware interrupts and software interrupt are located at the top of the memory map, from $3FF0 to $3FFF. Each vector location consists of two bytes containing the start address of the reset or interrupt routine (See Table 4-1).



Port A Data Register	$0000
Port B Data Register	$0001
Port C Data Register	$0002
Port D Data Register	$0003
Port A Data Direction Register	$0004
Port B Data Direction Register	$0005
Port C Data Direction Register	$0006
Port D Data Direction Register	$0007
Core Timer Control & Status Register	$0008
Core Timer Control Register	$0009
Unused	$000A–$000B
Port E Data Register	$000C
Port E Data Direction Register	$000D
Port F Data Register	$000E
Port F Data Direction Register	$000F
Input Capture A High Register	$0010
Input Capture A Low Register	$0011
Output Compare A High Register	$0012
Output Compare A Low Register	$0013
Input Capture B High Register	$0014
Input Capture B Low Register	$0015
Unused	$0016–$0017
Counter High Register	$0018
Counter Low Register	$0019
Alternate Counter High Register	$001A
Alternate Counter Low Register	$001B
Timer Control Register	$001C
Unused	$001D
Timer Status Register	$001E
Reserved	$001F
Port E Mode Register	$0020
Unused	$0021–$0022
SCI 1 Baud Register	$0023
SCI 1 Control 1 Register	$0024
SCI 1 Control 2 Register	$0025
SCI 1 Status Register	$0026
SCI 1 Data Register	$0027
Unused	$0028–$002A
SCI 2 Baud Register	$002B
SCI 2 Control 1 Register	$002C
SCI 2 Control 2 Register	$002D
SCI 2 Status Register	$002E
SCI 2 Data Register	$002F
Channel A Data Write Register	$0030
Channel B Data Write Register	$0031
Channel C Data Write Register	$0032
Channel D Data Write Register	$0033
Channel A Data Read Register	$0034
Channel B Data Read Register	$0035
Channel C Data Read Register	$0036
Channel D Data Read Register	$0037
Channel A Status Register	$0038
Channel B Status Register	$0039
Channel C Status Register	$003A
Channel D Status Register	$003B
Interrupt Status Register	$003C
Interrupt Mask Register	$003D
Mode Register	$003E
Unused	$003F

Figure 5-1 Memory Map of the MC68HC05i8

Table 5-1(a) Register Outline

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name	Reset Status
$0000									Port A Data	not affected
$0001									Port B Data	
$0002									Port C Data	
$0003									Port D Data	
$0004									Port A Data Direction (Port A DDR)	0000 0000
$0005									Port B Data Direction (Port B DDR)	0000 0000
$0006									Port C Data Direction (Port C DDR)	0000 0000
$0007									Port D Data Direction (Port D DDR)	0000 0000
$0008	CTOF	RTIF	CTOFE	RTIE	0	0	RT1	RT0	Core Timer Control & Status (CTCSR)	0000 0011
$0009									Core Timer Counter (CTCR)	0000 0000
$000A $000B									Unused	
$000C									Port E Data	not affected
$000D									Port E Data Direction (Port E DDR)	0000 0000
$000E									Port F Data	not affected
$000F									Port F Data Direction (Port F DDR)	0000 0000
$0010									Input Capture A High (ICRA)	not affected
$0011									Input Capture A Low (ICRA)	
$0012									Output Compare High (OCR)	
$0013									Output Compare Low (OCR)	
$0014									Input Capture B High (ICRB)	
$0015									Input Capture B Low (ICRB)	
$0016 $0017									Unused	
$0018									Counter High	not affected
$0019									Counter Low	
$001A									Alternate Counter High	
$001B									Alternate Counter Low	
$001C	ICAIE	ICBIE	OCIE	TOIE	0	IEDGA	IEDGB	OLV	Timer Control (TCR)	0000 0uu0
$001D									Unused	
$001E	ICAF	ICBF	OCF	TOF	0	0	0	0	Timer Status (TSR)	uuu0 0000
$001F	–	–	–	–	–	–	–	–	Reserved	

u = unaffected

Table 5-1(b) Register Outline (continued)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name	Reset Status
$0020	CK68	TCAPB	TCAPA	TCMP	WOM3	WOM2	WOM1	WOM0	Port E Mode	0000 0000
$0021 $0022									Unused	
$0023	–	0	SCP1	SCP0	–	SCR2	SCR1	SCR0	SCI 1 Baud (BAUD)	-000 -uuu
$0024	R8	T8	0	M	WAKE	0	0	0	SCI 1 Control 1 (SCCR1)	uu00 0000
$0025	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	SCI 1 Control 2 (SCCR2)	0000 0000
$0026	TDRE	TC	RDRF	IDLE	OR	NF	FE	0	SCI 1 Status (SCSR)	1100 0000
$0027									SCI 1 Data (SCDR)	0000 0000
$0028 $0029 $002A									Unused	
$002B	–	0	SCP1	SCP0	–	SCR2	SCR1	SCR0	SCI 2 Baud (BAUD)	-000 -uuu
$002C	R8	T8	0	M	WAKE	0	0	0	SCI 2 Control 1 (SCCR1)	uu00 0000
$002D	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	SCI 2 Control 2 (SCCR2)	0000 0000
$002E	TDRE	TC	RDRF	IDLE	OR	NF	FE	0	SCI 2 Status (SCSR)	1100 0000
$002F									SCI 2 Data (SCDR)	0000 0000
$0030									MCU Channel A Data Write (ADRWO5)	not affected
$0031									MCU Channel B Data Write (BDRWO5)	
$0032									MCU Channel C Data Write (CDRWO5)	
$0033									MCU Channel D Data Write (DDRWO5)	
$0034									MCU Channel A Data Read (ADRRO5)	
$0035									MCU Channel B Data Read (BDRRO5)	
$0036									MCU Channel C Data Read (CDRRO5)	
$0037									MCU Channel D Data Read (DDRRO5)	
$0038	ROR	RRDY	RFULL	REMTY	TOR	TRDY	TFULL	TEMTY	MCU Channel A Status (ASRO5)	0001 0001
$0039	ROR	RRDY	RFULL	REMTY	TOR	TRDY	TFULL	TEMTY	MCU Channel B Status (BSRO5)	0001 0001
$003A	ROR	RRDY	RFULL	REMTY	TOR	TRDY	TFULL	TEMTY	MCU Channel C Status (CSRO5)	0001 0001
$003B	ROR	RRDY	RFULL	REMTY	TOR	TRDY	TFULL	TEMTY	MCU Channel D Status (DSRO5)	0001 0001
$003C	RA	TA	RB	TB	RC	TC	RD	TD	MCU Interrupt Status (ISRO5)	0000 0000
$003D	RMA	TMA	RMB	TMB	RMC	TMC	RMD	TMD	MCU Interrupt Mask (IMRO5)	0000 0000
$003E	IMOD	0	0	0	AEN	BEN	CEN	DEN	MCU Mode (MRO5)	0000 0000
$003F									Unused	

u =unaffected

SECTION 6
PARALLEL INPUT/OUTPUT PORTS

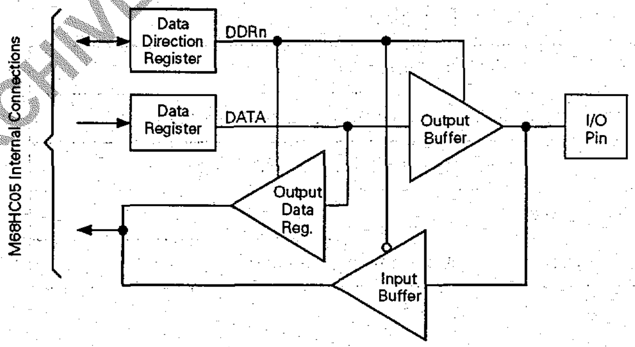
In Single Chip mode there are 42 lines arranged as five 8-bit and one 2-bit I/O ports. The I/O ports are programmable as either inputs or outputs under software control of the data direction registers. Port E also controls the timer functions, the external clock input and the wired-OR capability. Control of these functions is done via the Port E Mode register, explained later in this section.

To avoid glitches on the output pins data should be written to the I/O Port Data Register before setting the pin to output mode, by writing a "1" to the corresponding Data Direction Register.

6.1 INPUT/OUTPUT PROGRAMMING

Bi-directional port lines may be programmed as inputs or outputs under software control. The direction of each pin is determined by the state of the corresponding bit in the port data direction register (DDR). Each port has an associated DDR. Any I/O port 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, thus configuring all port pins as inputs. The data direction registers can be 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. Refer to Figure 6-1 and Table 6-1.



DDRn	DATA	I/O Pin
1	0	0
1	1	1
0	0	tri-state
0	1	tri-state

Figure 6-1 Bi-directional I/O Port Structure Block Diagram

6.2 PORTS A–D

Ports A, B, C and D are 8-bit bi-directional ports. The port data registers are located from $0000 to $0003 and the data direction registers (DDR) from $0004 to $0007. Reset does not affect the data registers, but clears the data direction registers, thereby returning the ports to inputs. Writing a "1" to a DDR bit sets the corresponding port bit to output mode.

6.3 PORT E

Port E is an 8-bit bi-directional I/O port. The port E data register is at $000C and the data direction register (DDR) is at $000D. Reset does not affect the data register, but clears the data direction register, thereby returning the port to inputs. This port can be reconfigured to provide the signals needed for the Serial Communication Interface units, the 16-bit timer and the M68000 Interface unit. Furthermore it has wired-OR capability on pins 0–3. The function of each pin is determined by the Port E Mode register at location $0020. With the exception of the SCI subsystems, a logic one in the mode register selects the associated subsystem, while a zero configures it as an I/O line. Reset clears the Port E Mode register, causing Port E to function as a standard I/O port. The action of setting the Port E Mode bits is described below and summarised in Table 6-2. To configure the port to support the SCIs, bits 2 and 3 of the SCI Control register 2 should be used. This register is described in Section 9.

6.4 PORT F

Port F is a 2-bit bi-directional port. The port F data register is at location $000E and the data direction register (DDR) at $000F. Reset does not affect the data register, but clears the data direction registers thereby returning the port to inputs. Writing a "1" to a DDR bit sets the corresponding port bit to output mode.

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 to the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in output mode. The output data latch is read.

† Note that $R/\overline{W}$ is an internal signal, not available to the user.

6.5 PORT REGISTERS

The following sections explain in detail the individual bits in the data and control registers associated with the I/O ports.

6.5.1 Port A–Port D Data Registers ($0000–$0003)

Each bit can be configured as input or output via the corresponding data direction bit in the Port DDR.

Reset does not affect the state of these registers.

6.5.2 Port E Data Register ($000C)

Each bit can be configured as input or output via the corresponding data direction bit in the Port E DDR. The alternate functions of Port E are controlled via the Port E Mode register and are described below.

Reset does not affect the state of this register.

6.5.3 Port F Data Register ($000E)

Each bit can be configured as input or output via the corresponding data direction bit in the Port F DDR.

Reset does not affect the state of this register.

6.5.4 Ports A–D Data Direction Register (Port x DDR) ($0004–$0007)

Writing a "1" to any bit configures the corresponding bit in the port data register as an output; conversely, writing any bit to "0" configures the corresponding port bit as an input.

Reset clears this register.

6.5.5 Port E Data Direction Register (Port E DDR) ($000D)

Writing a "1" to any bit configures the corresponding bit in the Port E data register as an output. Conversely, writing any bit to "0" configures the corresponding Port E bit as an input.

Reset clears this register.

6.5.6 Port E Mode Register ($0020)

In addition to the DDR, Port E has a mode register which controls the function of each pin in Port E. A "1" in a particular bit selects the "subsystem" while a "0" configures the pin as standard I/O.

Reset clears this register configuring all pins in Port E as standard I/O.

	7	6	5	4	3	2	1	0	
$0020	CK68	TCAPB	TCAPA	TCMP	WOM3	WOM2	WOM1	WOM0	Port E Mode Register
Reset	0	0	0	0	0	0	0	0	

6.5.6.1 CK68 — External clock mode for the M68000 unit

1 (set) – Selects external clock mode for $\overline{\text{DTACK}}$ synchronization in the M68000 system and configures Port E pin 7 as the external clock input CLK68K to the M68000 subsystem.

0 (clear) – Port E pin 7 is configured as a standard I/O pin. $\overline{\text{DTACK}}$ is synchronised with the internal MCU clock.

6.5.6.2 TCAPB — Input Capture B

1 (set) – Port E pin 6 is configured as the Timer Input Capture B input pin TCAPB.

0 (clear) – Port E pin 6 is configured as a standard I/O pin.

6.5.6.3 TCAPA — Input Capture A

1 (set) – Port E pin 5 is configured as the Timer Input Capture A input pin TCAPA.

0 (clear) – Port E pin 5 is configured as a standard I/O pin.

6.5.6.4 TCMP — Output Compare

1 (set) – Port E pin 4 is configured as the Timer Output Compare output pin TCMP.

0 (clear) – Port E pin 4 is configured as a standard I/O pin.

6.5.6.5 WOM0–3 — Wired-OR mode

1 (set) – The corresponding Port E line (bits 0–3) is configured for Wired-OR mode operation.

0 (clear) – Wired-OR mode is disabled

Note: Wired-OR mode can be selected when the pin functions as a standard I/O pin or as an SCI pin. Bits 2 and 3 of the relevant SCI Control register 2 are used to configure the Port E bits 0–3 to support the SCI functions. See Section 9 for further details.

Table 6-2 Port E Mode Table

Bit	Value	Function
WOM0	0 1	No wired-OR capability on pin PE0 Wired-OR capability on this pin
WOM1	0 1	No wired-OR capability on pin PE1 Wired-OR capability on this pin
WOM2	0 1	No wired-OR capability on pin PE2 Wired-OR capability on this pin
WOM3	0 1	No wired-OR capability on pin PE3 Wired-OR capability on this pin
TCMP	0 1	PE4 functions as standard I/O Timer compare function on this pin
TCAPA	0 1	PE5 functions as standard I/O Timer compare function on this pin
TCAPB	0 1	PE6 functions as standard I/O Timer compare function on this pin
CK68	0 1	DTACK synchronised with MCU clock DTACK synchronised with external clock

6.5.7 Port F Data Direction Register (Port F DDR) ($000F)

Writing a "1" to any bit configures the corresponding bit in the Port F data register as an output. Conversely, writing any bit to "0" configures the corresponding Port F bit as an input.

Reset clears this register.

6.6 OTHER PORT CONSIDERATIONS

All Input/Output ports can emulate open-drain outputs. This is achieved by writing a zero to the relevant output port latch. By toggling the corresponding data direction bit, the port pin will either be an output zero or tri-state (an input). Refer to Figure 6-2.

When using a port pin as an open-drain output, certain precautions must be taken in the user software. If a read-modify-write instruction is used on a port where the open-drain is assigned and the pin at this time is programmed as an input, it will read it as a "1". The read-modify-write instruction will then write this "1" into the output data latch on the next cycle. This would cause the open-drain pin not to output a "0" when desired.

Note: "Open-drain" outputs should not be pulled above V_{DD}.

DDRn	Data	I/O Pin	
1	0	0	Normal operation
1	1	1	
0	0	tri-state	
0	1	tri-state	
1	0	low	"Open-drain"
1	1	—	
0	0	high	
0	1	high	

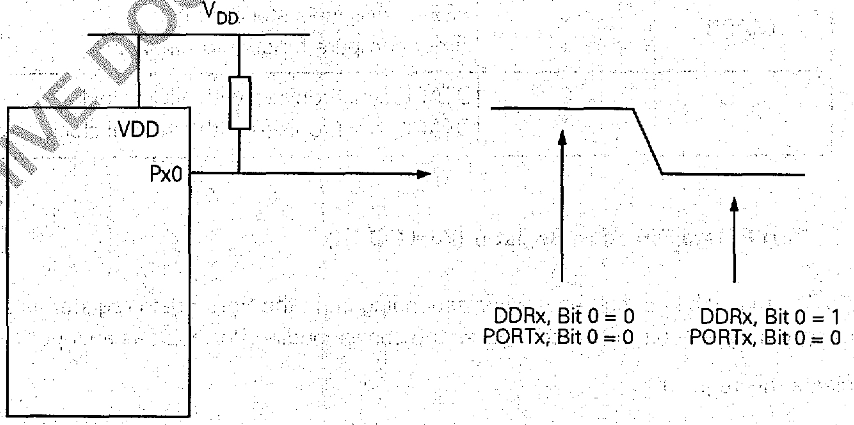


Figure 6-2 Port Logic Levels

SECTION 7
PROGRAMMABLE TIMER

The programmable timer on the MC68HC05i8 consists of a 16-bit read-only free-running counter, with a fixed divide-by-four prescaler, plus the input capture/output compare circuitry. Selected input edges cause the current counter value to be latched into a 16-bit input capture register so that software can later read this value to determine when the edge occurred. When the free running counter value matches the value in the output compare registers, the programmed pin action takes place. Refer to Figure 7-1 for a block diagram of the Timer.

