HC05

MC68HC05L10

TECHNICAL
DATA

MOTOROLA

MC68HC05L10

HCMOS Microcomputer Unit

All Trade Marks recognized. This document contains information on new products. Specifications and information herein are subject to change without notice.

TABLE OF CONTENTS

Paragraph	Title	Page
	SECTION 1 INTRODUCTION	
1.1	General	1-1
1.2	Features	1-1
	SECTION 2 FUNCTIONAL PIN DESCRIPTION, INPUT/OUTPUT PROGRAMMING, MEMORY AND CPU REGISTERS	
2.1	Functional Pin Description	2-1
2.1.1	VDD and VSS	2-1
2.1.2	IRQ1#, IRQ2#	2-1
2.1.3	RESET#	2-1
2.1.4	OSC1, OSC2	2-1
2.1.5	PA0-PA7	2-1
2.1.6	PB0-PB7	2-1
2.1.7	PC0-PC7	2-2
2.1.8	PD0-PD7	2-2
2.1.9	PE0-PE3	2-2
2.1.10	P02	2-2
2.1.11	R/W#	2-2
2.1.12	AD0-AD19	2-2
2.1.13	D0-D7	2-2
2.1.14	BP0-BP40	2-2
2.1.15	BPCLK	2-2
2.1.16	FRM#	2-3
2.1.17	CS1#, CS2#, CS3#, CS4#	2-3
2.1.18	Vlcd	2-3
2.1.19	Vout	2-3
2.1.20	V1, V4	2-3
2.1.21	TONE	2-3
2.1.22	VCOIN, PCOUT	2-3
2.1.23	RDIS	2-3
2.2	Input/Output Programming	2-4
2.2.1	Parallel Ports	2-4
2.2.2	Fixed Port	2-4
2.3	Memory	2-6
2.3.1	Control Port Register	2-7
2.3.2	Control Registers	2-7
2.4	CPU Registers	2-9
2.4.1	Accumulator (A)	2-10
2.4.2	Index Register (X)	2-10
2.4.3	Program Counter (PC)	2-10
2.4.4	Stack Pointer (SP)	2-10

TABLE OF CONTENTS
(Continued)

Paragraph	Title	Page
2.4.5	Condition Code Register (CC)	2-11
2.4.5.1	Half Carry Bit (H)	2-11
2.4.5.2	Interrupt Mask Bit(I)	2-11
2.4.5.3	Negative (N)	2-11
2.4.5.4	Zero (Z)	2-11
2.4.5.5	Carry/Borrow (C)	2-11
	SECTION 3 RESETS, INTERRUPTS, LOW POWER, AND DATA RETENTION MODES	
3.1	Resets	3-1
3.1.1	Reset Pin	3-1
3.1.2	Power-On Reset	3-1
3.2	Interrupts	3-4
3.2.1	Hardware Controlled Sequences	3-5
3.2.2	Software Interrupt (SWI)	3-5
3.2.3	External Interrupt	3-5
3.2.4	Timer Interrupt	3-9
3.2.5	SCI Interrupt	3-9
3.2.6	Keyboard Interrupt	3-9
3.2.7	RTC Interrupt	3-10
3.2.8	SPI Interrupt	3-10
3.3	Low Power Modes	3-10
3.3.1	Low Power Modes Introduction	3-10
3.3.2	STOP Mode	3-11
3.3.2.1	TIMER during STOP Mode	3-11
3.3.2.2	SCI during STOP Mode	3-11
3.3.2.3	RTC during STOP Mode	3-11
3.3.2.4	SPI during STOP Mode	3-11
3.3.3	WAIT Mode	3-12
3.3.3.1	SCI during WAIT Mode	3-12
3.3.3.2	TIMER during WAIT Mode	3-12
3.3.3.3	SPI during WAIT Mode	3-12
3.3.3.4	REAL TIME CLOCK during WAIT Mode	3-12
3.4	Data Retention Mode	3-13
	SECTION 4 PROGRAMMABLE TIMER	
4.1	Introduction	4-1
4.2	Counter	4-1
4.3	Output Compare Register	4-7
4.4	Input Capture Register	4-8
4.5	Timer Control Register (TCR)	4-8
4.6	Other Timer Function Control Bits	4-9
4.7	Timer Status Register (TSR)	4-9

TABLE OF CONTENTS
(Continued)

Paragraph	Title	Page
	SECTION 5 **SERIAL COMMUNICATIONS INTERFACE (SCI)**	
5.1	Introduction	5-1
5.1.1	SCI Two-Wire System Features	5-1
5.1.2	SCI Receiver Features	5-1
5.1.3	SCI Transmitter Features	5-1
5.2	Data Format	5-2
5.3	Wake-Up Feature	5-2
5.4	Receive Data In	5-2
5.5	Start Bit Detection Following A Framing Error	5-3
5.6	Transmit Data out	5-6
5.7	Registers	5-6
5.7.1	Serial Communications Data Register (SCDAT)	5-6
5.7.2	Serial Communications Control Register 1 (SCCR1)	5-6
5.7.3	Serial Communications Control Register 2 (SCCR2)	5-7
5.7.4	Serial Communications Status Register (SCSR)	5-8
5.7.5	Baud Rate Register	5-10
	SECTION 6 **LIQUID CRYSTAL DISPLAY DRIVER, AUTO DISPLAY OFF, REAL TIME CLOCK, PHASE LOCK LOOP**	
6.1	Introduction	6-1
6.2	Liquid Crystal Display Driver	6-1
6.3	Auto Display Off	6-4
6.4	Real Time Clock	6-4
6.5	Phase Lock Loop	6-7
	SECTION 7 **SERIAL PERIPHERAL INTERFACE (SPI)**	
7.1	Introduction	7-1
7.2	SPI Signal Description	7-1
7.2.1	Master Out Slave In (MOSI)	7-1
7.2.2	Master In Slave Out (MISO)	7-4
7.2.3	Slave Select (SS#)	7-4
7.2.4	Serial Clock (SCK)	7-5
7.3	Functional Description	7-5
7.4	Registers	7-7
7.4.1	Serial Peripheral Control Register (SPCR)	7-7
7.4.2	Serial Peripheral Status Register (SPSR)	7-8
7.4.3	Serial Peripheral I/O Register (SPDR)	7-10
7.5	SPI During WAIT Mode	7-11
7.6	SPI During STOP Mode	7-11

TABLE OF CONTENTS
(Continued)

Paragraph	Title	Page
	SECTION 8 MEMORY MANAGEMENT UNIT (MMU)	
8.1	Introduction	8-1
8.2	Physical Address Translation	8-2
8.3	Registers	8-4
8.3.1	Bank Area Register	8-4
8.3.2	Bank 1 Page Offset Register	8-5
8.3.3	Bank 2 Page Offset Register	8-5
	SECTION 9 INSTRUCTIONS SET AND ADDRESSING MODES	
9.1	Instruction Set	9-1
9.1.1	Register/Memory Instructions	9-2
9.1.2	Read-Modify-Write Instructions	9-2
9.1.3	Branch Instructions	9-3
9.1.4	Bit Manipulation Instructions	9-3
9.1.5	Control Instructions	9-4
9.2	Addressing Modes	9-4
9.2.1	Inherent	9-4
9.2.2	Immediate	9-4
9.2.3	Direct	9-5
9.2.4	Extended	9-5
9.2.5	Indexed, No Offset	9-5
9.2.6	Indexed, 8-bit Offset	9-5
9.2.7	Indexed, 16-bit Offset	9-6
9.2.8	Relative	9-6
9.2.9	Bit Set/Clear	9-6
9.2.10	Bit Test and Branch	9-6
	SECTION 10 ELECTRICAL CHARACTERISTICS	
10.1	Introduction	10-1
10.2	Maximum Ratings	10-1
10.3	DC Electrical Characteristics (VDD = 5.0 Vdc)	10-2
10.4	DC Electrical Characteristics (VDD = 3.0 Vdc)	10-3
10.5	Bus Timing	10-4
10.6	Port/Slave Control Timing	10-5
	SECTION 11 SELF-CHECK	
11.1	Self-Check Mode	11-1

LIST OF APPENDICES

Title	Page
APPENDIX A	
Mechanical Data	A-1
APPENDIX B	
MC141511	B-1

LIST OF APPENDICES

Title	Page
APPENDIX A	
Mechanical Data	A-1
APPENDIX B	
MC141511	B-1

LIST OF FIGURES

Figure	Title	Page
1-1	MC68HC05L10 Microcontroller Block Diagram	1-2
1-2	MC68HC05L10 Pin Assignment	1-3
2-1	Parallel Port I/O Circuitry	2-5
2-2	MC68HC05L10 Memory Map	2-6
2-3	Programming Model	2-9
2-4	Stacking Order	2-10
3-1	Power-On Reset and RESET#	3-3
3-2	Hardware Interrupt Flow Chart	3-6
3-3	Stop/Wait Flowcharts	3-7
3-4	External Interrupt	3-8
4-1	Programmable Timer Block Diagram	4-2
4-2	Timer State Timing Diagram for Reset	4-3
4-3	Timer State Timing Diagram for Input Capture	4-4
4-4	Timer State Timing Diagram for Output Compare	4-5
4-5	Timer State Diagram for Timer Overflow	4-6
5-1	Data Format	5-2
5-2	Example of Start-Bit Sampling Technique	5-3
5-3	Sampling Technique Used on All Bits	5-3
5-4	SCI Artificial Start Following A Framing Error	5-4
5-5	SCI Start Bit Following A Break	5-4
5-6	Serial Communications Interface Block Diagram	5-5
5-7	Rate Generator Division	5-12
6-1	LCD Driver - Functional Block Diagram	6-2
6-2	Backplane Waveform of 1/5 bias, 1/32 or 1/41 Multiplex	6-2
6-3	Display RAM Configuration at 1:32 and 1:41 Multiplex Ratio	6-3
6-4	Phase Lock Loop Block Diagram	6-8
6-5	Typical Waveforms for PLL	6-8
7-1	Master-Slave System Configuration	7-2
7-2	Data Clock Timing Diagram	7-3
7-3	Serial Peripheral Interface Block Diagram	7-6
7-4	Serial Peripheral Interface Master-Slave Interconnection	7-7
8-1	Memory Paging On Bank 2	8-1
8-2	MMU Block Diagram	8-2
8-3	Logical to Physical Address Translation	8-3
8-4	An Example of Paging	8-6
10-1	Bus Timing	10-4
10-2	Port/Slave CS Timing	10-5
11-1	MC68HC05L10 Self-Check Circuit Diagram	11-3

LIST OF TABLES

Table	Title	Page
2-1	I/O Pin Functions	2-4
2-2	PD6, PD7 Configuration	2-9
3-1	Reset Action on Internal Circuit	3-2
3-2	Vector Address and Priority of Interrupts and Reset	3-4
5-1	Prescaler Highest Baud Rate Frequency Output	5-12
5-2	Transmit Baud Rate Output For a Given Prescaler Output	5-12
11-1	Self-Check Results	11-2

SECTION 1
INTRODUCTION

1.1 GENERAL

The MC68HC05L10 HCMOS Microcontroller is a member of the M68HC05 Family of low-cost single-chip microcontrollers and it is an enhanced version of MC68HC05L9. This 8-bit microcontroller unit (MCU) contains two on-chip oscillators, CPU, RAM, ROM, I/O, timer, serial interface system, serial peripheral interface, liquid crystal display driver circuitry, real time clock, alarm, auto display off, external parallel address, data bus, and a tone generator.

1.2 FEATURES

The following are some of the hardware and software highlights of the MC68HC05L10 single-chip microcontroller.

HARDWARE FEATURES

- 8-bit architecture
- Power saving stop, wait modes
- 352 bytes of on-chip RAM (64 bytes for stack)
- 13.344K bytes of on chip ROM (12.878K bytes of user ROM)
- 28 bidirectional I/O lines and 8 input only lines
- LCD driver circuitry with a selection of 1:32 or 1:41 multiplex
- Capable of connecting to four slave LCD drivers to increase LCD segment drive to 512
- Memory Management Unit to increase address bus from 16 to 20 bit
- 8-bit bidirectional data bus
- Internal 16-bit timer
- Serial Communications Interface System
- Serial Peripheral Interface
- 2 on-chip oscillators - RC osc. for MCU, and crystal osc. for real time clock & LCD driver
- Self-check mode
- Four selectable bus speeds. The maximum bus speed at 5V supply is up to 3.6864 MHz
- 128-pin QFP or in Die form

SOFTWARE FEATURES

- Similar to MC6800
- 8 x 8 Unsigned Multiply Instruction
- Efficient Use of Program Space
- Versatile Interrupt Handling
- True Bit Manipulation
- Addressing Modes with Index Addressing for Tables
- Efficient Instruction Set
- Memory Mapped I/O
- Two Power-Saving Standby Modes
- Upward Software Compatible with the M146805 CMOS Family

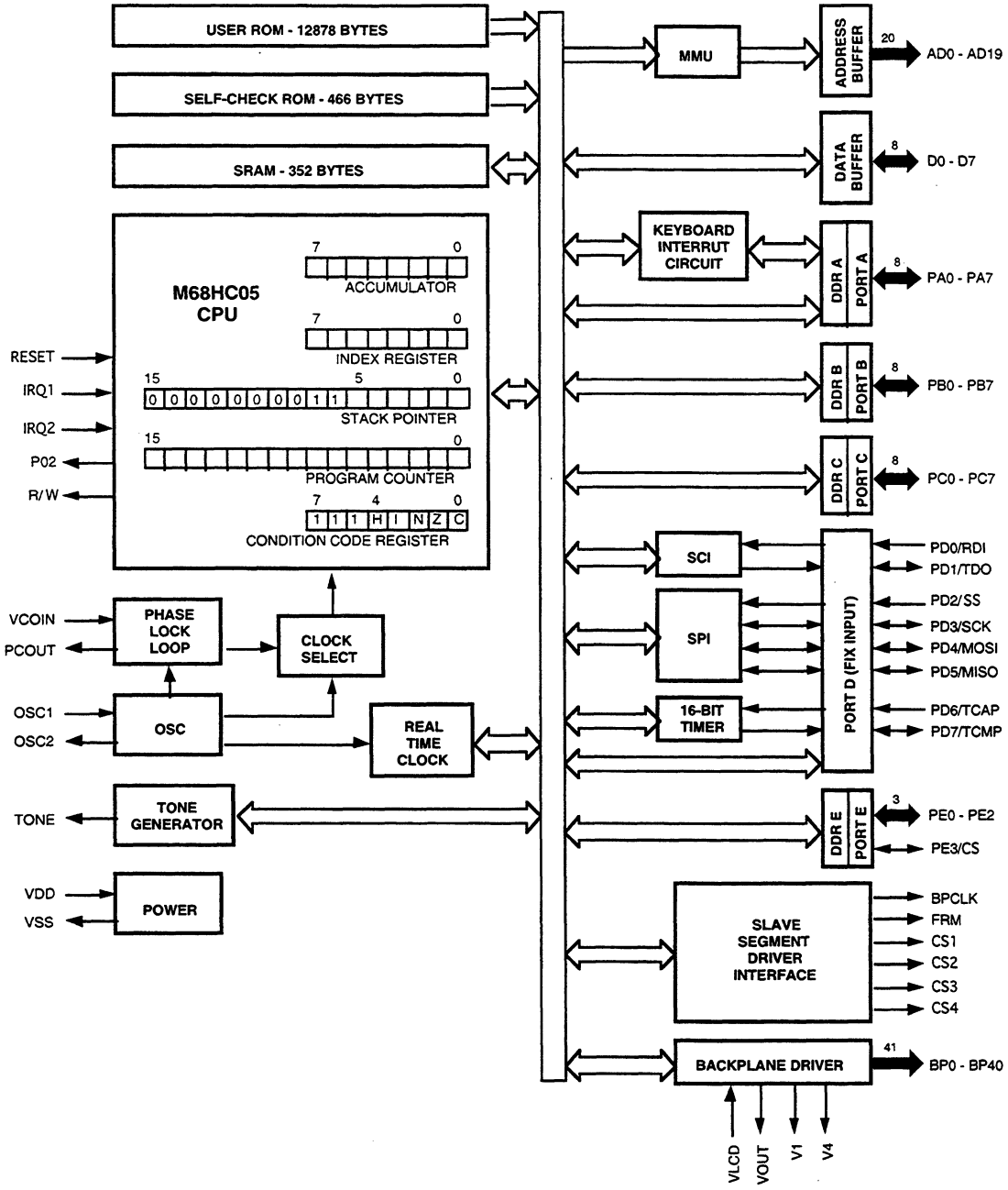


FIG. 1-1 MC68HC05L10 Block Diagram

128 PIN QFP

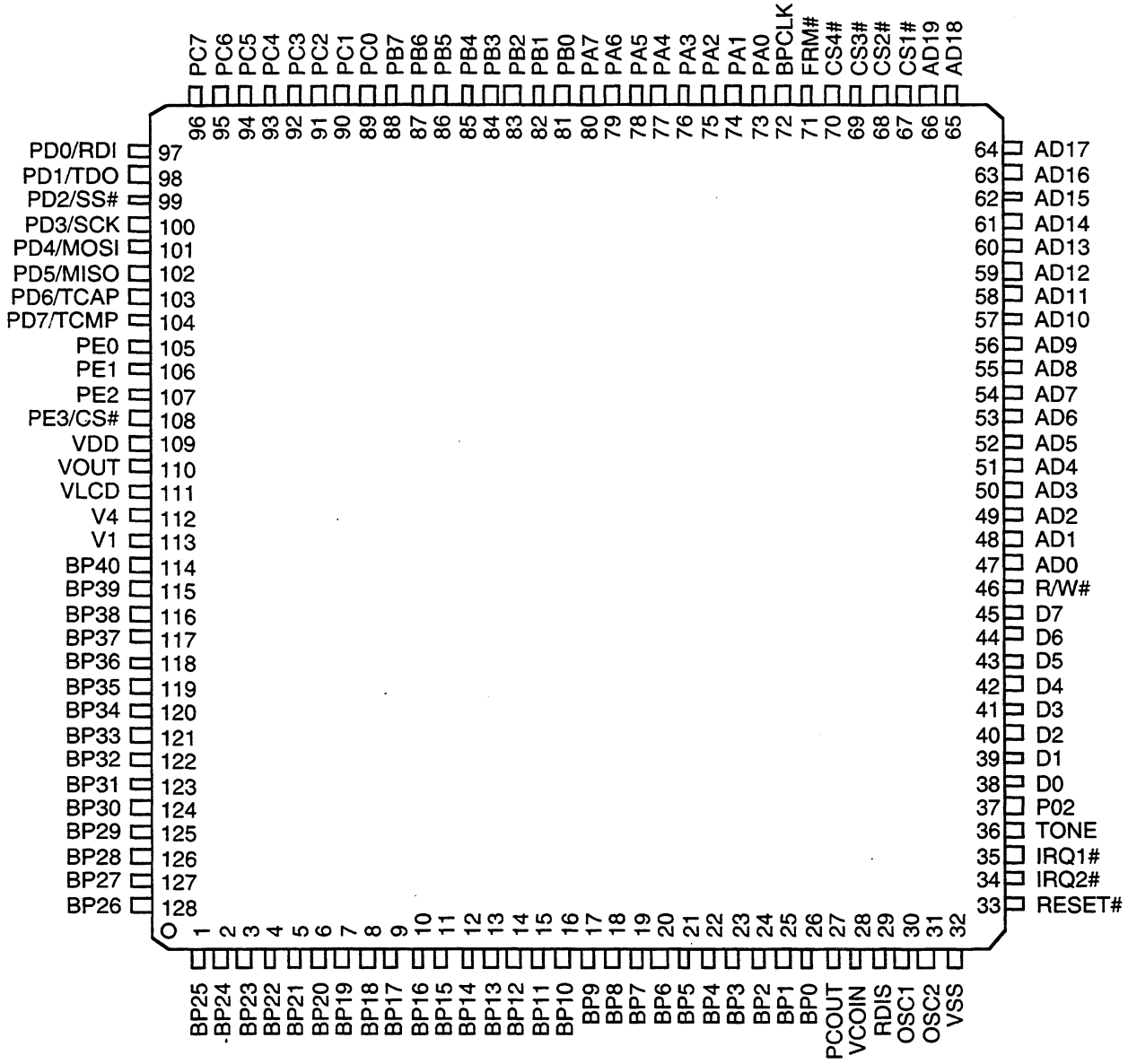


FIG. 1-2 MC68HC05L10 Pin Assignment

SECTION 2
FUNCTIONAL PIN DESCRIPTION, INPUT/OUTPUT PROGRAMMING, MEMORY AND CPU REGISTERS

This section provides a description of the functional pins, input/output programming, memory and CPU registers.

2.1 FUNCTIONAL PIN DESCRIPTION

2.1.1 V_{DD} AND V_{SS}

Power is supplied to the MCU using these two pins. V_{DD} is power and V_{SS} is ground.

2.1.2 IRQ1#, IRQ2#

IRQ1#, IRQ2# are software programmable to provide two different choices of interrupt triggering sensitivity. These options are: 1) negative edge-sensitive triggering only, or 2) both negative edge-sensitive and level-sensitive triggering. In the latter case, either type of input to the IRQ1# or IRQ2# pins will produce the interrupt.

2.1.3 RESET#

The RESET# input is not required for startup but can be used to reset the MCU internal state and provide an orderly software startup procedure.

2.1.4 OSC1, OSC2

These pins provide connections to the on chip crystal oscillator. The crystal frequency is 32.768KHz. OSC1 is an input of external clock source if the on chip oscillator is not used.

2.1.5 PA0-PA7

These eight I/O lines comprise of port A. The state of any pin is software programmable and all port A lines are configured as input during power up or after an external reset.

2.1.6 PB0-PB7

These eight I/O lines comprise of port B. The state of any pin is software programmable and all port B lines are configured as input during power up or after an external reset.

2.1.7 PC0-PC7

These eight I/O lines comprise of port C. The state of any pin is software programmable and all port C lines are configured as input during power up or after an external reset.

2.1.8 PD0-PD7

These eight lines comprise of port D. This port performs a dual function for the MCU; an 8-bit input only or peripheral I/O port. When sub-systems are disabled, the corresponding lines become an input port. A read of port D with sub-system enabled will always read as zero.

2.1.9 PE0-PE3

These four I/O lines comprise of port E. The state of any pin is software programmable and all port E lines are configured as input during power up or after an external reset. PE3 can be selected as a chip select signal if address $0A is properly configured (see section 2.3.1).

2.1.10 P02

This is a bus clock output from the processor that indicates when the data on the external data bus is to be accessed by either the processor or the peripheral. It will be in 'LO' state during internal access, wait and stop mode.

2.1.11 R/W#

R/W# is the processor output that indicates to peripheral which direction the data is to be passed over the data bus. When R/W# is high and address bus is addressing memory beyond the first 16K bytes, the processor will be reading data from the external peripheral or memory. When R/W# is low, the processor will be writing data to the external peripheral or memory.

2.1.12 AD0-AD19

AD0-AD19 is 20-bit wide address bus from the processor. All address lines will be in 'LO' state if the MCU is accessing internal address locations.

2.1.13 D0-D7

D0-D7 is an 8-bit wide bidirectional data bus used to connect external peripherals and memory to the processor. They will be in high impedance when address bus is selecting the MCU internal ROM.

2.1.14 BP0-BP40

These 41 output lines provide the backplane drive signals to the LCD unit.

2.1.15 BPCLK

A 2.048KHz 50% duty cycle signal output pin.

2.1.16 FRM#

A periodic reset signal pin for slave LCD driver.

2.1.17 CS1#, CS2#, CS3#, CS4#

These four chip select pins are used to select slave LCD drivers.

For 1: 32 mux
CS1# = 0 when addressing memory $200-$3FF
CS2# = 0 when addressing memory $400-$5FF
CS3# = 0 when addressing memory $600-$7FF
CS4# = 0 when addressing memory $800-$9FF

For 1: 41 mux
CS1# = 0 when addressing memory $1C0-$1CF, $200-$47F
CS2# = 0 when addressing memory $1D0-$1DF, $480-$6FF
CS3# = 0 when addressing memory $1E0-$1EF, $700-$97F
CS4# = 0 when addressing memory $1F0-$1FF, $980-$BFF

2.1.18 Vlcd

This input pin provides the power for the LCD backplane driver circuitry and Vout pin.

2.1.19 Vout

This output provides the power for the external voltage divider.

2.1.20 V1, V4

These two pins provide the voltage levels for internal LCD driver circuitry.

2.1.21 TONE

This output pin provides an audio frequency of either 500 Hz or 2 KHz.

2.1.22 VCOIN, PCOUT

These pins provide connections to the on chip RC oscillator.

2.1.23 RDIS

This input pin is used to enable or disable the internal 13.3K ROM (i.e. $0C00 - $3FFF). When this pin is pulled high, internal ROM is disabled. And is enabled when pulled low.

2.2 INPUT/OUTPUT PROGRAMMING

2.2.1 PARALLEL PORTS

Ports A*, B, C and E** (PE0-PE3) may be programmed as an input or an output under software control. The direction of the pins is determined by the state of corresponding bit in the port data direction register (DDR). Each 8-bit port has an associated 8-bit data direction register. Any port A, port B, port C or port E (PE0-PE3) 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-up or after a reset, all DDRs are cleared, which configure all port A, B, C and port E (PE0-PE3) pins as inputs. The data direction registers are capable of being written to or read by the processor (see Table 2-1). 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.

* When KEYE (bit 4 of $27) is set and port A is configured as an input, there will be 250 K OHM pull up resistors associate with each pin of port A. And this implies port A DDR bit set to one to become an output port will override the KEYE bit.

** PE3 becomes chip select signal when $0A is not equal to zero.

Table 2-1 I/O Pin Functions

R/W#	DDR	I/O Pin Functions
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 an output mode. The output data latch is read.

2.2.2 FIXED PORT

Port D*** (PD0-PD7) is an 8-bit fixed input that continually monitors the external pins whenever the SCI, SPI, TIMER systems are disabled. During power-on reset or external reset, TIMER is enabled, PD7 and PD6 are configured as TCMP and TCAP pins.

*** PD0-PD7 will perform as sub-system (Timer, SCI, SPI) functional pins when the corresponding sub-systems are selected. Refer to **SECTION 4 PROGRAMMABLE TIMER, SECTION 5 SERIAL COMMUNICATIONS INTERFACE, SECTION 7 SERIAL PERIPHERAL INTERFACE** for detail descriptions of PD0-PD7 when the peripheral sub-systems are enabled.

NOTE

It is recommended that all unused inputs and I/O ports be tied to an appropriate logic level (e.g., either VDD or VSS).

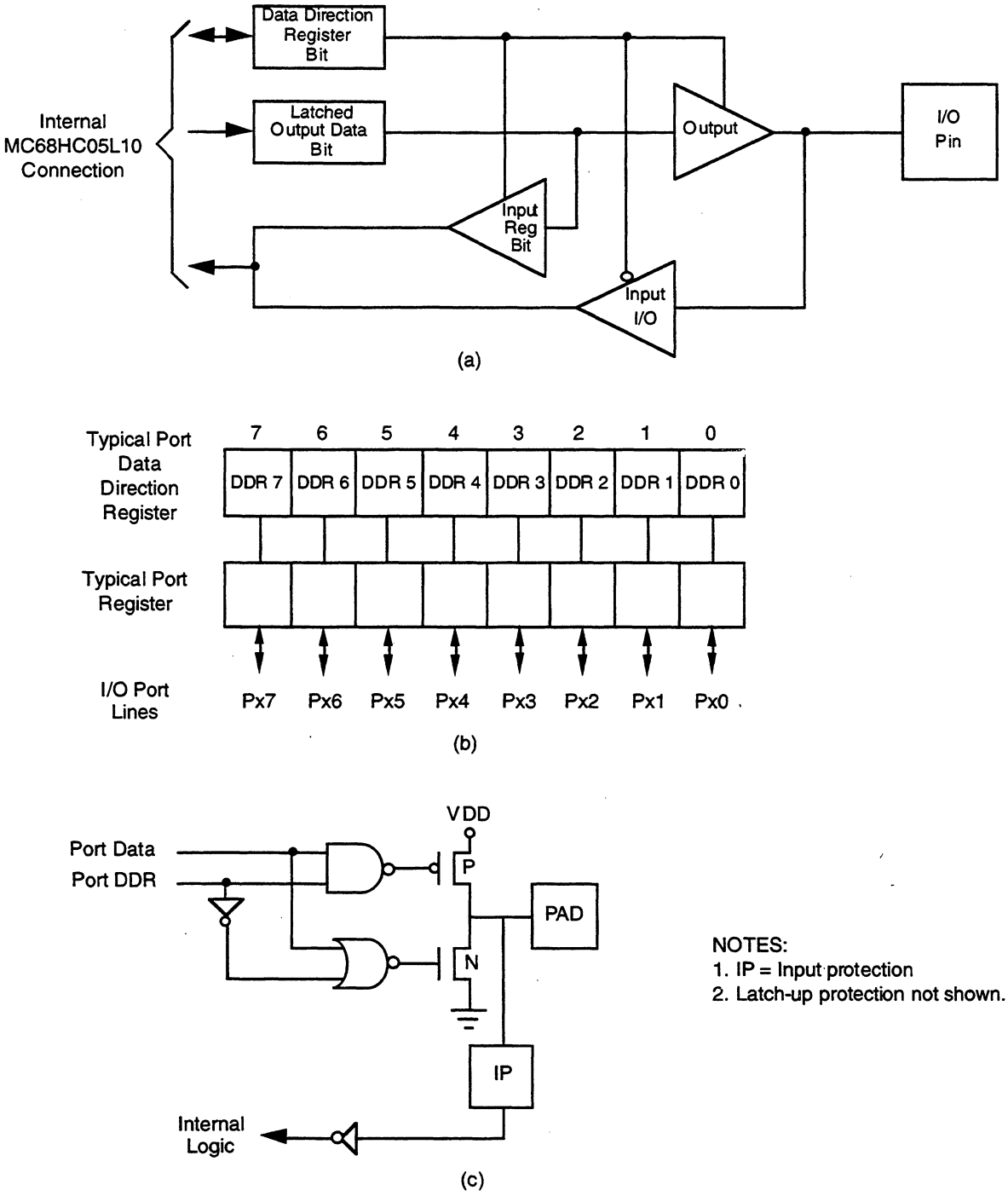


FIG. 2-1 Parallel Port I/O Circuitry

2.3 MEMORY

As shown in Fig. 2-2, the MCU internal memory address line is 16 bits but is capable of addressing one mega bytes of memory with its memory management unit (MMU).