The timer has a 16-bit architecture hence each specific functional segment is represented by two 8-bit registers. These registers contain the high and low byte of that functional segment. Accessing the low byte of a specific timer function allows full control of that function; however, an access of the high byte inhibits that specific timer function until the low byte is also accessed.

Note: The I-bit in the 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.

The various timer functions are enabled (on their respective Port E pins) by setting the relevant bits in the Port E Mode register (see Section 6.5.7).

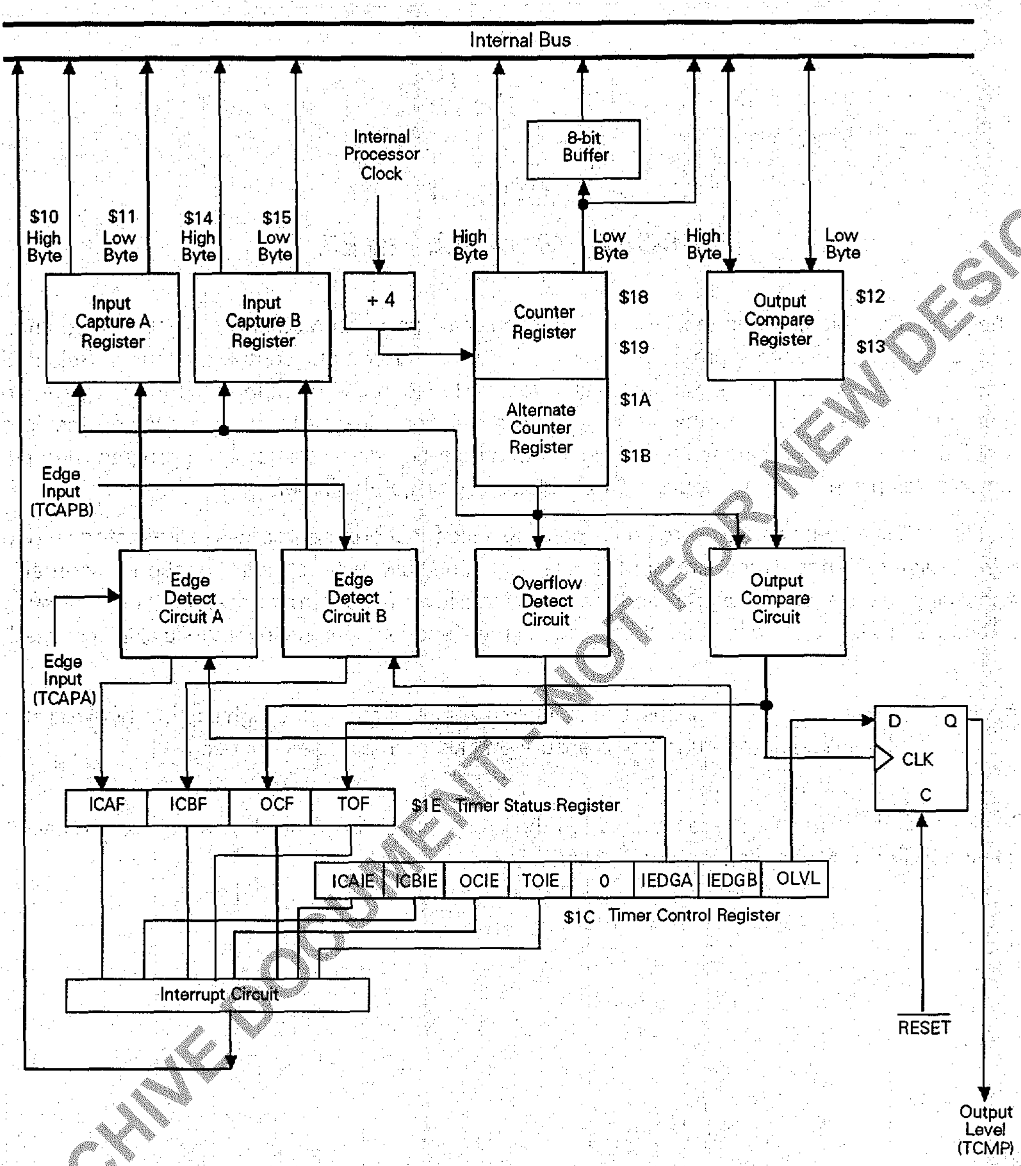


Figure 7-1 16-bit Programmable Timer Block Diagram

7.1 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 2μs if the internal bus clock is 2 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.

7.1.1 Counter Register ($18/19) and Alternate Counter Register ($1A/1B)

	7	6	5	4	3	2	1	0	
$0018									Counter High Register
$0019									Counter Low Register
$001A									Alternate Counter High Register
$001B									Alternate Counter Low Register
Reset:	Register states not affected by Reset								

The double-byte, free-running counter can be read from either of two locations, $18-$19 (counter register) or $1A-$1B (alternate counter register). A read from only the less significant byte (LSB) of the free-running counter ($19 or $1B) receives the count value at the time of the read. If a read of the free-running counter or alternate counter register first addresses the more significant byte (MSB) ($18 or $1A), the LSB 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 alternate counter register LSB and thus completes a read sequence of the total counter value. In reading either the free-running counter or alternate counter register, if the MSB is read, the LSB must also be read to complete the sequence. If the timer overflow flag (TOF) is set when the counter register LSB is read then a read of the TSR will clear the flag.

The Alternate Counter register differs from the Counter register only in that a read of the LSB does not clear TOF. Therefore, to avoid the possibility of missing timer overflow interrupts due to clearing of TOF, the Alternate Counter register should be used where this is a critical issue.

The free-running counter is set 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. TOF is set when the counter overflows (from $FFFF to $0000); this will cause an interrupt if TOIE is set.

7.2 TIMER FUNCTIONS

The 16-bit programmable timer is monitored and controlled by a group of ten registers, full details of which are contained in the following paragraphs. An explanation of the Timer functions is also given.

7.2.1 Timer Control Register (TCR)

The Timer Control register ($1C) is used to enable the Input Captures (ICAIE and ICBIE), Output Compare (OCIE), and Timer Overflow (TOIE) functions as well as selecting input edge sensitivity (IEDGA and IEDGB) and output level polarity (OLVL).

	7	6	5	4	3	2	1	0	
$001C	ICAIE	ICBIE	OCIE	TOIE	0	IEDGA	IEDGB	OLVL	Timer Control Register
Reset	0	0	0	0	0	U	U	0	

7.2.1.1 ICAIE — Input Capture A Interrupt Enable

1 (set) – Interrupt enabled

0 (clear) – Interrupt disabled

7.2.1.2 ICBIE — Input Capture B Interrupt Enable

1 (set) – Interrupt enabled

0 (clear) – Interrupt disabled

7.2.1.3 OCIE — Output Compare Interrupt Enable

1 (set) – Interrupt enabled

0 (clear) – Interrupt disabled

7.2.1.4 TOIE — Timer Overflow Interrupt Enable

1 (set) – Interrupt enabled

0 (clear) – Interrupt disabled

7.2.1.5 IEDGA — Input Edge A

When IEDGA is set, a positive-going edge on the TCAPA pin will trigger a transfer of the free-running counter value to the input capture register. When clear a negative-going edge triggers the transfer.

1 (set) – TCAPA is positive-going edge sensitive.

0 (clear) – TCAPA is negative-going edge sensitive.

7.2.1.6 IEDGB — Input Edge B

When IEDGB is set, a positive-going edge on the TCAPB pin will trigger a transfer of the free-running counter value to the input capture register. When clear a negative-going edge triggers the transfer.

1 (set) – TCAPB is positive-going edge sensitive.

0 (clear) – TCAPB is negative-going edge sensitive.

7.2.1.7 OLVL — Output Level

When OLVL is set a high output level will be clocked into the output level register by the next successful output compare, and will appear on the TCMP pin. When clear, it will be a low level which will appear on the TCMP pin.

1 (set) – A high output level will appear on the TCMP pin.

0 (clear) – A low output level will appear on the TCMP pin.

7.2.2 Timer Status Register (TSR)

The Timer Status register ($1E) contains the status bits corresponding to the four timer interrupt conditions – ICAF, ICBF, OCF and TOF.

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

	7	6	5	4	3	2	1	0	
$001E	ICAF	ICBF	OCF	TOF	0	0	0	0	Timer Status Register
Reset	U	U	U	0	0	0	0	0	

7.2.2.1 ICAF — Input Capture A Status Flag

This bit is set when the selected polarity of edge is detected by the input capture edge detector; an input capture interrupt will be generated, if ICAIE is set. ICAF is cleared by reading the TSR and then the Input Capture A Low register ($11).

1 (set) – A valid input capture has occurred.

0 (clear) – No input capture has occurred.

7.2.2.2 ICBF — Input Capture B Status Flag

This bit is set when the selected polarity of edge is detected by the input capture edge detector; an input capture interrupt will be generated, if ICBIE is set. ICBF is cleared by reading the TSR and then the Input Capture B Low register ($15).

1 (set) – A valid input capture has occurred.

0 (clear) – No input capture has occurred.

7.2.2.3 OCF — Output Compare Status Flag

This bit is set when the output compare register contents match those of the free-running counter; an output compare interrupt will be generated, if OCIE is set. OCF is cleared by reading the TSR and then the Output Compare Low register ($13).

1 (set) – A valid output compare has occurred.

0 (clear) – No output compare has occurred.

7.2.2.4 TOF — Timer Overflow Status Flag

This bit is set when the free-running counter overflows from $FFFF to $0000; a timer overflow interrupt will occur, if TOIE is set. TOF is cleared by reading the TSR and the counter low register ($19).

1 (set) – Timer Overflow has occurred.

0 (clear) – No timer overflow has occurred.

When using the timer overflow function and reading the free-running counter at random times to measure an elapsed time, a problem may occur whereby the timer overflow flag is unintentionally cleared if:

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

Reading the alternate counter register instead of the counter register will avoid this potential problem.

7.2.3 Input Capture Function

'Input Capture' is a technique whereby an external signal is used to trigger a read of the free running counter. In this way it is possible to relate the timing of an external signal to the internal counter value, and hence to elapsed time.

Note: The pins used for monitoring the external signals are the shared PE5/TCAPA and PE6/TCAPB pins, hence when this function is used the corresponding port pin cannot be used as a standard I/O line.

There are two input capture registers: Input Capture Register A (ICRA) and Input Capture Register B (ICRB).

7.2.3.1 Input Capture Register A (ICRA)

	7	6	5	4	3	2	1	0	
$0010									Input Capture A High Register
$0011									Input Capture A Low Register
Reset:	Register states not affected by Reset								

The two 8-bit registers that make up the 16-bit Input Capture Register A are read-only, and are used to latch the value of the free-running counter after the input capture edge detector senses a valid transition. The level transition that triggers the counter transfer is defined by the input edge bit (IEDGA). The 8 most significant bits are stored in the Input Capture A High Register at $10, the 8 least significant bits in the Input Capture A Low register at $11.

The result obtained from an input capture will be one greater 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 A register on each valid signal transition whether the input capture A flag (ICAF) is set or clear. The input capture A register always contains the free-running counter value that corresponds to the most recent input capture A. After a read of the input capture A register MSB ($10), the counter transfer is inhibited until the LSB ($11) 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 A register LSB ($11) does not inhibit the free-running counter transfer since the two actions occur on opposite edges of the internal bus clock.

Reset does not affect the contents of the Input Capture A register, except when exiting STOP mode. (See Section 7.4)

7.2.3.2 Input Capture Register B (ICRB)

	7	6	5	4	3	2	1	0	
$0014									Input Capture B High Register
$0015									Input Capture B Low Register
Reset:	Register states not affected by Reset								

The two 8-bit registers that make up the 16-bit Input Capture Register B are read-only, and are used to latch the value of the free-running counter after the input capture edge detector senses a valid transition. The level transition that triggers the counter transfer is defined by the input edge bit (IEDGB). The 8 most significant bits are stored in the Input Capture B High Register at $14, the 8 least significant bits in the Input Capture B Low Register at $15.

The result obtained from an input capture will be one greater 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 B on each valid signal transition whether the Input Capture B Flag (ICBF) is set or clear. The Input Capture Register B always contains the free-running counter value that corresponds to the most recent input capture B. After a read of the Input Capture Register B MSB ($14), 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 B LSB ($15) does not inhibit the free-running counter transfer since the two actions occur on opposite edges of the internal bus clock.

Reset does not affect the contents of the Input Capture B Register, except when exiting STOP mode. (See Section 7.4)

7.2.4 Output Compare Function

'Output Compare' is a technique which may be used, for example, to generate an output waveform, or to signal when a specific time period has elapsed, by presetting the Output Compare Register to the appropriate value.

Note: The output pin used is the shared PE4/TCMP, hence the PE4 I/O pin function will be lost when the Output Compare function is enabled.

7.2.4.1 Output Compare Register (OCR)

	7	6	5	4	3	2	1	0	
$0012									Input Compare A High Register
$0013									Input Compare A Low Register
Reset:	Register states not affected by Reset								

The 16-bit Output Compare Register is made up of two 8-bit registers at locations $12 (MSB) and $13 (LSB). The contents of the Output Compare Register are compared with the contents of the free-running counter continually and, if a match is found, the corresponding Output Compare Flag (OCF) in the Timer Status Register is set and the output level (OLVL) bit clocked to the 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 (OCIE) is set. (The free-running counter is updated every four internal bus clock cycles.)

After a processor write cycle to the Output Compare Register containing the MSB ($12), the output compare function is inhibited until the LSB ($13) is also written. The user must write both bytes (locations) if the MSB is written first. A write made only to the LSB ($13) will not inhibit the compare function. The processor can write to either byte of the Output Compare Register without affecting the other byte. The output level (OLVL) bit is clocked to the Output Level Register whether the Output Compare Flag (OCF) is set or clear. The minimum time required to update the Output Compare Register is a function of the program rather than the internal hardware. Because the Output Compare Flag and the Output Compare Register are not defined at power on, and not affected by reset, care must be taken when initializing output compare functions with software. The following procedure is recommended:

- Write to Output Compare High to inhibit further compares;
- Read the Timer Status Register to clear OCF (if set);
- Write to Output Compare Low to enable the output compare function.

All bits of the Output Compare Register are readable and writable and are not altered by the timer hardware or reset. If the compare function is not needed, the two bytes of the output compare register can be used as storage locations.

7.3 TIMER DURING WAIT MODE

The CPU clock halts during the WAIT mode, hence the Timer stops, and holds the last count value. On exit from WAIT mode the timer will resume counting from where it had stopped. If RESET is used to exit the WAIT mode then the counters are forced to $FFFC.

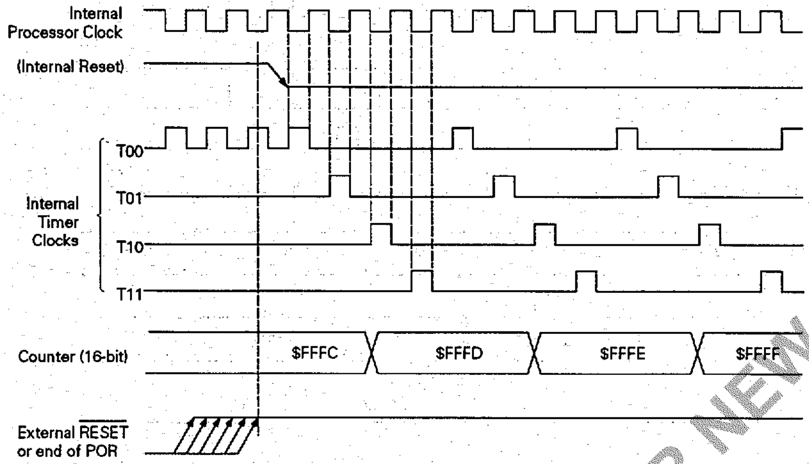
7.4 TIMER DURING STOP MODE

In the STOP mode all MCU clocks are stopped, hence the Timer stops counting. If STOP is exited by an interrupt the counter retains the last count value. If the device is reset then the counter is forced to $FFFC.

During STOP, if at least one valid input capture edge occurs at either the TCAPA or TCAPB pin, the input capture detect circuit is armed. This does not set any timer flags nor wake up the MCU. When the MCU does wake up however, there is an active input capture flag and data from the first valid edge that occurred during the STOP period. If the device is reset to exit STOP mode, then no input capture flag or data remains, even if a valid input capture edge occurred.

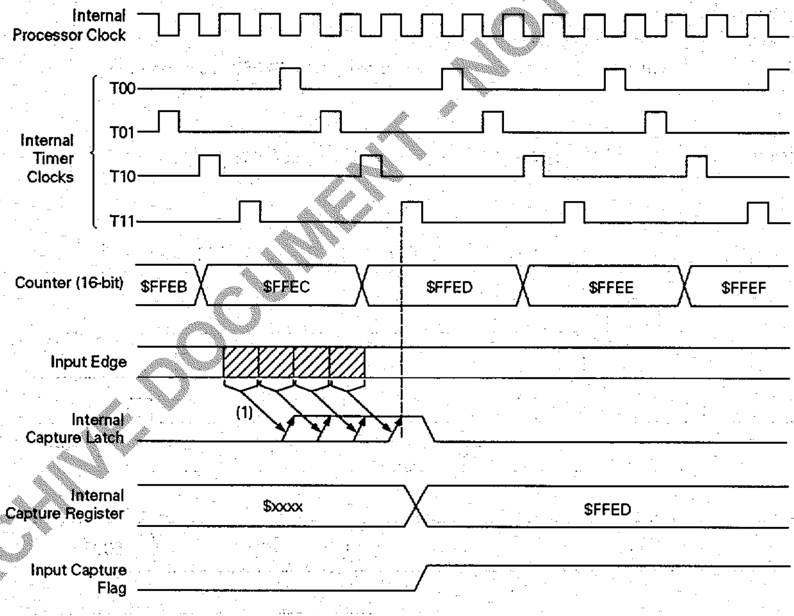
7.5 TIMER STATE DIAGRAMS

The relationships between the internal clock signals, the counter contents and the status of the flag bits are shown in the following figures. It should be noted that the signals labelled "internal" (Processor clock, timer clocks and Reset) are not available to the user.



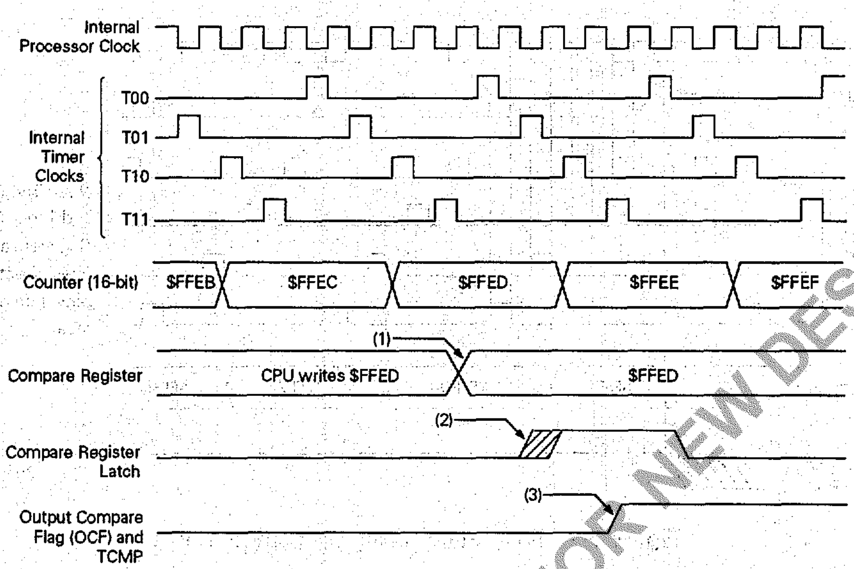
Note: The Counter and Timer Control Registers are the only ones affected by power-on or external reset.

Figure 7-2 Timer State Timing Diagram for Reset



Note: If the input edge occurs in the shaded area from one timer state T10 to the next timer state T10, then the input capture flag will be set during the next T11 state.

Figure 7-3 Timer State Timing Diagram for Input Captures

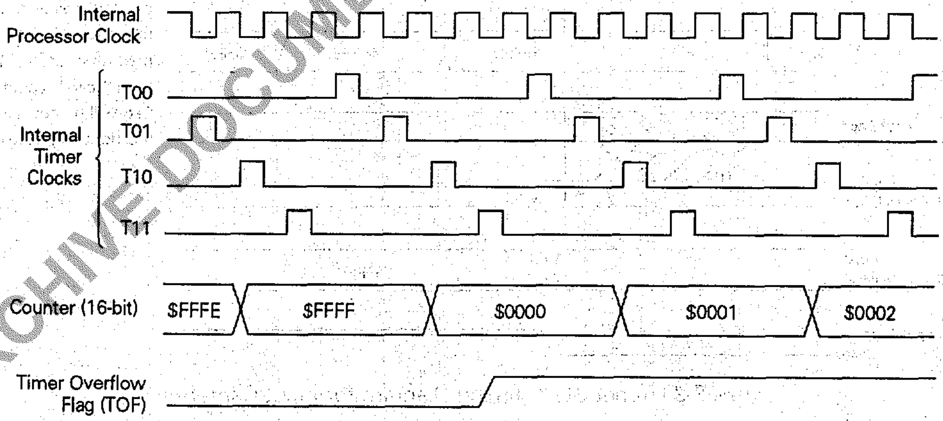


NOTE 1: The CPU write to the compare registers may take place at any time, but a compare only occurs at timer state T01. Thus a four cycle difference may exist between the write to the compare register and the actual compare.

NOTE 2: Internal compare takes place during timer state T01.

NOTE 3: OCF is set at the timer state T11 that follows the comparison match ($FFED in this example).

Figure 7-4 Timer State Timing Diagram for Output Compare



NOTE: The TOF-bit is set at timer state T11 (transition of counter from $FFFF to $0000). It is cleared by a read of the timer status register during the internal processor clock high time followed by a read of the counter low register.

Figure 7-5 Timer State Timing Diagram for Timer Overflow

SECTION 8
CORE TIMER

Besides the 16-bit timer on the MC68HC05i8, there is a 15-stage ripple counter called the Core Timer (CTIMER). Features of this timer are: Timer Overflow; Power-On Reset (POR); Real Time interrupt (RTI), with four selectable interrupt rates; and a Computer Operating Properly (COP) watchdog timer.

As seen in Figure 8.1, the timer is driven by the internal bus clock divided by four with a fixed prescaler. This signal drives an 8-bit ripple counter. The value of this 8-bit ripple counter can be read by the CPU at any time, by accessing the CTIMER Counter Register (CTCR) at address $09. A timer overflow function is implemented on the last stage of this counter, giving a possible interrupt at the rate of f_{OP}/1024. (The POR signal is also derived from this register, at f_{OP}/4064.) The Counter Register circuit is followed by four 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 8 to drive the COP Watchdog Timer circuit. The RTI rate selector bits, and the RTI and CTIMER overflow enable bits and flags are located in the CTIMER Control and Status Register (CTCSR) at location $08.

CTOF (Core Timer Overflow Flag) 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 CTOFE is set. Clearing the CTOF is done by writing a "0" to it. Writing a "1" to CTOF has no effect on the bit's value. Reset clears CTOF.

When CTOFE (Core Timer Overflow Enable) is set, a CPU interrupt request is generated when the CTOF bit is set. Reset clears CTOFE.

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

The power-on cycle clears the entire counter chain and begins clocking the counter. After 4064 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 at any time during operation (other than POR), the counter chain will be cleared.

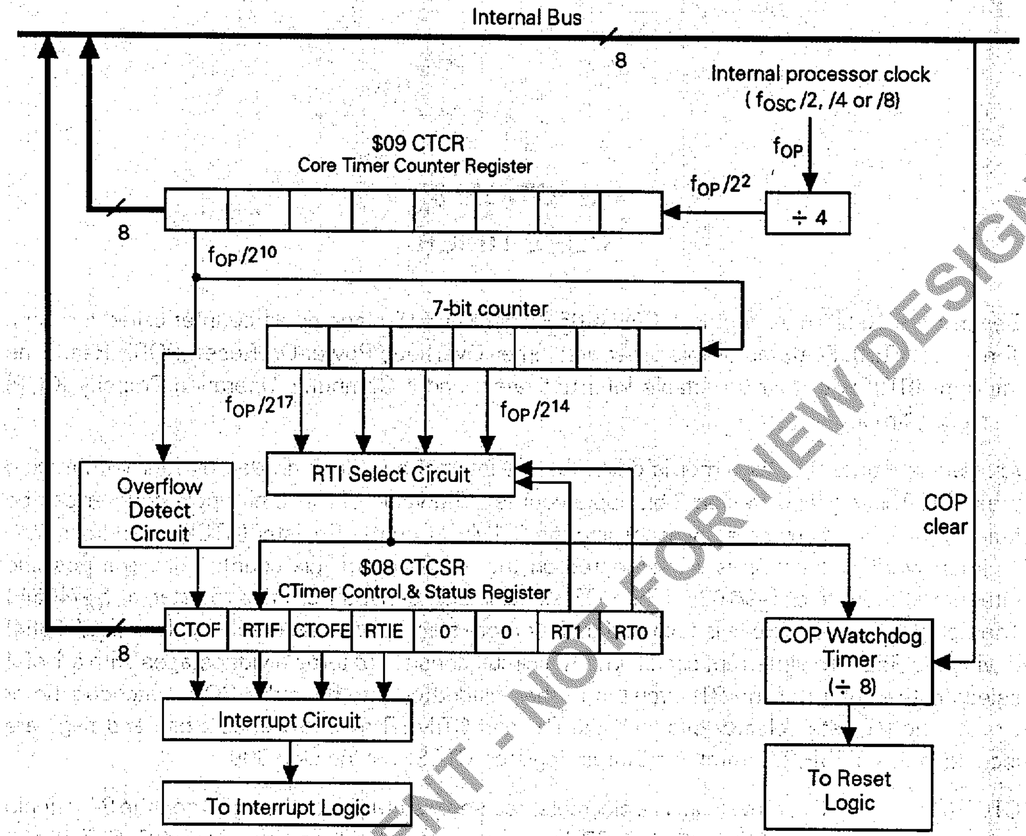


Figure 8-1 Core Timer Block Diagram

8.1 REAL TIME INTERRUPTS (RTI)

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 $f_{OP}/2^{14}$ (or $f_{OP}/16384$), with three additional divider stages, giving a maximum interrupt period of 4 seconds at a bus frequency (f_{OP}) of 32kHz. Register details are given in Section 8.3.

8.2 COMPUTER OPERATING PROPERLY (COP) WATCHDOG TIMER

The COP watchdog timer function is implemented by taking the output of the RTI circuit and further dividing it by eight, as shown in Figure 8-1. Note that the minimum COP timeout period is seven times the RTI period. This is because the COP will be cleared asynchronously with respect to the value in the Core Timer counter register/RTI divider, hence the actual COP timeout period will vary between 7x and 8x the RTI period.

The COP function is a mask option, selectable via the Mask Option register during device manufacture.

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 is cleared (see Figure 8-1).

8.3 CORE TIMER REGISTERS

8.3.1 Core Timer Control and Status Register (CTCSR) ($08)

$08	CTOF	RTIF	CTOFE	RTIE	0	0	RT1	RT0
Reset	0	0	0	0	0	0	1	1

8.3.1.1 CTOF — Core Timer Overflow

This bit is set when the Core Timer Counter register rolls over from $FF to $00; an interrupt request will be generated if CTOFE is set. When set, the bit may be cleared by writing a "0" to it.

1 (set) – Core Timer Overflow has occurred.

0 (clear) – No Core Timer Overflow interrupt has been generated.

8.3.1.2 RTIF — Real Time Interrupt Flag

This bit is set when the output of the chosen stage becomes active; an interrupt request will be generated if RTIE is set. When set, the bit may be cleared by writing a "0" to it.

1 (set) – A Real Time interrupt has occurred.

0 (clear) – No Real Time interrupt has been generated.

8.3.1.3 CTOFE — Core Timer Overflow Enable

Setting this bit enables the Core Timer Overflow Interrupt. A CPU interrupt request will then be generated whenever the CTOF bit becomes set. Clearing this bit disables the Core Timer Overflow interrupt capability.

1 (set) – Core Timer Overflow interrupt is enabled.

0 (clear) – Core Timer Overflow interrupt is disabled.

8.3.1.4 RTIE — Real Time Interrupt Enable

Setting this bit enables the Real Time Interrupt. A CPU interrupt request will then be generated whenever the RTIF bit becomes set. Clearing this bit disables the Real Time interrupt capability.

1 (set) – Real Time interrupt is enabled.

0 (clear) – Real Time interrupt is disabled.

8.3.1.5 RT1:RT0 — Real Time Interrupt Rate Select

These two bits select one of four taps from the Real Time Interrupt circuitry. Reset sets both RT0 and RT1 to one, selecting the lowest periodic rate and therefore the maximum time in which to alter them if necessary. The COP reset times are also determined by these two bits. Care should be taken when altering RT0 and RT1 if a timeout is imminent, or the timeout period is 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 the RTI taps. See Table 8-1 for some example RTI periods.

Table 8-1 Example RTI Periods

RT1	RT0	Division Ratio	RTI Period at a Bus Frequency of:	
			16.384kHz	2.097MHz
0	0	2^{14}	1 s	7.8 ms
0	1	2^{15}	2 s	15.6 ms
1	0	2^{16}	4 s	31.3 ms
1	1	2^{17}	8 s	62.5 ms

8.3.2 Core Timer Counter Register (CTCR) ($09)

The Core Timer Counter register is a read-only register, which contains the current value of the 8-bit ripple counter at the beginning of the timer chain.

Reset clears this register.

8.4 CORE TIMER DURING WAIT

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

8.5 CORE TIMER DURING STOP

The timer is cleared when going into STOP mode. When STOP is exited by an external interrupt or an external reset, the internal oscillator will restart, followed by an internal processor stabilisation delay (t_{PORL}). The timer is then cleared and operation resumes.

SECTION 9
SERIAL COMMUNICATIONS INTERFACE

This section describes the UART type serial communications interface system (SCI). The SCI can be used, for example, to connect a CRT terminal or personal computer to the MCU or to form a serial communication network connecting several widely distributed MCUs. The MC68HC05i8 has two independent SCI subsystems — SCI1 and SCI2. The internal operation of each SCI is identical, with each unit having its own set of control registers. SCI1 Control registers 1 and 2 are at locations $0024 and $0025 respectively, and SCI2 Control registers 1 and 2 are at locations $002C and $002D. Each of the SCI units has its own Baud Rate register, SCI1 at location $0023 and SCI2 at location $002B, which allows each unit to receive and transmit data independently. For example, SCI1 can be set up to receive and transmit 8-bit data at 9600 baud while SCI2 can receive and transmit 9-bit data at 19,200 baud. Because of the identical nature and operation of each unit, only a description of SCI1 is provided in this manual. Both units are covered in the register summary.

The external signals for the SCI blocks are routed through Port E bits 0–3. Bit 0 and bit 1 are the receive and transmit pins for SCI1 (RX1 and TX1), while bit 2 and bit 3 are the receive and transmit pins for SCI2 (See Figure 9-1). When either or both of the transmitters or receivers are disabled, the respective pins can be used as standard I/O. If open drain operation is required for either of the transmitters, it can be obtained by writing to the WOMx bit in the Port E mode register. (See Section 6.2.)

9.1 OVERVIEW AND FEATURES

The SCI on the MC68HC05i8 is a full duplex UART type asynchronous system. The SCI uses standard non-return-to-zero (NRZ) format (one start bit, eight or nine data bits, and one stop bit). An on-chip baud rate generator derives standard baud rate frequencies from the MCU oscillator. Both the transmitter and the receiver are double buffered so back-to-back characters can be handled easily, even if the CPU is delayed in responding to the completion of an individual character. The SCI transmitter and receiver are functionally independent, but use the same data format and baud rate.