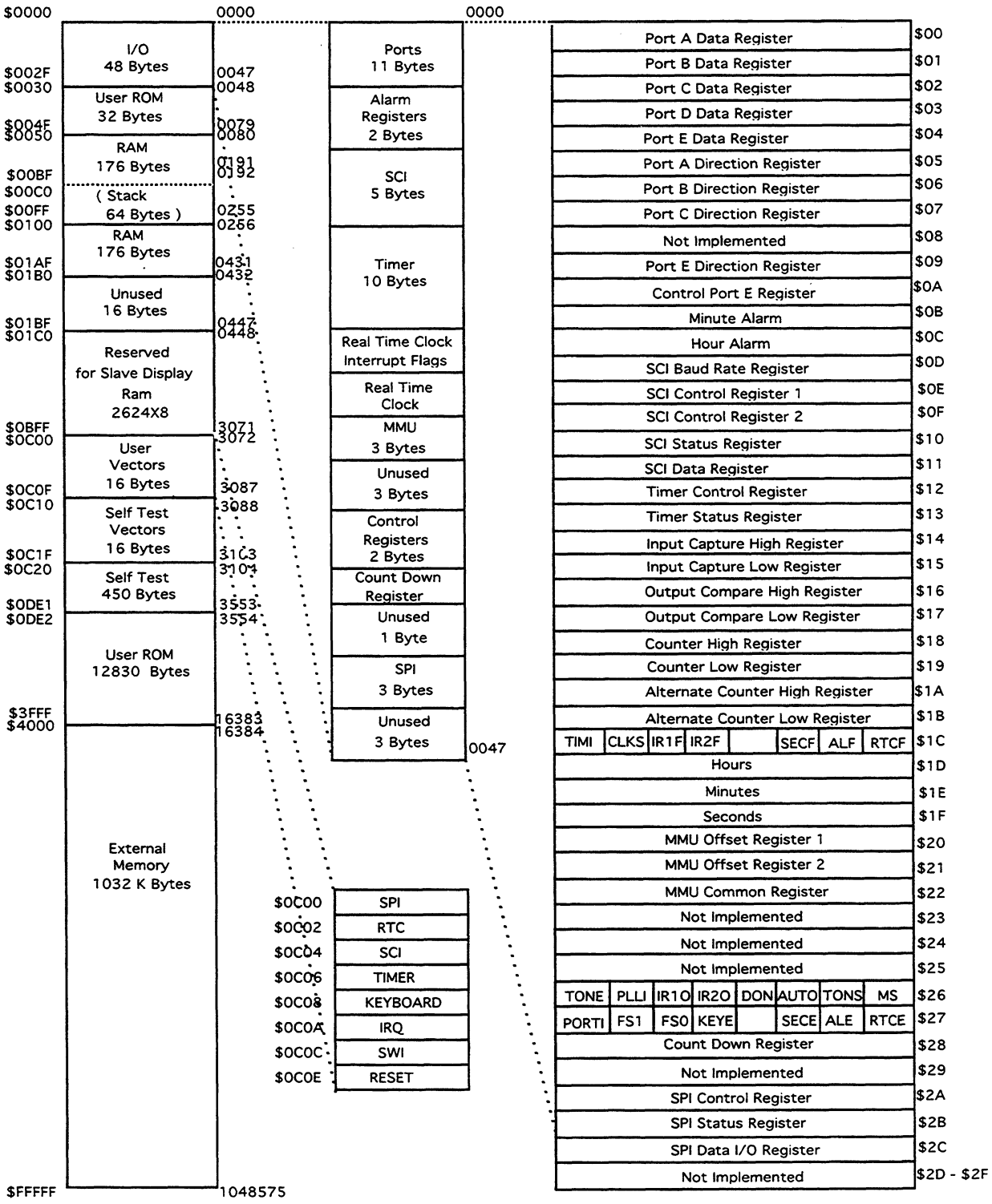


FIG. 2-2 MC68HC05L10 Memory Map

2.3.1 CONTROL PORT REGISTER

Address $0A

Bit 1-0 00 = PE3 is general I/O port
01 = PE3 will output zero when addressing memory $8000 - $9FFF (8K)
10 = PE3 will output zero when addressing memory $8000 - $BFFF (16K)
11 = PE3 will output zero when addressing memory $8000 - $FFFF (32K)

2.3.2 CONTROL REGISTERS

Address $26, $27 and $1C are the control registers to control the special function of the MCU and they are described as follow:

Address $26

MS	BIT0	0 = 1 : 32 multiplex 1 = 1 : 41 multiplex
TONS	BIT1	0 = Tone output frequency of 500 Hz 1 = Tone output frequency of 2 KHz
AUTO	BIT2	0 = Disable auto display off feature 1 = Enable auto display off feature
DON	BIT3	0 = Turn off LCD 1 = Turn on LCD
IR2O	BIT4	0 = Both negative edge-sensitive and level-sensitive triggering for IRQ2 1 = Negative edge-sensitive triggering only for IRQ2
IR1O	BIT5	0 = Both negative edge-sensitive and level-sensitive triggering for IRQ1 1 = Negative edge-sensitive triggering only for IRQ1
PLLI	BIT6	0 = Indicate PLL is unlocked 1= Indicate PLL is locked
TONE	BIT7	0 = No audio output at TONE pin 1 = Audio output at TONE pin

Address $27

RTCE	BIT 0	0 = Real time clock once a day interrupt disable 1 = Real time clock once a day interrupt enable
ALE	BIT1	0 = Alarm interrupt disable 1 = Alarm interrupt enable
SECE	BIT2	0 = Real time clock once a second interrupt disable 1 = Real time clock once a second interrupt enable
KEYE	BIT4	0 = Keyboard interrupt disable 1 = Keyboard interrupt enable

FS1,FS0	BIT6,5	00 = 307.2 KHz internal bus frequency 01 = 1.2288 MHz internal bus frequency 10 = 2.4576 MHz internal bus frequency 11 = 3.6864 MHz internal bus frequency
PORTI	BIT7	This bit, together with $1C bit 7, determine the function of pin 6 and 7 of port D (see table 2-2)

Address $1C

RTCF	BIT 0	0 = Indicates no once a day interrupt has occurred. 1= Indicates once a day interrupt has occurred. After serving this interrupt, user is responsible to clear this bit, otherwise the CPU will keep on serving this once a day interrupt when a new RTC interrupt occurs (even there is no once a day interrupt occurs).
ALF	BIT1	0 = Indicates no alarm interrupt has occurred. 1 = Indicates alarm interrupt has occurred. After serving this interrupt, user is responsible to clear this bit, otherwise the CPU will keep on serving this alarm interrupt when a new RTC interrupt occurs (even there is no alarm interrupt occurs).
SECF	BIT2	0 = Indicates no once a second interrupt has occurred. 1 = Indicates once a second interrupt has occurred. After serving this interrupt, user is responsible to clear this bit, otherwise the CPU will keep on serving this once a second interrupt when a new RTC interrupt occurs (even there is no once a second interrupt occurs).
IR2F	BIT4	0 = Indicates no IRQ2 interrupt has occurred. 1 = Indicates IRQ2 interrupt has occurred. After serving this interrupt, user is responsible to clear this bit, otherwise the CPU will keep on serving IRQ2 interrupt when a new IRQ interrupt occurs (even there is no IRQ2 interrupt occurs).
IR1F	BIT5	0 = Indicates no IRQ1 interrupt has occurred. 1 = Indicates IRQ1 interrupt has occurred. After serving this interrupt, user is responsible to clear this bit, otherwise the CPU will keep on serving IRQ1 interrupt when a new IRQ interrupt occurs (even there is no IRQ1 interrupt occurs).
CLKS	BIT6	0 = 16 KHZ internal bus frequency 1 = PLL clock for CPU
TIMI	BIT7	This bit, together with $27 bit 7, determine the function of pin 6 and 7 of port D (see the following table)

TIMI	PORTI	TIMER	PD6	PD7
0	0	Enable	TCAP	TCMP
0	1	*Enable	INPUT	INPUT
1	0	Disable	INPUT	INPUT
1	1	Disable	INPUT	INPUT

*TCAP and TCMP not available to external

Table 2-2 PD6, PD7 Configuration

NOTE

All bits in addresses $1C, $26 and $27 are readable and writable, except bit 6 of address $26 that is read only. All bits are cleared during power on or external reset.

2.4 CPU REGISTERS

The MC68HC05L10 CPU contains five registers, as shown in the programming model of Fig. 2-3. The interrupt stacking order is shown in Fig. 2-4.

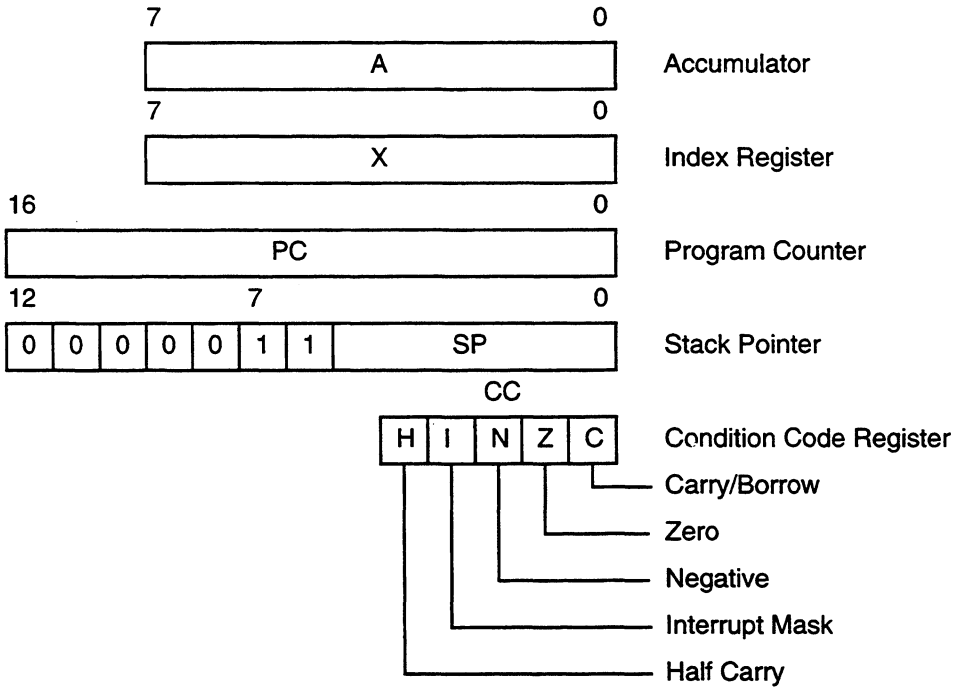


Fig. 2-3 Programming Model

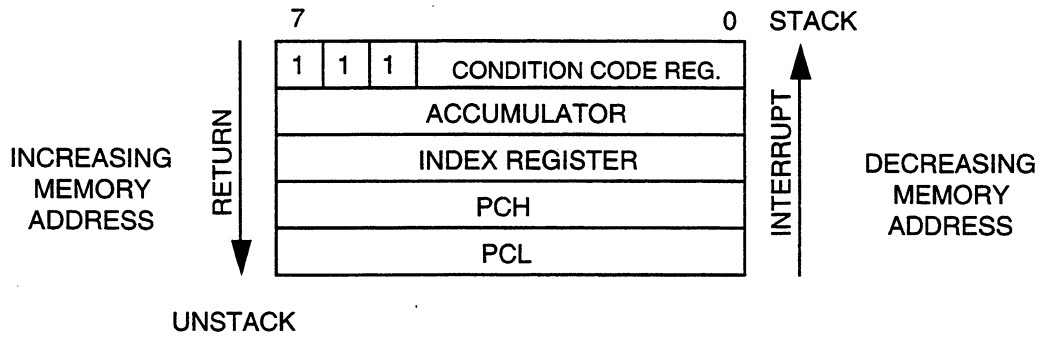


Fig. 2-4 Stacking Order

2.4.1 Accumulator (A)

The accumulator is an 8-bit general purpose register used to hold operands, results of arithmetic calculations, and data manipulation.

2.4.2 Index Register (X)

The X register is an 8-bit register which is used during the index modes of addressing. It provides an 8-bit value which is used to create an effective address. The index register is also used for data manipulations with the read-modify-write type of instructions and as a temporary storage register when not performing addressing operations.

2.4.3 Program Counter (PC)

The Program counter is a 16-bit register that contains the address of the next instruction to be executed by the processor.

2.4.4 Stack Pointer (SP)

The stack pointer is a 13-bit register containing the address of the next free locations on the push-down/pop-up stack. When accessing memory, the seven most significant bits are permanently configured to 0000011. These seven bits are appended to the six least significant register bits to produce an address within the range of $00FF to $00C0. The stack area of RAM is used to store the return address on subroutine calls and the machine state during interrupts. During external or power-on reset, and during a reset stack pointer (RSP) instruction, the stack pointer is set to its upper limit ($00FF). Nested interrupt and/or subroutines may use up to 64 (decimal) locations. When the 64 locations are exceeded, the stack pointer wraps around and points to its upper limit ($00FF), thus, losing the previously stored information. A subroutine call occupies two RAM bytes on the stack, while an interrupt uses five RAM bytes.

2.4.5 Condition Code Register (CC)

The condition code register is a 5-bit register which indicates the results of the instruction just executed as well as the state of the processor. These bits can be individually tested by a program and specified action taken as a result of their state. Each bit is explained in the following paragraphs.

2.4.5.1 HALF CARRY BIT (H) - The H bit is set to a one when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instruction. The H bit is useful in binary coded decimal subroutines.

2.4.5.2 INTERRUPT MASK BIT (I) - When the I bit is set, all interrupts are disabled. Clearing this bit enables the interrupts. If an external interrupt occurs while the I bit is set, the interrupt is latched and is processed after the I bit is next cleared; therefore, no interrupts are lost because of the I bit being set. An internal interrupt can be lost if it is cleared while the I bit is set (refer to **Section 4 Programmable Timer, Section 5 Serial Communications Interface and Section 7 Serial Peripheral Interface** for more information).

2.4.5.3 NEGATIVE (N) - When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation is negative (bit 7 in the result is a logic one).

2.4.5.4 ZERO (Z) - When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation is zero.

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

SECTION 3
RESETS, INTERRUPTS, LOW POWER, AND DATA RETENTION MODES

3.1 RESETS

The MC68HC05L10 has two reset modes; an active low external reset pin (RESET#) and a power-on reset function; refer to Fig. 3-1.

3.1.1 RESET# Pin

The RESET# input pin is used to reset the MCU to provide an orderly software startup procedure. When using the external reset mode, the RESET# pin must stay low for a minimum of one and one half tcyc. The RESET# pin contains an internal Schmitt Trigger as part of its input to improve noise immunity. There is no internal pull high on the RESET# pin. It has to be taken care of externally.

3.1.2 Power-On Reset (POR)

The power-on reset occurs when a positive transition is detected on VDD. The power-on reset is used strictly for power turn-on conditions and should not be used to detect any drops in the power supply voltage. There is no provision for a power-down reset. The power-on circuitry provides for a tpor tcyc delay from the time that the oscillator becomes active. If external RESET# pin is low at the end of the tpor tcyc time out, the processor remains in the reset condition until RESET# go high. The user must ensure that VDD has risen to a point where the MCU can operate properly prior to the time the tpor tcyc POR reset cycles have elapsed. If there is doubt, the external RESET# pin should remain low until such time that VDD has risen to the minimum operating voltage specified.

tpor is defined by a user specified mask option as either 16 or 4064 cycles.

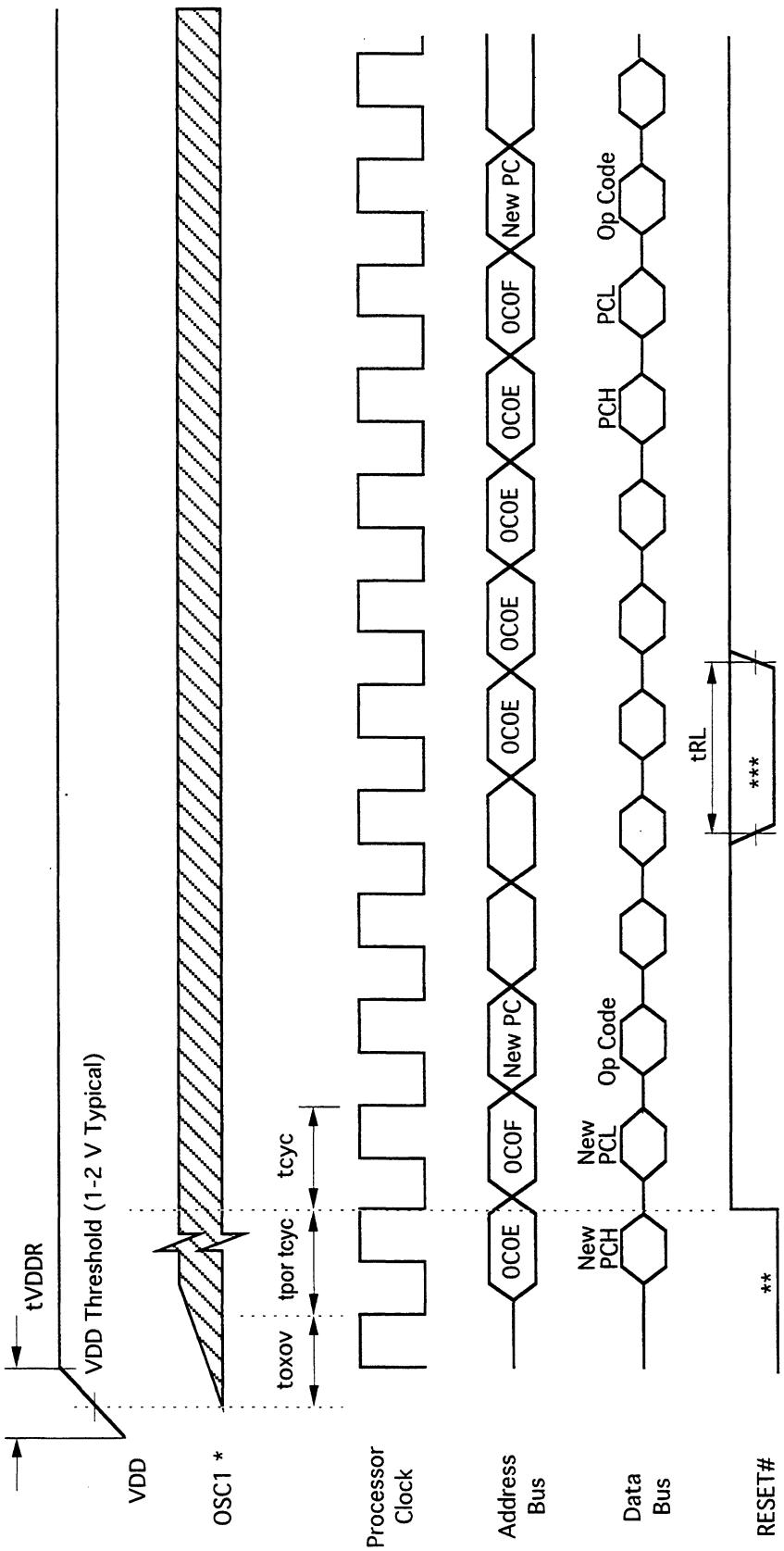
Table 3-1 shows the actions of two resets on internal circuits, but not necessarily in order of occurrence (X indicates that the condition occurs for the particular reset).

During the POR reset cycles, the RESET# pin will pull itself low until the reset cycles has elapsed.

Table 3-1 Reset Action on Internal Circuit

Condition	RESET Pin	Power-On Reset
Timer Prescaler reset to zero state	X	X
Timer counter configures to $FFFC	X	X
Timer output compare (TCMP) bit reset to zero	X	X
All timer interrupt enable bits cleared (ICIE, OCIE, and TOIE) to disable timer interrupts The OLVL timer bit is also cleared by reset	X	X
All data direction registers cleared to zero (input)	X	X
Count Down register are set to three	X	X
Hour and Minute alarm register are set to zero	X	X
Hours, minutes and seconds registers are set to zero	X	X
Configure stack pointer to $00FF	X	X
Force internal address bus to restart vector ($0C0E-$0C0F)	X	X
Set I bit in condition code register to a logic one	X	X
Clear STOP latch	X	X
Clear external interrupt latch	X*	X
Clear WAIT latch	X	X
Disable SCI (serial control bits TE=0 and RE=0). Other SCI bits cleared by reset include: TIE, TCIE, RIE, ILIE, RWU, SBK, RDRF, IDLE, OR, NF, and FE	X	X
Set serial status bits TDRE and TC	X	X
Clear all serial interrupt enable bits (TIE, and TCIE)	X	X
Clear SCI prescaler rate control bits SCP0-SCP1	X	X
Keyboard interrupt enable bit is cleared	X	X
RTC interrupt enable bit is cleared. Other RTC bit cleared by reset include: RTCF, ALF, SECF	X	X
Disable SPI (Serial Output Enable Control bit SPE=0)-other SPI bits cleared by Reset include: SPIE, MSTR, SPIF, WCOL, and MODF	X	X
Clear Serial Interrupt Enable Bit (SPIE)	X	X
Place SPI system in slave mode	X	X
All bits in address $0A,$1C, $26 and $27 are cleared	X	X
Clear MMU Offset Register1, Offset Register2	X	X
Set MMU Common Bank Register to $FF	X	X

* Indicate that timeout still occurs



* OSC1 line is not meant to represent frequency. It is only used to represent time.
** The reset pin will be pulled low by the POR circuitry until the end of the tpor tcyc.
*** The next rising edge of the processor clock following the rising edge of RESET# initiates the reset sequence.

FIG. 3-1 Power-On reset and RESET#

3.2 INTERRUPTS

The MC68HC05L10 is capable of operating with eight different interrupts, seven hardware and one software. The "I" bit in the Condition Code Register, if set, blocks all interrupts except the software interrupt, SWI. Interrupts such as Timer, RTC, SCI and SPI have several flags which will cause the interrupt. Generally, interrupt flags are found in "read only" status registers (except RTC) while their enables are in associated control registers. They are never mixed in the same register. If the enable bit is "0", it blocks the interrupt from occurring but does not inhibit the flag from being set. RESET clears all enable bits. The general sequence for clearing an interrupt is a software sequence of reading the status register while the flag is set followed by a read or write of an associated register (except RTC). When any of these interrupts occur, and if enabled, normal processing is suspended at the end of the current instruction execution. The state of the machine is pushed onto the stack (see Figure 2-4 for stacking order) and the appropriate vector points to the starting address of the interrupt service routine (see Table 3-2). Also, the interrupt mask bit in the condition code register is set. This masks further interrupts. At the completion of the service routine, the software normally contains an RTI instruction which, when executed, restores the machine state and continues executing the interrupted program. Note that the interrupt mask bit (I bit) will be reset if and only if the corresponding bit stored on the stack is zero.

Table 3-2 Vector Address and Priority of Interrupts and Reset

Register	Flag Name	Interrupts	CPU Interrupt	Priority	Vector Address
X	X	Reset	RESET	1	$0C0E-$0C0F
X	X	Software	SWI	2	$0C0C-$0C0D
X	X	External Interrupt	IRQ	3	$0C0A-$0C0B
X	X	Keyboard	KEYBOARD	4	$0C08-$0C09
Timer Status	ICF	Input Capture	TIMER	5	$0C06-$0C07
	OCF	Output Compare			
	TOF	Timer Overflow			
SCI Status	TDRE	Transmit Buffer Empty	SCI	6	$0C04-$0C05
	TC	Transmit Complete			
	RDRF	Receive Buffer Full			
	IDLE	Idle Line Detect			
	OR	Overrun			
RTC Status	SECF	Per Second	RTC	7	$0C02-$0C03
	ALF	Alarm			
	RTCF	Per Day			
SPI Status	SPIF	Data Transfer Complete	SPI	8	$0C00-$0C01
	MODF	Mode Fault			

NOTE

There is no internal pull high on the IRQ pins. It has to be taken care of externally.

3.2.1 HARDWARE CONTROLLED SEQUENCES

The following three functions are not strictly interrupts, however, they are tied very closely to the interrupts. These functions are RESET, STOP, WAIT.

(a) RESET# The RESET# input pin causes the program to go to its starting address. This address is specified by the contents of memory locations $0C0E and $0C0F. The interrupt mask of the condition code register is also set. Much of the MCU is set to some known state.

(b) STOP The STOP instruction causes the internal processor clock to be turned off and the processor to "sleep" until external IRQ1#, IRQ2#, RTC, or Keyboard interrupt, or RESET# occurs. A mask option of OSC being stop or running during 'STOP' mode is offered. RTC will be out of function if OSC stops in 'STOP' mode.

(c) WAIT The WAIT instruction causes all processor clocks to stop, but leaves the Timer, SCI and SPI clock running. This "rest" state of the processor can be cleared by RESET#, external IRQ1#, IRQ2#, RTC, Keyboard, Timer, SCI or SPI interrupt. There are no special wait Vectors for these individual interrupts.

3.2.2 SOFTWARE INTERRUPT (SWI)

The software interrupt is an executable instruction. The action of the SWI instruction is similar to the hardware interrupts. The SWI is executed regardless of the state of the interrupt mask in the condition code register. The service routine address is specified by the contents of memory location $0C0C and $0C0D.

3.2.3 EXTERNAL INTERRUPT

The IRQ1# and IRQ2# interrupt signals will be explained in this paragraph. If the signal of the external interrupt pins (IRQ1#, IRQ2#) satisfy the condition selected by the IR1O and IR2O in the control register (location $26), then an external interrupt occurs. The actual processor interrupt is generated only if the interrupt mask bit of the condition code register is cleared. When the interrupt is recognised, the current state of the machine is pushed onto the stack and the interrupt mask bit in the condition code register is set. This masks further interrupts until the present one is serviced. The service routine address is specified by the contents of $0C0A and $0C0B. The interrupt logic recognises negative edge transitions and pulses (special case of negative edges) on the external interrupt lines. Figure 3-4 shows both a block diagram and timing for the interrupt lines (IRQ1#, IRQ2#) to the processor. It indicates pulses on the interrupt line to be serviced. The minimum time between pulses is a function of the number of cycles required to execute the interrupt service routine plus 21 cycles. Once a pulse occurs, the next pulse should not occur until the MCU software has exited the routine (an RTI occurs).

NOTE

The internal interrupt latch is cleared in the first part of the service routine; therefore, one (and only one) external interrupt pulse could be latched during t_{ILIL} and serviced as soon as the I bit is cleared.

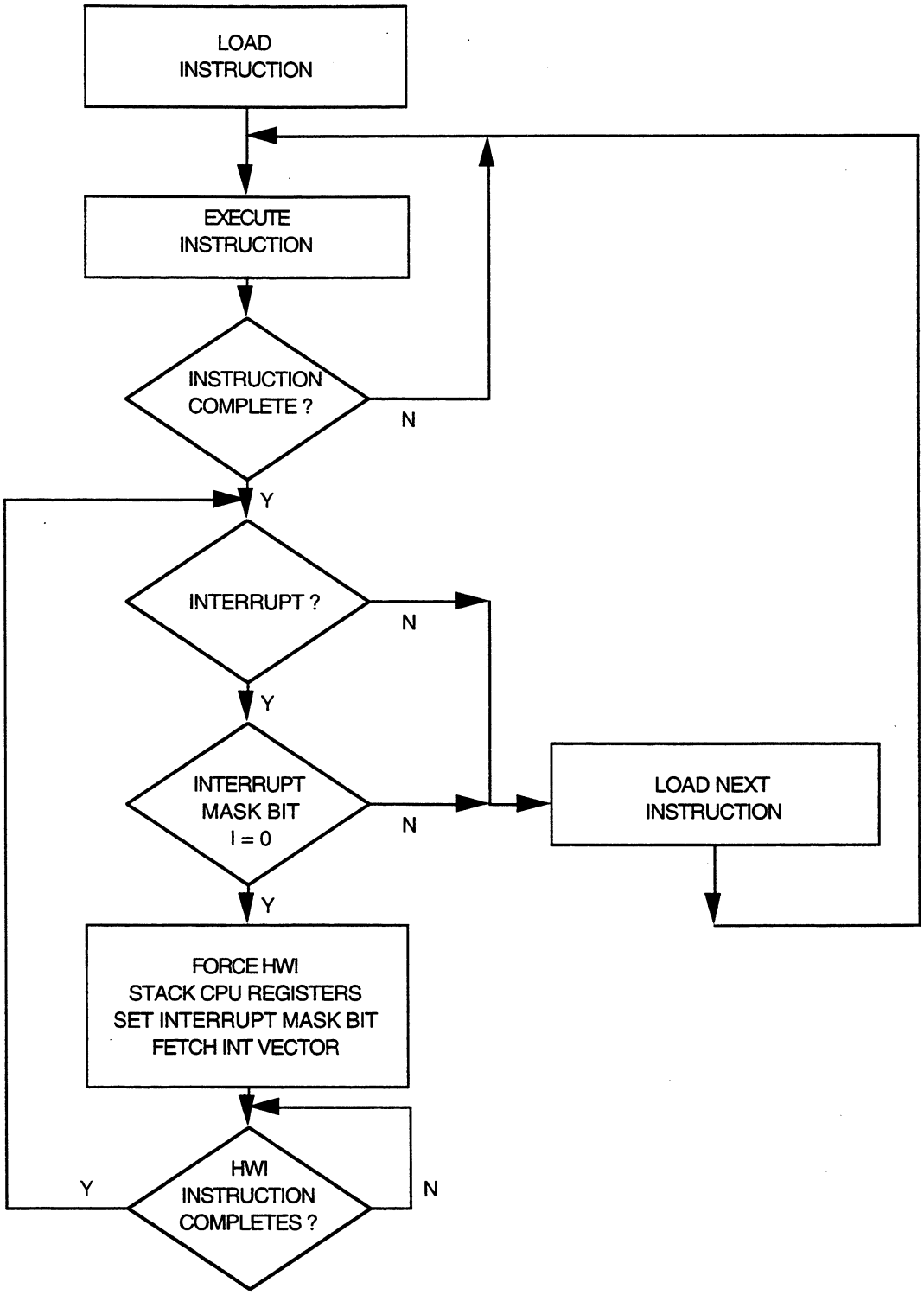


Fig. 3-2 HARDWARE INTERRUPT FLOWCHART

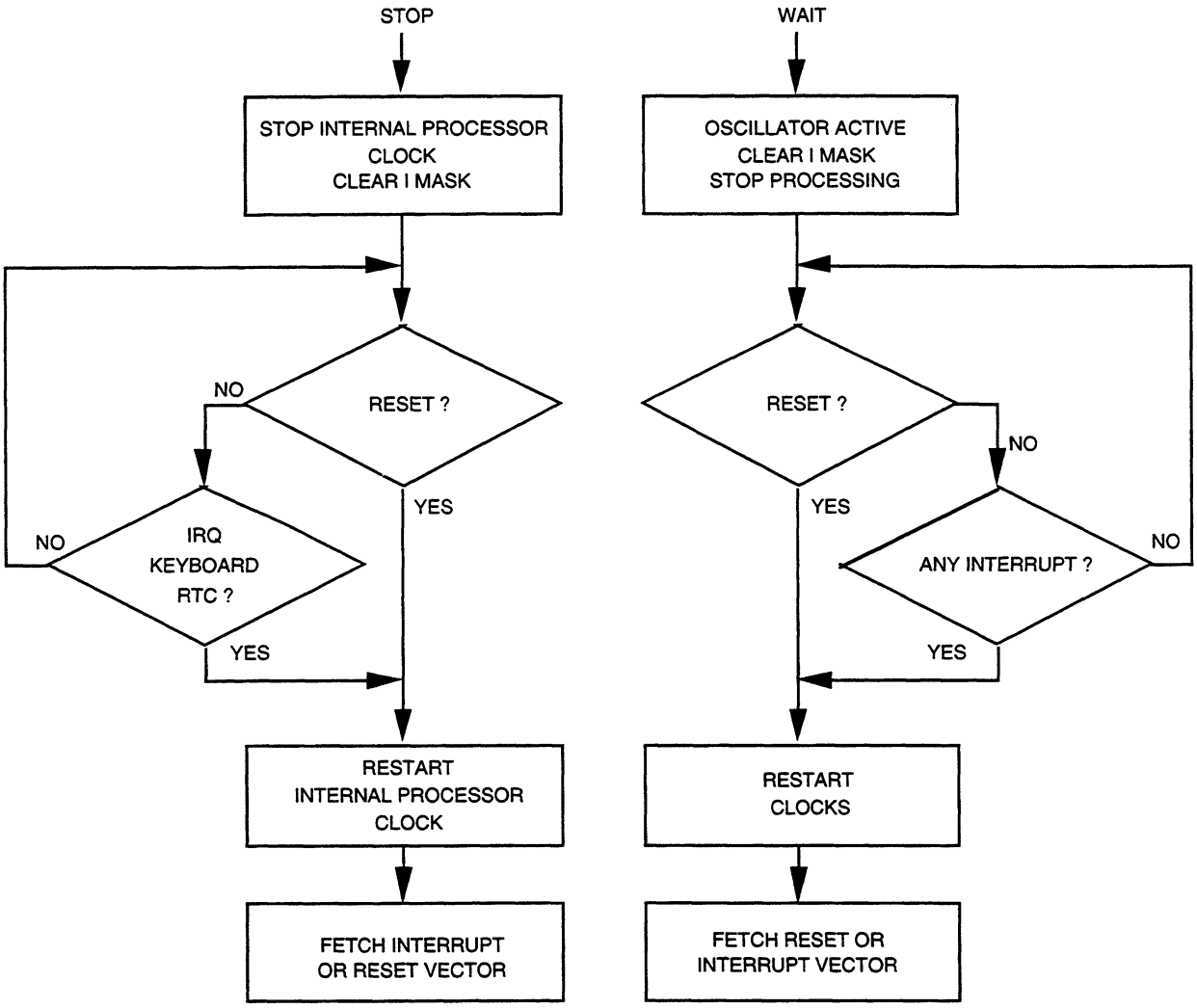


FIG. 3-3 STOP/WAIT FLOWCHARTS

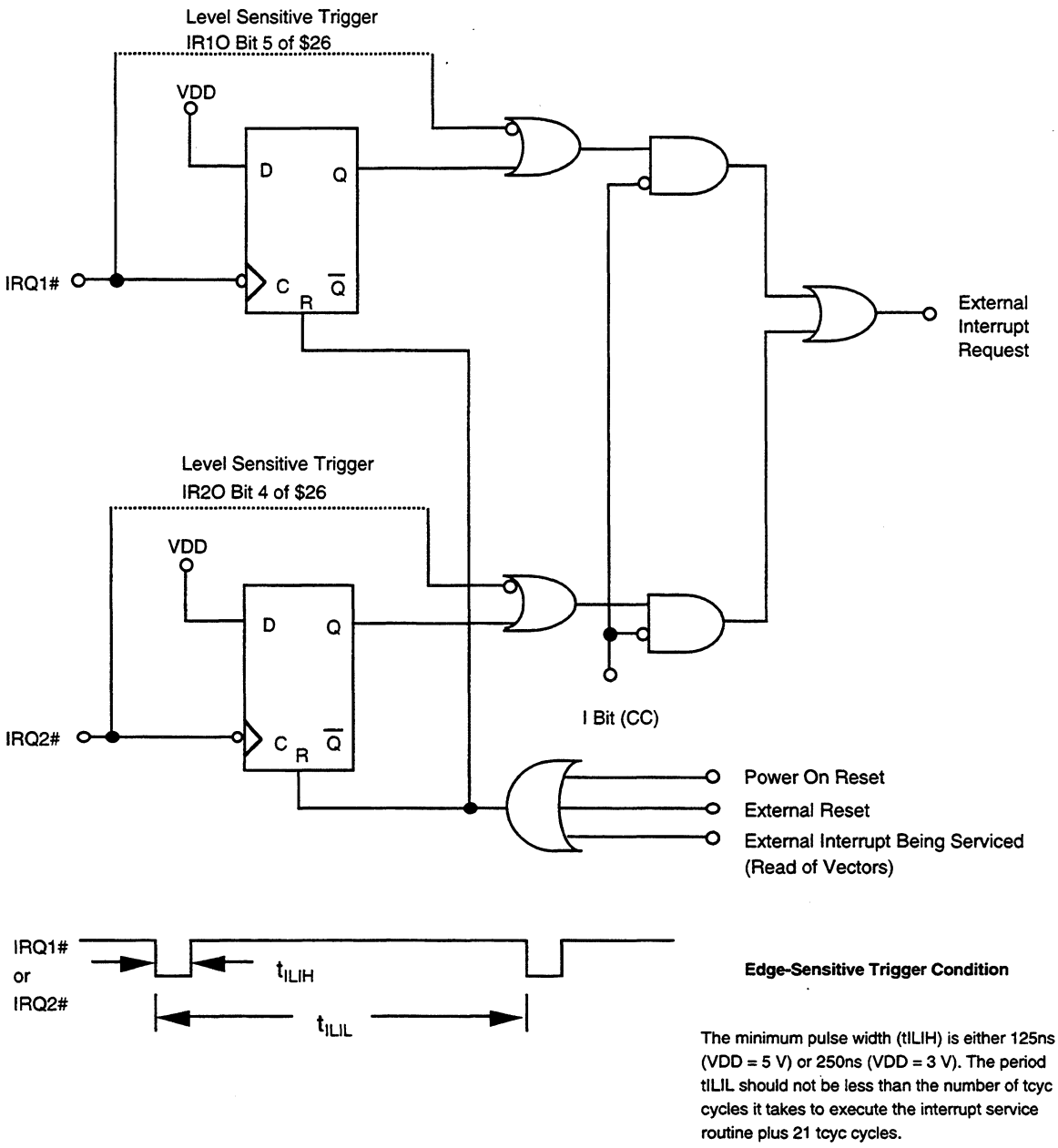


FIG. 3-4 External Interrupt

3.2.4 TIMER INTERRUPT

The three Timer interrupt flags are found in the three most significant bits of the Timer Status Register (TSR). All three interrupts will vector to the same service routine location.