9.1.1 SCI Two-wire System Features

- Standard NRZ (mark/space) format
- Advanced error detection method includes noise detection for noise duration of up to 1/16th bit time
- Full-duplex operation
- Software programmable for one of 32 different baud rates
- Software selectable word length (eight or nine bits)
- Separate transmitter and receiver enable bits
- Capable of being interrupt driven
- Four separate enable bits available for interrupt control

9.1.2 SCI Receiver Features

- Receiver wake-up function (idle line or address bit)
- Idle line detect
- Framing error detect
- Noise detect
- Overrun detect
- Receiver data register full flag

9.1.3 SCI Transmitter Features

- Transmit data register empty flag
- Transmit complete flag
- Send break

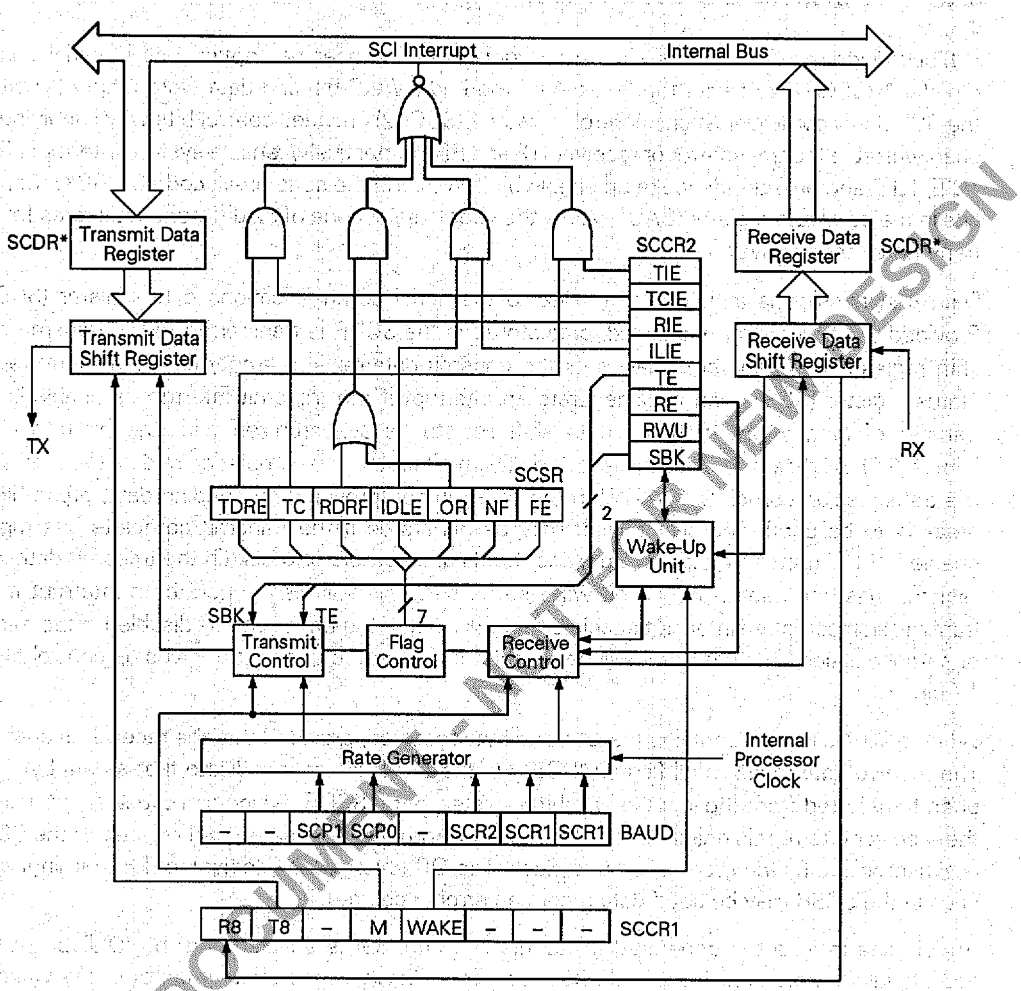
9.2 FUNCTIONAL DESCRIPTION

A block diagram of the SCI is shown in Figure 9-1. The user has option bits in serial control register 1 (SCCR1) to select the "wake-up" method (WAKE bit) and data word length (M bit) of the SCI. Serial communications control register 2 (SCCR2) provides control bits which individually enable/disable the transmitter or receiver (TE and RE, respectively), enable system interrupts (TIE, TCIE, ILIE) and provide the wake-up enable bit (RWU) and the send break code bit (SBK). Control bits in the baud rate register (BAUD) allow the user to select one of 32 different baud rates for the transmitter and receiver.

Data transmission is initiated by a write to the serial communications data register (SCDR). Provided the transmitter is enabled, data stored in the SCDR is transferred to the transmit data shift register. This transfer of data sets the transmit data register empty flag (TDRE) in the SCI status register (SCSR) and may generate an interrupt if the transmit interrupt is enabled. The transfer of data to the transmit data shift register is synchronized with the bit rate clock (Figure 9-2). All data is transmitted least significant bit first. Upon completion of data transmission, the transmission complete flag (TC) in the SCSR is set (provided no pending data, preamble or break is to be sent), and an interrupt may be generated if the transmit complete interrupt is enabled. If the transmitter is disabled, and the data, preamble or break (in the transmit data shift register) has been sent, the TC bit will also be set. This will also generate an interrupt if the transmission complete interrupt enable bit (TCIE) is set. If the transmitter is disabled in the middle of a transmission, that character will be completed before the transmitter gives up control of the TX pin.

When SCDR is read, it contains the last data byte received, provided that the receiver is enabled. The receive data register full flag bit (RDRF) in the SCSR is set to indicate that a data byte has been transferred from the input serial shift register to the SCDR, which can cause an interrupt if the receiver interrupt is enabled. The data transfer from the input serial shift register to the SCDR is synchronized by the receiver bit rate clock. The OR (overrun), NF (noise), or FE (framing) error flags in the SCSR may be set if data reception errors occurred.

An idle line interrupt is generated if the idle line interrupt is enabled and the IDLE bit (which detects idle line transmission) in SCSR is set. This allows a receiver that is not in the wake-up mode to detect the end of a message or the preamble of a new message, or to resynchronize with the transmitter. A valid character must be received before the idle line condition or the IDLE bit will not be set and idle line interrupt will not be generated.



Note: * The Serial Communications Data Register (SCDR) is controlled by the internal R/$\overline{W}$ signal. It is the transmit data register when written and the receive data register when read.

Figure 9-1 Serial Communications Interface Block Diagram

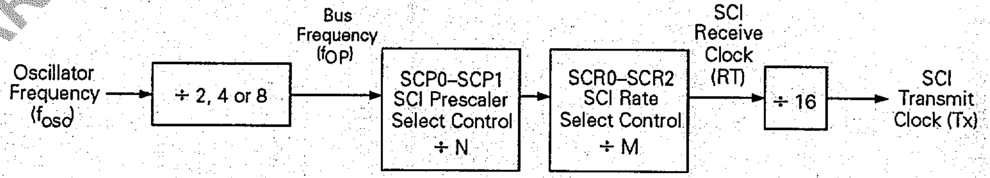

Oscillator Frequency (f_{osc})
÷ 2, 4 or 8
Bus Frequency (f_{OP})
SCP0–SCP1 SCI Prescaler Select Control ÷ N
SCR0–SCR2 SCI Rate Select Control ÷ M
SCI Receive Clock (RT)
÷ 16
SCI Transmit Clock (Tx)

Figure 9-2 Rate Generator Division

9.3 DATA FORMAT

Receive data or transmit data is the serial data that is transferred to the internal data bus from the receive data input pin (RX) or from the internal bus to the transmit data output pin (TX). The non-return-to-zero (NRZ) data format shown in Figure 9-3 is used and must meet the following criteria:

- The idle line is brought to a logic one state prior to transmission/reception of a character.
- A start bit (logic zero) is used to indicate the start of a frame.
- The data is transmitted and received least significant bit first.
- A stop bit (logic one) is used to indicate the end of a frame. A frame consists of a start bit, a character of eight or nine data bits, and a stop bit.
- A break is defined as the transmission or reception of a low (logic zero) for at least one complete frame time.

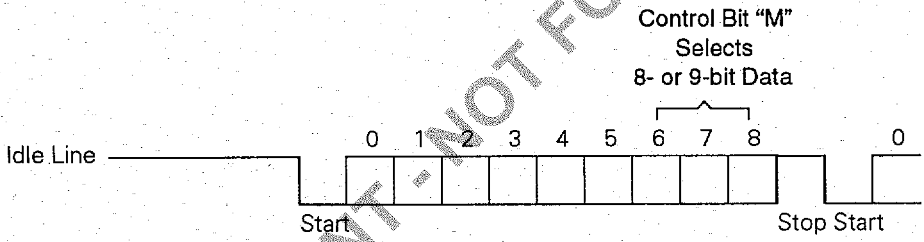


Figure 9-3 Data Format

9.4 RECEIVER WAKE-UP OPERATION

The receiver logic hardware also supports a receiver wake-up function which is intended for systems having more than one receiver. With this function a transmitting device directs messages to an individual receiver or group of receivers by passing addressing information as the initial byte(s) of each message. The wake-up function allows receivers not addressed to remain in a dormant state for the remainder of the unwanted message. This eliminates any further software overhead to service the remaining characters of the unwanted message and thus improves system performance.

The receiver is placed in wake-up mode by setting the receiver wake-up bit (RWU) in the SCCR2 register. While RWU is set, all of the receiver related status flags (RDRF, IDLE, OR, NF, and FE) are inhibited (cannot become set). Note that the idle line detect function is inhibited while the RWU bit is set. Although RWU may be cleared by a software write to SCCR2, it would be unusual to do so. Normally RWU is set by software and is cleared automatically in hardware by one of the two methods described below.

9.4.1 Idle Line Wake-up

In idle line wake-up mode, a dormant receiver wakes up as soon as the RX line becomes idle. Idle is defined as a continuous logic high level on the RX line for ten (or eleven) full bit times. Systems using this type of wake-up must provide at least one character time of idle between messages to wake up sleeping receivers, but must not allow any idle time between characters within a message.

9.4.2 Address Mark Wake-up

In address mark wake-up, the most significant bit (MSB) in a character is used to indicate whether it is an address (1) or a data (0) character. Sleeping receivers will wake up whenever an address character is received. Systems using this method for wake-up would set the MSB of the first character of each message and leave it clear for all other characters in the message. Idle periods may be present within messages and no idle time is required between messages for this wake-up method.

9.5 RECEIVE DATA (RX)

Receive data is the serial data that is applied through the input line and the serial communications interface to the internal bus. The receiver circuitry clocks the input at a rate equal to 16 times the baud rate and this time is referred to as the RT clock.

Once a valid start bit is detected the start bit, each data bit and the stop bit are sampled three times at RT intervals 8 RT, 9 RT and 10 RT (1 RT is the position where the bit is expected to start), as shown in Figure 9-4. The value of the bit is determined by voting logic which takes the value of the majority of the samples.

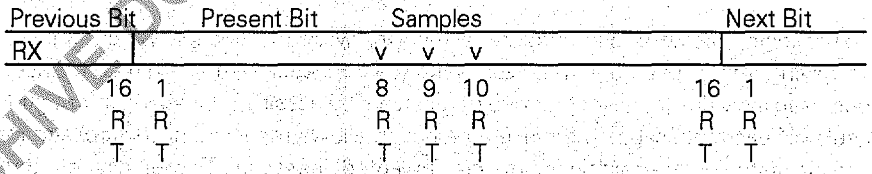


Figure 9-4 Sampling Technique Used On All Bits

9.6 START BIT DETECTION

When the RX input is detected low, it is tested for three more sample times (referred to as the start edge verification samples in Figure 9-5). If at least two of these three verification samples detect a logic zero, a valid start bit has been detected, otherwise the line is assumed to be idle. A noise flag is set if all three verification samples do not detect a logic zero. A valid start bit could be assumed with a set noise flag present.

If there has been a framing error without detection of a break (10 zeros for 8-bit format or 11 zeros for 9-bit format), the circuit continues to operate as if there actually was a stop bit and the start edge will be placed artificially. The last bit received in the data shift register is inverted to a logic one, and the three logic one start qualifiers (shown in Figure 9-5) are forced into the sample shift register during the interval when detection of a start bit is anticipated (see Figure 9-6); therefore, the start bit will be accepted no sooner than it is anticipated.

If the receiver detects that a break produced the framing error, the start bit will not be artificially induced and the receiver must actually detect a logic one before the start bit can be recognised (see Figure 9-7).

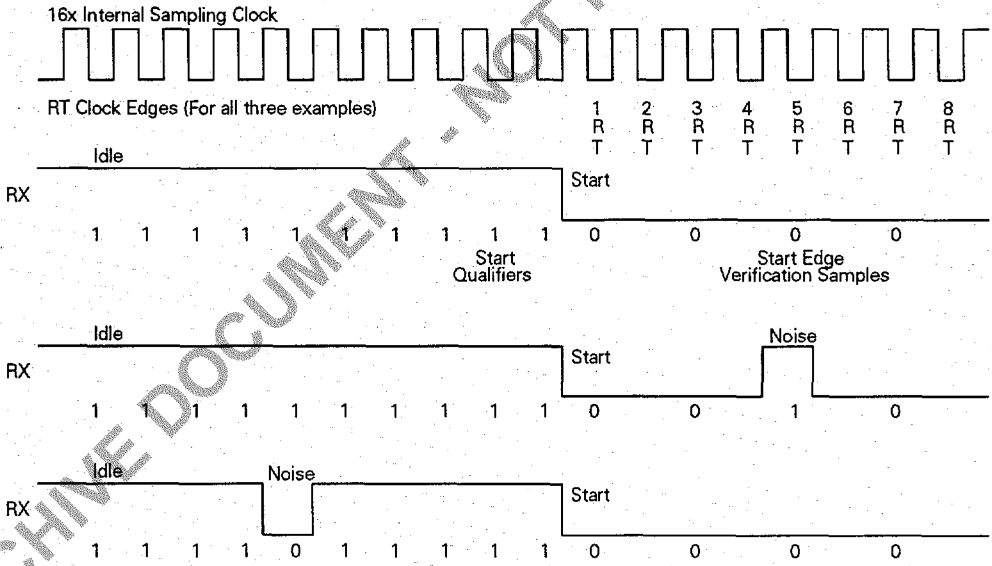
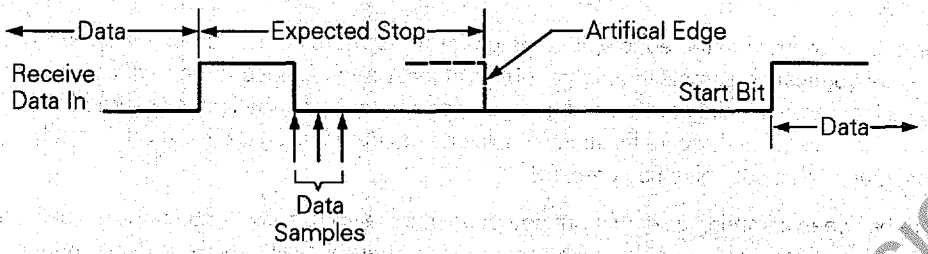
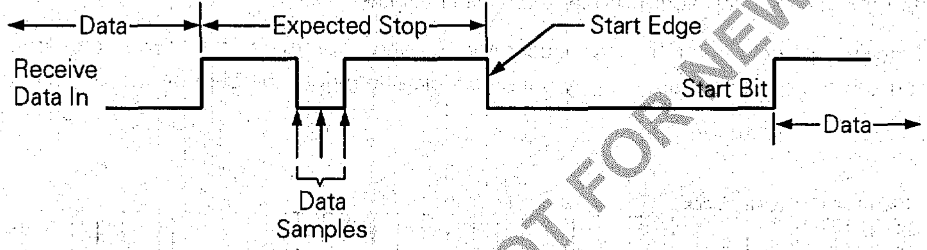


Figure 9-5 Examples of Start Bit Sampling Techniques



(a) Case 1, Receive Line Low During Artificial Edge



(b) Case 2, Receive Line High During Expected Start Edge

Figure 9-6 SCI Artificial Start Following a Framing Error

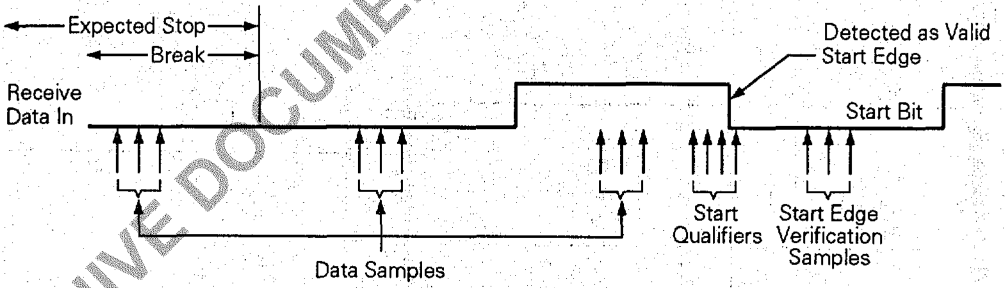


Figure 9-7 SCI Start Bit Following a Break

9.7 TRANSMIT DATA (TX)

Transmit data is the serial data from the internal data bus that is applied through the serial communications interface to the output line. The transmitter generates a bit time by using a derivative of the RT clock, thus producing a transmission rate equal to 1/16th that of the receiver sample clock.

9.8 SCI REGISTERS

Primarily the SCI system is configured and controlled by five registers; BAUD, SCCR1, SCCR2, SCSR, and SCDR. Reference should be made to the block diagram shown in Figure 9-1.

9.8.1 Serial Communications Data Register (SCDR)

The SCI data register (SCDR) shown in the following figure is actually two separate registers. When SCDR is read, the read-only receive data register is accessed and when SCDR is written, the write-only transmit data register is accessed.

	7	6	5	4	3	2	1	0	
$0027									SCDR
Reset	0	0	0	0	0	0	0	0	

9.8.2 Serial Communications Control Register 1 (SCCR1)

The SCI control register 1 (SCCR1) contains control bits related to the nine data bit character format and the receiver wake-up feature.

Note: Bits 5, 2, 1 and 0 are not implemented and always read as zeros.

	7	6	5	4	3	2	1	0	
$0024	R8	T8	0	M	WAKE	0	0	0	SCCR1
Reset	U	U	0	0	0	0	0	0	

9.8.2.1 R8 — Receive Data Bit 8

This read-only bit is the ninth serial data bit received when the SCI system is configured for nine data bit operation (M = 1). The most significant bit (bit 8) of the received character is transferred into this bit at the same time as the remaining eight bits (bits 0–7) are transferred from the serial receive shifter to the SCI receive data register.

9.8.2.2 T8 — Transmit Data Bit 8

This read/write bit is the ninth data bit to be transmitted when the SCI system is configured for nine data bit operation (M = 1). When the eight low order bits (bits 0–7) of a transmit character are transferred from the SCI data register to the serial transmit shift register, this bit (bit 8) is transferred to the ninth bit position of the shifter.

9.8.2.3 M — Mode (select character format)

The read/write M-bit controls the character length for both the transmitter and receiver at the same time. The 9th data bit is most commonly used as an extra stop bit or in conjunction with the "address mark" wake-up method. It can also be used as a parity bit.

1 (set) – start bit, 8 data, 9th data bit, 1 stop bit.

0 (clear) – start bit, 8 data bits, 1 stop bit.

9.8.2.4 WAKE — Wake-up Mode Select

The WAKE bit can be read or written to any time.

1 (set) – Wake-up on address mark.

0 (clear) – Wake-up on idle line.

9.8.3 Serial Communications Control Register 2 (SCCR2)

The SCI control register 2 (SCCR2) is a read/write register that provides the control bits which enable/disable individual SCI functions.

	7	6	5	4	3	2	1	0	
$0025	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	SCCR2
Reset	0	0	0	0	0	0	0	0	

9.8.3.1 TIE — Transmit Interrupt Enable

1 (set) – TDRE interrupts enabled

0 (clear) – TDRE interrupts disabled

9.8.3.2 TCIE — Transmit Complete Interrupt Enable

1 (set) – TC interrupts enabled

0 (clear) – TC interrupts disabled

9.8.3.3 RIE — Receiver Interrupt Enable

1 (set) – RDRF and OR interrupts enabled

0 (clear) – RDRF and OR interrupts disabled

9.8.3.4 ILIE — Idle Line Interrupt Enable

1 (set) – IDLE interrupts enabled.

0 (clear) – IDLE interrupts disabled.

9.8.3.5 TE — Transmitter Enable

1 (set) – Port E pin 1 is configured as the SCI1 transmit output pin TX1

0 (clear) – Port E pin 1 is configured as a standard I/O pin.

When the transmit enable bit is set, the transmit shift register output is applied to the TX line. Depending on the state of control bit M (SCCR1), a preamble of 10 (M = 0) or 11 (M = 1) consecutive ones is transmitted when software sets the TE bit from a cleared state. After loading the last byte in the serial communications data register and receiving the TDRE flag, the user can clear TE. Transmission of the last byte will then be completed before the transmitter gives up control of the TX pin. While the transmitter is active, the data direction register control for Port E bit 1 is overridden and the line is forced to be an output.

9.8.3.6 RE — Receiver Enable

1 (set) – Port E pin 0 is configured as the SCI1 receive input pin RX1

0 (clear) – Port E pin 0 is configured as a standard I/O pin.

When the receiver enable bit is set, the receiver is enabled. When RE is clear, the receiver is disabled and all the status bits associated with the receiver (RDRF, IDLE, OR, NF and FE) are inhibited. While the receiver is enabled, the data direction register control for Port E bit 0 is overridden and the line is forced to be an input.

9.8.3.7 RWU — Receiver Wake-up

When the receiver wake-up bit is set by the user software, it puts the receiver to sleep and enables the wake-up function. If the WAKE bit is cleared, RWU is cleared by the SCI logic after receiving 10 (M = 0) or 11 (M = 1) consecutive ones. If the WAKE bit is set, RWU is cleared by the SCI logic after receiving a data word whose MSB is set.

9.8.3.8 SBK — Send Break

If the send break bit is toggled set and cleared, the transmitter sends 10 (M = 0) or 11 (M = 1) zeros and then reverts to idle sending data. If SBK remains set, the transmitter will continually send whole blocks of zeros (sets of 10 or 11) until cleared. At the completion of the break code, the transmitter sends at least one high bit to guarantee recognition of a valid start bit. If the transmitter is currently empty and idle, setting and clearing SBK is likely to queue two character times of break because the first break transfers almost immediately to the shift register and the second is then queued into the parallel transmit buffer.

9.8.4 Serial Communications Status Register (SCSR)

The serial communications status register (SCSR) provides inputs to the interrupt logic circuits for generation of the SCI system interrupt.

	7	6	5	4	3	2	1	0	
$0026	TDRE	TC	RDRF	IDLE	OR	NF	FE	0	SCSR
Reset	1	1	0	0	0	0	0	0	

9.8.4.1 TDRE — Transmit Data Register Empty Flag

This bit is set when the byte in the transmit data register is transferred to the serial shift register. New data will not be transmitted unless the SCSR register is read before writing to the transmit data register. Reset sets this bit.

9.8.4.2 TC — Transmit Complete Flag

This bit is set to indicate that the SCI transmitter has no meaningful information to transmit (no data in shifter, no preamble, no break). When TC is set the serial line will go idle (continuous MARK). Reset sets this bit.

9.8.4.3 RDRF — Receive Data Register Full Flag

This bit is set when the contents of the receiver serial shift register are transferred to the receiver data register.

9.8.4.4 IDLE — Idle Line Detected Flag

This bit is set when a receiver idle line is detected (the receipt of a minimum of ten/eleven consecutive "1"s). This bit will not be set by the idle line condition when the RWU bit is set. Once cleared, IDLE will not be set again until after RDRF has been set, (until after the line has been active and becomes idle again).

9.8.4.5 OR — Overrun Error Flag

This bit is set when a new byte is ready to be transferred from the receiver shift register to the receiver data register and the receive data register is already full (RDRF bit is set). Data transfer is inhibited until this bit is cleared.

9.8.4.6 NF — Noise Error Flag

This bit is set if there is noise on a "valid" start bit, any of the data bits or on the stop bit. The NF bit is set during the same cycle as the RDRF bit but does not get set in the case of an overrun (OR).

9.8.4.7 FE — Framing Error Flag

This bit is set when the word boundaries in the bit stream are not synchronized with the receiver bit counter (generated by the reception of a logic zero bit where a stop bit was expected). The FE bit reflects the status of the byte in the receive data register and the transfer from the receive shifter to the receive data register is inhibited by an overrun. The FE bit is set during the same cycle as the RDRF bit but does not get set in the case of an overrun (OR). The framing error flag inhibits further transfer of data into the receive data register until it is cleared.

9.8.5 Baud Rate Register (BAUD)

The baud rate register (BAUD) is used to set the bit rate for the SCI system. Normally this register is written once, during initialization, to set the baud rate for SCI communications. Both the receiver and the transmitter use the same baud rate which is derived from the MCU bus rate clock. A two stage divider is used to develop custom baud rates from normal MCU crystal frequencies so it is not necessary to use special baud rate crystal frequencies.

	7	6	5	4	3	2	1	0	
$0023	–	0	SCP1	SCP0	–	SCR2	SCR1	SCR0	BAUD
Reset	–	0	0	0	–	U	U	U	

9.8.5.1 SCP1, SCP0 — Serial Prescaler Select bits

These read/write bits determine the factor by which the E-clock is divided as shown in Table 9-1. This prescaled output provides an input to a divider that is controlled by the SCI rate select bits (SCR2–SCR0).

Table 9-1 First Prescaler Stage

SCP1	SCP0	SCI Prescaler Division Ratio (N)
0	0	1
0	1	3
1	0	4
1	1	13

9.8.5.2 SCR2, SCR1, SCR0 — SCI Rate Select bits

These three read/write bits select the baud rates for both the transmitter and the receiver. The prescaler output described above is divided by the factors shown in Table 9-2.

Table 9-2 Second Prescaler Stage

SCR2	SCR1	SCR0	SCI Rate Select Division Ratio (M)
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

SECTION 10
M68000 INTERFACE MODULE

The M68000 interface module is designed to allow efficient communication between the MC68HC05i8 MCU and an M68000 bus. Throughout the following section, the MC68HC05i8 device is referred to as the as the "MCU" and the M68000 as the "Host". Within the module are four pairs of independent transmit and receive channels, each consisting of a 4-byte asynchronous buffer operating as a FIFO system (see Figure 10-1). The MCU or the Host can be interrupted, under software control, when the buffer is full or when a byte is received. Figure 10-1 shows the buffer system between the MCU and the Host. Further information on how data is transferred between the MCU and Host can be obtained from Section 10.4.

10.1 M68000 BUS INTERFACE PIN DESCRIPTIONS

10.1.1 $\overline{\text{DTACK}}$

The tri-state active-low open-drain Data Transfer Acknowledge ($\overline{\text{DTACK}}$) output signal is asserted during read and write cycles to indicate the proper transfer of data between the MCU and the Host.

10.1.2 $\overline{\text{IRQ68K}}$

The active-low open-drain output pin signals to the Host that one or more of the maskable interrupt conditions is true.

10.1.3 RS1–RS4

The register select lines select the MCU internal registers during read/write operations. These lines are usually driven by four of the M68000 address bus lines.

10.1.4 RW68K

The read/write input pin, RW68K, is driven high by the Host when it reads and low when it writes. Read or write cycles are initiated by asserting the chip select input ($\overline{\text{CS}}$).

10.1.5 CLK68K

When configured to support the M68000 interface module, Port E bit 7 serves as the CLK68K input. This pin selects the external clock mode of the M68000 subsystem whereby the $\overline{\text{DTACK}}$ pin is synchronised to an external clock signal applied to this pin.

10.1.6 $\overline{\text{CS}}$

The active low $\overline{\text{CS}}$ (Chip Select) input enables data transfers between the Host and MCU on data lines D0 to D7. Data transfers are controlled by RW68K, $\overline{\text{CS}}$ and the register select inputs RS1 to RS4. When $\overline{\text{CS}}$ is high the data lines D0 to D7 are placed in the high impedance state.

10.1.7 Data Bus Lines (D0–D7)

These bi-directional tri-state data lines are used to transfer commands, data and status between the M68000 processor and MCU. D0 is the least significant bit.

10.2 DATA ACKNOWLEDGE ($\overline{\text{DTACK}}$) DELAY

The total Host bus access time is the time from $\overline{\text{CS}}$ asserted to $\overline{\text{DTACK}}$ asserted. Since the Host bus accesses are asynchronous with respect to the MCU it is necessary to provide a defined maximum delay from $\overline{\text{CS}}$ asserted to $\overline{\text{DTACK}}$ asserted. To achieve this an on-chip clock delay circuit is incorporated.

This circuit is normally clocked by the internal MCU clock. However, by setting the CK68 bit in the Port E Mode Register to "1", the circuit will be clocked by an external clock applied to Port E pin 7. This clock delay circuit guarantees the total delay from $\overline{\text{CS}}$ to $\overline{\text{DTACK}}$ (t_{CSDT}) to be less than three clock cycles.

Note: For a write cycle, if $\overline{\text{CS}}$ is negated within one clock cycle of it being recognised (i.e. the first rising edge of the clock), then $\overline{\text{DTACK}}$ may not be generated. In this case data will be latched on the negating edge of $\overline{\text{CS}}$.

10.3 INTERRUPT ACKNOWLEDGE CYCLE

When the $\overline{\text{IRQ68K}}$ pin is asserted it indicates that an interrupt condition has occurred. This pin remains asserted until the Host responds to the interrupt and reads the data register of the channel which issued the interrupt.

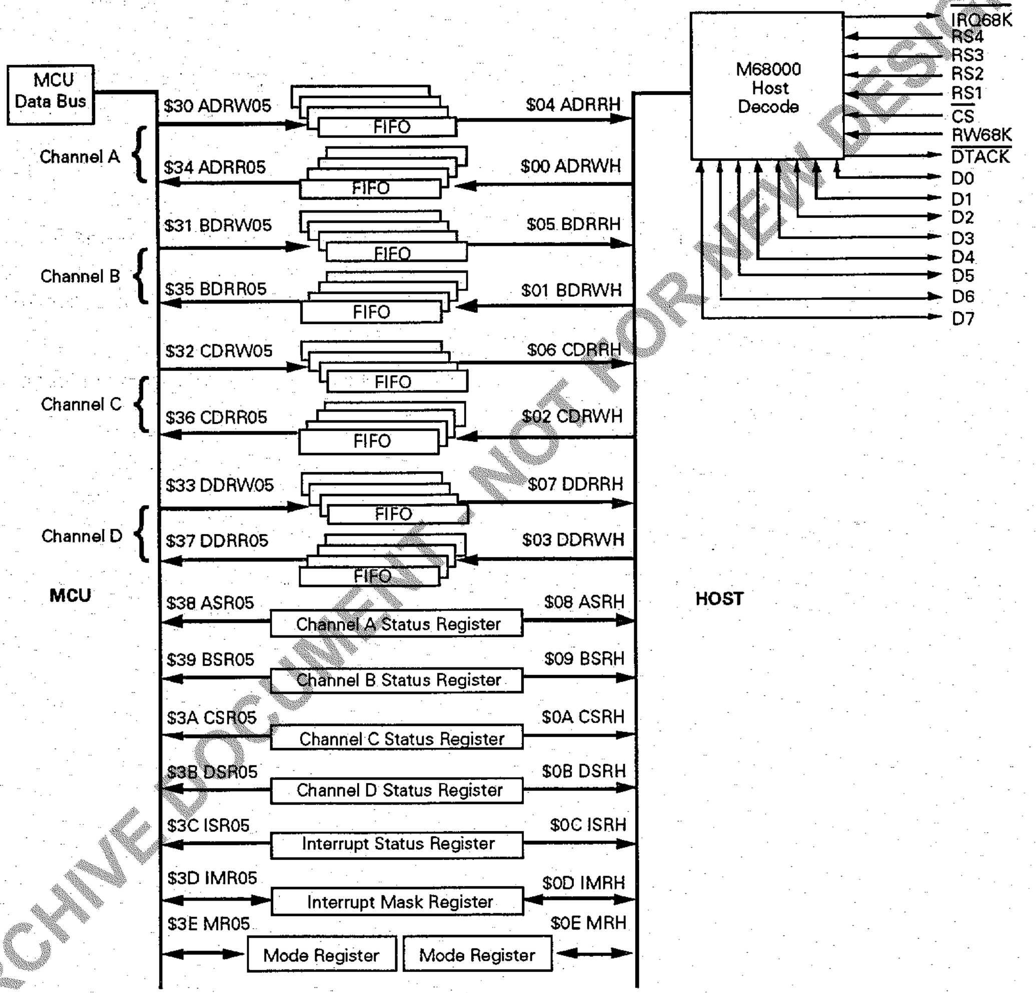


Figure 10-1 M68000 Interface Module block diagram

10.4 DATA AND CONTROL/STATUS REGISTERS

Communication between the MCU and the Host is achieved via four independent transmit channels (A–D) and four independent receive channels (A–D). Data is sent by one party to a transmit channel (Data Write Register) and can be read from the corresponding receive channel (Data Read Register) by the other party. Control of the data being transmitted and received is achieved via a set of seven control/status registers – Mode, Interrupt Mask, Interrupt Status and four Channel Status Registers (See Figure 10-1).