Each flag bit is defined as follows:

Bit 5 TOF — Timer Overflow Flag - TOF is set during the Counter transition of $FFFF to $0000. It is cleared by reading the TSR (with TOF set) followed by reading the counter least significant byte ($19). Reset does not affect this bit.

Bit 6 OCF — Output Compare Flag - OCF is set when the Output Compare Register matches the Counter Register. It is cleared by reading the TSR (with OCF set) and then accessing the Output Compare Register least significant byte ($17). Reset does not affect this bit.

Bit 7 ICF — Input Capture Flag - ICF is set when a proper edge has been sensed by the input capture edge detector. It is cleared by reading the TSR (with ICF set) followed by reading the Input Capture Register least significant byte ($15). Reset does not affect this bit.

All interrupt flags have corresponding enable bits (ICIE, OCIE, and TOIE) found in the Timer Control Register. Reset clears all enable bits preventing an interrupt from occurring. The actual processor interrupt is generated only if the interrupt mask bit of the condition code register is cleared. When the interrupt is recognised, the current state of the machine is pushed onto the stack and the interrupt mask bit in the condition code register is set. This masks further interrupts until the present one is serviced. The service routine address is specified by the contents of $0C06 and $0C07. Refer to **SECTION 4 PROGRAMMABLE TIMER** for additional information about the timer circuitry.

3.2.5 SCI INTERRUPTS

An interrupt in the SCI system will occur when one of the interrupt bits in the SCI Status Register (SCSR) is set, provided the interrupt mask bit of the condition code register is cleared and the enable bit in the SCI Control Register 2 (SCCR2) is enabled. When the interrupt is recognised, the current state of the machine is pushed onto the stack and the interrupt mask bit in the condition code register is set. This masks further interrupts until the present one is serviced. The interrupt causes the program counter to vector to memory location $0C04-$0C05 which contains the service routine's starting address. Software in the serial interrupt service routine must determine the priority and cause of the SCI interrupt by examination of the interrupt flags and status bits located in the SCI Status Register. The general sequence for clearing an interrupt is a software sequence of reading the status register while the flag is set followed by a read or write of an associated register. Refer to **SECTION 5 SERIAL COMMUNICATION INTERFACE** for a description of the SCI system and its interrupt.

3.2.6 KEYBOARD INTERRUPT

A keyboard interrupt is enabled when KEYE bit in the control register $27 is set, provided the interrupt mask bit of the condition code register is cleared. When the interrupt is recognised, the current state of the machine is pushed onto the stack and the interrupt mask bit in the condition code register is set. This masks further interrupts until the present one is serviced. The interrupt causes the program counter to vector to memory location $0C08 - $0C09 which contains the service routine's starting address. When KEYE bit is set and port A data direction register configures as input port, a 250 K OHM pull up resistor will associate with each pin of port A. A write to port A resets keyboard interrupt.

Any transition of one of the PORT A pins from 1 to 0 would cause a keyboard interrupt. This interrupt will be latched internally and will not be cleared even if the cause of interrupt pin return to 1 or the KEYE bit is reset to zero. The interrupt flag can only be cleared by performing a 'write' to PORT A. So it is advised that in the keyboard interrupt service routine, the CPU should perform a 'write' to PORT A before it clears the interrupt mask flag or execute an RET instruction.

3.2.7 RTC INTERRUPT

A RTC interrupt is enabled when either RTCE , ALE or SECE bit in the control register $27 is set, provided the interrupt mask bit of the condition code register is cleared. When RTCE bit is set, the real time clock will interrupt the CPU once a day. This will occur when the hours register in the real time clock register changes from twenty-three to zero. When the SECE bit is set, the real time clock will interrupt the CPU once every second. When the ALE bit is set, the RTC interrupt will occur when the value of the hour alarm and hours are equal, and the value of the minute alarm and minutes are equal. When the interrupt is recognised, the current state of the machine is pushed onto the stack and the interrupt mask bit in the condition code register is set. This masks further interrupts until the present one is serviced. The interrupt causes the program counter to vector to $0C02 - $0C03, which contains the service routine's starting address. Interrupt per day, per second, or alarm interrupt can be distinguished by the RTCF, SECF and ALF flags. In order to reset the interrupt, the user is responsible for clearing the appropriate flags when executing the interrupt routine. Refer to **SECTION 6.4 REAL TIME CLOCK** for additional information about the Real Time Clock (RTC) interrupt.

3.2.8 SPI INTERRUPT

An interrupt in the serial peripheral interface (SPI) occurs when one of the interrupt flag bits in the serial peripheral status register (location $2B) is set, provided the I bit in the condition code register is cleared and the enable bit in the serial peripheral control register (location $2A) is enabled. When the interrupt is recognised, the current state of the machine is pushed onto the stack and the I bit in the condition code register is set. This masks further interrupts until the present one is serviced. The SPI interrupt causes the program counter to vector to $0C00 -$0C01 which contains the starting address of the interrupts service routine. Software in the serial peripheral interrupt service routine must determine the priority and the cause of the SPI interrupt by examining the interrupt flag bits located in the SPI status register. The general sequence of clearing an interrupt is a software sequence of accessing the status register while the flag is set, followed by a read or write of an associated register. Refer to **SECTION 7 SERIAL PERIPHERAL INTERFACE** for a description of the SPI system and its interrupts.

3.3 LOW POWER MODES

3.3.1 LOW POWER MODES INTRODUCTION

The STOP and WAIT instructions have different effects on the programmable timer, the serial communications interface, serial peripheral interface and real time clock. These different effects are discussed separately in the following sections.

3.3.2 STOP MODE

The STOP instruction places the MC68HC05L10 in its lowest power consumption mode. In the STOP mode the internal processor clock is turned off, causing all internal processing to be halted; refer to Fig. 3-3. During the STOP mode, the I bit in the condition code register is cleared to enable external IRQ1# or IRQ2#, RTC and Keyboard interrupt. All other registers and memory remain unaltered and all input/output lines remain unchanged. This continues until an external IRQ1#, IRQ2#, RTC or Keyboard interrupt, or reset is sensed at which time the internal processor clock is turned on. The external IRQ1#, IRQ2#, RTC, keyboard interrupt, or reset causes the program counter to vector to memory location (depends on the kind of interrupt or reset) which contains the starting address of the interrupt or reset service routine respectively. The effects of the stop mode on each of the MCU systems (Timer, RTC, SCI and SPI) are described separately.

3.3.2.1 TIMER DURING STOP MODE

When the MCU enters the stop mode, the timer counter stops counting (the internal processor clock is stopped) and remains at that particular count value until the stop mode is exited by an interrupt (if exited by reset the counter is forced to $FFFC). If the stop mode is exited by an external interrupt on the IRQ1#, IRQ2# pin, or interrupt from KEYBOARD or RTC, then the counter resumes from its stop value as if nothing had happened. Another feature of the programmable timer, in the stop mode, is that if at least one valid input capture edge occurs at the TCAP pin, the input capture detect circuitry is armed. This action does not set any timer flags or "wake up" the MCU, but when the MCU does "wake up" there will be an active input capture flag (and data) from that first valid edge which occurred during stop mode. If the stop mode is exited by an external reset (logic low on RESET# pin), then no such input capture flag or data action takes place even if there was a valid input capture edge (at the TCAP pin) during MCU stop mode.

3.3.2.2 SCI DURING STOP MODE

When the MCU enters the stop mode, the baud rate generator which drives the receiver and transmitter is shut down. This essentially stops all SCI activity. The receiver is unable to receive and the transmitter is unable to transmit. If the STOP instruction is executed during a transmitter transfer, that transfer is halted. When the stop mode is exited, that particular transmission resumes (if the exit mode is the result of an external low on the IRQ1#, IRQ2# pin, interrupt from KEYBOARD or RTC). Since the previous transmission resumes after an IRQ1#, IRQ2#, KEYBOARD or RTC interrupt stop mode exit, the user should ensure that the SCI transmitter is in the idle state when the STOP instruction is executed. If the receiver is receiving data when STOP instruction is executed, received data sampling is stopped (baud rate generator stops) and the rest of the data is lost. For the above reasons, the SCI should be in the idle state when the STOP instruction is executed.

3.3.2.3 RTC DURING STOP MODE

Since RTC runs on a 32KHz oscillator clock (if MASK OPTION of OSC is chosen to be running during 'STOP' mode), STOP instruction has no effect on the OSC and RTC.

3.3.2.4 SPI DURING STOP MODE

When the processor executes the STOP instruction, the internal oscillator is turned off. This halts all internal CPU processing, including the operation of the serial peripheral interface. The only way for the MCU to "wake up" from the stop mode is by receipt of an external interrupt (logic low on IRQ1# or IRQ2# pin), interrupt from keyboard, RTC, or by the detection of a reset (logic low on RESET# pin or a power-on reset).

When the MCU enters the stop mode, the baud rate generator which drives the SPI shuts down. This essentially stops all master mode SPI operation, thus the master SPI is unable to transmit or receive any data. If the STOP instruction is executed during an SPI transfer, that transfer is halted until the MCU exits stop mode (provided it is an exit resulting from a logic low on the IRQ1#, IRQ2# pin, interrupt from keyboard, RTC, or by the detection of a reset of logic low on reset pin or a power on reset). If the stop mode is exited by a reset, then the appropriate control/status bits are cleared and the SPI is disable. If the device is in the slave mode when the STOP instruction is executed, the slave SPI will still operate. It can still accept data and clock information in addition to transmitting its own data back to a master device.

3.3.3 WAIT MODE

The WAIT instruction places the MC68HC05L10 in a low power consumption mode, but the WAIT mode consumes somewhat more power than the STOP mode. In the WAIT mode, the internal clock remains active, but all CPU processing is stopped; however, the programmable timer, serial communications interface, serial peripheral interface and real time clock remain active. Refer to Fig. 3-3. During the WAIT mode, the I bit in the condition code register is cleared to enable all interrupts. In fact an interrupt from timer, SCI, SPI, KEYBOARD or RTC (in addition to a logic low on the IRQ# or RESET# pin) cause the processor to exit the wait mode. All other registers and memory remain unaltered and all parallel input/output lines remain unchanged. This continues until a interrupt or reset is sensed. At this time the program counter vectors to the memory location ($0C00 through $0C0F) which contains the starting address of interrupt or reset service routine.

The wait mode power consumption depends on how many systems are active. The power consumption will be the highest when all systems (timer, TCMP, SCI and SPI) are active. The power consumption will be the least when the SCI and SPI are disabled (timer operation cannot be disabled in the wait mode). If a non-reset exit from the wait mode is performed (eg. timer overflow interrupt exit), the state of the remaining peripherals will be unchanged. If a reset exit from the wait mode is performed all systems will revert to the disabled reset state.

3.3.3.1 SCI DURING WAIT MODE

The SCI system is not affected by the WAIT mode and continues regular operation. Any valid SCI interrupt will wake the system up.

3.3.3.2 TIMER DURING WAIT MODE

The TIMER system is not affected by the WAIT mode, it continues regular operation. Any valid TIMER interrupt will wake the system up.

3.3.3.3 SPI DURING WAIT MODE

When the MCU enters the wait mode, the CPU clock is halted, all CPU action is suspended; however, the SPI system remains active. In fact, an interrupt from the SPI (in addition to a logic low on the IRQ1#, IRQ2# or interrupt from keyboard or RTC or a logic low on the RESET# pin or a power on reset) causes the processor to exit the wait mode.

3.3.3.4 REAL TIMER CLOCK DURING WAIT MODE

The RTC system is not affected by the WAIT mode, it continues regular operation. Any valid RTC interrupt will wake the system up.

3.4 DATA RETENTION MODE

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

SECTION 4
PROGRAMMABLE TIMER

4.1 INTRODUCTION

The timer consists of a 16-bit free-running counter driven by a fixed divide-by-four prescaler. This timer can be used for many purposes, including input waveform measurements while simultaneously generating an output waveform. Pulse widths can vary from several microseconds to many seconds.

Because of the 16-bit timer architecture, the I/O registers for the input capture and output compare functions are pairs of 8-bit registers (high byte and low byte). Generally, accessing the low byte of a specific timer function allows full control of that function. However, an access of the high byte inhibits that specific timer function until the low byte is also accessed.

NOTE

The I bit in the 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 programmable timer capabilities are provided by using the following ten addressable 8-bit registers (note the high and low represent the significant of the byte). A description of each register is provided in the following sections.

Timer Control Register (TCR) - location $12
Timer Status Register (TSR) - location $13
Input Capture High Register - location $14
Input Capture Low Register - location $15
Output Compare High Register - location $16
Output Compare Low Register - location $17
Counter High Register - location $18
Counter Low Register - location $19
Alternate Counter High Register - location $1A
Alternate Counter Low Register - location $1B

4.2 COUNTER

The key element in the programmable timer is a 16-bit, free-running counter or counter register, preceded by a prescaler that divides the internal processor clock by four. The prescaler gives the timer a resolution of 3.255 μs if the internal bus clock is 1.2288 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.

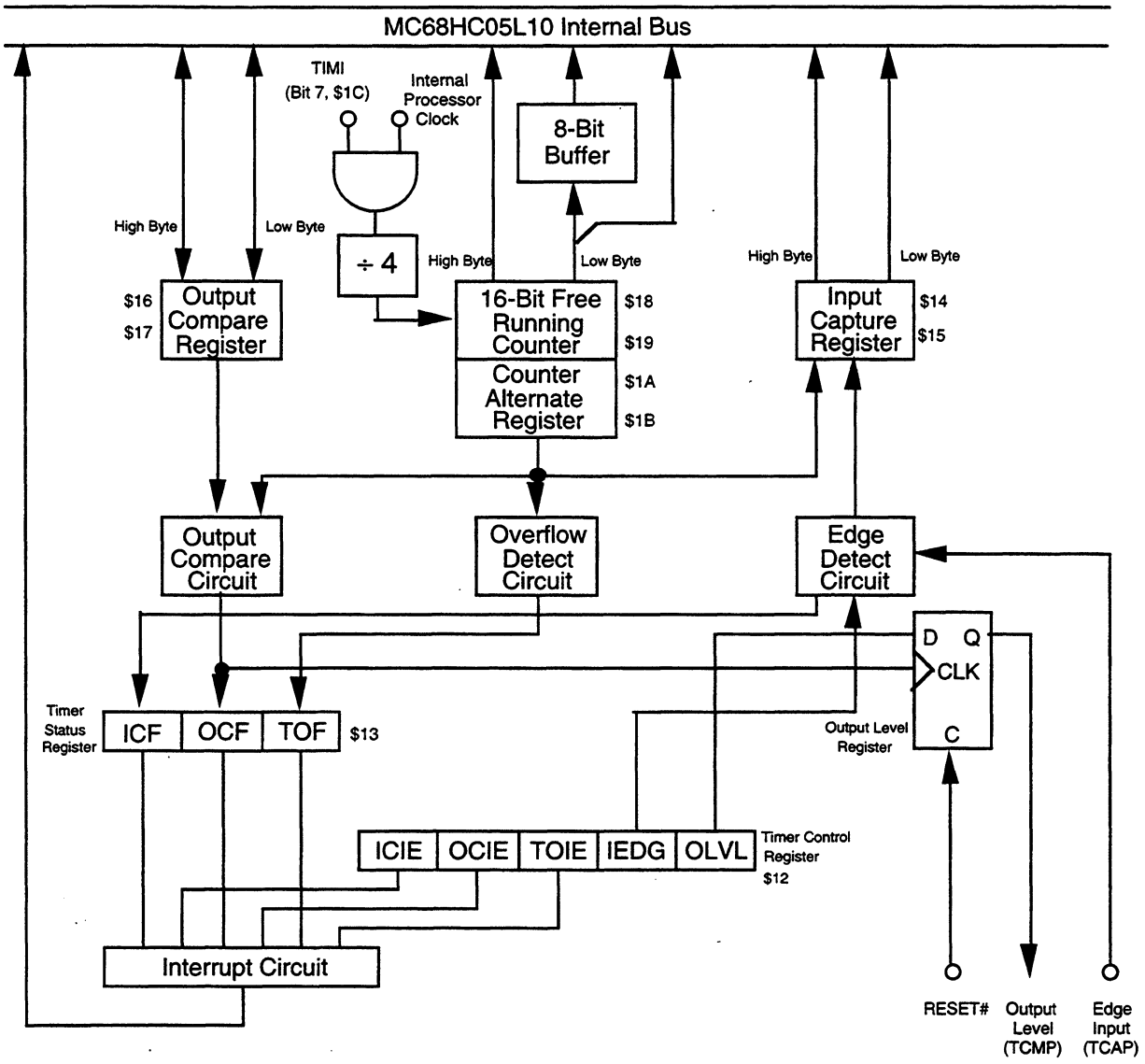
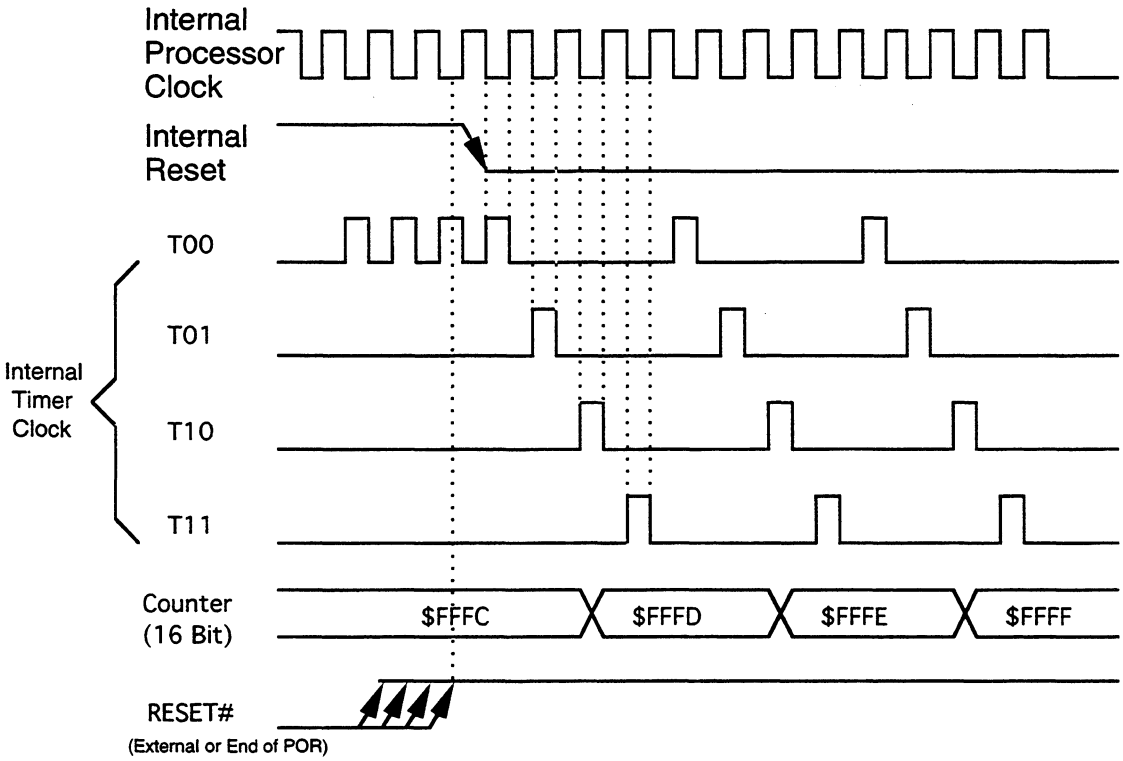
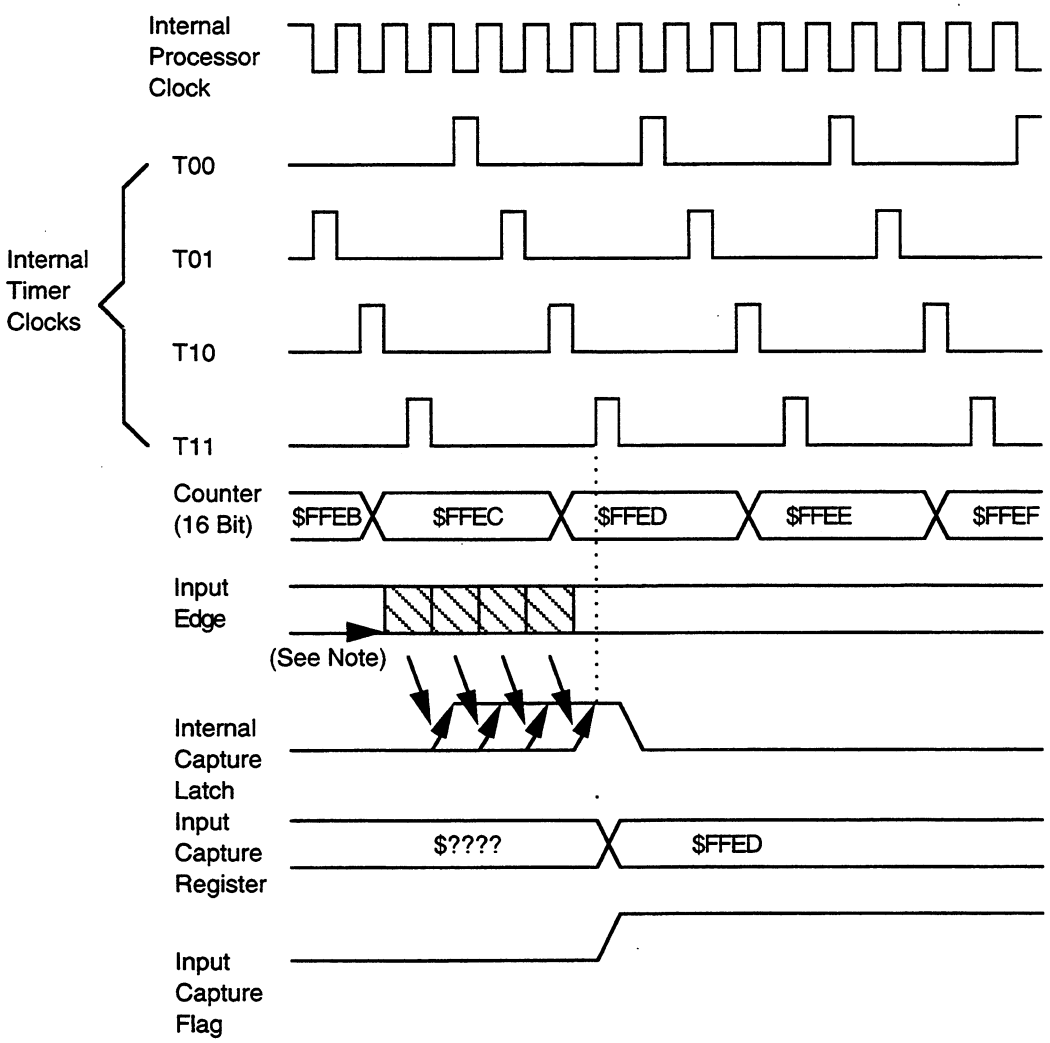


Fig. 4-1 Programmable Timer Block Diagram



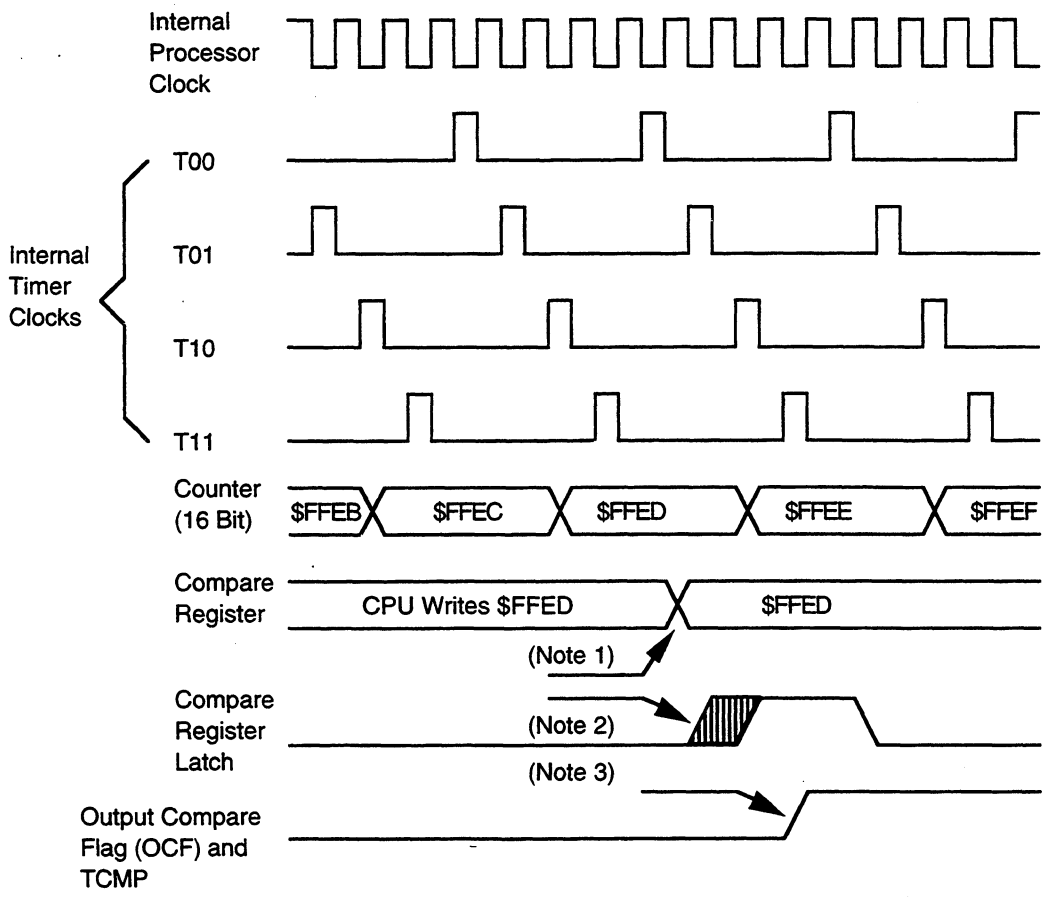
NOTE: The Counter Register and Timer Control Register are the only ones affected by RESET

Fig. 4-2 Timer State Timing Diagram For Reset



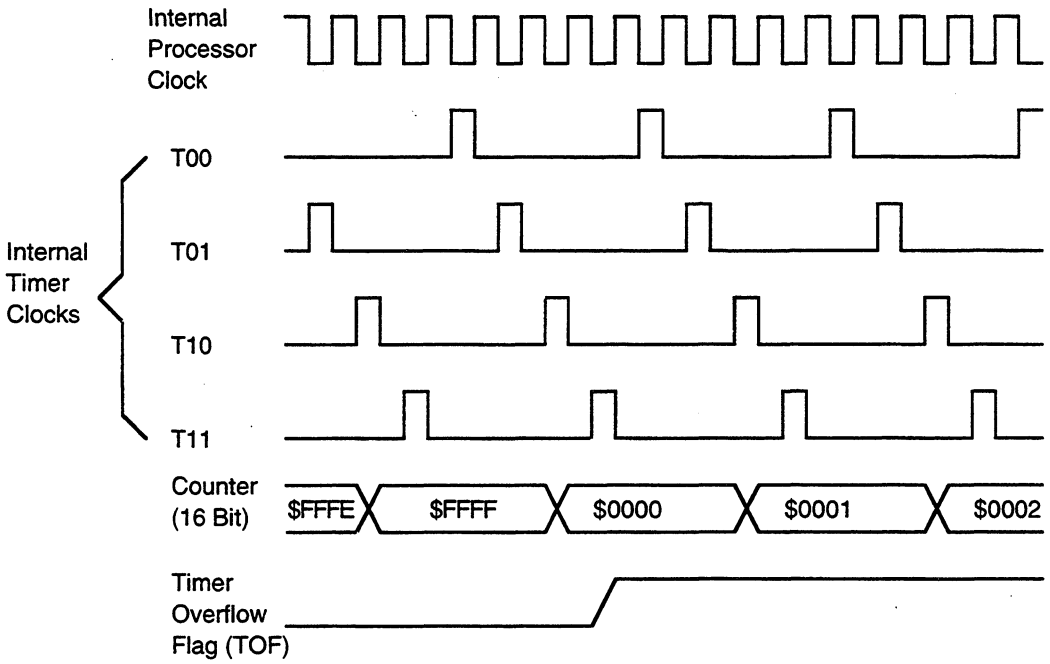
NOTE: If the input edge occurs in the shaded area from one timer state T10 to the other timer state T10 the input capture flag is set during the next state T11.

Fig. 4-3 Timer State Timing Diagram for Input Capture



NOTES: 1. The CPU write to the compare register may take place at any time, but a compare only occurs at timer state T01. Thus, a 4-cycle difference may exist between the write to the compare register and the actual compare.
2. Internal compare takes place during timer state T01.
3. OCF is set at timer state T11 which follows the comparison match ($FFED in this example).

Fig. 4-4 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.

Fig.4-5 Timer State Diagram for Timer Overflow

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

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

The free-running counter is configured 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 divide-by-four prescaler, the value in the free-running counter repeats every 262,144 internal bus clock cycles. When the counter rolls over from $FFFF to $0000, the TOF bit is set. An interrupt can also be enabled when counter rollover occurs by setting its interrupt enable bit (TOIE).

In some particular timing control applications it may be desirable to reset the 16-bit free running counter under software control. When the low byte of the counter ($19 or $1B) is written to, the counter is configured to its reset value ($FFFC).

The divide-by-4 prescaler is also reset and the counter resumes normal counting operation. All of the flags and enable bits remain unaltered by this operation. If access has previously been made to the high byte of the free running counter ($18 or $1A), then the reset counter operation terminates the access sequence.

4.3 OUTPUT COMPARE REGISTER

The 16-bit output compare register is made up of two 8-bit registers at locations $16 (MSB) and $17 (LSB). The output compare register is used for several purposes, such as indicating when a period of time has elapsed. All bits are readable and writable, and are not altered by the timer hardware or reset. If the compare function is not needed, the two bytes of the output compare register can be used as storage locations.

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

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

4.4 INPUT CAPTURE REGISTER

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

The result obtained by an input capture will be one more than the value of the free-running counter on the rising edge of the internal bus clock preceding the external transition. This delay is required for internal synchronisation. Resolution is one count of the free-running counter, which is four internal bus clock cycles.