Each party has access to the control and status registers at separate address locations within its own memory map. With the exception of the Mode Registers, where there is one physical register for the MCU and another for the Host, all the registers are implemented as single registers simultaneously accessible by both the MCU and the Host. Within each register, each bit is implemented as a dual ported RAM bit and acts as a receive bit for one party and the corresponding transmit bit for the other party. In this way, both devices can access the same control and status information. For example, an action by the Host which sets a transmitter ready (TRDY) bit in one of its channel status registers, can be read immediately by the MCU as a receiver ready (RRDY) bit set in its corresponding channel status register.

The MCU control and data registers are mapped into the MC68HC05i8 memory map at location $0030 to $003E as shown in Table 10-1.

The Host registers are mapped into the M68000 address space as shown in Table 10-2 and can be accessed using the Register Select lines (RS1–RS4) on the M68000 interface module.

Note: In the following sections there is only one description for each physical register, or bit. Where applicable, the MCU name is given first, followed by the Host name.

Table 10-1 MCU Data and Control/Status Registers

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$0030									MCU Channel A Data Write Register (ADRWO5)
$0031									MCU Channel B Data Write Register (BDRWO5)
$0032									MCU Channel C Data Write Register (CDRWO5)
$0033									MCU Channel D Data Write Register (DDRWO5)
$0034									MCU Channel A Data Read Register (ADRRO5)
$0035									MCU Channel B Data Read Register (BDRRO5)
$0036									MCU Channel C Data Read Register (CDRRO5)
$0037									MCU Channel D Data Read Register (DDRRO5)
$0038	ROR	RRDY	RFULL	REMTY	TOR	TRDY	TFULL	TEMTY	MCU Channel A Status Register (ASR05)
$0039	ROR	RRDY	RFULL	REMTY	TOR	TRDY	TFULL	TEMTY	MCU Channel B Status Register (BSR05)
$003A	ROR	RRDY	RFULL	REMTY	TOR	TRDY	TFULL	TEMTY	MCU Channel C Status Register (CSR05)
$003B	ROR	RRDY	RFULL	REMTY	TOR	TRDY	TFULL	TEMTY	MCU Channel D Status Register (DSR05)
$003C	RA	TA	RB	TB	RC	TC	RD	TD	MCU Interrupt Status Register (ISRO5)
$003D	RMA	TMA	RMB	TMB	RMC	TMC	RMD	TMD	MCU Interrupt Mask Register (IMRO5)
$003E	IMOD	0	0	0	AEN	BEN	CEN	DEN	MCU Mode Register (MRO5)
$003F									Unused

Table 10-2 Host Data and Control/Status Registers

RS4 - RS1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
0000									Host Channel A Data Write (ADRWH)
0001									Host Channel B Data Write (BDRWH)
0010									Host Channel C Data Write (CDRWH)
0011									Host Channel D Data Write (DDRWH)
0100									Host Channel A Data Read (ADRRH)
0101									Host Channel B Data Read (BDRRH)
0110									Host Channel C Data Read (CDRRH)
0111									Host Channel D Data Read (DDRRH)
1000	ROR	RRDY	RFULL	REMTY	TOR	TRDY	TFULL	TEMTY	Host Channel A Status (ASRH)
1001	ROR	RRDY	RFULL	REMTY	TOR	TRDY	TFULL	TEMTY	Host Channel B Status (BSRH)
1010	ROR	RRDY	RFULL	REMTY	TOR	TRDY	TFULL	TEMTY	Host Channel C Status (CSRH)
1011	ROR	RRDY	RFULL	REMTY	TOR	TRDY	TFULL	TEMTY	Host Channel D Status (DSRH)
1100	RA	TA	RB	TB	RC	TC	RD	TD	Host Interrupt Status (ISRH)
1101	RMA	TMA	RMB	TMB	RMC	TMC	RMD	TMD	Host Interrupt Mask (IMRH)
1110	IMOD	FCLR	0	0	AEN	BEN	CEN	DEN	Host Mode (MRH)
1111									Unused

10.4.1 MCU Channel Data Write Registers (ADRW05–DDRW05)
MCU Channel Data Read Registers (ADRR05–DDRR05)
Host Channel Data Write Registers (ADRWH–DDRWH)
Host Channel Data Read Registers (ADRRH–DDRRH)

When the MCU wishes to send a byte of data via channel A, say, it writes that information to its Channel Data Write Register (ADRW05). This information can then be read by the Host from its corresponding Channel A Data Read Register (ADRRH). Similarly, data sent from the Host to the MCU via channel C, say, would be written by the Host to its Channel C Data Write Register (CDRWH) and read by the MCU from its corresponding Channel C Data Read Register (CDRR05).

The Read registers and the Write registers are implemented as 4-stage FIFO locations and can therefore queue four bytes of data at any one time. Each byte wide FIFO register has a transmit counter and a receive counter which point at the next available location to be either written to or read from. In this way the data is always transmitted and received in the same order. It is the information transferred in these channels which affects the bits and flags of the Control and Status Registers.

10.4.2 MCU Channel Status Registers (ASR05–DSR05)
Host Channel Status Registers (ASRH–DSRH)

Each pair of transmit/receive channels has its own Channel Status Register shared between the MCU and the Host. This register comprises four transmit status bits and four receive status bits, all of which can be read by both the MCU and the Host. Figure 10-2 shows how these bits are used by both the MCU and the Host. Note that in Figure 10-2, only the MCU Channel Status Register (xSR05) bits are shown in their correct sequence.

The four bits interpreted as receiver status bits by the MCU (ROR, RRDY, RFULL and REMTY) are the same four bits seen by the Host as the transmitter status bits (TOR, TRDY, TFULL and TEMTY), and vice versa. Therefore, when the MCU transmits, the status of that transmission is available to the MCU in the transmitter status bits of the xSR05 and to the Host in the receiver status bits of the xSRH (see Figure 10-2). Similarly, when the Host transmits, the status of that transmission is available to the Host in the transmitter status bits of the xSRH and to the MCU in the receiver status bits of the xSR05.

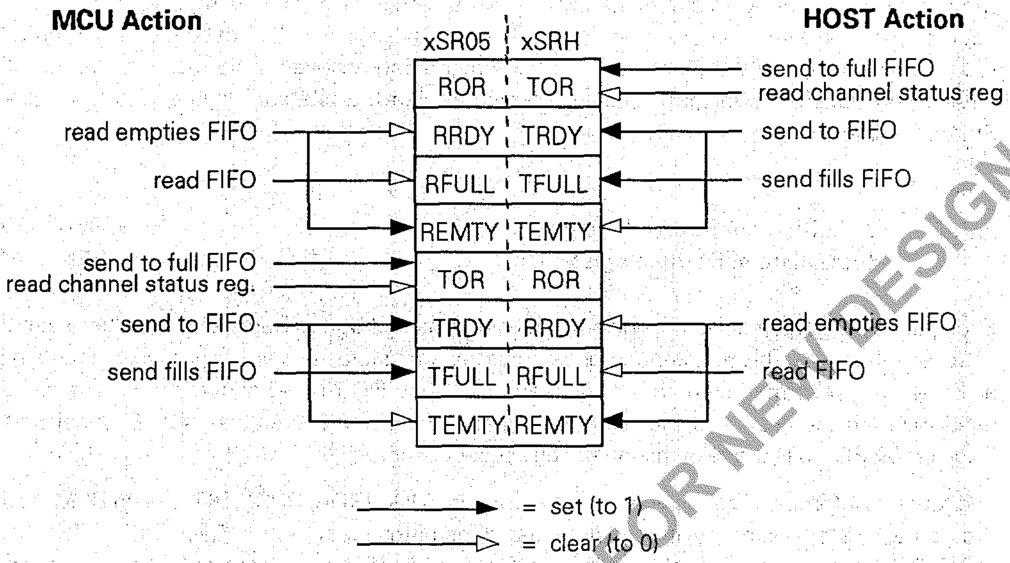
	7	6	5	4	3	2	1	0	
	ROR	RRDY	RFULL	REMTY	TOR	TRDY	TFULL	TEMTY	Channel Status Register A–D
Reset	0	0	0	1	0	0	0	1	

ROR — Receiver Overrun Error (MCU) Transmitter Overrun Error (Host) — TOR

This bit is set when the Host sends a byte to a FIFO that is already full. It is cleared on reset or when the Host reads its corresponding Channel Status Register.

1 (set) – Overrun has occurred; one or more bytes in the data stream has been lost.

0 (clear) – No overrun has occurred.



Note: All bits are read-only. This figure also illustrates how the bits are set and cleared by various MCU and Host actions.

Figure 10-2 Status Register

RRDY — Receiver Ready (MCU) **Transmitter Ready (Host) — TRDY**

This bit is set when a byte is transferred from the Host to the FIFO, and is waiting to be read by the MCU. It is cleared when the MCU reads the last byte of information from the FIFO, leaving it empty.

1 (set) – A byte is waiting in the FIFO to be read by the MCU.

0 (clear) – No byte waiting; FIFO is empty.

RFULL — Receiver Full (MCU) **Transmitter Full (Host) — TFULL**

This bit is set when a byte is transferred by the Host to the FIFO and the transfer causes the buffer to become full.

1 (set) – All four FIFO register positions are occupied; FIFO is full.

0 (clear) – FIFO is not full.

REMTY — Receiver Empty (MCU) Transmitter Empty (Host) — TEMTY

This bit is the complement of Receiver Ready. It is set on reset or when the MCU reads the FIFO, leaving it empty. It is cleared when the Host transfers a byte to the FIFO. If an attempt is made to read a FIFO when it is empty, the data will be undefined.

1 (set) – FIFO is empty.

0 (clear) – FIFO contains data.

TOR — Transmitter Overrun Error (MCU) Receiver Overrun Error (Host) — ROR

This bit is set when the MCU sends a byte to a FIFO that is already full. It is cleared on reset or when the MCU reads the corresponding Channel Status Register.

1 (set) – Overrun has occurred; one or more bytes in the data stream have been lost.

0 (clear) – No overrun has occurred.

TRDY — Transmitter Ready (MCU) Receiver Ready (Host) — RRDY

This bit is set when a byte is transferred by the MCU to the FIFO. It is cleared on reset or when the Host reads the last byte of information from the FIFO, leaving it empty.

1 (set) – A byte is waiting in the FIFO to be read by the Host.

0 (clear) – No byte waiting; FIFO is empty

TFULL — Transmitter Full (MCU) Receiver Full (Host) — RFULL

This bit is set when a byte is transferred by the MCU to the FIFO and the transfer causes the buffer to become full. It is cleared on reset or when the Host reads the FIFO.

1 (set) – All four FIFO register positions are occupied.

0 (clear) – FIFO is not full.

TEMTY — Transmitter Empty (MCU) Receiver Empty (Host) — REMTY

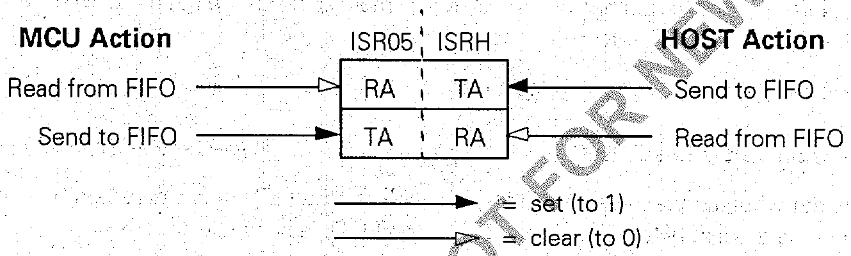
This bit is the complement of Transmitter Ready. It is set on reset or when the Host reads the transmit FIFO, leaving it empty. It is cleared when the MCU transmits a byte to the FIFO. If an attempt is made to read a FIFO when it is empty, the data will be undefined.

1 (set) – FIFO is empty.

0 (clear) – FIFO contains data.

10.4.3 MCU Interrupt Status Register (ISR05)
Host Interrupt Status Register (ISRH)

This read-only register serves as the MCU Interrupt Status Register (ISR05) and the Host Interrupt Status Register (ISRH). Just like the Channel Status Register bits described earlier in this section, the Interrupt Status Register bits seen by the MCU in ISR05 as receiver bits are seen by the Host as transmitter bits in ISRH, and vice versa. Figure 10-3 shows the relationship between the transmit interrupt status bit TA and the receive interrupt status bit RA as seen by the MCU and by the Host. These bits are set and cleared according to the status of the transmit and receive FIFOs and the criterion selected by the IMOD bit in the MCU and Host Mode Registers (see Sections 10.4.5 and 10.4.6).



Note: The MCU IMOD bit selects the criterion for setting and clearing the MCU RA/Host TA bit. The HOST IMOD bit selects the criterion for setting and clearing the Host RA/MCU TA bit. All bits are read only.

Figure 10-3 Interrupt Status Register

	7	6	5	4	3	2	1	0	
	RA	TA	RB	TB	RC	TC	RD	TD	Interrupt Status Register
Reset	0	0	0	0	0	0	0	0	

RA–RD — Receiver A–D (MCU) Transmitter A–D (Host) — TA–TD

When the IMOD bit in the MCU Mode Register is set, the Rx/Tx bit will be set when RRDY/TRDY is set, i.e. a byte is transmitted from the Host to the FIFO. It is cleared when the MCU reads the last byte of information from the FIFO, leaving it empty.

When the IMOD bit in the MCU Mode Register is clear, the Rx/Tx bit will be set when RFULL/TFULL is set, i.e. a byte is transmitted by the Host to the FIFO and the transfer causes the FIFO to become full.

In both instances, Rx/Tx set indicates that data has been received on channel x and is waiting in the FIFO to be read by the MCU. This information is used in conjunction with the mask bits (RMA-RMD/TMA-TMD) in the Interrupt Mask Register to determine whether or not an interrupt will be issued to the MCU as a result of the data being received. Further explanation of how this register operates is given in the Interrupt Mask Register description.

MC68HC05i8

TA–TD — Transmitter A–D (MCU) Receiver A–D (Host) — RA–RD

When the IMOD bit in the Host Mode Register is set, the Tx/Rx bit will be set when TRDY/RRDY is set, i.e., when a byte has been transmitted by the MCU to the FIFO. It is cleared when the Host reads the last byte of information from the FIFO, leaving it empty.

When the IMOD bit in the MCU Mode Register is clear, the Tx/Rx bit will be set when TFULL/RFULL is set, i.e., when a byte is transmitted by the MCU and the transfer causes the FIFO to become full.

In both instances, Tx set indicates that a byte has been transmitted on channel x by the MCU and is waiting in the transmitter FIFO to be read by the Host. This information is used in conjunction with the mask bits (TMA-TMD/RMA-RMD) in the Interrupt Mask Register to determine whether or not an interrupt will be issued to the Host as a result of the byte being transmitted by the MCU. Further explanation of how this register operates is given in the Interrupt Mask Register description.

10.4.4 MCU Interrupt Mask Register (IMR05)
Host Interrupt Mask Register (IMRH)

This read/write register serves as the Host Interrupt Mask Register (IMRH) and the MCU Interrupt Mask Register (IMR05). As with the Channel Status Register bits and the Interrupt Status Register bits, the Interrupt Mask Register bits seen by the MCU in IMR05 as Receiver Interrupt Mask bits are seen by the Host in IMRH as Transmitter Interrupt Mask bits, and vice versa. As an example, Figure 10-4 shows this relationship between the Receiver Mask bit, RMA, and the Transmitter Mask bit, TMA, as seen by the MCU and by the Host.

As shown in Figure 10-4, the Receiver Mask bits can be set or cleared only by the device acting as the receiver on that channel. The Transmit Mask bits are set or cleared as a result of the other device setting or clearing its own Receiver Mask bits.

	7	6	5	4	3	2	1	0	
	RMA	TMA	RMB	TMB	RMC	TMC	RMD	TMD	Interrupt Mask Register
Reset	0	0	0	0	0	0	0	0	

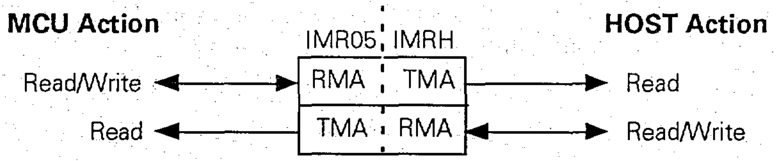


Figure 10-4 Interrupt Mask Register

RMA–RMD — Receive Mask A–D (MCU) Transmit Mask A–D (Host) — TMA–TMD

The RMx bits are the read/write interrupt mask bits for the four receive channels. The RMx bits should be set if an MCU interrupt is required whenever the Rx bit becomes set, i.e. on receipt of data on channel x. When RMx is set, the corresponding read-only TMx bit in the Host Interrupt Mask Register is set, thus informing the Host that transmitting data on this channel can interrupt the MCU.

1 (set) – An MCU interrupt will be asserted when the corresponding Rx flag in the ISR05 becomes set.

0 (clear) – No MCU interrupt will be asserted regardless of the state of the Rx flag.

TMA–TMD — Transmit Mask A–D (MCU) Receive Mask A–D (Host) — RMA–RMD

The TMx bits are the read-only interrupt mask bits for the four transmit channels. The TMx bit in the MCU Interrupt Mask Register is set when the corresponding RMx bit in the Host Interrupt Mask Register is set. These bits indicate whether the Host will be interrupted when the MCU transmits data on a certain channel.

1 (set) – Transmitting data on channel x will cause a Host interrupt to be generated when the Rx flag in the Host Interrupt Status Register becomes set.

0 (clear) – No Host interrupt will be asserted regardless of the state of the Rx flag in the Host Interrupt Status Register.

10.4.5 MCU Mode Register (MR05)

	7	6	5	4	3	2	1	0	
$003E	IMOD	0	0	0	AEN	BEN	CEN	DEN	MCU Mode Register
Reset	0	0	0	0	0	0	0	0	

IMOD — Interrupt Mode

This bit controls the criteria by which the MCU Interrupt Status Register (ISR05) flags are set.

1 (set) – Rx and Tx bits in ISR05 will be set according to the "Receiver Ready" and "Transmitter Ready" criteria respectively.

0 (clear) – Rx and Tx bits in ISR05 will be set according to the "Receiver Full" and "Transmitter Full" criteria respectively.

AEN, BEN, CEN and DEN — Channel A, B, C and D Enable

In order to read from or write to a channel its enable bit (xEN) must be set.

1 (set) – Channel x is enabled.

0 (clear) – Channel x is disabled.

10.4.6 Host Mode Register (MRH)

	7	6	5	4	3	2	1	0	
$000E	IMOD	FCLR	0	0	AEN	BEN	CEN	DEN	Host Mode Register
Reset	0	0	0	0	0	0	0	0	

IMOD — Interrupt Mode

This bit controls the criteria by which the Host Interrupt Status Register (ISRH) flags are set.

1 (set) – Rx and Tx bits in ISRH will be set according to the "Receiver Ready" and "Transmitter Ready" criteria respectively.

0 (clear) – Rx and Tx bits in ISRH will be set according to the "Receiver Full" and "Transmitter Full" criteria respectively.

FCLR — FIFO Data Clear

Setting this bit clears all data channels. When the channels are cleared, the FIFO read and write pointers are set to zero for both directions and the channels are returned to their initial state. The control registers are unaffected by FCLR.

Note: FIFO clear (FCLR) can only be done from the Host. There is no FCLR bit in the MCU Mode Register.

AEN, BEN, CEN and DEN — Channel A, B, C and D Enable

To read from or write to a channel, its enable bit (xEN) must be set.

1 (set) – Channel x is enabled.

0 (clear) – Channel x is disabled.

SECTION 11
ELECTRICAL SPECIFICATIONS

This section contains the electrical specifications and associated timing information for the MC68HC05i8.

11.1 MAXIMUM RATINGS

Table 11-1 Maximum Ratings

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
Operating Temperature Range	T_A	T_L to T_H 0 to 70	°C
Storage Temperature Range	T_{stg}	– 65 to +150	°C
Current Drain per pin (note 2) – excluding VDD and VSS	I_D	25	mA

Note 1) All voltages are with respect to V_{SS}.

Note 2) Maximum current drain per pin is for one pin at a time, limited by an external resistor.

Note 3) This device contains circuitry designed to protect against damage due to high electrostatic voltages or electric fields. However, it is recommended that normal precautions be taken to avoid the application of any voltages higher than those given in the Maximum Ratings table to this high impedance circuit. For maximum reliability all unused inputs should be tied to either V_{SS} or V_{DD}.

11.2 THERMAL CHARACTERISTICS AND POWER CONSIDERATIONS

Table 11-2 Package Thermal Characteristics

Characteristics	Symbol	Value	Unit
Thermal Resistance – Plastic 64-pin Quad Flat Pack (QFP)	θ_{JA}	50	°C/W

The average chip junction temperature, T_J, in degrees Celsius can be obtained from the following equation:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \qquad [1]$$

where:

T_A = Ambient Temperature (°C)

θ_{JA} = Package Thermal Resistance, Junction-to-ambient (°C/W)

$P_D = P_{INT} + P_{I/O}$ (W)

P_{INT} = Internal Chip Power = $I_{DD} \bullet V_{DD}$ (W)

$P_{I/O}$ = Power Dissipation on Input and Output pins (User determined)

An approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is:

$$P_D = \frac{K}{T_J + 273} \qquad [2]$$

Solving equations [1] and [2] for K gives:

$$K = P_D \bullet (T_A + 273) + \theta_{JA} \bullet P_D^2 \qquad [3]$$

where K is a constant for a particular part. K can be determined by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained for any value of T_A by solving the above equations. The package thermal characteristics are shown in Table 11-2.

Pins	R1	R2	C
All Port Pins, D0–D7, $\overline{\text{IRQ68K}}$ & $\overline{\text{DTACK}}$	3.26kΩ	2.38kΩ	50pF

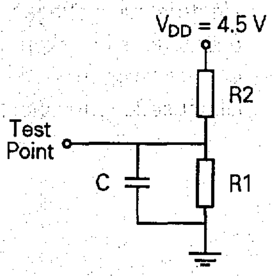


Figure 11-1 Equivalent Test Load

11.3 DC ELECTRICAL CHARACTERISTICS

Table 11-3 DC Electrical Characteristics

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = T_L to T_H)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage					
I_{LOAD} = – 10 µA	V_{OH}	V_{DD} – 0.1	—	—	V
I_{LOAD} = +10 µA	V_{OL}	—	—	0.1	V
Output High Voltage (I_{LOAD} = – 0.8 mA) Port A – Port F	V_{OH}	V_{DD} – 0.8	—	—	V
Output Low Voltage (I_{LOAD} = +1.6 mA) Port A – Port F	V_{OL}	—	—	0.4	V
Input High Voltage All Ports, OSC1, $\overline{\text{IRQ}}$ & $\overline{\text{RESET}}$	V_{IH}	0.7V_{DD}	—	—	V
Input Low Voltage All Ports, OSC1, $\overline{\text{IRQ}}$ & $\overline{\text{RESET}}$	V_{IL}	—	—	0.2V_{DD}	V
Output High Voltage (I_{LOAD} = –0.8 mA) D0–D7	V_{OH}	V_{DD} – 0.8	—	—	V
Output Low Voltage (I_{LOAD} = +1.6 mA) D0–D7, OSC1, $\overline{\text{IRQ68K}}$ & $\overline{\text{DTACK}}$	V_{OL}	—	—	0.4	V
Input High Voltage D0–D7, RS1–RS4, $\overline{\text{CS}}$ & $\overline{\text{RW68K}}$	V_{IH}	0.7V_{DD}	—	—	V
Input Low Voltage D0–D7, RS1–RS4, $\overline{\text{CS}}$ & $\overline{\text{RW68K}}$	V_{IL}	—	—	0.2V_{DD}	V
Supply Current (see Notes)	I_{DD}				
RUN		—	TBD	TBD	mA
WAIT		—	TBD	TBD	mA
STOP		—	TBD	TBD	µA
High-Z Leakage Current Port A – Port F	I_{IL}	—	—	±10	µA
Input Current $\overline{\text{RESET}}$, OSC1 & $\overline{\text{IRQ}}$	I_{IN}	—	—	±1	µA
Capacitance					
Ports (as input or output)	C_{OUT}	—	—	12	pF
$\overline{\text{RESET}}$, $\overline{\text{IRQ}}$	C_{IN}	—	—	8	pF
Hysteresis $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$	V_{HYST}	—	1.0	—	V

Note 1) All I_{DD} measurements taken with suitable decoupling capacitors across the power supply to suppress the transient switching currents inherent in CMOS designs (see Section 2.1).

Note 2) Typical values are at mid point of voltage range and at 25°C only.

Note 3) RUN and WAIT I_{DD}: measured using an external square-wave clock source (f_{OP} = 2MHz); all inputs 0.2 V from rail; no DC loads; maximum load on outputs 50pF (except OSC2 load 20pF).

Note 4) WAIT I_{DD}: only the timer system active; current varies linearly with the OSC2 capacitance.

Note 5) STOP and WAIT I_{DD}: all ports configured as inputs; V_{IL} = 0.2 V and $V_{IH} = V_{DD} - 0.2$ V.

Note 6) STOP I_{DD}: measured with OSC1 = V_{DD}.

11.4 AC ELECTRICAL CHARACTERISTICS

Table 11-4 AC Electrical Characteristics

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Crystal External Clock	 f_{OSC} f_{OSC}	 — dc	 8.0 16.0	 MHz
Internal Operating Frequency	f_{OP}	dc	2.0	MHz
Processor cycle time	t_{CYC}	250	—	ns
Crystal Oscillator startup time	t_{OXOV}	—	100	ms
$\overline{\text{RESET}}$ pulse width	t_{RL}	1.5	—	t_{CYC}
Power-on Reset delay	t_{PORL}	4064	4064	t_{CYC}
Interrupt Pulse Width Low (edge-triggered)	t_{ILIH}	125	—	ns
Interrupt Pulse Period	t_{ILIL}	**	—	t_{CYC}
OSC1 Pulse Width	t_{OH}, t_{OL}	90	—	ns

** The minimum period t_{ILIL} should not be fewer than the number of cycles it takes to execute the interrupt service routine plus 21 t_{cyc}.

11.5 M68000 INTERFACE MODULE – AC ELECTRICAL CHARACTERISTICS

11.5.1 Host Read Cycle Bus Timing

Table 11-5 Read Cycle Bus Timing

Characteristic	Symbol	Min	Max	Unit
$\overline{\text{CS}}$ Setup Time to CLK68K High	t_{CSC}	90	—	ns
RS1–RS4 Setup Time to $\overline{\text{CS}}$ Asserted	t_{RSS}	10	—	ns
RW68K Setup Time to $\overline{\text{CS}}$ Asserted	t_{RWS}	10	—	ns
$\overline{\text{CS}}$ Pulse Width Asserted	t_{CSWL}	130	—	ns
Data Valid from $\overline{\text{CS}}$ Asserted	t_{DD}	—	100	ns
$\overline{\text{DTACK}}$ Asserted from CLK68K High	t_{DCR}	—	50	ns
$\overline{\text{CS}}$ Asserted to $\overline{\text{DTACK}}$ Asserted	t_{CSDT}	—	3	t_{CYC}
$\overline{\text{CS}}$ Negated from $\overline{\text{DTACK}}$ Asserted	t_{CSD}	20	—	ns
RS1–RS4 Hold Time from $\overline{\text{CS}}$ Negated	t_{RSH}	0	—	ns
RW68K Hold Time from $\overline{\text{CS}}$ Negated	t_{RWH}	0	—	ns
Data Hold Time from $\overline{\text{CS}}$ Negated	t_{DH}	0	—	ns
Data Bus floating from $\overline{\text{CS}}$ Negated	t_{DF}	—	50	ns
$\overline{\text{DTACK}}$ Negated from $\overline{\text{CS}}$ Negated	t_{DAH}	—	50	ns
$\overline{\text{DTACK}}$ Hi-Z from $\overline{\text{CS}}$ Negated	t_{DAT}	—	75	ns
$\overline{\text{CS}}$ Negated Pulse Width	t_{CSWH}	70	—	ns
CLK68K frequency	t_{CLK}	—	10	MHz

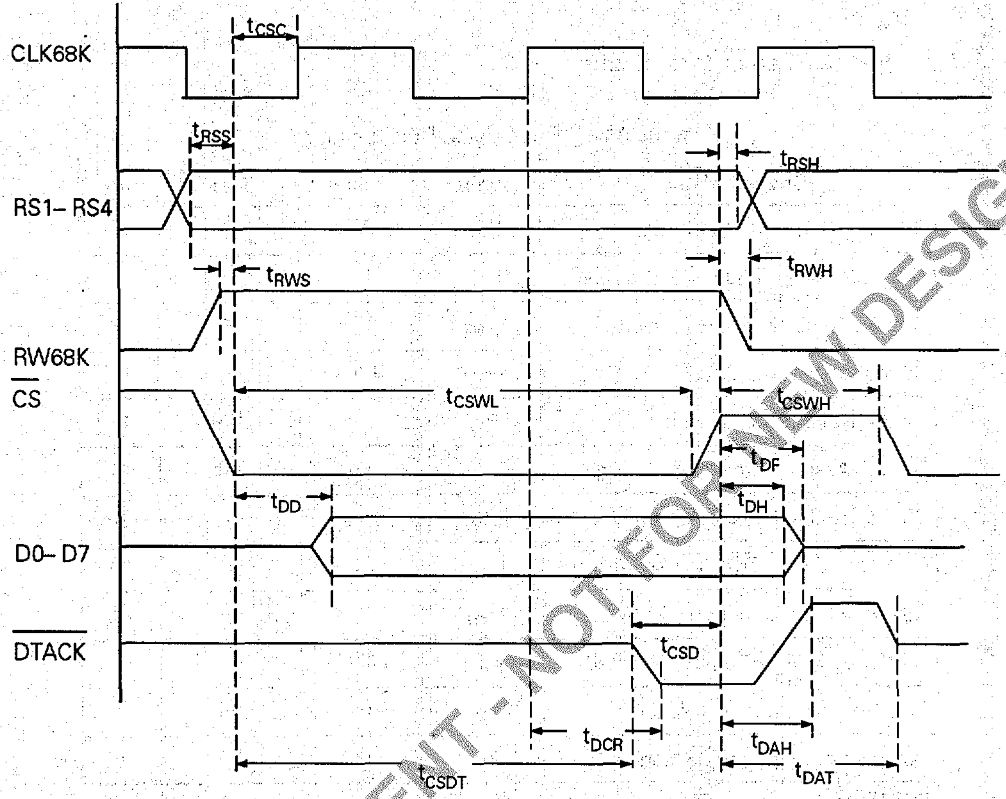


Figure 11-2 Read Cycle Bus Timing Diagram

11.5.2 Host Write Cycle Bus Timing

Table 11-6 Write Cycle Bus Timing

Characteristic	Symbol	Min	Max	Unit
$\overline{CS}$ Setup Time to CLK68K High	t_{CSC}	90	—	ns
RS1–RS4 Setup Time to $\overline{CS}$ Asserted	t_{RSS}	10	—	ns
RW68K Setup Time to $\overline{CS}$ Asserted	t_{RWS}	10	—	ns
$\overline{CS}$ Pulse Width Asserted	t_{CSWL}	130	—	ns
Data Setup time to CLK68K High	t_{DS}	50	—	ns
$\overline{DTACK}$ Asserted from CLK68K High	t_{DCW}	—	50	ns
$\overline{CS}$ Asserted to $\overline{DTACK}$ Asserted	t_{CSDT}	—	3	t_{CYC}
$\overline{CS}$ Negated from $\overline{DTACK}$ Asserted	t_{CSD}	20	—	ns
RS1–RS4 Hold Time from $\overline{CS}$ Negated	t_{RSH}	0	—	ns
RW68K Hold Time from $\overline{CS}$ Negated	t_{RWH}	0	—	ns
Data Hold Time from $\overline{CS}$ Negated	t_{DH}	10	—	ns
Data Setup time to $\overline{CS}$ Negated	t_{DSCS}	—	50	ns
$\overline{DTACK}$ Negated from $\overline{CS}$ Negated	t_{DAH}	—	50	ns
$\overline{DTACK}$ Hi-Z from $\overline{CS}$ Negated	t_{DAT}	—	75	ns
$\overline{CS}$ Negated Pulse Width	t_{CSWH}	70		ns
CLK68K frequency	t_{CLK}	—	10	MHz