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

After a read of the input capture register ($14) MSB, the counter transfer is inhibited until the LSB ($15) is also read. This characteristic causes the time used in the input capture software routine and its interaction with the main program to determine the minimum pulse period.

A read of the input capture register LSB ($15) does not inhibit the free-running counter transfer since they occur on opposite edges of the internal bus clock.

4.5 TIMER CONTROL REGISTER (TCR)

The TCR is a read/write register containing five control bits. Three bits control interrupts associated with each of the three flag bits found in the timer status register (discussed below). The other two bits control: 1) which edge is significant to the input capture edge detector (ie., negative or positive), and 2) the next value to be clocked to the output level register in response to a successful output compare. The timer control register and the free running counter are the only sections of the timer affected by reset. The TCMP pin is forced low during external reset and stays low until a valid compare changes it to high. The timer control register is illustrated below by a definition of each bit.

7	6	5	4	3	2	1	0	
ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL	$12

B7, ICIE — If the input capture interrupt enable (ICIE) bit is set, a timer interrupt is enable whenever the ICF status flag (in the timer status register) is set. If the ICIE bit is cleared, the interrupt is inhibited. The ICIE bit is cleared by reset.

B6, OCIE — If the output compare interrupt enable (OCIE) bit is set, a timer interrupt is enabled whenever the OCF status flag is set. If the OCIE bit is cleared, the interrupt is inhibited. The OCIE bit is cleared by reset.

B5, TOIE — If the timer overflow interrupt enable (TOIE) bit is set, a timer interrupt is enabled whenever the TOF status flag is set. If the TOIE bit is cleared, the interrupt is inhibited. The TOIE bit is cleared by reset.

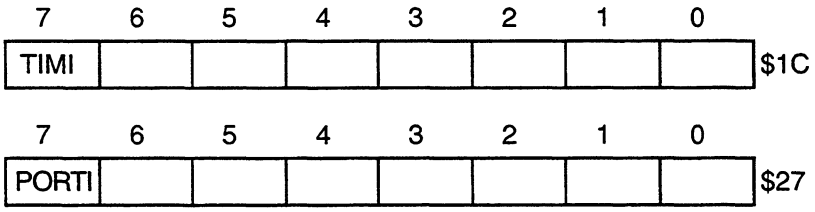
B1, IEDG — The value of the input edge (IEDG) bit determines which level transition on TCAP pin will trigger a free running counter transfer to the input capture register. Reset does not affect the IEDG bit.

0 = negative edge
1 = positive edge

B0, OLVL — The value of the output level (OLVL) bit is clocked into the output level register by the next successful output compare and will appear at TCMP pin. This bit and the output level register are cleared by reset.

0 = low output
1 = high output

4.6 OTHER TIMER FUNCTION CONTROL BITS



B7 ($27), PORTI, B7 ($1C), TIMI — These bits are cleared by power on reset or external reset.

TIMI	PORTI	TIMER	PD6	PD7
0	0	ENABLE	TCAP	TCMP
0	1	*ENABLE	INPUT PORT	INPUT PORT
1	0	DISABLE	INPUT PORT	INPUT PORT
1	1	DISABLE	INPUT PORT	INPUT PORT

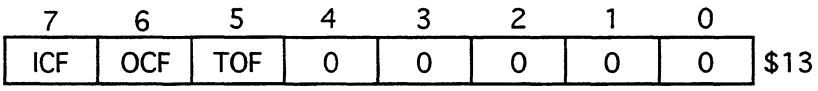
* TCAP and TCMP not available to external

4.7 TIMER STATUS REGISTER (TSR)

The TSR is a read-only register containing three status flag bits. These three bits indicate the following:

1. A proper transition has taken place at TCAP pin with an accompanying transfer of the free running counter contents to the input capture register,
2. A match has been found between the free running counter and the output compare register, and
3. A free running counter transition from $FFFF to $0000 has been sensed (timer overflow).

The timer status register is illustrated below followed by a definition of each bit. Refer to timing diagrams shown in Figures 4-2, 4-3, and 4-4 for timing relationship to the timer status register bits.



B7, ICF — The input capture flag (ICF) is set when a proper edge has been sensed by the input capture edge detector. It is cleared by a processor access of the timer status register (with ICF set) followed by accessing the low byte ($15) of the input capture register. Reset does not affect the input compare flag.

B6, OCF — The output compare flag (OCF) is set when the output compare register contents matches the contents of the free running counter. The OCF is cleared by accessing the timer status register (with OCF set) and then accessing the low byte (($17) of the output compare register. Reset does not affect the output compare flag.

B5, TOF — The timer overflow flag (TOF) bit is set by transition of the free running counter from $FFFF to $0000. It is cleared by accessing the timer status register (with TOF set) followed by an access of the free running counter least significant byte ($19). Reset does not affect the TOF bit.

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

A problem can occur when using the timer overflow function and reading the free-running counter at random times to measure an elapsed time. Without incorporating the proper precautions into software, the timer overflow flag could unintentionally be cleared if: 1)The timer status register is read or written when TOF is set, and 2)The LSB of the free-running counter is read but not for the purpose of servicing the flag. The counter alternate register at address $1A and $1B contains the same value as the free-running counter (at address $18 and $19); therefore, this alternate register can be read at any time without affecting the timer overflow flag in the timer status register.

During STOP and WAIT instructions, the programmable timer functions as follows: during the wait mode, the timer continues to operate normally and may generate an interrupt to trigger the CPU out of the wait state; during the stop mode, the timer holds at its current state, retaining all data, and resumes operation from this point when external interrupt (IRQ), RTC or Keyboard interrupt is received.

SECTION 5
SERIAL COMMUNICATIONS INTERFACE (SCI)

5.1 INTRODUCTION

A full-duplex asynchronous SCI is provided with a standard NRZ format and a variety of baud rates. The SCI transmitter and receiver are functionally independent, but uses the same data format and baud rate. The terms baud and bit rate are used synonymously in the following description.

5.1.1 SCI TWO-WIRE SYSTEM FEATURES

- Standard NRZ (mark/space) format
- Advanced error detection method includes noise detection for noise duration of up to 1/16 bit time
- Full-duplex operation (simultaneous transmit and receive)
- Software programmable for different baud rates
- Software-selectable word length (eight or nine bit words)
- Separate transmitter and receiver enable bits
- SCI may be interrupt driven
- Four separate interrupt conditions

5.1.2 SCI RECEIVER FEATURES

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

5.1.3 SCI TRANSMITTER FEATURES

- Transmit data register empty flag
- Transmit complete flag
- Break send

Any SCI two-wire system requires receive data in (RDI) and transmit data out (TDO).

5.2 DATA FORMAT

Receive data in (RDI) or transmit data out (TDO) is the serial data presented between the internal data bus and the output pin (TDO) and between the input pin (RDI) and the internal data bus. Data format is as shown for the NRZ in Fig. 5-1 and must meet the following criteria:

1. A high level indicates a logic one and a low level indicates a logic zero.
2. The idle line is in a high (logic one) state prior to transmission/reception of a message.
3. A start bit (logic zero) is transmitted/received indicating the start of a message.
4. The data is transmitted and received least-significant-bit first.
5. A stop bit (high in the tenth or eleventh bit position) indicates the byte is complete.
6. A break is defined as the transmission or reception of a low (logic zero) for some multiple of the data format.

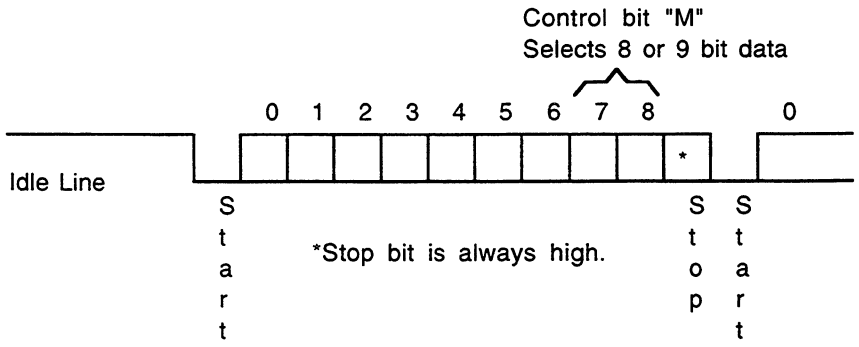


FIG. 5-1 Data Format

5.3 WAKE-UP FEATURE

In a typical multiprocessor configuration, the software protocol will usually identify the addressee(s) at the beginning of the message. To permit uninterested MPUs to ignore the remainder of the message, a wake-up feature is included, whereby all further SCI receiver flag (and interrupt) processing can be inhibited until its data line returns to the idle state. An SCI receiver is re-enabled by an idle string of at least ten (or eleven) consecutive ones. Software for the transmitter must provide for the required idle string between consecutive messages and prevent it from occurring within messages.

A second wake-up method is available in which sleeping SCI receivers can be awakened by a logic one in the high-order bit of a received character.

5.4 RECEIVE DATA IN

Receive data in (RDI) is the serial data which is presented from the input pin via the SCI to the receive data register (RDR). While waiting for a start bit, the receiver samples the input at a rate 16 times higher than the set baud rate. This increased rate is referred to as the RT rate. When the input (idle) line is detected low, it is tested for three more sample times. If at least two of these three samples detect a logic low, a valid start bit is assumed to be detected. If in two or more samples, a logic high is detected, the line is assumed to be idle. The receive clock generator is controlled by the baud rate register (see Fig. 5-6 and 5-7); however, the SCI is synchronised by the start bit independent of the transmitter.

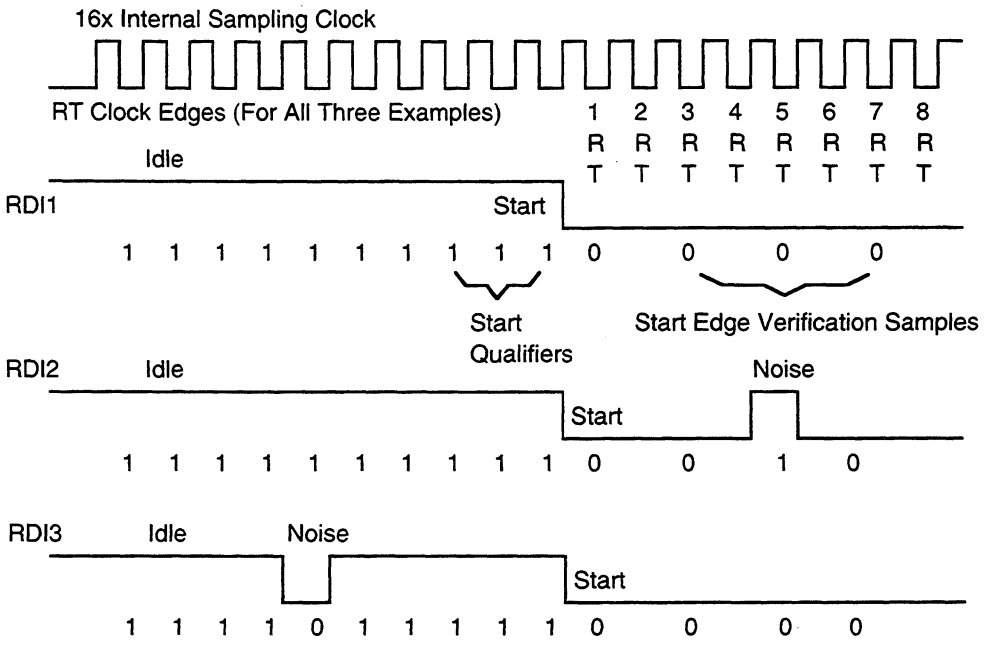


FIG. 5-2 Example of Start-Bit Sampling Technique

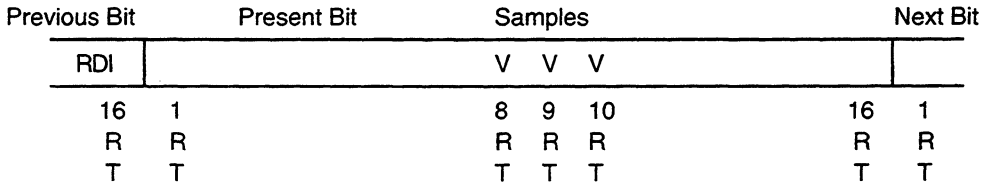


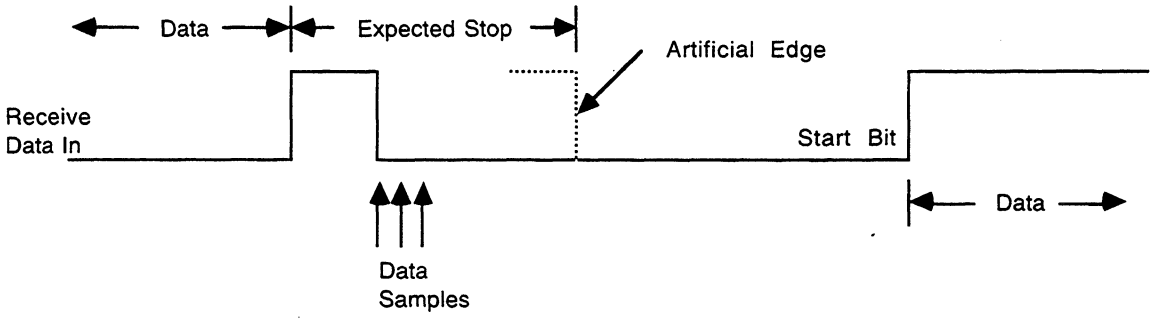
FIG. 5-3 Sampling Technique Used on All Bits

Once a valid start bit is detected, the start bit, each data bit, and the stop bit are each sampled three times. The value of the bit is determined by voting logic, which takes the value of a majority of samples. A noise flag is set when all three samples on a valid start bit, data bit, or stop bit do not agree. A noise flag is also set when the start verification samples do not agree.

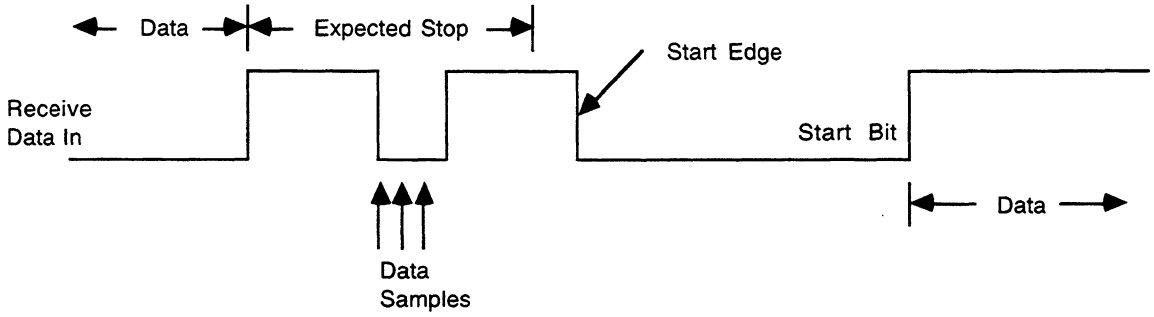
5.5 START BIT DETECTION FOLLOWING A FRAMING ERROR

If there has been a framing error (FE) without detection of a break (10 zeros for a 8-bit format or 11 zeros for a 9-bit format), the circuit continues to operate as if there actually were 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 Fig. 5-2) are forced into the sample shift register during the interval when detection of a start bit is anticipated (see Fig. 5-4); therefore, the start bit will be accepted no sooner than it is anticipated.

If the receiver detects that a break (RDRF=1, FE=1, receiver data register=$00) produced the framing error, the start bit will not be artificially induced, and the receiver must actually receive a logic one before start. See Fig. 5-5.



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



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

FIG. 5-4 SCI Artificial Start Following A Framing Error

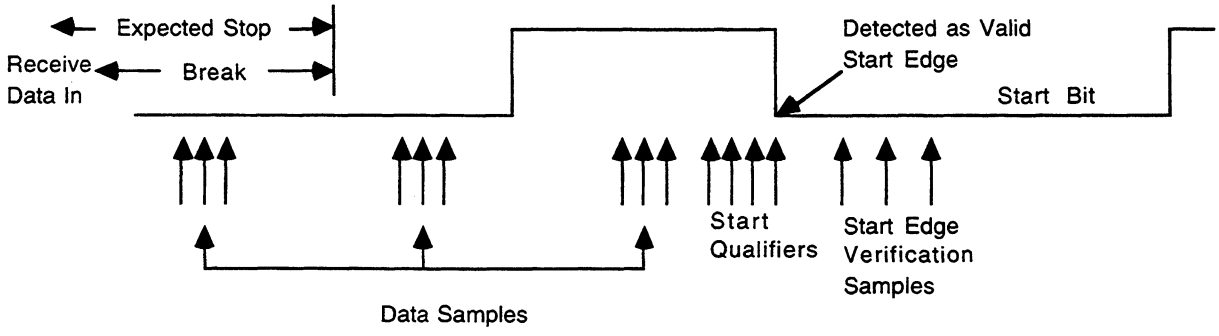
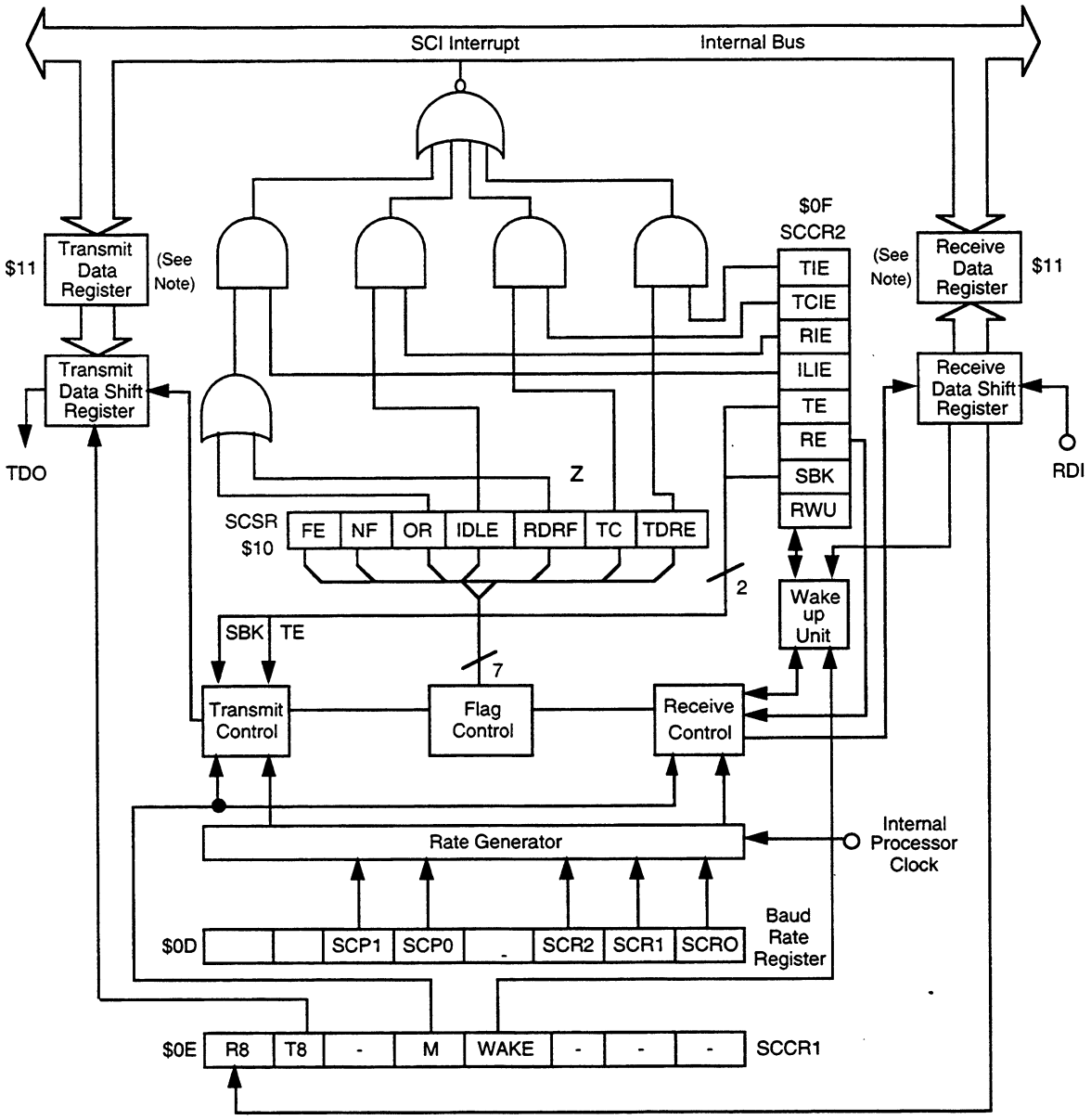


FIG. 5-5 SCI Start Bit Following A Break



NOTE: The Serial Data Communications Data Register (SCDAT) is controlled by the R/W signal. It is the transmit data register when written and receive data register when read.

FIG. 5-6 Serial Communications Interface Block Diagram

5.6 TRANSMIT DATA OUT (TDO)

Transmit data out is the serial data which is presented from the internal data bus via the SCI and then to the output pin. Data format is as discussed above and shown in Fig. 5-1. The transmitter generates a bit time by using a derivative of the RT clock, thus producing a transmission rate equal to 1/16 that of the receiver sample clock.

5.7 REGISTERS

There are five registers used in the SCI; the internal configuration of these registers is discussed in the following paragraphs. A block diagram of the SCI system is shown in Fig. 5-6.

5.7.1 Serial Communications Data Register (SCDAT)

7	6	5	4	3	2	1	0	
Serial Communications Data Register								$11

The serial communications data register performs two functions in the serial communications interface; i.e. it acts as the receive data register when it is read and as the transmit data register when it is written. Fig. 5-6 shows this register as two separate registers, namely: the receive data register (RDR) and the transmit data register (TDR). As shown in Fig. 5-6, the TDR (transmit data register) provides the parallel interface from the internal data bus to the transmit shift register and the receive data register (RDR) provides the interface from the receive shift register to the internal data bus.

When SCDAT is read, it becomes the receive data register and contains the last byte of data received. The receive data register, represented above, is a read-only register containing the last byte of data received from the shift register for the internal data bus. The RDRF bit (receive data register full bit in the serial communications status register) is set to indicate that a byte has been transferred from the input serial shift register to the serial communications data register. The transfer is synchronised with the receiver bit rate clock (from the receive control) as shown in Fig. 5-6. All data is received least-significant-bit first.

When SCDAT is written, it becomes the transmit data register and contains the next byte of data to be transmitted. The transmit data register, also represented above, is a write-only register containing the next byte of data to be applied to the transmit shift register from the internal data bus. As long as the transmitter is enabled, data stored in the serial communications data register is transferred to the transmit shift register (after the current byte in the shift register has been transmitted). The transfer from the SCDAT to the transmit shift register is synchronised with the bit rate clock (from the transmit control) as shown in Fig. 5-6. All data is transmitted least-significant-bit first.

5.7.2 Serial Communications Control Register 1 (SCCR1)

7	6	5	4	3	2	1	0	
R8	T8	-	M	WAKE	-	-	-	$0E

The serial communications control register 1 (SCCR1) provides the control bits which: 1) determine the word length (either 8 or 9 bits), and 2) selects the method used for the wake-up feature. Bits 6 and 7 provide a location for storing the ninth bit for longer bytes.

B7, R8 — If the M bit is a one, then this bit provides a storage location for the ninth bit in the receive data byte. Reset does not affect this bit.

B6, T8 — If the M bit is a one, then this bit provides a storage location for the ninth bit in the transmit data byte. Reset does not affect this bit.

B4, M — The option of the word length is selected by the configuration of this bit and is shown below. Reset does not affect this bit.

0 = 1 start bit, 8 data bits, 1 stop bit
1 = 1 start bit, 9 data bits, 1 stop bit

B3, WAKE — This bit allows the user to select the method for receive "wake up". If the WAKE bit is a logic zero, an idle line condition will "wake up" the receiver. If the WAKE bit is set to a logic one, the system acknowledges an address bit (most significant bit). The address bit is dependent on both the WAKE bit and the M bit level (table shown below). (Additionally, the receiver does not use the wake-up feature unless the RWU control bit in serial communications control register 2 is set as discussed below.) Reset does not affect this bit.

Wake	M	Method of Receiver "Wake-Up"
0	X	Detection of an idle line allows the next data byte received to cause the receive data register to fill and produce an RDRF flag.
1	0	Detection of a received one in the eighth data bit allows an RDRF flag and associated error flags.
1	1	Detection of a received one in the ninth data bit allows an RDRF flag and associated error flags.

5.7.3 Serial Communications Control Register 2 (SCCR2)

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

The serial communications control register 2 (SCCR2) provides the control bits which: individually enable/disable the transmitter or receiver, enable the system interrupts, and provide the wake-up enable bit and a "send break code" bit. Each of these bits is described below. (The individual flags are discussed in the 5.7.4 Serial Communications Status Register).

B7, TIE — When the transmit interrupt enable bit is set, the SCI interrupt occurs provided TDRE is set (see Fig. 5-6). When TIE is cleared, the TDRE interrupt is disabled. Reset clears the TIE bit.

B6, TCIE — When the transmission complete interrupt enable bit is set, the SCI interrupt occurs provided TC is set (see Fig. 5-6). When TCIE is clear, the TC interrupt is disabled. Reset clears the TCIE bit.

B5, RIE — When the receive interrupt enable bit is set, the SCI interrupt occurs provided OR is set or RDRF is set (see Fig. 5-6). When RIE is cleared, the OR and RDRF interrupts are disabled. Reset clears the RIE bit.

B4, ILIE — When the idle line interrupt enable bit is set, the SCI interrupt occurs provided IDLE is set (see Fig. 5-6). When ILIE is cleared, the IDLE interrupt is disabled. Reset clears the ILIE bit.

B3, TE — When the transmit enable bit is set, the transmit shift register output is applied to the TDO line. Depending on the state of control bit M in serial communications control register 1, a preamble of 10 (M=0) or 11 (M=1) consecutive ones is transmitted when software sets the TE bit from a cleared state. If a transmission is in progress, and TE is written to a zero, then the transmitter will wait until after the present byte has been transmitted before placing the TDO pin in the idle high-impedance state. If the TE bit has been written to a zero and then set to a one before the current byte is transmitted, the transmitter will wait until that byte is transmitted and will then initiate transmission of a new preamble. After the preamble is transmitted, and provided the TDRE bit is set (no new data to transmit), the line remains idle (driven high while TE = 1); otherwise, normal transmission occurs. This function allows the user to "neatly" terminate a transmission sequence. After loading the last byte in the serial communications data register and receiving the interrupt from TDRE, indicating the data has been transferred into the shift register, the user should clear TE. The last byte will then be transmitted and the line will go idle (high impedance). Reset clears the TE bit.

B2, RE — When the receive enable bit is set, the receiver is enabled. When RE is cleared, the receiver is disabled and all of the status bits associated with the receiver (RDRF, IDLE, OR, NF, and FE) are inhibited. Reset clears the RE bit.

B1, RWU — When the receiver wake-up bit is set, it enables the "wake up" function. The type of "wake up" mode for the receiver is determined by the WAKE bit discussed above (in the SCCR1). When the RWU bit is set, no status flags will be set. Flags which were set previously will not be cleared with RWU set. If the WAKE bit is cleared, RWU is cleared after receiving 10 (M=0) or 11 (M=1) consecutive ones. Under these conditions, RWU cannot be set if the line is idle. If the WAKE bit is set, RWU is cleared after receiving an address bit. The RDRF flag will then be set and the address byte will be stored in the receiver data register. Reset clears the RWU bit.

B0, SBK — When the send break bit is set the transmitter sends zeros in some number equal to a multiple of the data format bits. If the SBK bit is toggled set and clear, the transmitter sends 10 (M=0) or 11 (M=1) zeros and then reverts to idle or sending data. The actual number of zeros sent when SBK is toggled depends on the data format set by the M bit in the serial communications control register 1; therefore, the break code will be synchronous with respect to the data stream. At the completion of the break code, the transmitter sends at least one high bit to guarantee recognition of a valid start bit. Reset clears the SBK bit.

5.7.4 Serial Communications Status Register (SCSR)

7	6	5	4	3	2	1	0	
TDRE	TC	RDRF	IDLE	OR	NF	FE	-	$10

The serial communications status register (SCSR) provides inputs to the interrupt logic circuits for generation of the SCI system interrupt. In addition, a noise flag bit and a framing error bit are also contained in the SCSR.

B7, TDRE — The transmit data register empty bit is set to indicate that the contents of the serial communications data register have been transferred to the transmit serial shift register. If the TDRE bit is clear, it indicates that the transfer has not yet occurred and a write to the serial communications data register will overwrite the previous value. The TDRE bit is cleared by accessing the serial communications status register (with TDRE set), followed by writing to the serial communications data register. Data can not be transmitted unless the serial communications status register is accessed before writing to the serial communications data register to clear the TDRE flag bit. Reset sets the TDRE bit.

B6, TC — The transmit complete bit is set at the end of a data frame, preamble, or break condition if:
1. TE = 1, TDRE = 1, and no pending data, preamble, or break is to be transmitted; or
2. TE = 0, and the data, preamble, or break (in the transmit shift register) has been transmitted.

The TC bit is a status flag which indicates that one of the above conditions has occurred. The TC bit is cleared by accessing the serial communications status register (with TC set), followed by writing to the serial communications data register. It does not inhibit the transmitter function in any way. Reset sets the TC bit.

B5, RDRF — When the receive data register full bit is set, it indicates that the receiver serial shift register is transferred to the serial communications data register. If multiple errors are detected in any one received word, the NF, FE, and RDRF bits will be affected as appropriate during the same clock cycle. The RDRF bit is cleared when the serial communications status register is accessed (with RDRF set) followed by a read of the serial communications data register. Reset clears the RDRF bit.

B4, IDLE — When the idle line detect bit is set, it indicates that a receiver idle line is detected (receipt of a minimum number of ones to constitute the number of bits in the byte format). The minimum number of ones needed will be 10 (M=0) or 11 (M=1). This allows a receiver that is not in the wake-up mode to detect the end of a message, detect the preamble of a new message, or to resynchronize with the transmitter. The IDLE bit is cleared by accessing the serial communications status register (with IDLE set) followed by a read of the serial communications data register. The IDLE bit will not be set again until after an RDRF has been set; i.e., a new idle line occurs. The IDLE is not set by an idle line when the receiver "wakes up" from the wake-up mode. Reset clears the IDLE bit.

B3, OR — When the overrun error bit is set, it indicates that the next byte is ready to be transferred from the receive shift register to the serial communications data register when it is already full (RDRF it is set). Data transfer is then inhibited until the RDRF bit is cleared. Data in the serial communications data register is valid in this case, but additional data received during an overrun condition (including the byte causing overrun) will be lost. The OR bit is cleared when the serial communications status register is accessed (with OR set), followed by a read of the serial communications data register. Reset clears the OR bit.

B2, NF — The noise flag bit is set if there is noise on a "valid" start bit or if there is noise on any of the data bits or if there is noise on the stop bit. It is not set by noise on the idle line nor by invalid (false) start bits. If there is noise, the NF bit is not set until the RDRF flag is set. Each data bit is sampled three times as described above in RECEIVE DATA IN and shown in Fig. 5-3. The NF bit represents the status of the byte in the serial communications data register. For the byte being received (shifted in) there will also be a "working" noise flag the value of which will be transferred to the NF bit when the serial data is loaded into the serial communications data register. The NF bit does not generate an interrupt because the RDRF bit gets set with NF and can be used to generate the interrupt. The NF bit is cleared when the serial communications status register is accessed (with NF set), followed by a read of the serial communications data register. Reset clears the NF bit.

B1, FE — The framing error bit is set when the byte boundaries in the bit stream are not synchronised with the receiver bit counter (generated by a "lost" stop bit). The byte is transferred to the serial communications data register and the RDRF bit is set. The FE bit does not generate an interrupt because the RDRF bit is set at the same time as FE and can be used to generate the interrupt. Note that if the byte received causes a framing error and it will also cause an overrun if transferred to the serial communications data register, then the overrun bit will be set, but not the framing error bit, and the byte will not be transferred to the serial communications data register. The FE bit is cleared when the serial communications status register is accessed (with FE set) followed by a read of the serial communications data register. Reset clears the FE bit.

5.7.5 Baud Rate Register

7	6	5	4	3	2	1	0	
-	-	SCP1	SCP0	-	SCR2	SCR1	SCR0	$0D

The baud rate register provides the means for selecting different baud rates which may be used as the rate control for the transmitter and receiver. The SCP0-SCP1 bits function as a prescaler for the SCR0-SCR2 bits. Together, these five bits provide multiple, baud rate combinations for a given internal processor clock frequency.

B4,SCP0;B5,SCP1 — These two bits in the baud rate register are used as a prescaler to increase the range of standard baud rates controlled by the SCR0-SCR2 bits. A table of the prescaler internal processor clock division versus bit levels is provided below. Reset clears SCP1-SCP0 bit (divide-by-one).

SCP1	SCP0	Internal Processor Clock Divide By
0	0	1
0	1	3
1	0	4
1	1	13

B0,SCR0,B1,SCR1,B2,SCR2 — These three bits in the baud rate register are used to select the baud rates of both the transmitter and receiver. A table of baud rates versus bit levels is shown below. Reset does not affect the SCR2-SCR0 bits.

SCR2	SCR1	SCR0	Prescaler Output Divide By
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

Fig. 5-7 and Tables 5-1 and 5-2 illustrate the divided chain used to obtain the baud rate clock (transmit clock). Note that there is a fixed rate divide-by-16 between the receive clock (RT) and the transmit clock (Tx). The actual divider chain is controlled by the combined SCP0-SCP1 and SCR0-SCR2 bits in the baud rate register as illustrated. All divided frequencies shown in the first table represent the final transmit clock (the actual baud rate) resulting from the internal processor clock division shown in the "divide-by" column only (prescaler division only). The second table illustrates how the prescaler output can be further divided by action of the SCI select bits (SCR0-SCR2). For example, assume that a 9600 Hz baud rate is required with a 2.4576 MHz PLL output clock. In this case the prescaler bits (SCP0-SCP1) could be configured as a divide-by-one or a divide-by-four. If a divide-by-four prescaler is used, then the SCR0-SCR2 bits must be configured as a divide-by-two. This results in a divide-by-128 of the internal processor clock to produce a 9600 Hz baud rate clock. Using the same PLL output clock, the 9600 baud rate can be obtained with a prescaler divide-by-one and the SCR0-SCR2 bits configured for a divide-by-eight.

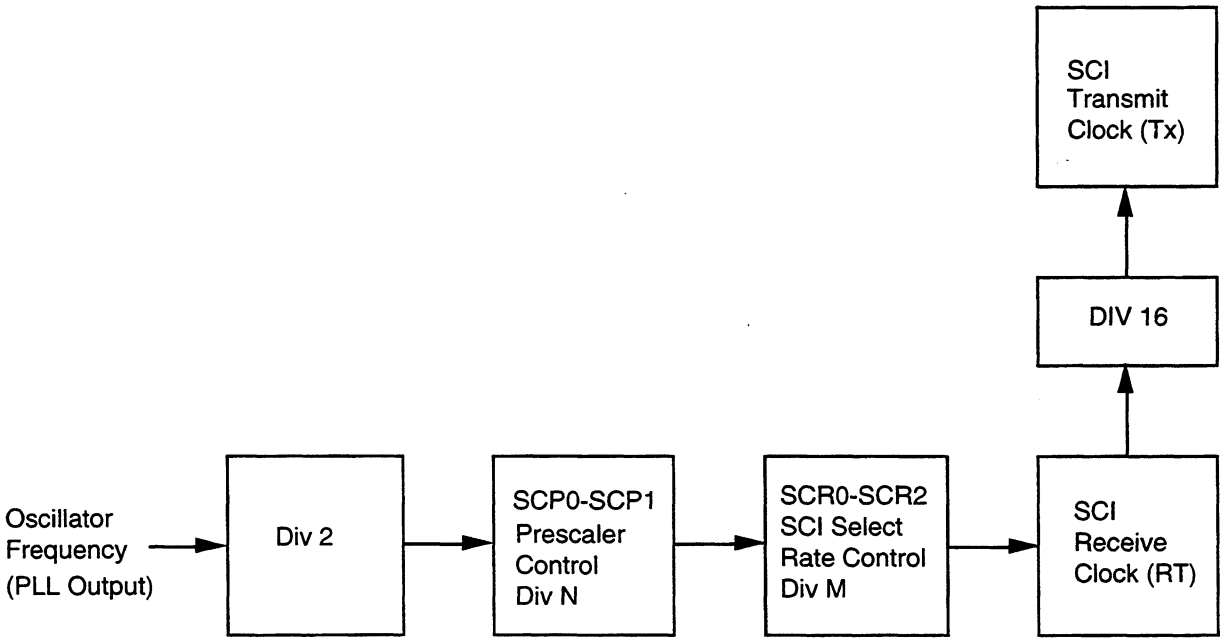


FIG. 5-7 Rate Generator Division

SCP Bit		Clock* Divided By	PLL Output Clock Frequency MHz			
1	0		7.3728	4.9125	2.4576	0.6144
0	0	1	230.4 KHz	153.5 KHz	76.80 KHz	19.20 KHz
0	1	3	76.8 KHz	51.17 KHz	25.60 KHz	6.40 KHz
1	0	4	57.6 KHz	38.38 KHz	19.20 KHz	4.80 KHz
1	1	13	17.72 KHz	11.81 KHz	5.907 KHz	1.477 KHz

* The Clock in the "Clock Divided By" column is the internal processor clock.

Table 5-1 Prescaler Highest Baud Rate Frequency Output

NOTE

The divided frequencies shown in Table 5-1 represent baud rates which are the highest transmit baud rate (Tx) that can be obtained by a specific clock frequency and only using the prescaler division. Lower baud rate may be obtained by providing a further division using the SCI rate select bits as shown below for some representative prescaler outputs.

SCR Bits			Divide By	Representative Highest Prescaler Baud Rate Output			
2	1	0		230.4 KHz	153.5 KHz	76.80 KHz	19.20 KHz
0	0	0	1	230.4 KHz	153.5 KHz	76.80 KHz	19.20 KHz
0	0	1	2	115.2 KHz	76.8 KHz	38.40 KHz	9600 Hz
0	1	0	4	57.6 KHz	38.4 KHz	19.20 KHz	4800 Hz
0	1	1	8	28.8 KHz	19.2 KHz	9600 Hz	2400 Hz
1	0	0	16	14.4 KHz	9600 Hz	4800 Hz	1200 Hz
1	0	1	32	7200 Hz	4800 Hz	2400 Hz	600 Hz
1	1	0	64	3600 Hz	2400 Hz	1200 Hz	300 Hz
1	1	1	128	1800 Hz	1200 Hz	600 Hz	150 Hz

Table 5-2 Transmit Baud Rate Output For a Given Prescaler Output

NOTE

Table 5-2 illustrates how the SCI select bits can be used to provide lower transmitter baud rates by further dividing the prescaler output frequency. The four examples are only representative samples. In all cases, the baud rates shown are transmit baud rates (transmit clock) and the receiver clock is 16 times higher in frequency than the actual baud rate.

SECTION 6
LIQUID CRYSTAL DISPLAY DRIVER, AUTO DISPLAY OFF, REAL TIME CLOCK AND PHASE LOCKED LOOP

6.1 INTRODUCTION

This section contains a description of the liquid crystal display driver, auto display off, real time clock, and phase locked loop.

6.2 LIQUID CRYSTAL DISPLAY DRIVER

The liquid crystal display driver is 1:5 bias and software selectable to either 1:32 or 1:41 multiplex; and consists of the following circuitry:

CONTROL LOGIC provides the control signals for display synchronisation.

LEVEL SELECTOR consists of switching circuit to select appropriate voltage levels from external voltage divider.

BACKPLANE DRIVER provides the backplane drive signal to the LCD. It can be selected either as 1:32 or 1:41 multiplex.

The LCD driver clock is derived from the 32KHz oscillator; and the frame frequency is 64Hz for 1:32 multiplex and 50 Hz for 1:41 multiplex.

There are several bits in the control register ($26) which are used to control the operation of the LCD driver and they are explained in the following paragraphs:

7	6	5	4	3	2	1	0	
				DON	AUTO		MS	$26

MS· BIT 0 When this bit is cleared, LCD driver is 1:32 multiplex, setting this bit changes the LCD driver to 1:41 multiplex. For detail description of display RAM configuration and slave LCD driver, refer to Fig 6-3 & MC141511 Data Sheet. This bit is cleared during power on or external reset.

AUTO BIT 2 This bit controls whether the auto display off feature is selected or not. When this bit is set, LCD will be turned off after CPU executes STOP instruction. The amount of time that is required for LCD to turn off after execute the STOP instruction depends on the value in the count down register. If this bit is cleared, there is no auto display off feature. This bit is cleared during power on or external reset.

DON BIT 3 This is the bit which controls the on/off of the LCD and is cleared during power on or external reset. LCD will not turn on if this bit is cleared. This is the bit which can be cleared by the auto display off feature. When auto display off feature is selected and count down register reaches zero, DON bit is cleared. In order to turn on the LCD again, a one has to be written into this bit. For example:

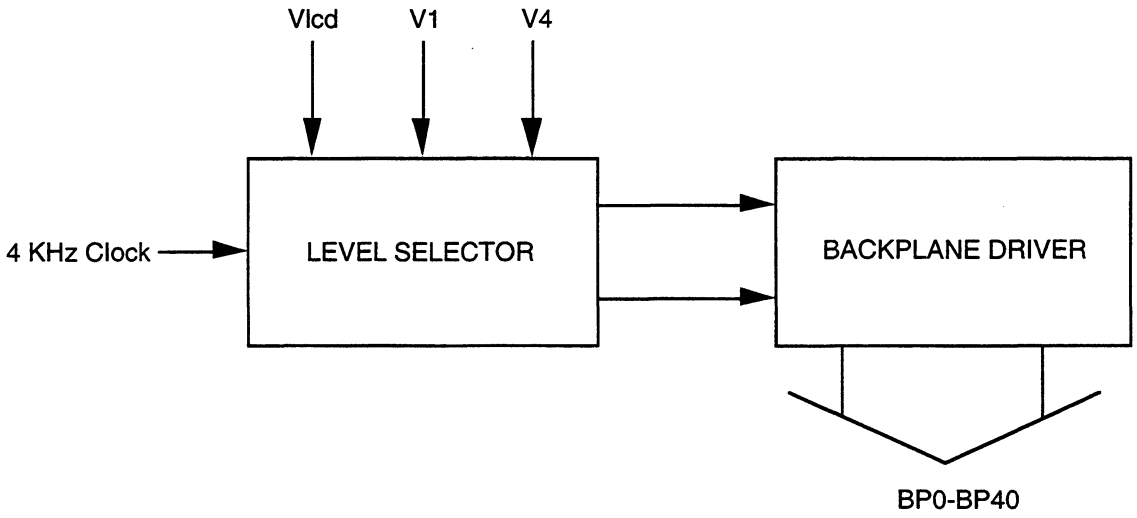


FIG. 6-1 LCD Driver - Functional Block Diagram

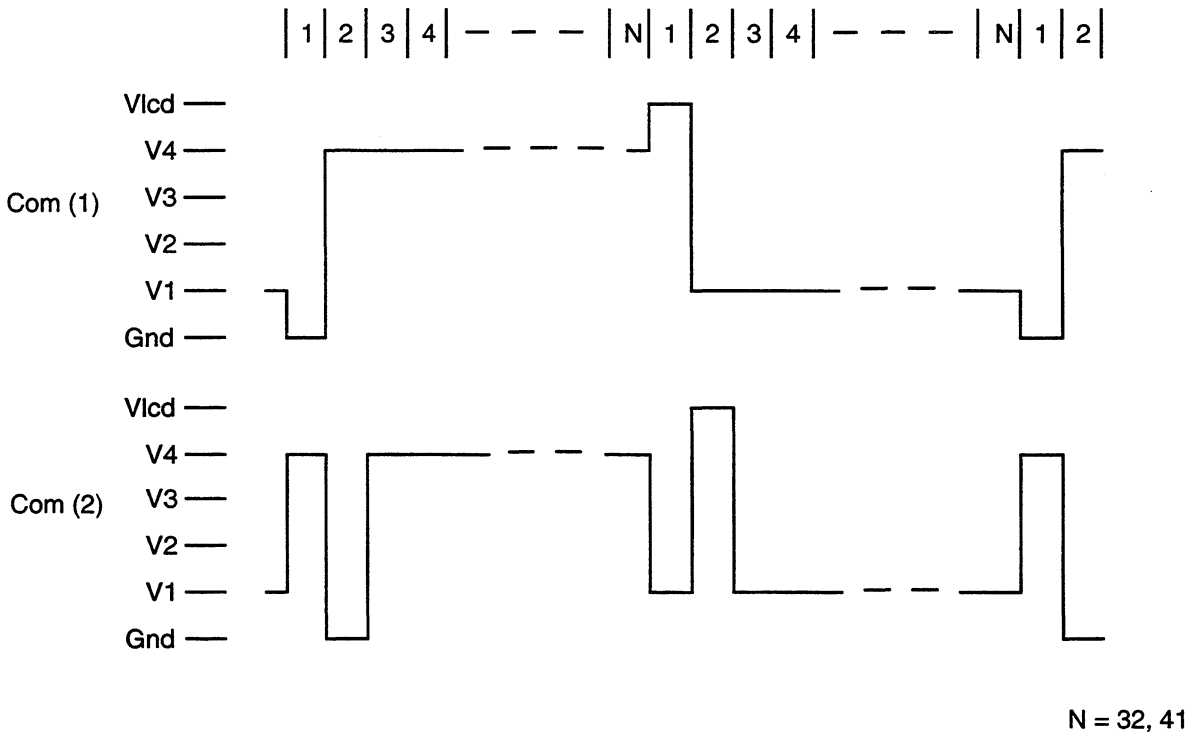


FIG. 6-2 Backplane waveform of 1/5 bias, 1/32 or 1/41 multiplex

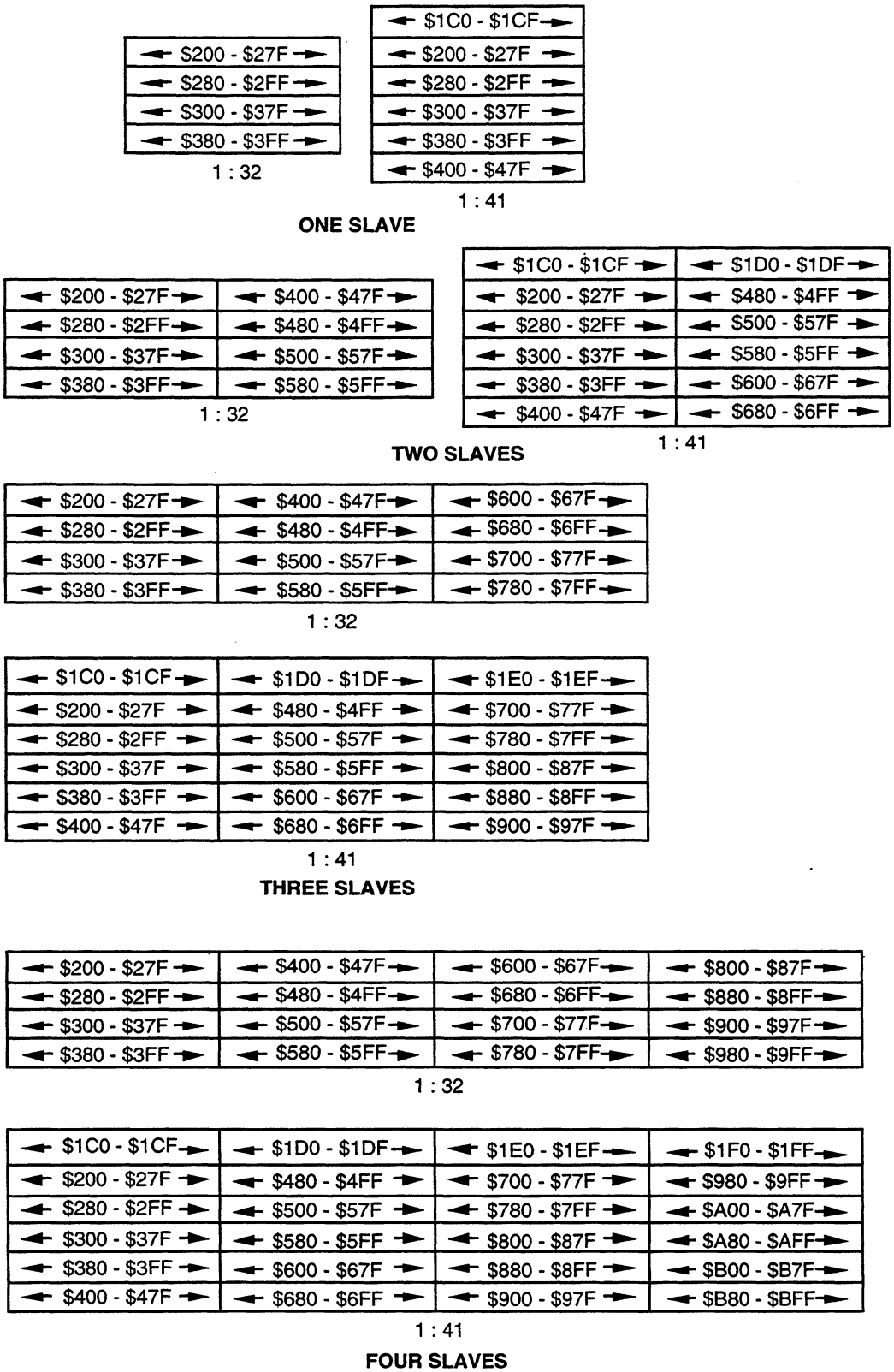


FIG. 6-3 Display RAM configuration at 1:32 and 1:41 multiplex ratio

* LCD is turned off after executing the STOP instruction when auto display off feature is selected.

```
BSET    2,$26       Select auto display off feature.
BSET    3,$26       Turn on LCD.
STOP                Enter stop mode to conserve power.
```

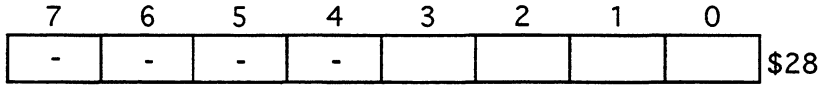
(LCD will turn off when the count down register reaches zero)

*MCU waked up by interrupt or reset.

```
BSET    3,$26       LCD is turned off by the auto display off feature, so turn it back on.
```

6.3 AUTO DISPLAY OFF

Address $28 represents the count down register which is used for the auto display off feature. This register represents the amount of time (in minute) that has to elapse before the LCD is turned off. This is a 4 bits register with default value equal to three during power on or external reset. When MCU is waked up from STOP mode, this register will resume to the previous value. Degree of accuracy of the count down register is 59 seconds and maximum elapse time is 15 minutes.

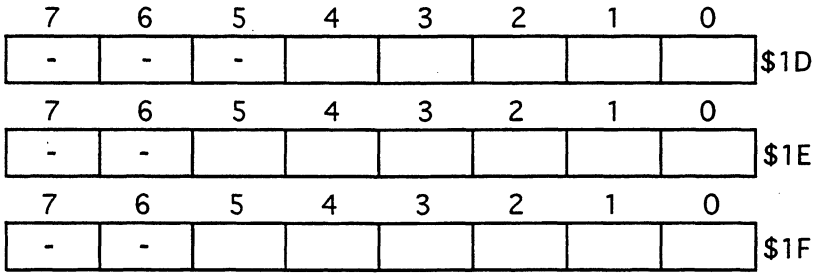


6.4 REAL TIME CLOCK

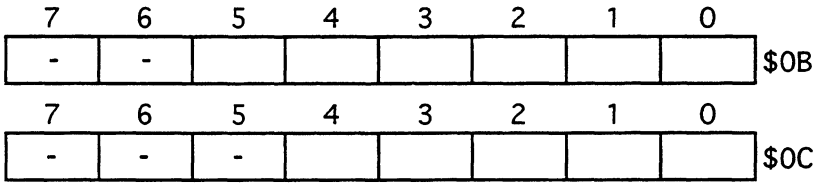
Real time clock is a mask programmable option (Crystal oscillator in STOP mode) which can be either enabled or disabled. Real time clock consists of three binary counters which divide down the clock source from the 32 KHz oscillator. There are three bits in control register ($27) and three bits in address ($1C) which are associated with the operation of the real time clock. Three locations are also reserved for the real time clock, address $1D that represents hours, address $1E represents minutes, and address $1F represents seconds.

As the hours, minutes and seconds counters are not inhibited during CPU write, if the CPU wants to set the time to, say 15:00:00, at the instant when the RTC counters is at the value of, say 3:*59:59*, it may immediately advance to the value 16:00:00, instead of the desired set value of 15:00:00. This will happen if the RTC advances ahead of the CPU in writing to the minutes counter. Therfore, it is always suggested to write to the RTC counters only after detecting a change of the second interrupt flag, so that there is approximately 1 second for the CPU to write to the RTC counter before the next RTC counter advance. This can be done by either CPU reading the SECOND INTERRUPT flag until it changes from 0 to 1 or by enabling the SECOND INTERRUPT so that the CPU only writes to the counter after it is interrupted by the RTC.

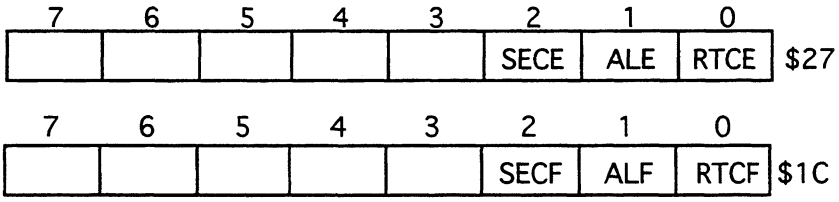
A similar problem may happen during reading of the RTC counters if the RTC counter advances between readings of the hours, minutes, and seconds counters. Therefore, it is advisable to read the RTC counters only after a change of the second interrupt flag is detected.



There are two locations associated with the alarm registers. They are address $0B which represents the minute of the alarm and address $0C which represents the hour of the alarm. These two registers contain random data on power up.



Corresponding to the real time clock, there are three interrupts, and they are controlled by the following bits:



RTCE BIT 0 When this bit is set, real time clock interrupts the CPU once a day, RTCF (bit 0 of address $1C) is the bit which indicates whether an once a day interrupt has just occurred. After serving this interrupt, user is responsible to clear this bit (RTCF), otherwise the CPU will keep on serving this once a day interrupt when a new RTC interrupt occurs (even there is no once a day interrupt occurs). Both RTCE and RTCF bits are cleared during power on or external reset.

ALE BIT 1 When this bit is set, the real time clock interrupts the CPU whenever the value of the real time clock matches the value of the alarm registers. ALF (bit 1 of address $1C) is the bit to indicate whether this is a real time clock interrupt caused by the matching of the value of real time clock and the alarm registers. After serving this interrupt, the user is responsible for clearing this bit (ALF), otherwise the CPU will keep on serving this alarm interrupt when a new RTC interrupt occurs (even there is no alarm interrupt occurs). Both ALE and ALF bits are cleared during power on or external reset. A write to the hour/minute registers ($1D, 1E) or alarm hour/minute register ($0B,0C) will reset the ALE bit to zero.

The ALF is set at the time when the second counter advances while the value programmed in the alarm hour and alarm minute register matches the value of the hour and minute counter respectively. This means that, for example, an alarm time of 3:45 is programmed in the alarm registers, then the RTC will start to set the alarm flag at the the time 3:45:01 and continues to do so until 3:46:00. This means that there is one second delay in the generation of the alarm interrupt to the CPU and the CPU can only clear the alarm interrupt one minute after the alarm time. If the CPU clear the alarm flag too early, say at 3:45:15, then the alarm flag will be set again at 3:45:16.

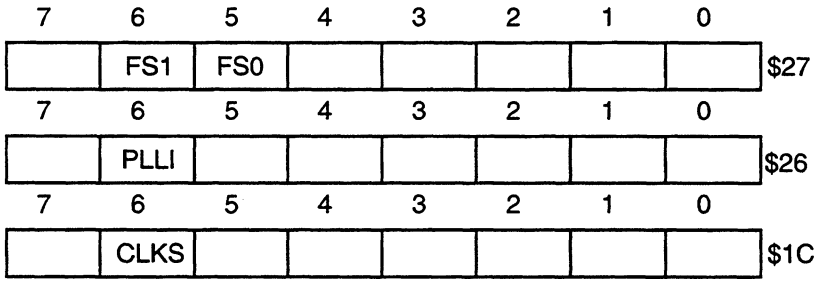
SECE BIT 2 When this bit is set, real time clock interrupts the CPU once a second, SECF (bit 2 of address $1C) is the bit which indicates whether an once a second interrupt has just occurred. After serving this interrupt, the user is responsible for clearing this bit (SECF), otherwise the CPU will keep on serving this once a second interrupt when a new RTC interrupt occurs (even there is no once a second interrupt occurs). Both SECE and SECF bits are cleared during power on or external reset.

The following is an example showing how to use the real time clock interrupt:

```
*Main program
        BSET    0,$27           Enable RTC (once a day) interrupt.
        BSET    1,$27           Enable RTC (alarm) interrupt.
        BSET    2,$27           Enable RTC (once a second) interrupt.
        STOP                    MCU execute STOP instruction for power conservation.

*Real time clock interrupt service routine
        BRSET   0,$1C,ODAY      This bit is set indicating this is an once a day RTC interrupt.
        BRSET   1,$1C,ALINT     This bit is set indicating this is a RTC interrupt caused by a
                                match of alarm registers and real time clock registers.
        BRSET   2,$1C,OSEC      This bit is set indicating this is an once a second RTC
                                interrupt.
ODAY    BCLR    0,$1C           Clear this bit so that this once a day interrupt will not be
                                recognised as a new one on next RTC interrupt.
        JSR     OADAY           Once a day interrupt service routine

        BRSET1,$1C,ALINT        This bit is set indicating alarm interrupt also occurs at the
                                same time.
        BRSET2,$1C,OSEC         This bit is set indicating once a second RTC interrupt also
                                occurs at the same time.
RTCR    RTI

ALINT   BCLR    1,$1C           Clear this bit so that this alarm interrupt will not be recognised
                                as a new one on next RTC interrupt.
        JSR     ALARM           Alarm service interrupt routine.
        BRSET   2,$1C,OSEC      This bit is set indicating once a second interrupt also occurs
                                at the same time.
        BRA     RTCR            Return from interrupt.

OSEC    BCLR    2,$1C           Clear this bit so that this once a second interrupt will not
                                be recognised as a new one on next RTC interrupt.
        JSR     OASEC           Once a second interrupt service routine.
        BRA     RTCR            Return from interrupt.
```

6.5 PHASE LOCK LOOP

System clock can be either obtained from the 32K Hz oscillator or from the PLL. During power on or external reset, MCU system clock comes from the 32K Hz clock. Setting bit 6 of address $1C selects PLL clock for the CPU. PLL clock frequency depends on FS1, FS0 (bit 6,5 of location $27). When FS1 and FS0 are set, a 4.9 MHz clock is output from the PLL. Phase Lock Loop Indicator (PLLI, Bit 6 of address $26) is a read only bit which indicates an accurate clock is ready when set to one. FS1, FS0 and bit 6 of address $1C are cleared during power on or external reset.



FS1	FS0	PLL Output Clock Frequency	P02 (Internal Bus Frequency)
0	0	0.6144 MHz	0.3072 MHz
0	1	2.4576 MHz	1.2288 MHz
1	0	4.9152 MHz	2.4576 MHz
1	1	7.3728 MHz	3.6864 MHz

CLKS (bit 6 of address $1C) is the bit to select CPU clock either coming from the phase lock loop or from the 32 KHz clock. During power on or external reset, this bit is cleared to indicate that the CPU clock is from the 32 KHz clock. The following is the procedure on how to use the PLL to obtain an accurate CPU clock.

1. Switch to the new frequency from original stable frequency.
2. Select 32 KHz as the clock source.
3. Delay 4 ms.
4. Select PLL frequency and check for the setting of PLLI.

Note: It takes 4 ms to reset the PLLI bit (indicates the frequency is unstable) and 16 ms to set it (indicates the frequency is stable).

The PLL consists of an on chip VCO, a phase comparator and a programmable divide-by-N counter. An external filter is required to filter the phase comparator output to provide a DC signal to control the VCO frequency (Fig. 6.4).

The phase comparator compares the rising edge of a 8 KHz reference signal derived from the 32 KHz crystal clock to the rising edge of the VCO clock after being divided by the divide-by-N counter. When there is phase difference between the two signals, the phase comparator output will adjust the DC level input to the VCO to change the VCO frequency (Fig. 6.5). The divide-by-N counter can be programmed to divide by different rates to obtain 4 different VCO frequencies; 614.4 KHz, 2.4576 MHz, 4.9152 MHz and 7.3728 MHz.

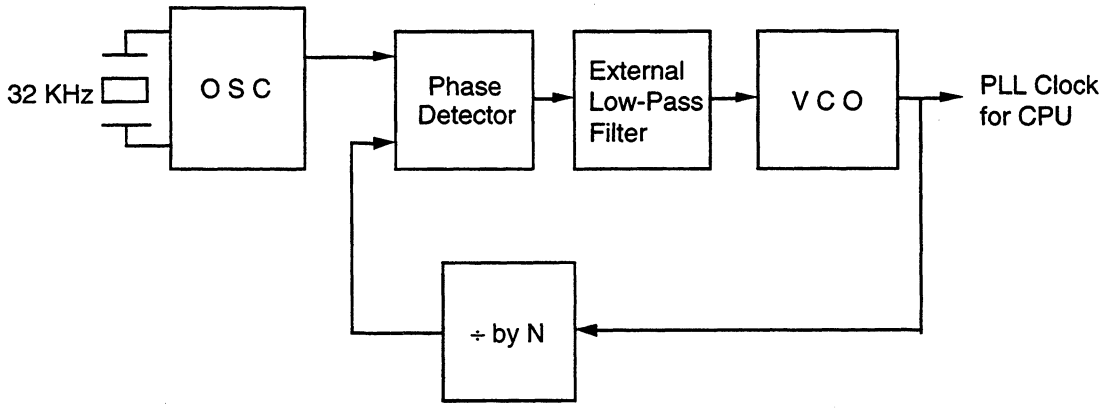


FIG. 6-4 Phase Lock Loop Block Diagram

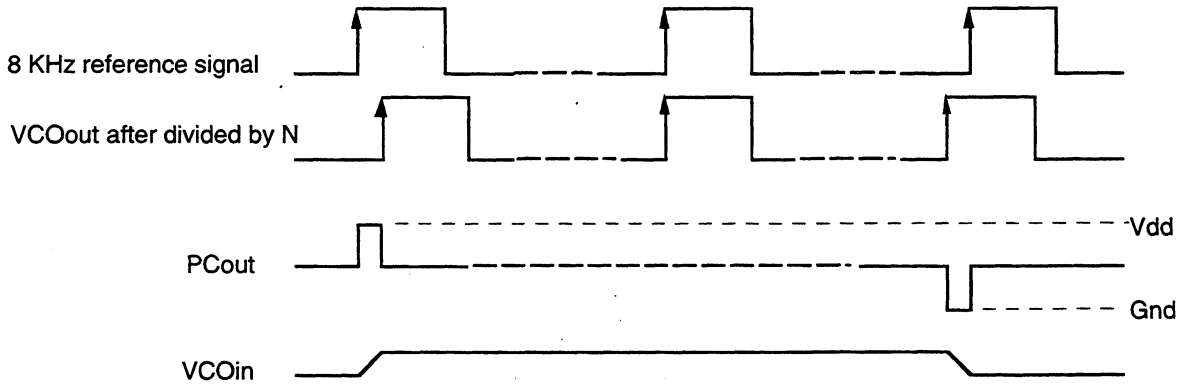


FIG. 6-5 Typical Waveform for PLL

SECTION 7
SERIAL PERIPHERAL INTERFACE (SPI)

7.1 INTRODUCTION

The serial peripheral interface (SPI) is an interface built into the MC68HC05L10 microcontroller which allows several SPI microcontrollers, or SPI-type peripherals to be interconnected within a single "black box" or on the same printed circuit board. In a serial peripheral interface, separate wires (signals) are required for data and clock. In the SPI format, the clock is not included in the data stream and must be furnished as a separate signal. An SPI system may be configured as one containing one master MCU and several slave MCUs, or as a system in which an MCU is capable of being either a master or slave.

Figure 7-1 illustrates two different system configurations. Figure 7-1(a) represents a system of five different microcontrollers in which there is one master and four slaves (0,1,2,3). In this system four basic lines (signals) are required for MOSI (master out slave in), MISO (master in slave out), SCK (serial clock), and SS# (slave select) lines. Figure 7-1(b) represents a system of five microcontrollers in which three can be either a master or a slave and two are slaves only.