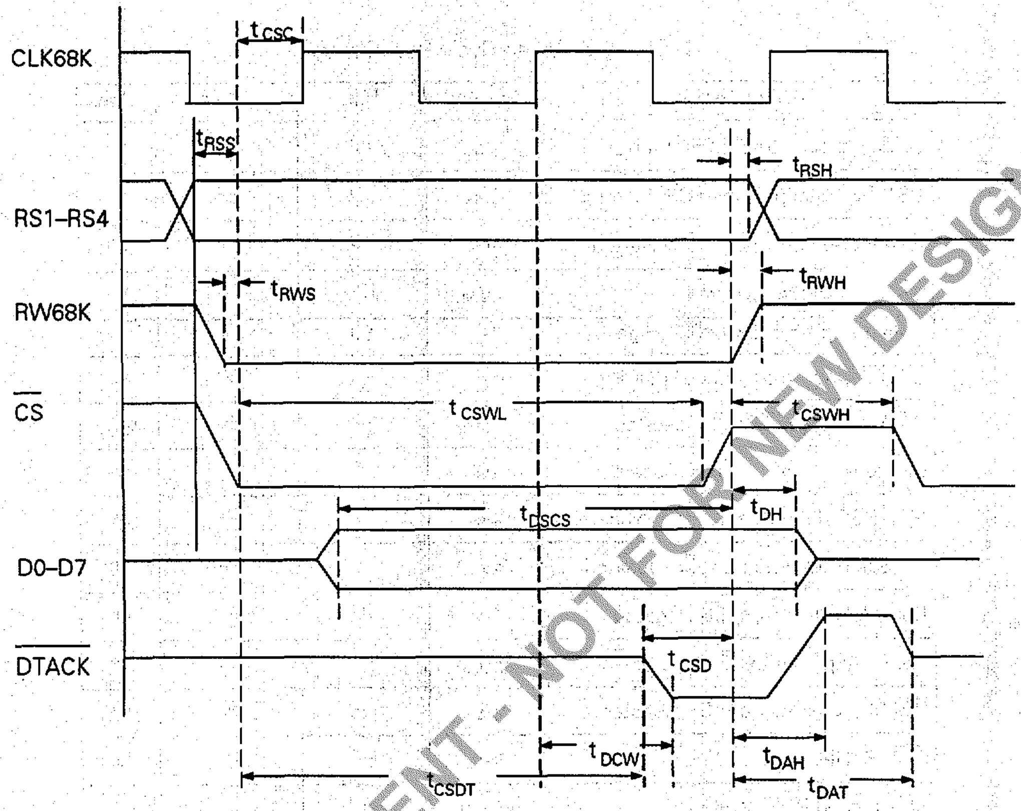


Figure 11-3 Write Cycle Bus Timing Diagram

SECTION 12
MECHANICAL DATA AND ORDERING INFORMATION

12.1 64-PIN QFP PACKAGE

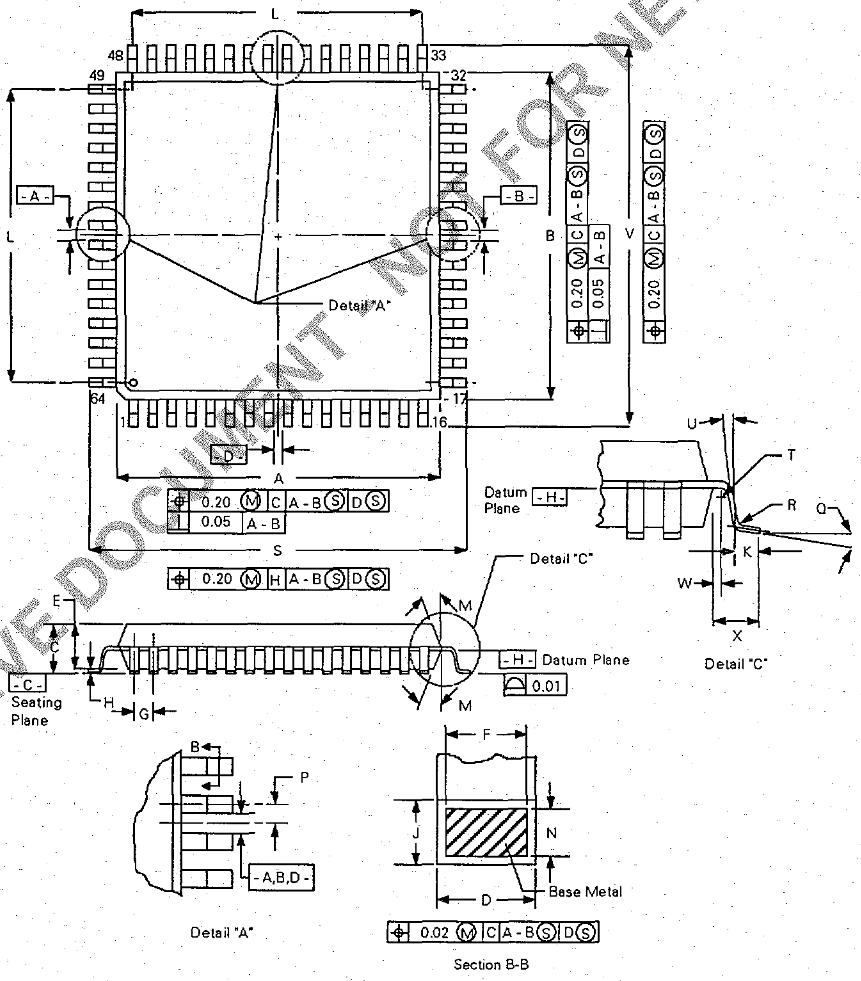


Figure 12-1 64-pin QFP Mechanical Specifications

Table 12-1 64-pin QFP Dimensions

Dim	Min	Max
A	13.95	14.10
B	13.95	14.10
C	2.217	2.457
D	0.30	0.40
E	2.15	2.25
F	0.30	—
G	0.80 BSC	
H	0.067	0.207
J	0.168	0.173
K	0.50	0.66
L	12.00 REF	
M	6°	8°
N	0.143	0.157
P	0.40 BSC	
Q	2°	8°
R	0.13	0.30
S	16.20	16.60
T	0.15	0.25
U	9°	15°
V	16.20	16.60
W	0.37	0.47
X	1.2 REF	

Notes

1. Dimensions and tolerancing per ANSI Y14.5M, 1982.
2. All dimensions in millimetres.
3. Datum Plane – H – is located at bottom of lead and is coincident with the lead where the lead exits the plastic body at the bottom of the parting line.
4. Datums A – B and – D – to be determined at Datum Plane – H –.
5. Dimensions S and V to be determined at Seating Plane – C –.
6. Dimensions A and B do not include mould protrusion. Allowable protrusion is 0.25mm per side. Dimensions A and B do include mould mismatch and are determined at Datum Plane – H –.
7. Dimension D does not include dambar protrusion. Allowable dambar protrusion shall be 0.08mm total in excess of the D dimension at maximum material condition. Dambar cannot be located on the lower radius or the foot.

12.2 ORDERING INFORMATION

This section describes the information needed to order the MCU.

To initiate a ROM pattern for the MCU, it is necessary to first contact your local field service office, local sales person or Motorola representative. Please note that you will need to supply details such as: mask option selections; temperature range; oscillator frequency; package type; electrical test requirements; and device marking details so that an order can be processed, and a customer specific part number allocated.

12.2.1 EPROMs

A 16 kbyte EPROM programmed with the customer's software (positive logic for address and data) should be submitted for pattern generation. All unused bytes should be programmed to zeros.

The EPROM should be clearly labelled, placed in a conductive IC carrier and securely packed.

12.2.2 Verification media

All original pattern media (EPROMs) are filed for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned with a listing verification form. The listing should be thoroughly checked and the verification form completed, signed and returned to Motorola. The signed verification form constitutes the contractual agreement for creation of the custom mask. If desired, Motorola will program blank EPROMs (supplied by the customer) from the data file used to create the custom mask, to aid in the verification process.

12.2.3 ROM Verification Units (RVUs)

Ten MCUs containing the customer's ROM pattern will be provided for program verification. These units will have been made using the custom mask but are for ROM verification only. For expediency, they are usually unmarked and are tested only at room temperature (25°C) and at 5 Volts. These RVUs are included in the mask charge and are not production parts. They are neither backed nor guaranteed by Motorola Quality Assurance.

12.2.4 MC Order Numbers

Package Type	Temperature	MC Order Number
Quad Flat Pack (QFP)	0 to 70°C	MC68HC05i8FU

– Cut along this line to remove –

CUSTOMER FEEDBACK QUESTIONNAIRE — MC68HC05i8

Motorola wishes to continue to improve the quality of its documentation. We would welcome your feedback on the publication you have just received. Having used the document, please complete this form (or a photocopy of it, if you prefer).

1. How would you rate the quality of the document? Check one box in each category.

	Excellent			Poor		Excellent			Poor
Organisation	□	□	□	□	Tables	□	□	□	□
Readability	□	□	□	□	Table of contents	□	□	□	□
Understandability	□	□	□	□	Index	□	□	□	□
Accuracy	□	□	□	□	Page size/binding	□	□	□	□
Illustrations	□	□	□	□	Overall impression	□	□	□	□

Comments: ______________________________

2. What is your intended use for this document? If more than one option applies, please rank them (1, 2, 3).

Selection of device for new application □ Other □ Please specify: ______________________________

System design □ ______________________________

Training purposes □ ______________________________

3. How well does this manual enable you to perform the task(s) outlined in question 2?

Completely □ □ □ □ Not at all

Comments: ______________________________

____

4. How easy is it to find the information you are looking for?

Easy □ □ □ □ Difficult

Comments ______________________________

____

5. Is the level of technical detail in the following sections sufficient to allow you to understand how the device functions?

	Too little detail				Too much detail
Section 1: Introduction	□	□	□	□	□
Section 2: Functional Pin Description	□	□	□	□	□
Section 3: CPU Core and Instruction Set	□	□	□	□	□
Section 4: Resets, Interrupts and Low Power Modes	□	□	□	□	□
Section 5: Memory and Registers	□	□	□	□	□
Section 6: Parallel Input/Output Ports	□	□	□	□	□
Section 7: Programmable Timer	□	□	□	□	□
Section 8: Core Timer	□	□	□	□	□
Section 9: Serial Communications Interface	□	□	□	□	□
Section 10: M68000 Interface Module	□	□	□	□	□
Section 11: Electrical Specifications	□	□	□	□	□
Section 12: Mechanical Data and Ordering Information	□	□	□	□	□

Comments: ______________________________

____

____

6. Have you found any errors? If so, please comment: ______________________________

____

7. From your point of view, is anything missing from the document? If so, please say what: ______________________________

____

8. How could we improve this document? ______________________

____

____

9. How would you rate Motorola's documentation?

	Excellent			Poor
– In general	□	□	□	□
– Against other semiconductor suppliers	□	□	□	□

10. Which semiconductor manufacturer provides the best technical documentation? ______________________

11. Which company (in any field) provides the best technical documentation? ______________________

12. How many years have you worked with microprocessors?

Less than 1 year □ 1–3 years □ 3–5 years □ More than 5 years □

– Second fold back along this line –

By air mail
Par avion

IBRS NUMBER PHQ-B/207/G
CCRI NUMERO PHQ-B/207/G

NE PAS AFFRANCHIR
NO STAMP REQUIRED

REPONSE PAYEE
GRANDE-BRETAGNE

Motorola Ltd.
Colvilles Road
Kelvin Industrial Estate
EAST KILBRIDE
GREAT BRITAIN
G75 8BR

Attn: Technical Publications Manager

MOTOROLA LTD.
Semiconductor Products Sector

– First fold back along this line –

– Cut along this line to remove –

– Third fold back along this line –

13. Recently there has been some discussion in the semiconductor industry regarding a move towards providing data sheets in electronic form. If you have any opinion on this subject, please comment: ______________________

____

14. We would be grateful if you would supply the following information (at your discretion), or attach your card.

Name: Phone No:
Position: FAX No:
Department:
Company:
Address:

Thank you for helping us to improve our documentation.

Graham Livey, Technical Publications Manager, Motorola Ltd., Scotland.

– Last tuck this edge into opposite flap –

CUSTOMER FEEDBACK QUESTIONNAIRE — MC68HC05i8

Motorola wishes to continue to improve the quality of its documentation. We would welcome your feedback on the publication you have just received. Having used the document, please complete this form (or a photocopy of it, if you prefer).

1. How would you rate the quality of the document? Check one box in each category.

	Excellent			Poor		Excellent			Poor
Organisation	□	□	□	□	Tables	□	□	□	□
Readability	□	□	□	□	Table of contents	□	□	□	□
Understandability	□	□	□	□	Index	□	□	□	□
Accuracy	□	□	□	□	Page size/binding	□	□	□	□
Illustrations	□	□	□	□	Overall impression	□	□	□	□

Comments: ______________________________

2. What is your intended use for this document? If more than one option applies, please rank them (1, 2, 3).

Selection of device for new application □ Other □ Please specify: ______________

System design □ ______________

Training purposes □ ______________

3. How well does this manual enable you to perform the task(s) outlined in question 2?

Completely □ □ □ □ Not at all Comments: ______________

____

4. How easy is it to find the information you are looking for?

Easy □ □ □ □ Difficult Comments ______________

____

5. Is the level of technical detail in the following sections sufficient to allow you to understand how the device functions?

	Too little detail				Too much detail
Section 1: Introduction	□	□	□	□	□
Section 2: Functional Pin Description	□	□	□	□	□
Section 3: CPU Core and Instruction Set	□	□	□	□	□
Section 4: Resets, Interrupts and Low Power Modes	□	□	□	□	□
Section 5: Memory and Registers	□	□	□	□	□
Section 6: Parallel Input/Output Ports	□	□	□	□	□
Section 7: Programmable Timer	□	□	□	□	□
Section 8: Core Timer	□	□	□	□	□
Section 9: Serial Communications Interface	□	□	□	□	□
Section 10: M68000 Interface Module	□	□	□	□	□
Section 11: Electrical Specifications	□	□	□	□	□
Section 12: Mechanical Data and Ordering Information	□	□	□	□	□

Comments: ______________________________

____

____

6. Have you found any errors? If so, please comment: ______________

____

7. From your point of view, is anything missing from the document? If so, please say what: ______________

____

8. How could we improve this document? ______________________________

____

____

9. How would you rate Motorola's documentation?

	Excellent			Poor
– In general	□	□	□	□
– Against other semiconductor suppliers	□	□	□	□

10. Which semiconductor manufacturer provides the best technical documentation? ______________________________

11. Which company (in any field) provides the best technical documentation? ______________________________

12. How many years have you worked with microprocessors?

Less than 1 year □ 1–3 years □ 3–5 years □ More than 5 years □

– Second fold back along this line –

By air mail
Par avion

IBRS NUMBER PHQ-B/207/G
CCRI NUMERO PHQ-B/207/G

NE PAS AFFRANCHIR
NO STAMP REQUIRED

REPONSE PAYEE
GRANDE-BRETAGNE

Motorola Ltd.
Colvilles Road
Kelvin Industrial Estate
EAST KILBRIDE
GREAT BRITAIN
G75 8BR

Attn: Technical Publications Manager

MOTOROLA LTD.
Semiconductor Products Sector

– First fold back along this line –

– Cut along this line to remove –

– Third fold back along this line –

13. Recently there has been some discussion in the semiconductor industry regarding a move towards providing data sheets in electronic form. If you have any opinion on this subject, please comment: ______________________________

____

14. We would be grateful if you would supply the following information (at your discretion), or attach your card.

Name: Phone No:

Position: FAX No:

Department:

Company:

Address:

Thank you for helping us to improve our documentation.

Graham Livey, Technical Publications Manager, Motorola Ltd., Scotland.

– Last tuck this edge into opposite flap –

Literature Distribution Centres:
EUROPE: Motorola Ltd., European Literature Centre, 88 Tanners Drive, Blakelands,
Milton Keynes, MK14 5BP, England.
ASIA PACIFIC: Motorola Semiconductors (H.K.) Ltd., Silicon Harbour Center, No. 2, Dai King Street,
Tai Po Industrial Estate, Tai Po, N.T., Hong Kong.
JAPAN: Nippon Motorola Ltd., 4-32-1, Nishi-Gotanda, Shinagawa-ku, Tokyo 141, Japan.
USA: Motorola Literature Distribution, P.O. Box 20912, Phoenix, Arizona 85036.

MOTOROLA

Printed in Great Britain by Tavistock Press (Bedford) Ltd. 2500 5/92 PO.2545