Feature of the SPI includes:

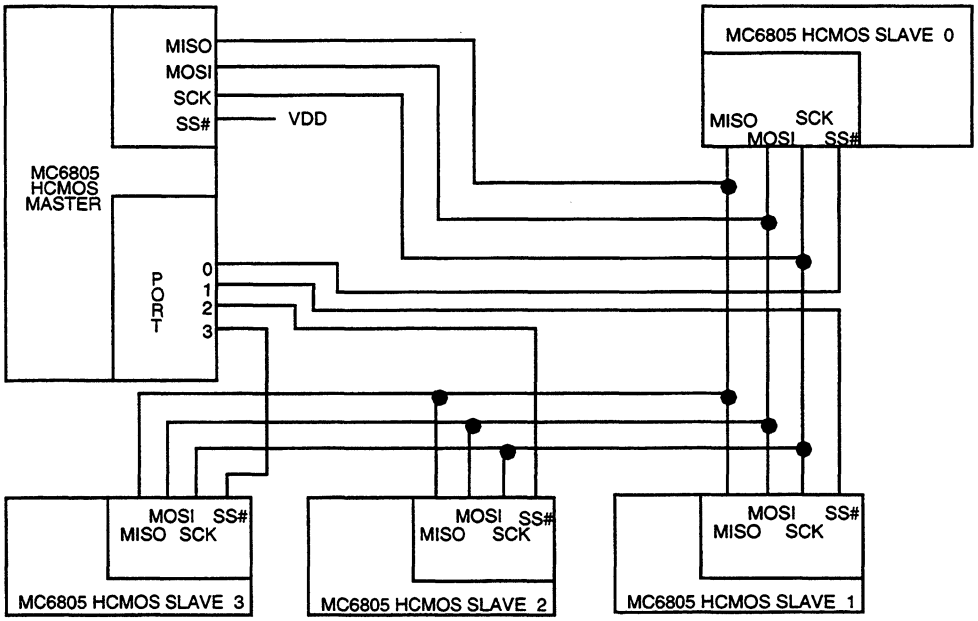
- Full Duplex, Three-Wire Synchronous Transfers
- Master or Slave Operation
- 1.84 MHz (Maximum) Master Bit Frequency
- 3.68 MHz (Maximum) Slave Bit Frequency
- Four programmable Master Bit Rates
- Programmable Clock Polarity and Phase
- End of Transmission Interrupt Flag
- Wire Collision Flag Protection
- Master-Master Mode Fault Protection Capability

7.2 SPI SIGNAL DESCRIPTION

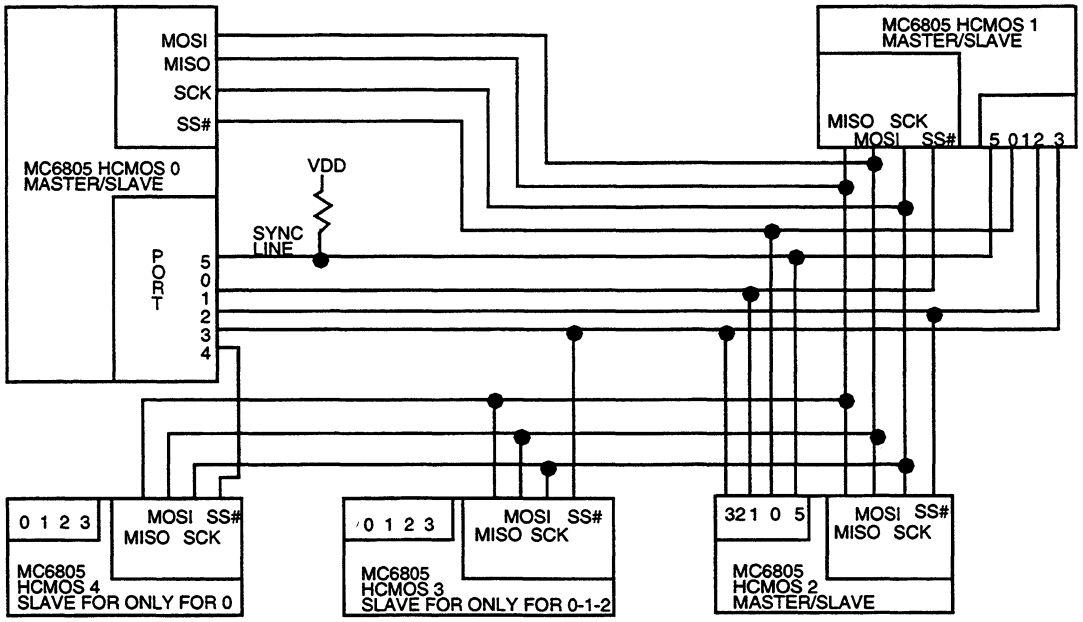
The four basic SPI signals (MOSI, MISO, SCK, and /SS) are described in the following paragraphs. Each signal is described for both the master and slave mode.

7.2.1 Master Out Slave In (MOSI)

The MOSI pin is configured as an output in a master (mode) device and as an input in a slave (mode) device. Data is transferred serially from a master to a slave on this line, most significant bit first. The timing diagram of Figure 7-2 shows the relationship between data and serial clock (SCK). As shown in Figure 7-2, four possible timing relationships may be chosen by using control bits CPOL and CPHA. The master device always allows data to be applied on the MOSI line a half-cycle before the serial clock edge (SCK) in order for the slave device to latch the data.



(a) SINGLE MASTER, FOUR SLAVE



(b) THREE MASTER/SLAVE, TWO SLAVES

FIG. 7-1 Master-Slave System Configuration

NOTE

Both the slave device(s) and a master device must be programmed to similar timing modes for proper data transfer.

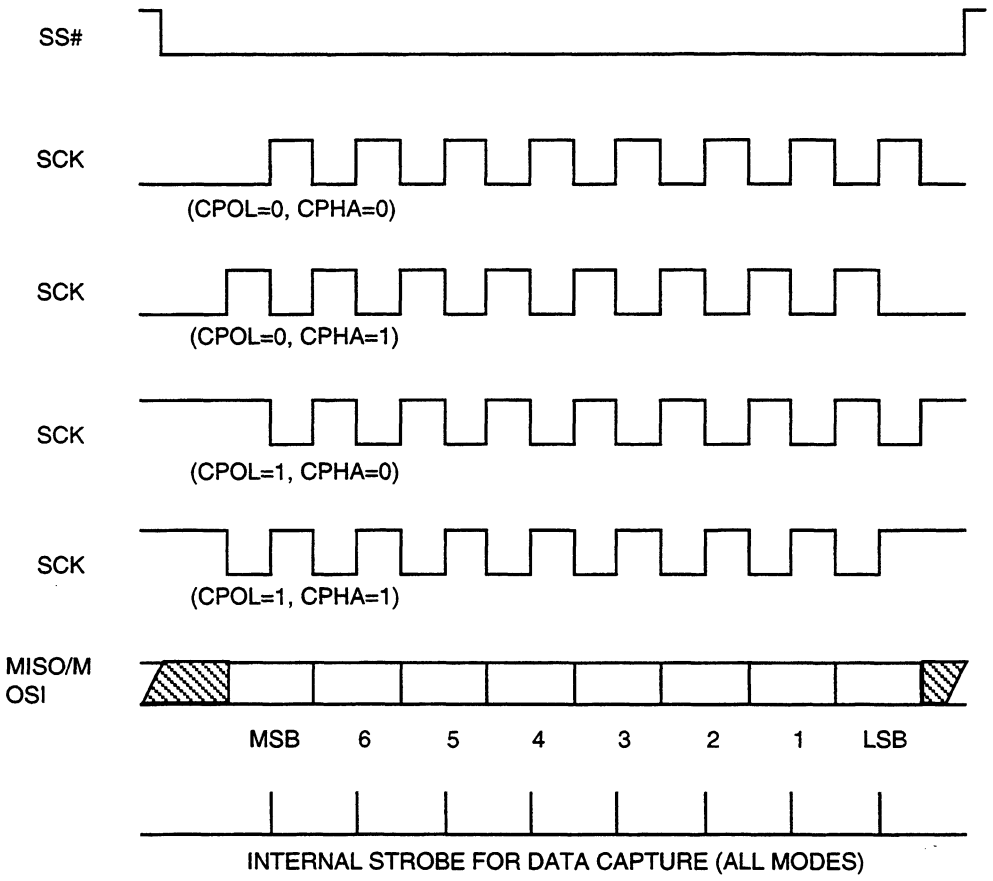


FIG. 7-2 Data Clock Timing Diagram

When the master device transmits data to a slave via a MOSI line, the slave device responds by sending data to the master device via MISO line. This implies full duplex transmission with both data out and data in synchronised with the same clock signal (one which is provided by the master device). Thus, the byte transmitted is replaced by the byte received and eliminates the need for separate transmit-empty and receiver-full status bits. A single status bit (SPIF) in the serial status register (SPSR, location $2B) is used to signify that the I/O operation is complete.

Configuration of the MOSI pin is a function of the MSTR bit in the serial peripheral control register (SPCR, location $2A). When a device is operating as a master, the MOSI pin is an output because the program in firmware sets the MSTR bit as a logic one.

7.2.2 Master In Slave Out (MISO)

The MISO pin is configured as an input in a master (mode) device and as an output pin in a slave (mode) device. Data is transferred serially from a slave to a master on this line, most significant bit first. The MISO pin of a slave is placed in the high-impedance state if it is not selected by a master; i.e., its SS# pin is a logic one. The timing diagram in Figure 7-2 shows the relationship between data and serial clock (SCK). As shown in Figure 7-2, four possible timing relationships may be chosen by using control bit CPOL and CPHA. The master device always allows data to be applied on the MISO line a half-cycle before the serial clock edge (SCK) in order for the slave device to latch the data.

NOTE

Both the slave device(s) and a master device must be programmed to similar timing modes for proper data transfer.

When a master device transmits data to a slave device via the MOSI line, the slave device responds by sending data to the master device via the MISO line. This implies full duplex transmission with both data out and data in synchronised with the same clock signal (one which is provided by the master device). Thus, the byte transmitted is replaced by the byte received and eliminates the need for separate transmit-empty and receiver-full status bits. A single status bit (SPIF) in the serial status register (SPSR, location $2B) is used to signify that the I/O operation is complete.

In the master device, the MSTR control bit in the serial peripheral control register (SPCR, location $2A) is set to a logic one (by the program) to allow the master device to receive data on its MISO pin. In the slave device, its MISO pin is enabled by the logic level of the SS# pin; i.e., if SS#=1 then the MISO pin is placed in the high impedance state, whereas, if SS#=0 the MISO pin is an output for the slave device.

7.2.3 Slave Select (SS#)

The slave select (SS#) pin is a fixed input which receives an active low signal that is generated by the master device to enable slave device(s) to accept data. To ensure that data will be accepted by a slave device, the SS# signal line must be a logic low prior to occurrence of serial SCK and must remain low until after the last (eighth) SCK cycle. Figure 7-2 illustrates the relationship between SCK and the data for two different level combinations of CPHA, when SS# is pulled low. These are: 1) with CPHA = 1 or 0, the first bit or data is applied to the MISO line for transfer, and 2) when CPHA = 0 the slave device is prevented from writing to its data register. Refer to the WCOL status flag in the serial peripheral status register (location $2B) description for further information on the effects that the SS# input and CPHA control bit have on the I/O register. A high level SS# signal forces the MISO line to the impedance state. Also, SCK and the MOSI line are ignored by a slave device when its SS# signal is high.

When the device is a master, it constantly monitors its SS# signal input for a logic low. the master device will become a slave device any time its SS# signal is detected low. This ensures that there is only one master controlling the SS# line for a particular system. When the SS# line is detected low, it clears the MSTR control bit (serial peripheral status register, location $2A). Also, control bit SPE in the SPCR is cleared and causes the serial peripheral interface to be disabled. The MODF flag bit in the serial peripheral status register (location $2B) is also set to indicate to the master device that another device is attempting to become a master. Two devices attempting to be outputs are normally the result of a software error; however, a system could be configured which would contain a default master which would automatically "take-over" and restart the system.

7.2.4 Serial Clock (SCK)

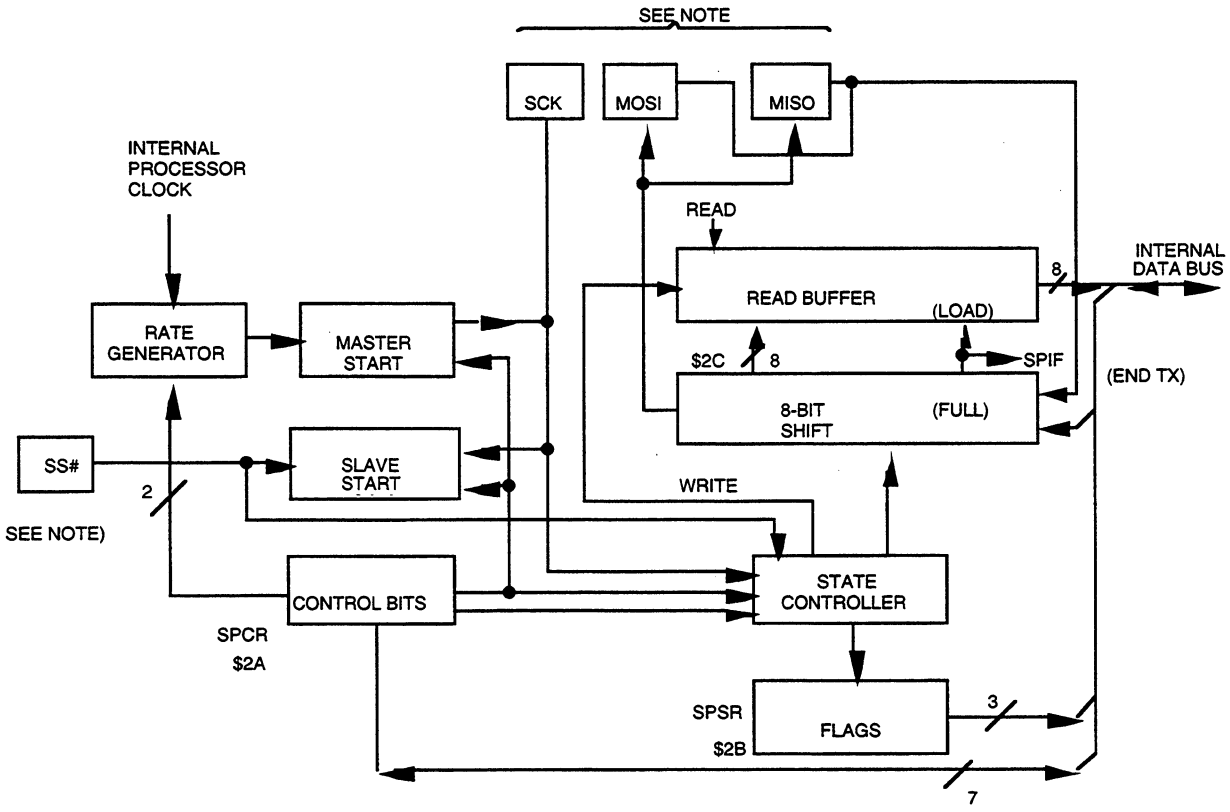
The serial clock is used to synchronise the movement of data both in and out the device through its MOSI and MISO pins. The master and slave devices are capable of exchanging a data byte of information during a sequence of eight clock pulses. Since SCK is generated by the master device, the SCK line becomes an input in all slave devices and synchronises slave data transfer. The type of clock and its relationship to data are controlled by the CPOL and CPHA bits in the SPCR (location $2A). Refer to Figure 7-2 for timing.

The master device generates the SCK through a circuit driven by the internal processor clock. Two bits (SPR0 and SPR1) in the SPCR (location $2A) of the master device select the clock rate. The master device uses the SCK to latch incoming slave device data on the MISO line and shifts out data to the slave device on the MOSI line. Both master and slave devices must be operated in the same timing mode as controlled by the CPOL and CPHA bit in the SPCR. In the slave device, SPR0, SPR1 have no effect on the operation of the serial peripheral interface.

7.3 FUNCTIONAL DESCRIPTION

A block diagram of the serial peripheral interface is shown in Figure 7-3. In a master configuration, the master start logic receives an input from the CPU (in the form of a write to the SPI rate generator) and originates the serial clock (SCK) based on the internal processor clock. This clock is also used internally to control the state controller as well as the 8-bit shift register. As a master device, data is parallel loaded into the 8-bit shift register (from internal bus) during a write cycle and then shifted out serially to the MOSI pin for application to the slave device(s). During a read cycle, data is applied serially from a slave device via the MISO pin to the 8-bit shift register. After the 8-bit shift register is loaded, its data is parallel transferred to the read buffer and then is made available to the internal data bus during a CPU read cycle.

In a slave configuration, the slave start logic receives a logic low (from a master device) at the SS# pin and serial clock input (from the same master device) at the SCK pin. Thus, the slave is synchronised with the master. Data from the master is received serially at the slave MOSI pin and loads the 8-bit shift register. After the 8-bit shift register is loaded, its data is parallel transferred to the read buffer and then is made available to the internal data bus during a CPU cycle. During a write cycle, data is parallel loaded into the 8-bit shift register from the internal data bus and then shifted out serially to the MISO pin for application to the master device.



NOTES:
The SS#,SCK,MOSI, and MISO are external pins which provide the following functions:

a. MOSI - Provides serial output to slave unit(s) when device is configured as a master. Receives serial input from master unit when device is configured as slave unit.

b. MISO - Receives serial input from slave unit(s) when device is configured as a master. Provides serial output to master when device is configured as a slave unit.

c. SCK - Provides system clock when device is configured as a master unit. Receives system clock when device is configured as a slave unit.

d. SS# - Provides a logic low to select a slave device for a transfer with a master device.

FIG. 7-3 Serial Peripheral Interface Block Diagram

Figure 7-4 illustrates the MOSI, MISO, and SCK master-slave interconnections. Note that in Figure 7-4 the master SS# pin is tied to a logic high and the slave SS# pin is a logic low. Figure 7-1 provides a larger system connection for these same pins. Note that in Figure 7-1, all SS# pins are connected to a port pin of a master/slave device. In this case any of the devices can be a slave.

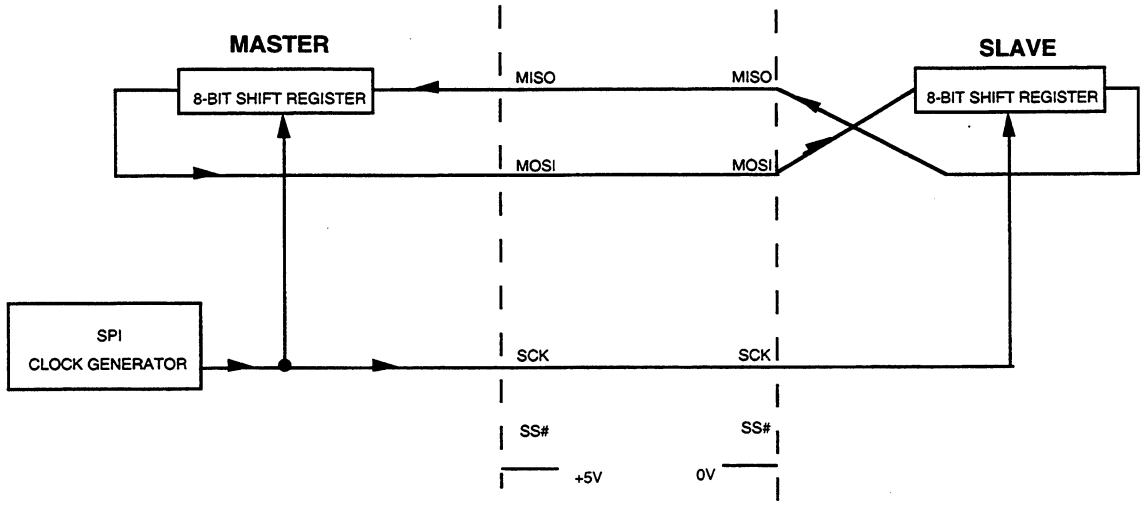


FIG. 7-4 Serial Peripheral Interface Master-Slave Interconnection

7.4 REGISTERS

There are three registers in the serial parallel interface which provide control, status, and data storage functions. These registers are called the serial peripheral control register (SPCR, location $2A), serial peripheral status register (SPSR, location $2B), and serial peripheral data I/O register (SPDR, location $2C) are described in the following paragraphs.

7.4.1 Serial Peripheral Control Register (SPCR)

	7	6	5	4	3	2	1	0
$002A	SPIE	SPE	-	MSTR	CPOL	CPHA	SPR1	SPR0

The serial peripheral control register bits are defines as follows:

SPIE - Serial Peripheral Interrupt Enable

When the serial peripheral interrupt enable bit is high, it allows the occurrence of a processor interrupt, and forces the proper vector to be loaded into the program counter if the serial peripheral status register flag bit (SPIF and/or MODF) is set to a logic one. It does not inhibit the setting of a status bit. The SPIE bit is cleared by reset.

SPE - Serial Peripheral Output Enable

When the serial peripheral output enable control bit is set, all output drive is applied to the external pins and the system is enabled. When the SPE bit is set, it enables the SPI system by connecting it to the external pins thus allowing it to interface with the external SPI bus. The pins that are defined as output depend on which mode (master or slave) the device is in. Because the SPI bit is cleared by reset, the SPI system is not connected to the external pins upon reset.

MSTR - Master

The master bit determines whether the device is a master or a slave. If the MSTR bit is a logic zero it indicates a slave device and a logic one denotes a master device. If the master mode is selected, the function of the SCK pin changes from an input to an output and the function of the MSIO and MOSI pins are reversed. This allows the user to wire device pins MISO to MISO, and MOSI to MOSI, and SCK to SCK without incident. The MSTR bit is cleared by reset; therefore, the device is always placed in the slave mode during reset.

CPOL - Clock Polarity

The clock polarity bit controls the normal or steady state value of the clock when data is not being transferred. The CPOL bit affects both the master and slave modes. It must be used in conjunction with the clock phase control bit (CPHA) to produce the wanted clock-data relationship between a master and a slave device. When the CPOL bit is a logic zero, it produces a steady state low value at the SCK pin of the master device. If the CPOL bit is a logic one, a high value is produced at the SCK pin of the master device when data is not being transferred. The CPOL bit is not affected by reset. Refer to Figure 7-2.

CPHA - Clock Phase

The clock phase bit controls the relationship between the data on the MISO and MOSI pins and the clock produced or received at the SCK pin. This control has effect in both the master and slave modes. It must be used in conjunction with the clock polarity control bit (CPOL) to produce the wanted clock-data relation. The CPHA bit in general selects the clock edge which captures data and allows it to change states. It has its greatest impact on the first bit transmitted (MSB) in that it does or does not allow a clock transition before the first data capture edge. The CPHA bit is not affected by reset. Refer to Figure 7-2.

SPR0, SPR1 - Serial Peripheral Rate

These two serial peripheral rate bits select one of four baud rates to be used as SCK if the device is a master; however, they have no effect in slave mode. The slave device is capable of shifting data in and out at a maximum rate which is equal to the CPU clock. A rate table is given below for the generation of the SCK from the master. The SPR1 and SPR0 bits are not affected by reset.

SPR1	SPR0	Internal Processor Clock Divide By
0	0	2
0	1	4
1	0	16
1	1	32

7.4.2 Serial Peripheral Status Register (SPSR)

	7	6	5	4	3	2	1	0
$002B	SPIF	WCOL	-	MODF	-	-	-	-

The status flags which generate a serial peripheral interface (SPI) interrupt may be blocked by the SPIE control bit in the serial peripheral control register. The WCOL bit does not cause an interrupt. The serial peripheral status register bits are defined as follows:

SPIF - Serial Peripheral Data Transfer Flag

The serial peripheral data transfer flag bit notifies the user that a data transfer between the device and an external device has been completed. With the completion of the data transfer, SPIF is set, and if SPIE is set, a serial peripheral interrupt is generated. During the clock cycle that SPIF is being set, a copy of the received data byte in the shift register is moved to a buffer. When the data register is read, it is the buffer that is read. During an overrun condition, when the master device has sent several bytes of data and the slave device has not responded to the first SPIF, only the first byte sent is contained in the receiver buffer and all other bytes are lost.

The transfer of data is initiated by the master device writing its serial peripheral data register.

Clearing the SPIF bit is accomplished by a software sequence of accessing the serial peripheral status register while SPIF is set and followed by a write to or a read of the serial peripheral data register. While SPIF is set, all writes to the serial peripheral data register are inhibited until the serial peripheral status register is read. This occurs in the master device. In the slave device, SPIF can be cleared (using a similar sequence) during a second transmission; however, it must be cleared before the second SPIF in order to prevent an overrun condition. The SPIF bit is cleared by reset.

WCOL - Write Collision Status

The function of the write collision status bit is to notify the user that an attempt was made to write the serial peripheral data register while a data transfer was taking place with an external device. The transfer continues uninterrupted; therefore, a write will be unsuccessful. A "read collision" will never occur since the received data byte is placed in a buffer in which access is always synchronous with the MCU operation. If a "write collision" occurs, WCOL is set but no SPI interrupt is generated. The WCOL is a status flag only.

Clearing the WCOL bit is accomplished by a software sequence of accessing the serial peripheral status register while WCOL is set, followed by 1), a read if the serial peripheral data register prior to the SPIF bit being set, or 2) a read or write of the serial peripheral data register after the SPIF bit is set. A write to the serial peripheral data register (SPDR) prior to the SPIF bit being set, will result in generation of another WCOL status flag. Both the SPIF and WCOL bits will be cleared in the same sequence. If a second transfer has started while trying to clear (the previously set) SPIF and WCOL bits with a clearing sequence containing a write to the serial peripheral data register, only the SPIF bit will be cleared.

A collision of a write to the serial peripheral data register while an external data transfer is taking place can occur in both the master mode and the slave mode, although with proper programming the master device should have sufficient information to preclude this collision.

Collision in the master device is defined as a write of the serial peripheral data register while the internal rate clock (SCK) is in the process of transfer. The signal in the SS# pin in always high on the master device.

A collision in a slave device is defined in two separate modes. One problem arises in a slave device when the CPHA control bit is a logic zero. When CPHA is a logic zero, data is latched with the occurrence of the first clock transition. The slave device does not have any way of knowing when that transition will occur; therefore, the slave device collision occurs when it attempts to write the serial peripheral data register after its SS# pin has been pulled low. The SS# pin of the slave device freezes the data in its serial peripheral data register and does not allow it to be altered if the CPHA bit is logic zero. The master device must raise the SS# pin of the slave device high between each byte it transfer to the slave device.

The second collision mode is defined for the state of the CPHA control bit being a logic one. With the CPHA bit set, the slave device will receive a clock (SCK) edge prior to the latch of the first data transfer. This first clock edge will freeze the data in the slave device I/O register and allow the most significant bit onto the external MISO pin of the slave device. The SS# pin low state enables the slave device but the drive onto the MOSI pin does not take place until the first data transfer clock edge. The WCOL bit will only be set if the I/O register is accessed while a transfer is taking place. By definition of the second collision mode, a master device might hold
a slave device SS# pin low during a transfer of several bytes of data without a problem.

A special case of write collision occurs in the slave device. This happens when the master device starts a transfer sequence (an edge of SCK for CPHA=1; or an active SS# transition for CPHA=0) at the same time the slave device CPU is writing to its serial peripheral interface data register. In this case it is assumed that the data byte written (in the slave device serial peripheral interface) is lost and the contents of the slave device read buffer becomes the byte that is transferred. Because the master device receives back the last byte transmitted, the master device can detect that a fatal write collision occurred.

Since the slave device is operating asynchronously with the master device, the WCOL bit may be used as an indicator of a collision occurrence. This helps alleviate the user from a strict real-time programming effort. The WCOL bit is cleared by reset.

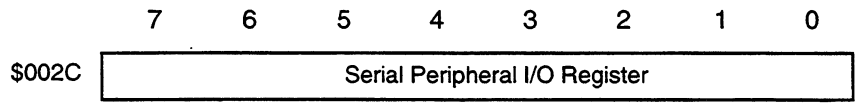
MODF - Mode Fault

The function of the mode fault flag bit is defined for the master mode (device). If the device is a slave device the MODF bit will be prevented from toggling from a logic zero to a logic one; however, this does not prevent the device from being in the slave mode with MODF bit set. The MODF bit is normally a logic zero and is set only when the master device has its SS pin pulled low. Toggling the MODF bit to a logic one affected the internal serial peripheral interface (SPI) system in the following ways:

1. MODF is set and SPI interrupt is generated if SPIE=1
2. The SPE bit is forced to a logic zero. This blocks all output drive from the device, disables the SPI system.
3. The MSTR bit is forced to a logic zero, thus forcing the device into the slave mode.

Clearing the MODF is accomplished by a software sequence of accessing the serial peripheral status register while MODF is set followed by a write to the serial peripheral control register. Control bits SPE and MSTR may restored to their original set state during this clearing sequence or after the MODF bit has been cleared. Hardware does not allow the user to set the SPE and MSTR bit while MODF is a logic one, unless it is during the proper clearing sequence. The MODF flag bit indicates that there might have been a multi-master conflict for system control and allows a proper exit from system operation to a reset or default system state. The MODF is cleared by reset.

7.4.3 Serial Peripheral Data I/O Register (SPDR)



The serial peripheral data I/O register is used to transmit and receive data on the serial bus. Only a write to this register will initiate transmission/reception of another byte and this will only occur in the master device. A slave device writing to this data I/O register will not initiate a transmission. At the completion of transmitting a byte of data, the SPIF status is set in both the master and slave devices. A write or read of the serial peripheral data I/O register, after accessing the serial peripheral status register with SPIF set, will clear SPIF.

During the clock cycle that the SPIF bit is set, a copy of the received data byte in the shift register is moved to a buffer. When the user reads the serial peripheral data I/O register, the buffer is actually being read. During an overrun condition, when the master device has sent several bytes of data and the slave device has not internally responded to clear the first SPIF, only the first byte is contained in the receive buffer of the slave device; all others are lost. The user may read the buffer at any time. The first SPIF must be cleared by the time a second transfer of data from the shift register to the read buffer is initiated, else an overrun condition will occur.

A write to the serial peripheral data I/O register is not buffered and places data directly into the shift register for transmission.

The ability to access the serial peripheral data I/O register is limited when a transmission is taking place. It is important to read the discussion defining the WCOL and SPIF status bits to understand the limits on using the serial peripheral data I/O register.

7.5 SPI DURING WAIT MODE

When the MCU enters the wait mode, the CPU clock is halted. All CPU action is suspended; however, the SPI system remains active. An interrupt from the SPI (in addition to a logic low on the IRQ1, IRQ2 or interrupt from keyboard or RTC or a logic low on the RESET pin or a power on reset) causes the processor to exit the wait mode.

7.6 SPI DURING STOP MODE

When the processor executes the STOP instruction, the internal oscillator is turned off. This halts all internal CPU processing, including the operation of the serial peripheral interface. The only way for the MCU to "wake-up" from the stop mode is by receipt of an external interrupt (logic low on IRQ1#, IRQ2#, RTC or keyboard interrupt), interrupt from keyboard, RTC ,or the detection of a reset (logic low on RESET# pin or a power-on reset).

When the MCU enters the stop mode, the baud rate generator which drives the SPI shuts down. This essentially stops all master mode SPI operation; the master SPI is unable to transmit or receive any data. If the STOP instruction is executed during an SPI transfer, that transfer is halted until the MCU exits the stop mode (provided it is an exit resulting from a logic low on the IRQ1#, IRQ2# pin, interrupt from keyboard or RTC or by the detection of a reset of logic low on reset pin or a power on reset). If the stop mode is exited by a reset, then the appropriate control/status bits are cleared and the SPI is disabled. If the device is in the slave mode when the STOP instruction is executed, the slave SPI will still operate. It can still accept data and clock information in addition to transmitting its own data back to a master device.

Since the MC68HC05 is the bus master, it internally controls the function of its MOSI and MISO lines; thus writing data to the slave devices on the MOSI and reading data from the slave devices on the MISO lines. The master device selects the individual slave devices by using four pins of a parallel port to control the four SS# pins of the slave devices. A slave device is selected when the master device pulls its SS# pin low. The SS# pins are pulled high during reset since the master device ports will be forced to be inputs at that time, thus disabling the slave devices. Note that the master can enable all slaves when writing to them, but can only read from one slave at a time. This is to prevent bus contention on the MISO line.

Example: in a one master, three slaves system, the master writes to the three slaves' display driver to clear a display with a single I/O operation. To ensure that proper data transmission between the master device and a slave device, the master device may have the slave device responding with a data byte previously sent by the master (this data byte could be inverted or at least be a byte that is different from the last one sent by the master device). The master device will always receive the byte back from the slave device if all MISO and MOSI lines are connected and the slave has not written to its data I/O register. Other transmission protocols may be defined using ports for handshake lines or data bytes with command fields.

A multi-master system may also be configured by the user. A system of this type is shown in Figure 7-1(b). An exchange of master control could be implemented using a handshake method through the I/O ports or by an exchange of code messages through the serial peripheral interface system. There are two bits which are important to this configuration, the MSTR bit in the serial peripheral control register and the MODF bit in the serial peripheral status register.

SECTION 8
MEMORY MANAGEMENT UNIT (MMU)

8.1 INTRODUCTION

The Memory Management Unit is to provide a means of memory segment relocation on a 1M byte physical addressing space. It translates internal 16 bit logical addresses to external 20 bit physical addresses of 1M byte space.

The 64K byte CPU logical address space can be configured into three separate logical address areas; a common area bank and 2 other banks. Each bank can be paged to make up a 1M byte space in total, as shown in Fig 8-1. The boundary between any two adjacent banks can be programmed at 4K byte resolution. The maximum space of any one bank is 64K byte.

The common area bank is always logical addresses, it by-passes the translation. At power-up, all bank areas are logical addresses. For external I/O device accessing, only 16 bit address space is available, the translation should be bypassed.

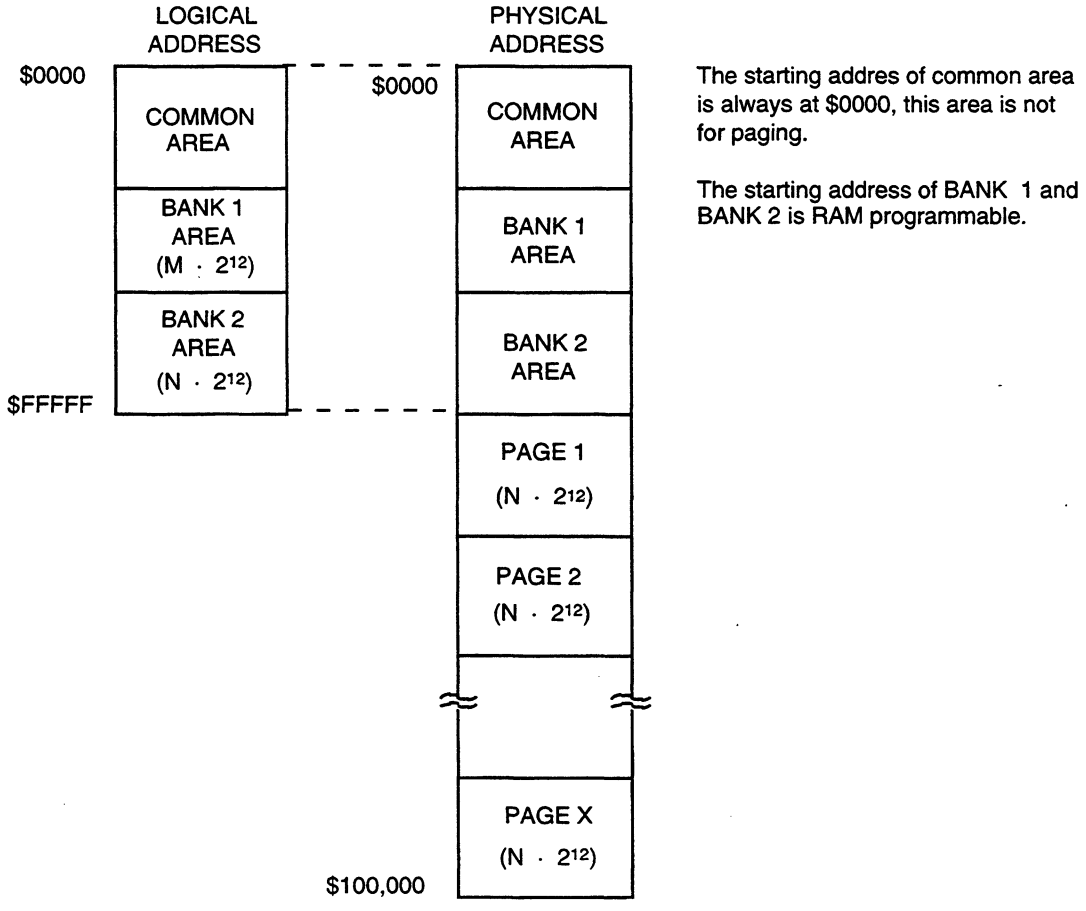


FIG. 8-1 Memory Paging On Bank 2

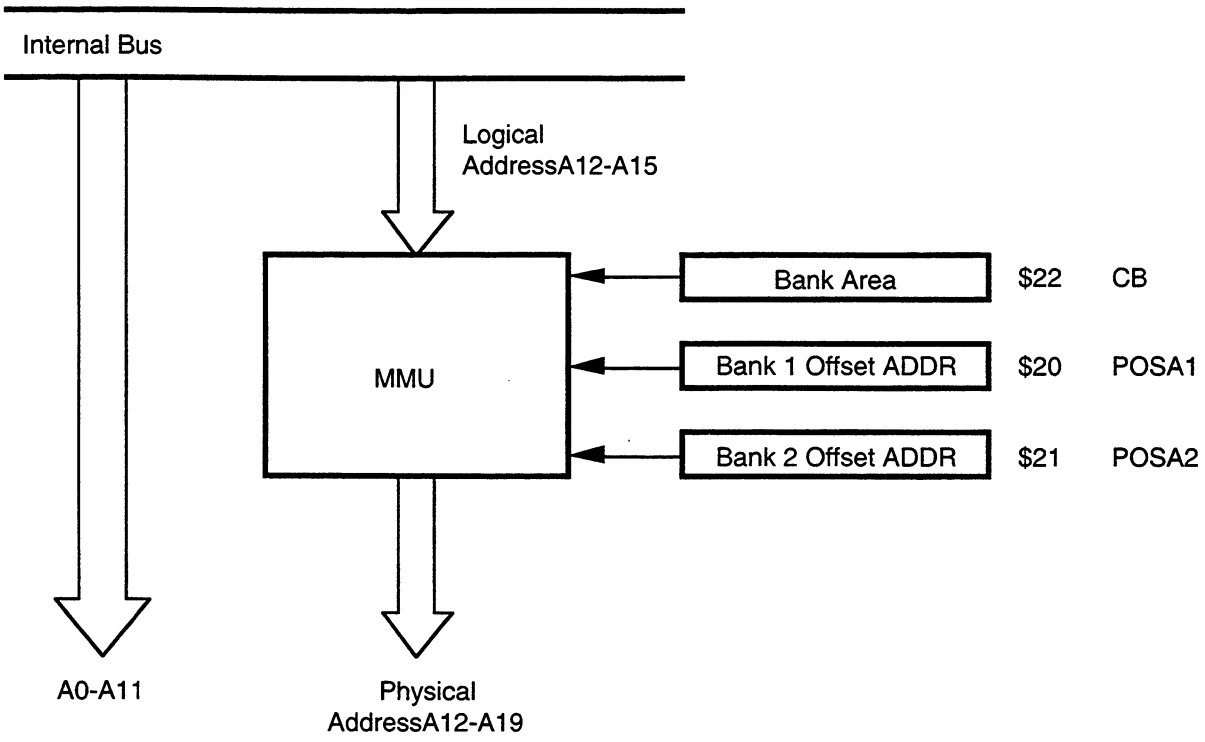


FIG. 8-2 MMU Block Diagram

8.2 PHYSICAL ADDRESS TRANSLATION

Three MMU registers are used to program a specific configuration of memory spacing. Fig 8-3 shows the address translation.

The comparator identifies which one of the two logical areas (Bank 1, Bank 2) is being translated by comparing the starting address specified in the BANK AREA register $22. The appropriate offset address specified in $20 or $21 is then added to the upper 4 bit of the logical address, resulting in a 20 bit physical address, an example is shown in Fig 8-4.

For programming, fetch the bank offset address register POSA1 or POSA2 first, then the logical address.

After a reset, the logical 64K byte address space is accessed.

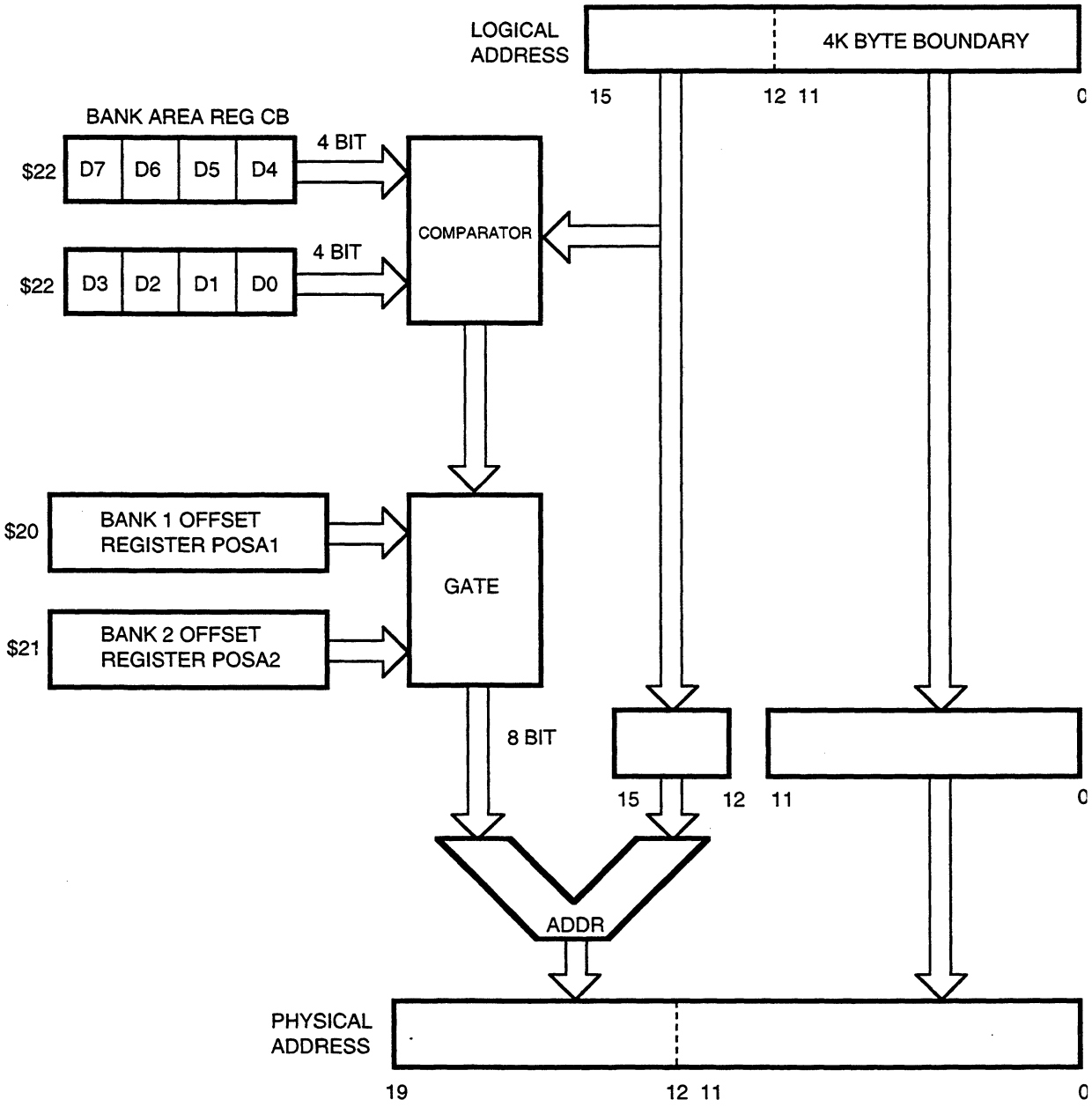


FIG. 8-3 Logical to Physical Address Translation

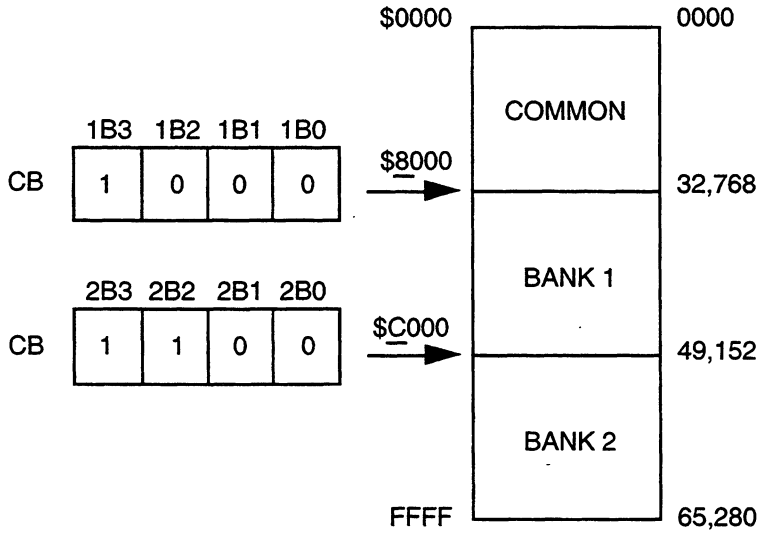
8.3 REGISTERS

8.3.1 Bank Area Register : CB ($22)

	7	6	5	4	3	2	1	0	
$22	2B3	2B2	2B1	2B0	1B3	1B2	1B1	1B0	CB
RESET	1	1	1	1	1	1	1	1	

1B0-1B3 Specifies the starting address of bank 1. This is the 1st hex digit of the starting address represented in binary. For example, 32768 = $8000; $8 = 1000 in binary.

2B0-2B3 Specifies the starting address of bank 2. This is the 1st hex digit of the starting address represented in binary. For example, 49152 = $C000; $C = 1100 in binary.



8.3.2 Bank 1 Page Offset Address Register : POSA1 ($20)

$20	D7	D6	D5	D4	D3	D2	D1	D0	POSA1
RESET	0	0	0	0	0	0	0	0	

POSA1 specifies the offset address of each page for bank 1.

The offset address is the first 2 hex digits resulted from the physical starting address of each page minus the starting address of bank 1 in page 0 specified by CB register.

An Example in Fig 8-4:

The Offset Address of Each Page

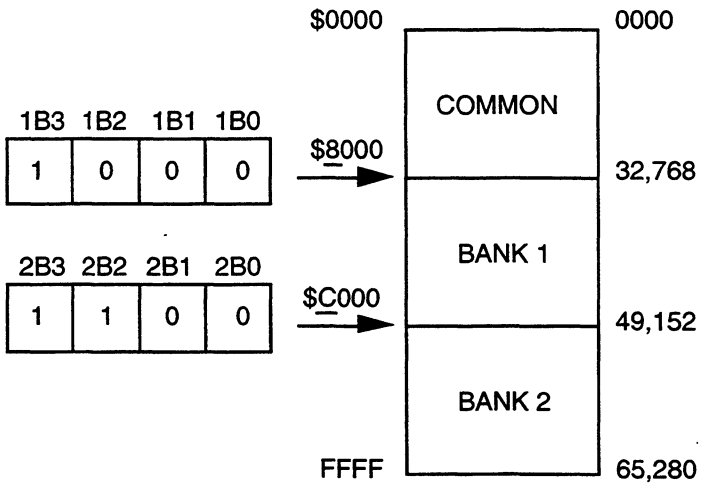
Page 1	$10,000	-	$8,000	=	$8000	=	$08
Page 2	$15,000	-	$8,000	=	$D000	=	$0D
Page 3	$1A,000	-	$8,000	=	$12,000	=	$12

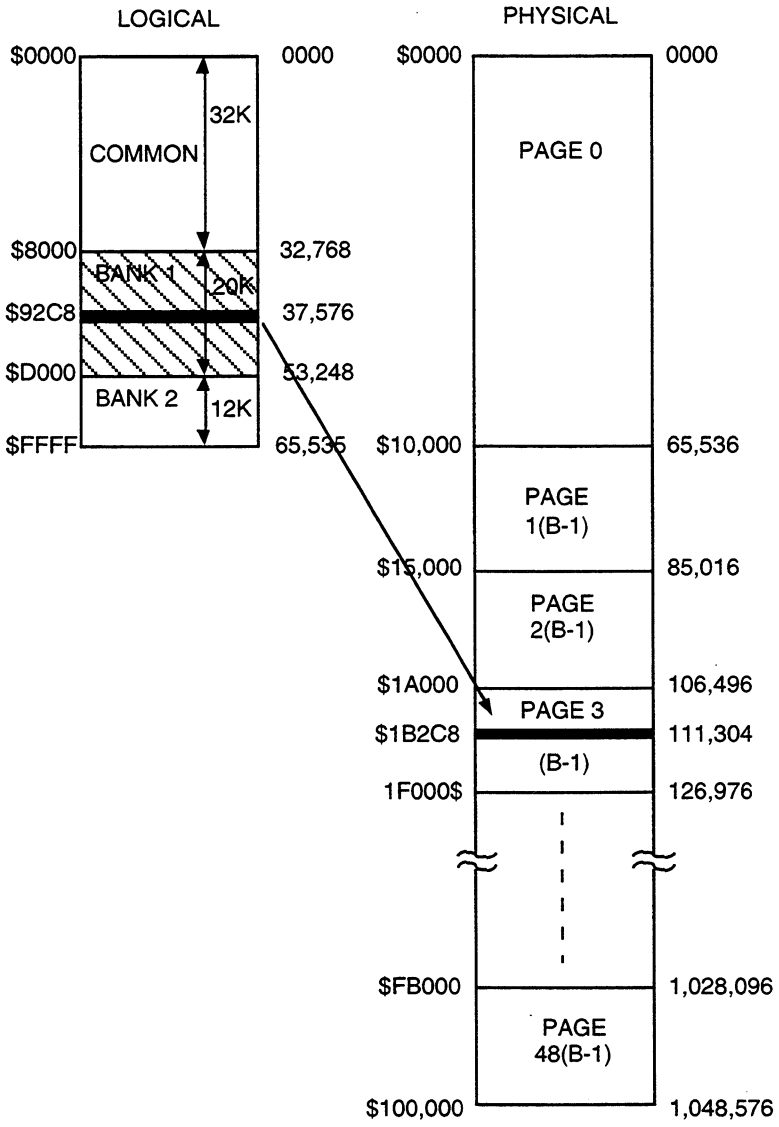
8.3.3 Bank 2 Page Offset Address Register : POSA2 ($21)

$21	D7	D6	D5	D4	D3	D2	D1	D0	POSA2
RESET	0	0	0	0	0	0	0	0	

POSA2 specifies the offset address of each page for bank 2.

The calculation of offset address of each page is same as in bank 1.





REG CB1B0-1B3 = $8 = 1000

REG POSA1 D0-D7 = $1A-$8 = $12 = 0001 , 0010

LOGICAL ADDR = $92C8 = 37,576 = 1001 , 0010 , 1100 , 1000

LOGICAL	1001 , 0010 , 1100 , 1000
PAGE 1 OFFSET	+ 0001 , 0010
PHYSICAL ADDR	0001 , 1011 , 0010 , 1100 , 1000 = $1B2C8 = 111,304

FIG. 8.4 An Example of Paging

SECTION 9
INSTRUCTION SET AND ADDRESSING MODES

9.1 INSTRUCTION SET

The MCU has a set of 62 basic instructions. They can be divided into five different types: register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type.

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

Operation: X:A ← X*A

Description: Multiplies the eight bits in the index register by the eight bits in the accumulator to obtain a 16-bit unsigned number in the concatenated accumulator and index register.

Condition Codes:

H : Cleared
I : Not affected
N : Not affected
Z : Not affected
C : Cleare

Source Form(s): MUL

Addressing Mode	Cycles	Bytes	Opcode
Inherent	11	1	$42

9.1.1 REGISTER/MEMORY INSTRUCTIONS

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

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

9.1.2 READ-MODIFY-WRITE INSTRUCTIONS

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

Function	Mnemonic
Increment	INC
Decrement	DEC
Clear	CLR
Complement	COM
Negate (2's Complement)	NEG
Rotate Left Thru Carry	ROL
Rotate Right Thru Carry	ROR
Logical Shift Left	LSL
Logical Shift Right	LSR
Arithmetic Shift Right	ASR
Test for Negative or Zero	TST
Multiply	MUL

9.1.3 BRANCH INSTRUCTIONS

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

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

9.1.4 BIT MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any bit which resides in the first 256 bytes of the memory space except for bit 6 of location $26, which is read only, port D data register ($03), serial communications status register ($10), timer status register ($13), timer input capture register ($14-$15) and serial peripheral status register ($2B). An additional feature allows the software to test and branch on the state of any bit within the first 256 locations. The bit set, bit clear, and bit test and branch functions are all implemented with a single instruction. For the test and branch instructions, the value of the bit tested is automatically placed in the carry bit of the condition code register. Refer to the following list for bit manipulation instructions.

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

9.1.5 CONTROL INSTRUCTIONS

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

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

9.2 ADDRESSING MODES

The MCU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code to 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, while the longest instructions (three bytes) permit accessing tables throughout memory. Short absolute (direct) and long absolute (extended) addressing are also included. One and two byte direct addressing permits jump instructions to reach all memory.

The term "effective address" (EA) is used in describing the various addressing modes, and is defined as the byte address to or from which the argument for an instruction is fetched or stored. The ten addressing modes of the processor are described in the following paragraphs. 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. 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.

9.2.1 Inherent

In inherent instructions, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator, and no other arguments, are included in this mode.

9.2.2 Immediate

In immediate addressing, the operand is contained in the byte immediately following the opcode. Immediate addressing is used to access constants which do not change during program execution (e.g. a constant used to initialise a loop counter).

EA = PC + 1; PC ← PC + 2

9.2.3 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. This includes all on-chip RAM and I/O registers, and 128 bytes of on chip ROM. Direct addressing is efficient in both memory and time.

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

9.2.4 Extended

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode. Instructions with extended addressing modes 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 most efficient addressing mode.

EA = (PC + 1):(PC + 2); PC ← PC + 3
Address Bus High ← (PC + 1); Address Bus Low ← (PC + 2)

9.2.5 Indexed, No Offset

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. Thus, this addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is used to move a pointer through a table or to address a frequently referenced RAM or I/O location.

EA = X; PC ← PC + 1
Address Bus High ← 0; Address Bus Low ← X

9.2.6 Indexed, 8-Bit Offset

Here the EA is obtained by adding the contents of the byte following the opcode to that of the index register; therefore, the operand is located anywhere within the lowest 511 memory locations. For example, this mode of addressing is useful for selecting the mth element in a n element table. All instructions are only two bytes. The content of the index register (X) is not changed. The content of (PC + 1) is an unsigned 8-bit integer. One byte offset indexing permits look-up tables to be easily accessed in either RAM or ROM.

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

where:

K = The carry from the addition of X + (PC + 2)

9.2.7 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 addressing 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 (e.g., jump tables in ROM). As with direct and extended, the M6805 assembler determines the most efficient form of indexed offset; 8- or 16-bit. The content of the index register is not changed.

EA = X + [(PC + 1):(PC + 2); PC ← PC + 3
Address Bus High ← (PC + 1) + K; Address Bus Low ← X + (PC + 2)

where:

K = The carry from the addition of X + (PC + 2)

9.2.8 Relative

Relative addressing is only used in branch instructions. In relative addressing, the content of the 8-bit signed byte following the opcode (the offset) is added to the PC if and only if the branch condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is limited to the range of -126 to +129 bytes from the branch instruction opcode location. The Motorola assembler calculates the proper offset and checks to see if it is within the span of the branch.

EA = PC + 2 + (PC + 1); PC ← EA if branch taken;
otherwise, EA = PC ← PC + 2

9.2.9 Bit Set/Clear

Direct addressing and bit addressing are combined in instructions which set and clear individual memory and I/O bits. In the bit set and clear instructions, the byte is specified as a direct address in the location following the opcode. The first 256 addressable locating are thus accessed. The bit to be modified within that byte is specified in the first three bits of the opcode. The bit set and clear instructions occupy two bytes, one for the opcode (including the bit number) and the other to address the byte which contains the bit of interest.

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

9.2.10 Bit Test and Branch

Bit test and branch is a combination of direct addressing, bit set/clear addressing, and relative addressing. The actual bit to be tested, within the byte, is specified within the low order nibble of the opcode. The address of the data byte to be tested is located via a direct address in the location following the opcode byte (EA1). The signed relative 8-bit offset is in the third byte (EA2) and 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 bit in the first 256 locations of memory.

EA1 = (PC +1)
Address Bus High ← 0; Address Bus Low ← (PC + 1)
EA2 = PC + 3 + (PC + 2); PC ← EA2 if branch taken;
otherwise, PC ← PC + 3

SECTION 10
ELECTRICAL CHARACTERISTICS

10.1 INTRODUCTION

This section contains the electrical specification of MC68HC05L10.

10.2 MAXIMUM RATINGS

(Voltages Referenced to V_{SS})

Ratings	Symbol	Value	Unit
Supply Voltage	V_{DD}	-0.3 to + 7.0	V
	V_{lcd}	-0.3 to + 12.0	V
Input Voltage (except V1, V4)	V_{in}	V_{SS}-0.3 to V_{DD} +0.3	V
Input Voltage (V1, V4)	V_{in}	V_{SS}-0.3 to V_{lcd}+0	V
Self-Check Mode ($\overline{IRQ1}$ Pin Only)	V_{in}	V_{SS}-0.3 to 2xV_{DD} + 0.3	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	25	mA
Operating Temperature Range	T_A	0 to 70	°C
Storage Temperature Range	T_{stg}	-65 to +150	°C

This device contains circuitry to protect its inputs against damage due to high static voltages or electric fields; however, it is advised that precautions to be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit. For proper operation it is recommended that Vin and Vout be constrained to the range VSS ≤ (Vin or Vout) ≤ VDD. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (e.g., either VSS or VDD).

10.3 DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, Temperature range = 0 to 70 °C)

Characteristics	Symbol	Min	Typ	Max	Unit
Output Voltage, $I_{Load} \leq$ 10.0µA	V_{OL} V_{OH}	- V_{DD}-0.1	- -	0.1 -	V V
Output High Voltage (I_{Load} = 1.6mA) PA0-PA7, PB0-PB7, PC0-PC7, PD1, PD3-PD5, PD7, PE0-PE3, TONE, P02, R/$\overline{W}$, D0-D7, AD0-AD19, $\overline{CS1}$-$\overline{CS4}$, FRM, BPCLK	V_{OH}	V_{DD}-0.8	-	-	V
Output Low Voltage (I_{Load}=1.6 mA) PA0-PA7, PB0-PB7, PC0-PC7, PD1, PD3-PD5,PD7, PE0-PE3, TONE,P02, R/$\overline{W}$, D0-D7, AD0-AD19, $\overline{CS1}$-$\overline{CS4}$, FRM, BPCLK	V_{OL}	-	-	0.4	V
Input High Voltage PA0-PA7, PB0-PB7, PC0-PC7 PD0-PD7, PE0-PE3, $\overline{IRQ1}$, $\overline{IRQ2}$, $\overline{RESET}$, OSC1, D0-D7	V_{IH}	0.7xV_{DD}	-	V_{DD}	V
Input Low Voltage PA0-PA7, PB0-PB7, PC0-PC7 PD0-PD7, PE0-PE3, $\overline{IRQ1}$, $\overline{IRQ2}$, $\overline{RESET}$, OSC1, D0-D7	V_{IL}	V_{SS}	-	0.2xV_{DD}	V
Data Retention Mode	V_{RM}	2.0	-	-	V
Supply Current (VDD = 5.5 Vdc) Run (3.6MHz) (2.4MHz) (1.2MHz) (307kHz)	 I_{DD} I_{DD} I_{DD} I_{DD}	 - - - -	 7.0 4.8 3.0 1.2	 9.0 6.0 5.0 2.0	 mA
Wait (3.6MHz) (2.4MHz) (1.2MHz) (307kHz)	I_{DD} I_{DD} I_{DD} I_{DD}	- - - -	2.0 1.5 1.2 0.9	2.5 2.0 1.5 1.2	mA
Stop (Oscillator on) (Oscillator off)	I_{DD} I_{DD}	- -	15 3	30 12	µA
I/O Ports Hi-Z Leakage Current PA0-PA7, PB0-PB7, PC0-PC7,PD0-PD7, PE0-PE3	I_{IL}	-	-	± 10	µA
Input Current TCAP, $\overline{IRQ1}$, $\overline{IRQ2}$, $\overline{RESET}$, PD0, OSC1	I_{in}	-	-	±1	µA
Capacitance Ports (as Input or Output) D0-D7, AD0-AD19, P02, TONE, R/$\overline{W}$ $\overline{RESET}$, $\overline{IRQ1}$, $\overline{IRQ2}$, OSC1, PD0-PD7	 C_{out} C_{in}	 - -	 - -	 12 8	 pF pF

10.4 DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 3.0 Vdc ± 10%, V_{SS} = 0 Vdc, Temperature range = 0 to 70 °C)

Characteristics	Symbol	Min	Typ	Max	Unit
Output Voltage, $I_{Load} \leq$ 10.0µA	V_{OL} V_{OH}	- V_{DD}-0.1	- -	0.1 -	V V
Output High Voltage (I_{Load} = 1.6mA) PA0-PA7, PB0-PB7, PC0-PC7, PD1, PD3-PD5, PD7, PE0-PE3, TONE, P02, R/$\overline{W}$, D0-D7, AD0-AD19, $\overline{CS1}$-$\overline{CS4}$, FRM, BPCLK	V_{OH}	V_{DD}-0.3	-	-	V
Output Low Voltage (I_{Load}=1.6 mA) PA0-PA7, PB0-PB7, PC0-PC7, PD1, PD3-PD5,PD7, PE0-PE3, TONE,P02, R/$\overline{W}$, D0-D7, AD0-AD19, $\overline{CS1}$-$\overline{CS4}$, FRM, BPCLK	V_{OL}	-	-	0.3	V
Input High Voltage PA0-PA7, PB0-PB7, PC0-PC7 PD0-PD7, PE0-PE3, $\overline{IRQ1}$, $\overline{IRQ2}$, $\overline{RESET}$, OSC1, D0-D7	V_{IH}	0.7xV_{DD}	-	V_{DD}	V
Input Low Voltage PA0-PA7, PB0-PB7, PC0-PC7 PD0-PD7, PE0-PE3, $\overline{IRQ1}$, $\overline{IRQ2}$, $\overline{RESET}$, OSC1, D0-D7	V_{IL}	V_{SS}	-	0.2xV_{DD}	V
Data Retention Mode	V_{RM}	2.0	-	-	V
Supply Current (VDD = 2.8V / 3.0 Vdc) Run (1.2MHz) (307kHz)	 I_{DD} I_{DD}	 - -	 1.2 0.5	 1.7 0.8	 mA
Wait (1.2MHz) (307kHz)	I_{DD} I_{DD}	- -	0.6 0.4	1.2 0.7	mA
Stop (Oscillator on) (Oscillator off)	I_{DD} I_{DD}	- -	7 2	15 8	µA
I/O Ports Hi-Z Leakage Current PA0-PA7, PB0-PB7, PC0-PC7,PD0-PD7, PE0-PE3	I_{IL}	-	-	± 10	µA
Input Current TCAP, $\overline{IRQ1}$, $\overline{IRQ2}$, $\overline{RESET}$, PD0, OSC1	I_{in}	-	-	±1	µA
Capacitance Ports (as Input or Output) D0-D7, AD0-AD19, P02, TONE, R/$\overline{W}$ $\overline{RESET}$, $\overline{IRQ1}$, $\overline{IRQ2}$, OSC1, PD0-PD7	 C_{out} C_{in}	 - -	 - -	 12 8	 pF pF

10.5 BUS TIMING

(Vdd = 5.0 Vdc ±10%, Vss = 0 Vdc,TA = 0 to 70°C, see figure 10-1)

Num	Characteristic	Symbol	307KHz		3.686MHz		Unit
			Min	Max	Min	Max	
1	Cycle Time	tcyc	3257	-	271	-	ns
2	Pulse Width, PO2 Low	PO2l	1603	-	110	-	ns
3	Pulse Width, PO2 High	PO2h	1598	-	105	-	ns
4	PO2 Rise Time	tr	-	25	-	25	ns
5	PO2 Fall Time	tf	-	25	-	25	ns
6	Address Delay Time from PO2 fall	tad	-	80	-	80	ns
7	Address Hold Time from PO2 rise	tah	10	-	10	-	ns
8	R/W Delay Time from PO2 fall	trwd	-	80	-	80	ns
9	R/W Hold Time from PO2 rise	trwh	10	-	10	-	ns
10	Write Data Delay Time	tddw	-	40	-	40	ns
11	Write Data Hold Time	tdhw	10	-	10	-	ns
12	Read Data Set Up Time	tdsr	30	-	30	-	ns
13	Read Data Hold Time	tdhr	0	-	0	-	ns

NOTES:
1. All timing is shown with respect to 20 % VDD and 70 % VDD.

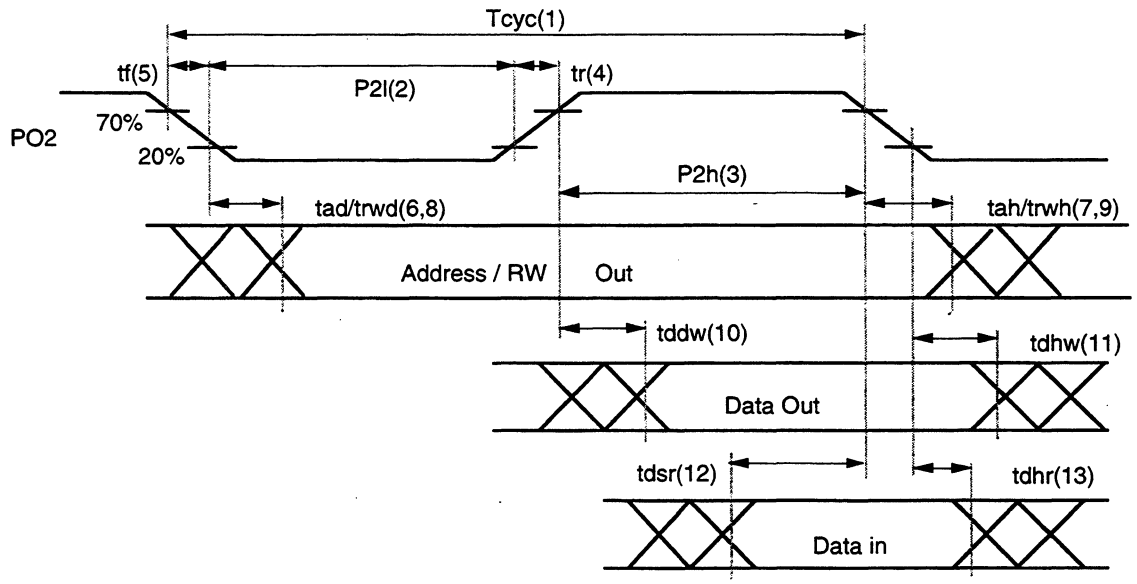


FIG. 10-1 Bus Timing

10.6 PORT/SLAVE CONTROL TIMING

(VDD = 5.0 Vdc ±10%, VSS = 0 Vdc, TA = 0 to 70°C, see figure 10-2)

Num	Characteristic	Symbol	307KHz		3.686MHz		Unit
			Min	Max	Min	Max	
1	Cycle Time	tcyc	3257	-	271	-	ns
2	Port Data In Setup Time	tpdsu	100	-	100	-	ns
3	Port Data In Hold Time	tpdh	350	-	50	-	ns
4	Port Data Write Delay Time	tpwd	-	40	-	40	ns
5	$\overline{CS1}$, $\overline{CS2}$, $\overline{CS3}$, $\overline{CS4}$ Output Delay	tcsd	-	50	-	50	ns
6	Backplane Clock	bpclk	4096	-	4096	-	Hz
7	Slave Counter Reset 1:32 multiplex 1:41 multiplex	$\overline{FRM}$	64 50	- -	64 50	- -	Hz

NOTES:
1. All timing is shown with respect to 20% VDD and 70 % VDD.
2. The CS becomes low when the Slave1, 2, 3 or 4 memory space is being addressed.
3. Accuracy of backplane clock and counter reset depend on OSC Xtal 32.768KHz

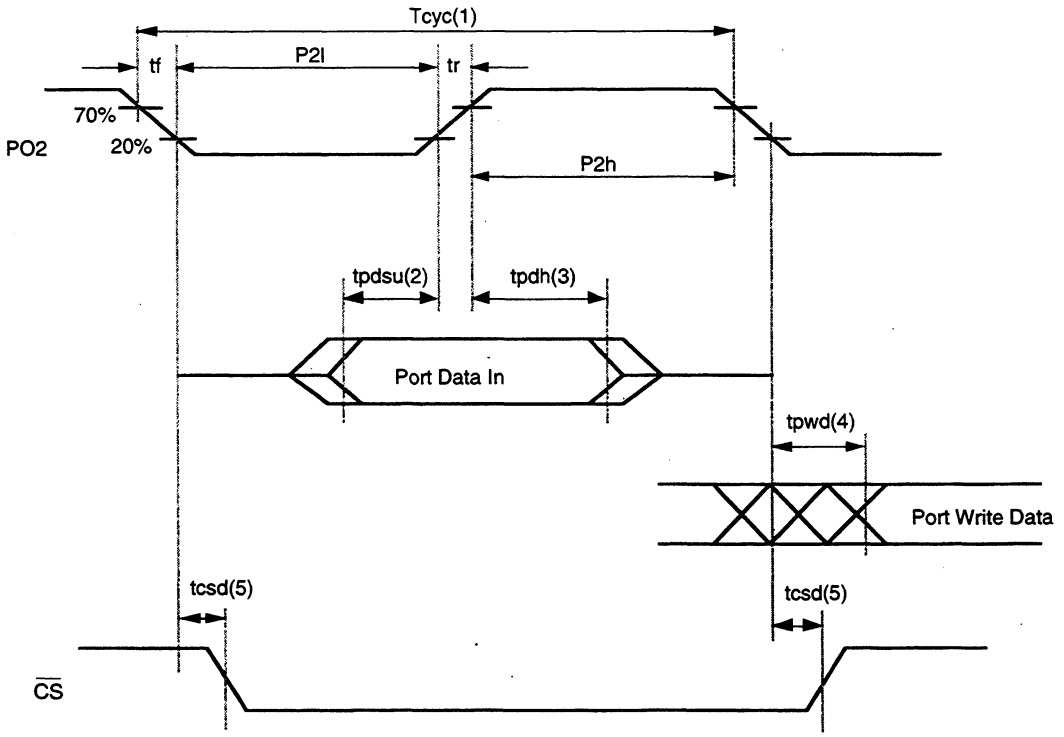


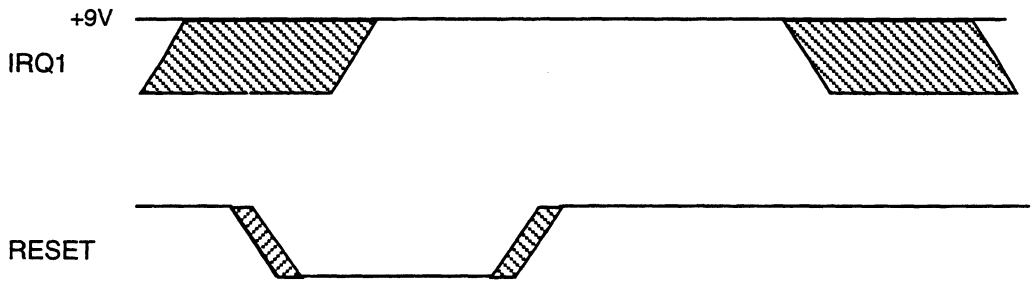
FIG. 10-2 Port/Slave CS Timing

SECTION 11
SELF-CHECK MODE

The MC68HC05L10 self-check mode is for the user to check the device functions with the on chip self-check program masked at location $0C20 to $0DE1 under minimum support hardware. Figure 11-1 shows the schematic drawing for performing self-check. The device enters self-check if the following conditions are met:

Reset	IRQ1	PB0	PB1	Mode
5V	VSS to VDD	-	-	Normal
5V	+9V Rising Edge *	VDD	VDD	Self-Check

* Minimum hold time should be 2 clock cycles, after that it can be used as IRQ normal function pin :



When the part is placed in the self-check mode the self-check vector will be fetched and the self- check firmware will start to execute.

The self-check result will be indicated by the LEDs connected to PC0 to PC3 (please refer to the following table).

PC3	PC2	PC1	PC0	REMARKS
1	0	0	1	Bad I/O
1	0	1	0	Bad RAM
1	0	1	1	Bad Timer
1	1	0	0	Bad SCI
1	1	0	1	Bad ROM
1	1	1	0	Bad SPI
1	1	1	1	Bad interrupts or IRQ Request
Flashing				Good Device
All Others				Bad Device, Bad Port A, etc.

0 indicates LED on; 1 indicates LED is off.

TABLE 11-1 SELF-CHECK RESULTS

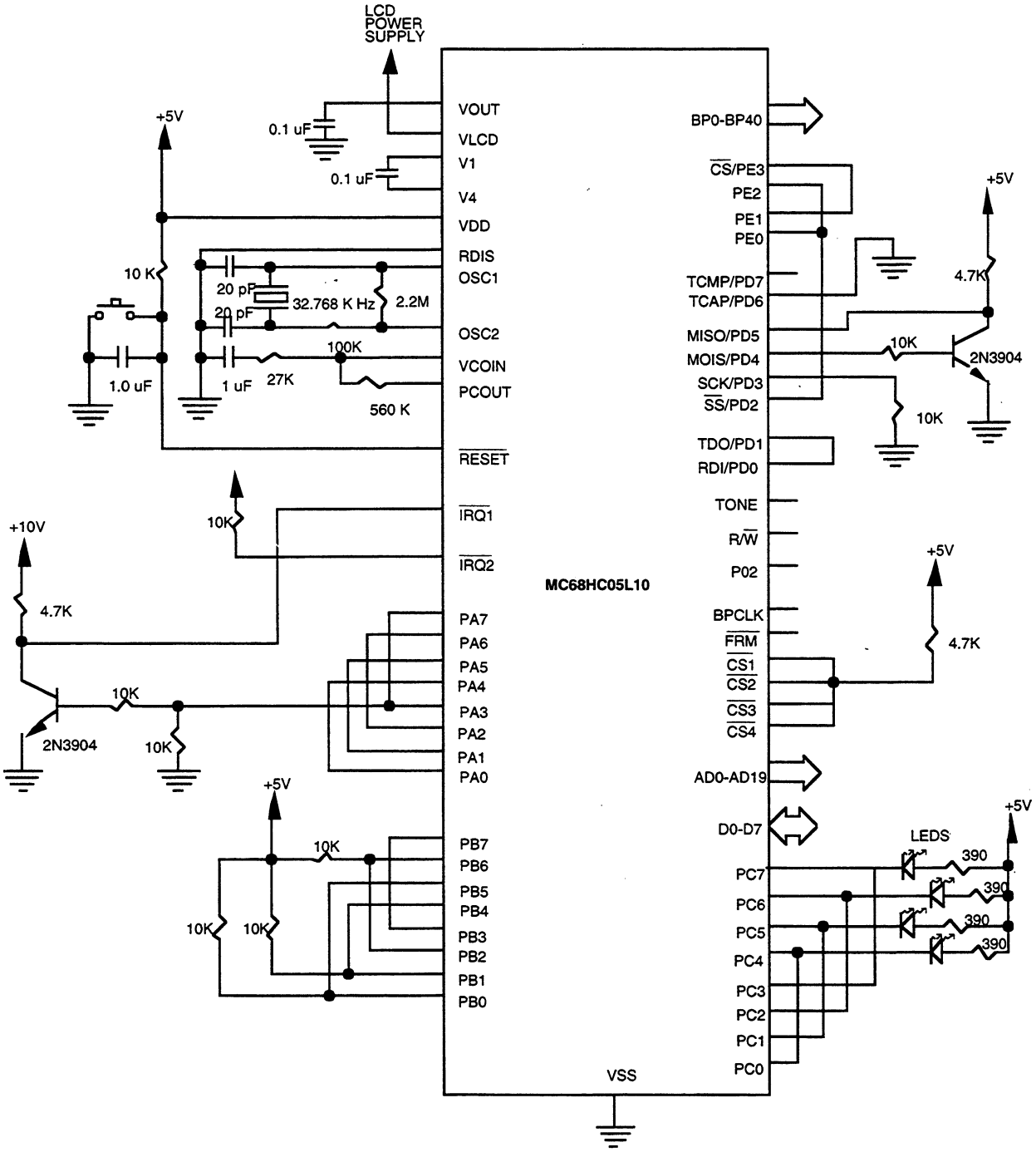
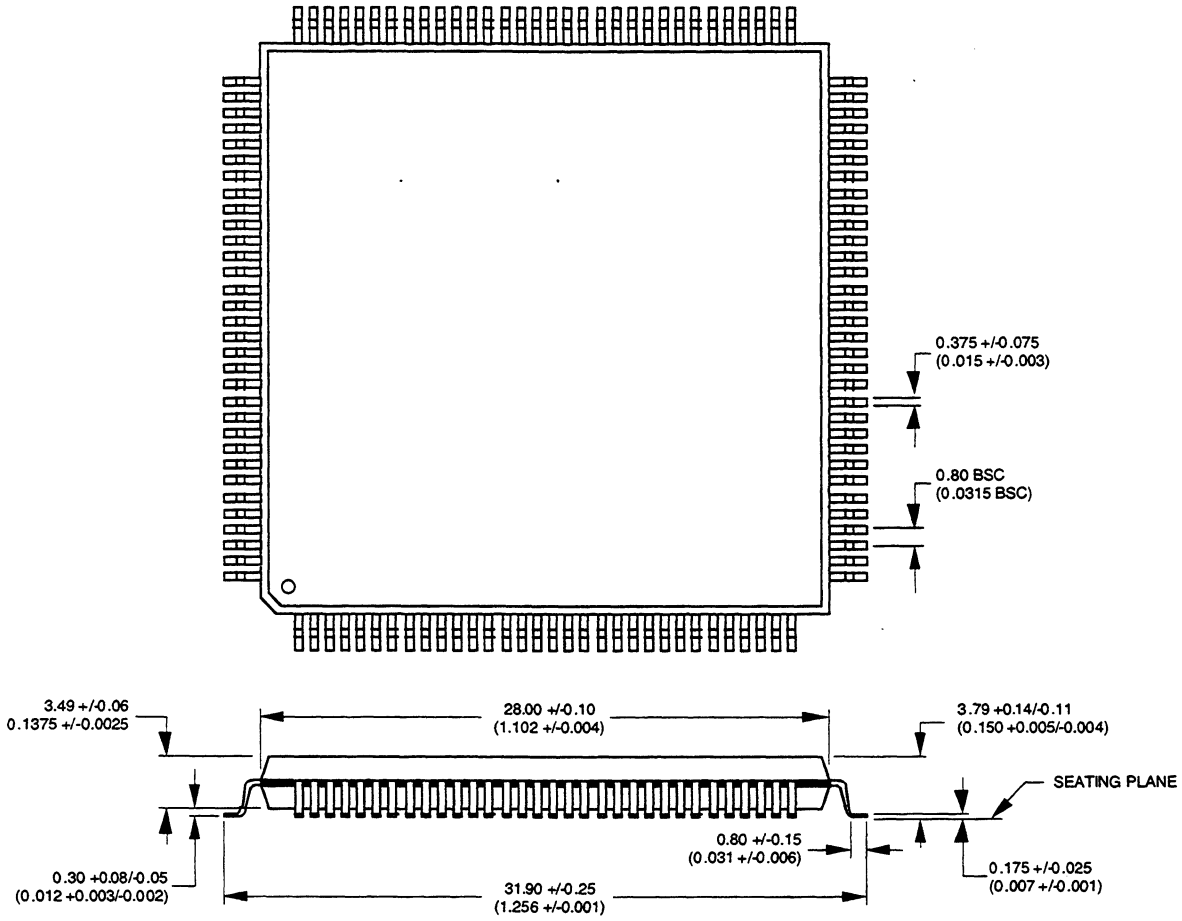


FIG 11-1 MC68HC05L10 SELF-CHECK CIRCUIT DIAGRAM

APPENDIX A
MECHANICAL DATA

A.1 PACKAGE DIMENSION

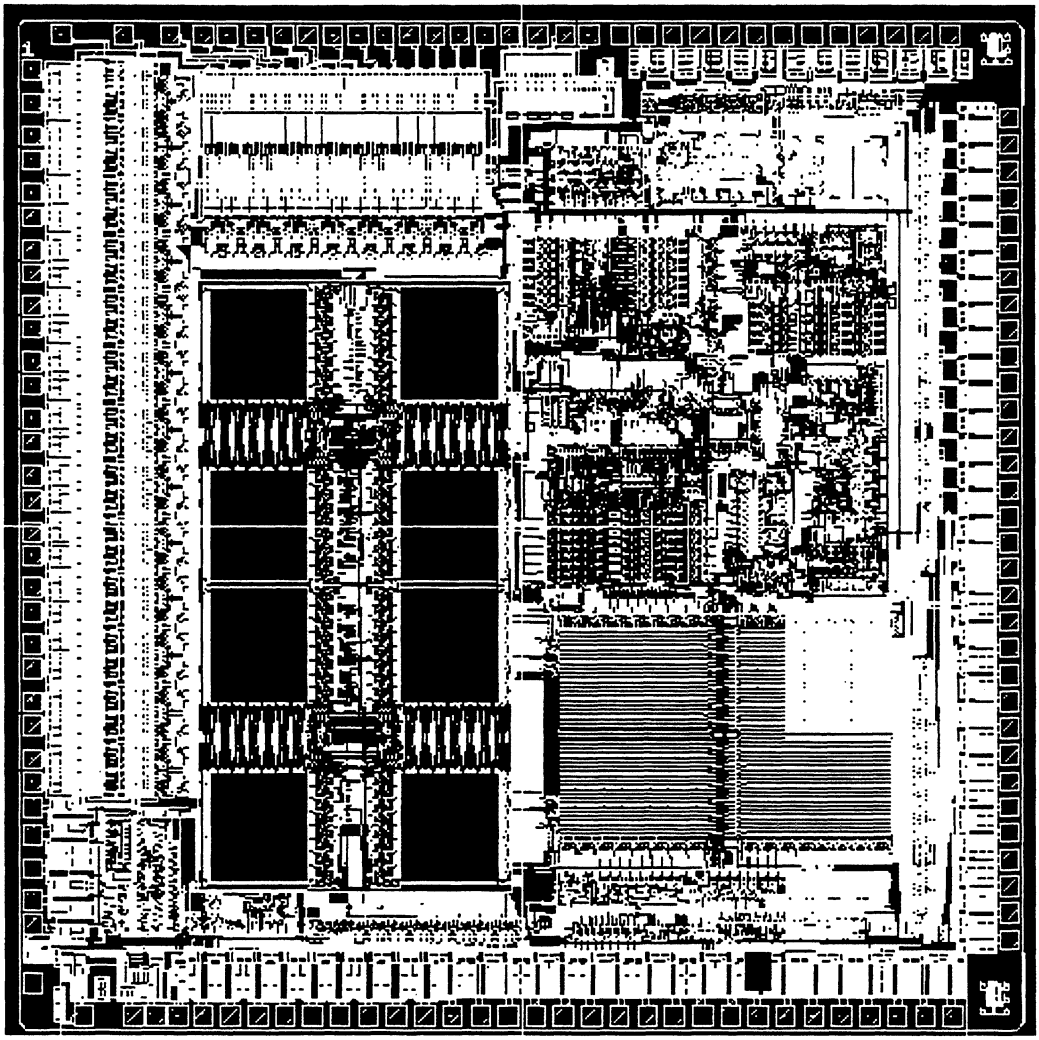


(DRAWING NOT TO SCALE)

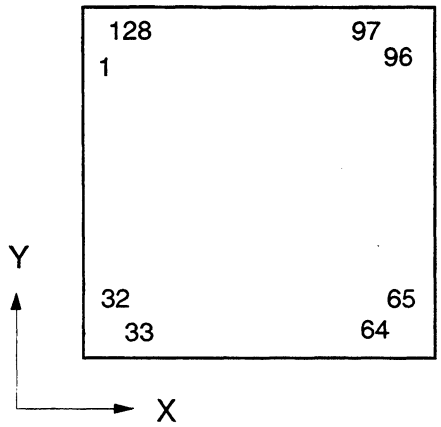
NOTE: PRINCIPAL DIMENSIONS IN MM, DIMENSIONS IN BRACKETS (INCH) FOR REFERENCE ONLY

MC68HC05L10 128 PINS QFP

A.2 BONDING DIAGRAM



A.3 BOND PAD COORDINATES

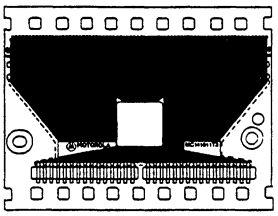


PIN NO.	PAD COORDINATES		PIN NO.	PAD COORDINATES		PIN NO.	PAD COORDINATES	
	X	Y		X	Y		X	Y
1	-3298.0	3086.0	51	471.0	-3331.0	101	2145.0	3331.0
2	-3298.0	2862.0	52	655.0	-3331.0	102	1958.0	3331.0
3	-3298.0	2639.0	53	845.0	-3331.0	103	1771.0	3331.0
4	-3298.0	2478.0	54	1030.0	-3331.0	104	1584.0	3331.0
5	-3298.0	2255.0	55	1220.0	-3331.0	105	1397.0	3331.0
6	-3298.0	2094.0	56	1404.0	-3331.0	106	1210.0	3331.0
7	-3298.0	1871.0	57	1594.0	-3331.0	107	1023.0	3331.0
8	-3298.0	1710.0	58	1779.0	-3331.0	108	836.0	3331.0
9	-3298.0	1487.0	59	1969.0	-3331.0	109	665.0	3331.0
0	-3298.0	1326.0	60	2153.0	-3331.0	110	456.0	3331.0
11	-3298.0	1103.0	61	2343.0	-3331.0	111	296.0	3331.0
12	-3298.0	942.0	62	2528.0	-3331.0	112	111.0	3331.0
13	-3298.0	719.0	63	2718.0	-3331.0	113	-69.0	3331.0
14	-3298.0	558.0	64	2902.0	-3331.0	114	-240.0	3331.0
15	-3298.0	335.0	65	3299.0	-2816.0	115	-410.0	3331.0
16	-3298.0	174.0	66	3299.0	-2636.0	116	-580.0	3331.0
17	-3298.0	-49.0	67	3299.0	-2456.0	117	-750.0	3331.0
18	-3298.0	-209.0	68	3299.0	-2276.0	118	-920.0	3331.0
19	-3298.0	-433.0	69	3299.0	-2096.0	119	-1090.0	3331.0
20	-3298.0	-593.0	70	3299.0	-1916.0	120	-1260.0	3331.0
21	-3298.0	-817.0	71	3299.0	-1736.0	121	-1430.0	3331.0
22	-3298.0	-977.0	72	3299.0	-1556.0	122	-1600.0	3331.0
23	-3298.0	-1201.0	73	3299.0	-1376.0	123	-1788.0	3331.0
24	-3298.0	-1361.0	74	3299.0	-1196.0	124	-1980.0	3331.0
25	-3298.0	-1585.0	75	3299.0	-1016.0	125	-2147.0	3331.0
26	-3298.0	-1745.0	76	3299.0	-836.0	126	-2368.0	3331.0
27	-3298.0	-1922.0	77	3299.0	-656.0	127	-2688.0	3331.0
28	-3298.0	-2112.0	78	3299.0	-476.0	128	-3103.0	3331.0
29	-3298.0	-2342.0	79	3299.0	-296.0			
30	-3298.0	-2550.0	80	3299.0	-116.0			
31	-3298.0	-2719.0	81	3299.0	64.0			
32	-3298.0	-3116.0	82	3299.0	244.0			
33	-2958.0	-3331.0	83	3299.0	424.0			
34	-2623.0	-3331.0	84	3299.0	604.0			
35	-2473.0	-3331.0	85	3299.0	784.0			
36	-2323.0	-3331.0	86	3299.0	964.0			
37	-2153.0	-3331.0	87	3299.0	1144.0			
38	-1968.0	-3331.0	88	3299.0	1324.0			
39	-1778.0	-3331.0	89	3299.0	1504.0			
40	-1593.0	-3331.0	90	3299.0	1684.0			
41	-1403.0	-3331.0	91	3299.0	1864.0			
42	-1218.0	-3331.0	92	3299.0	2044.0			
43	-1028.0	-3331.0	93	3299.0	2224.0			
44	-843.0	-3331.0	94	3299.0	2404.0			
45	-653.0	-3331.0	95	3299.0	2584.0			
46	-468.0	-3331.0	96	3299.0	2764.0			
47	-278.0	-3331.0	97	2893.0	3331.0			
48	-93.0	-3331.0	98	2706.0	3331.0			
49	96.0	-3331.0	99	2519.0	3331.0			
50	281.0	-3331.0	100	2332.0	3331.0			

APPENDIX B
MC141511

Product Preview *(Rev 3.2)*
LCD Segment Driver

MC141511



T2 SUFFIX
TAB

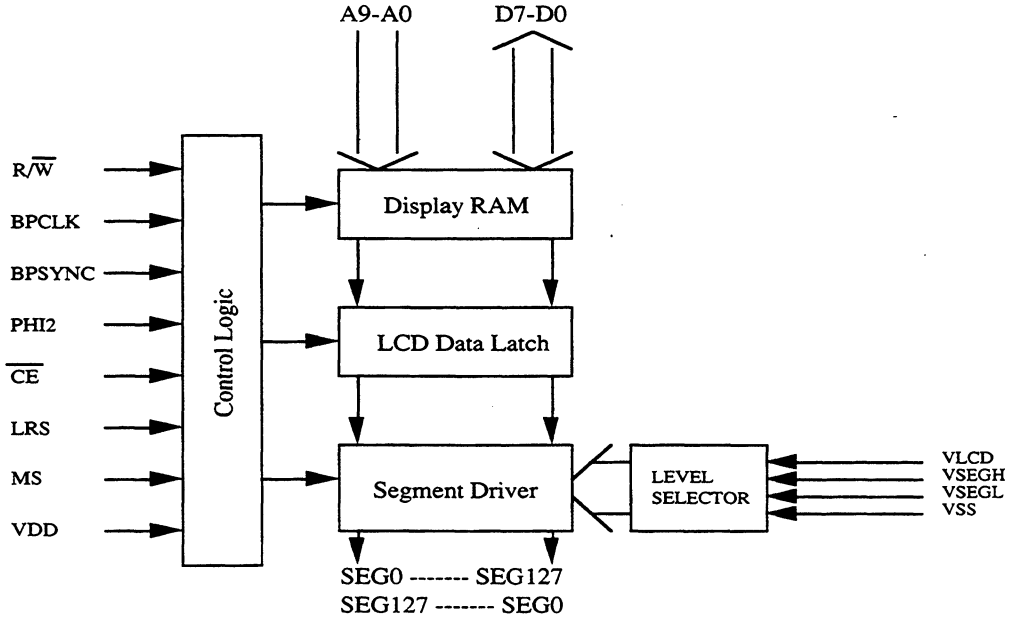
ORDERING INFORMATION

MCC141511	BARE DIE
MC141511T2	TAB

The MC141511 is a CMOS LCD driver chip which consists of 656 x 8 display RAM. It is used with MC68HC05L10 microcomputer for large LCD panel.

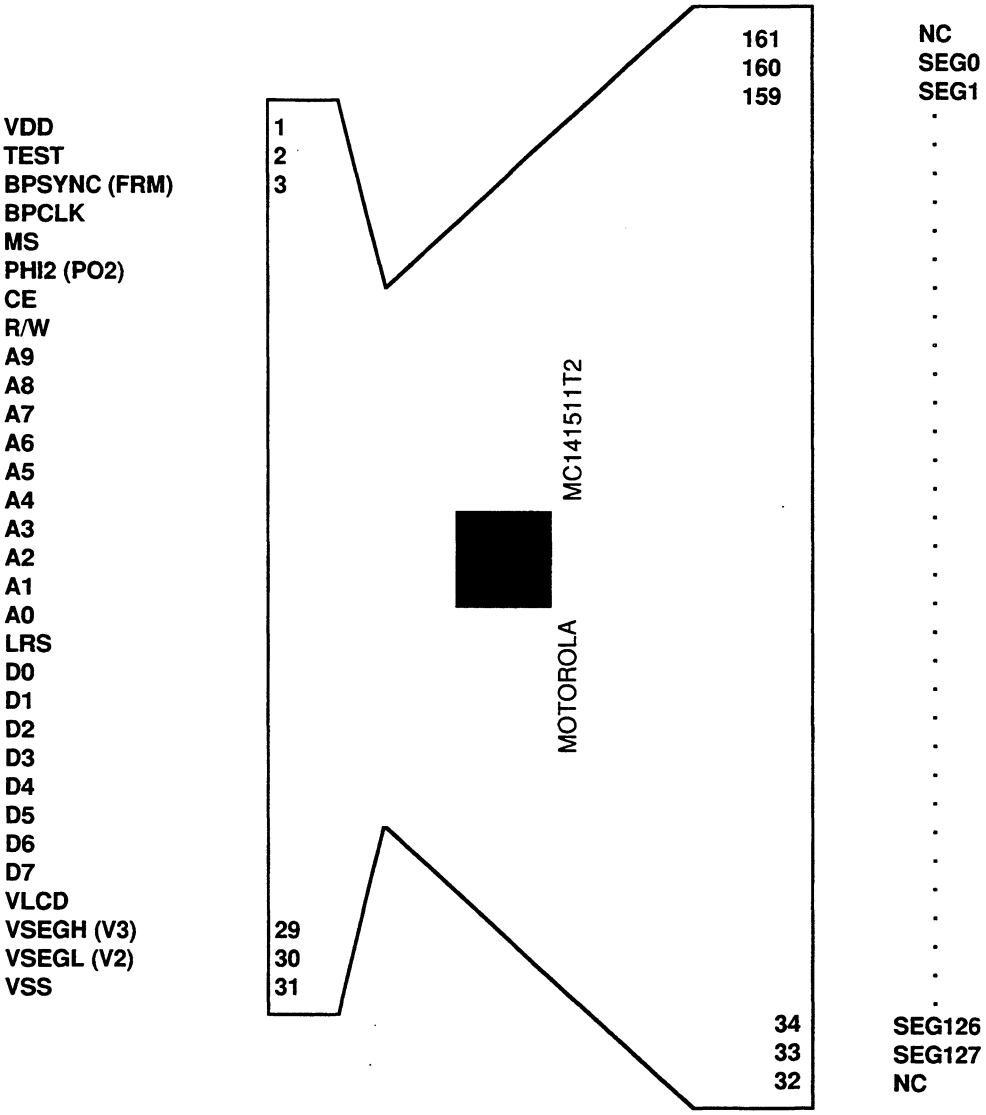
Features :
- Direct interface with MC68HC05L10
- 656 X 8 LCD static RAM (Display RAM)
- 128 LCD segment driving signal
- 10 bit address bus and 8 bit bidirectional data bus
- 1 : 32 or 1 : 41 multiplex
- TAB(161 pins) or die form (159 pins, 5 mil pad pitch, ~127μm)

BLOCK DIAGRAM



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

MC141511T2 (161 PINS TAB) PACKAGE PIN ASSIGNMENT



MAXIMUM RATINGS* (Voltages Referenced to V_{SS}, T_A = 25°C)

Ratings	Symbol	Value	Unit
Supply Voltage	VDD	-0.3 to + 7.0	V
	Vlcd	-0.3 to + 12.0	V
Input Voltage	Vin	VSS-0.3 to VDD +0.3	V
Current Drain Per Pin Excluding VDD and VSS	I	25	mA
Operating Temperature Range	TA	0 to 70	°C
Storage Temperature Range	Tstg	-65 to + 150	°C

* Maximum Ratings are those values beyond which damage to the device may occur. Functional operation should be restricted to the limits in the Electrical Characteristics tables or Pin Descriptions section.

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions to be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit. For proper operation it is recommended that Vin and Vout be constrained to the range VSS ≤ (Vin or Vout) ≤ VDD. Reliability of operation is enhanced if unused input are connected to an appropriate logic voltage level (e.g., either VSS or VDD). Unused outputs must be left open.

ELECTRICAL CHARACTERISTICS (Voltages Referenced to V_{SS}, T_A = 25°C)

Characteristics	Symbol	Min	Typ	Max	Unit
Operating Voltage Supply Voltage (reference to VSS) LCD Voltage (reference to VSS)	 VDD VLCD	 4.5 4.5	 - -	 5.5 12	 V V
Supply Current (at 5.5 V, 3.686MHz) RUN	 IDD ISB	 - -	 25 15	 200 20	 µA µA
Supply Current at VLCD	IVLCD	-	-	200	µA
Output Voltage, Iload < or = 10.0 uA	VOL VOH	- VLCD-0.1	- -	0.1 -	V V
Output High Voltage (Iload = 1.6 mA) D7-D0	VOH	VDD-0.8	-	-	V
Output Low Voltage (Iload = 1.6 mA) D7-D0	VOL	-	-	0.4	V
Input High Voltage R/$\overline{W}$, BPCLK, BPSYNC, PHI2, MS $\overline{CE}$, AD9-AD0, D7-D0	VIH	0.7xVDD	-	VDD	V
Input Low Voltage R/$\overline{W}$, BPCLK, BPSYNC, PHI2, MS $\overline{CE}$, AD9-AD0, D7-D0	VIL	VSS	-	0.2xVDD	V
Data Retention	VR	2.0	-	-	V
Input Current R/$\overline{W}$, PHI2, AD9-AD0, D7-D0	Iin	-	-	±1	µA
Capacitance R/$\overline{W}$, BPCLK, BPSYNC, PHI2, MS $\overline{CE}$, AD9-AD0, D7-D0	Cin	-	-	8	pF
Output current (VOH = 4.5V, VOL = 0.5V) D0-D7	IOH,IOL	-	±20	-	µA
Internal Pull Down Resistance TEST	Rdn	-	1	-	M Ohm

AC OPERATION CONDITIONS AND CHARACTERISTICS

WRITE CYCLE (VCC = 5.0 V ± 5%, VSS = 0)

Characteristics	Symbol	Min	Max	Unit
Write Cycle Time	tcyc (W)	400	-	ns
Address Set Up Time	tAS	100	-	ns
Address Hold Time	tAH	0	-	ns
Chip Select Pulse Width	tCS	150	-	ns
Write to Chip Select Delay Time	tWCS	100	-	ns
Data SetUp Time	tDSW	100	-	ns
Input Hold Time	tH	10	-	ns
Write Hold Time from Chip Select	tWH	0	-	ns

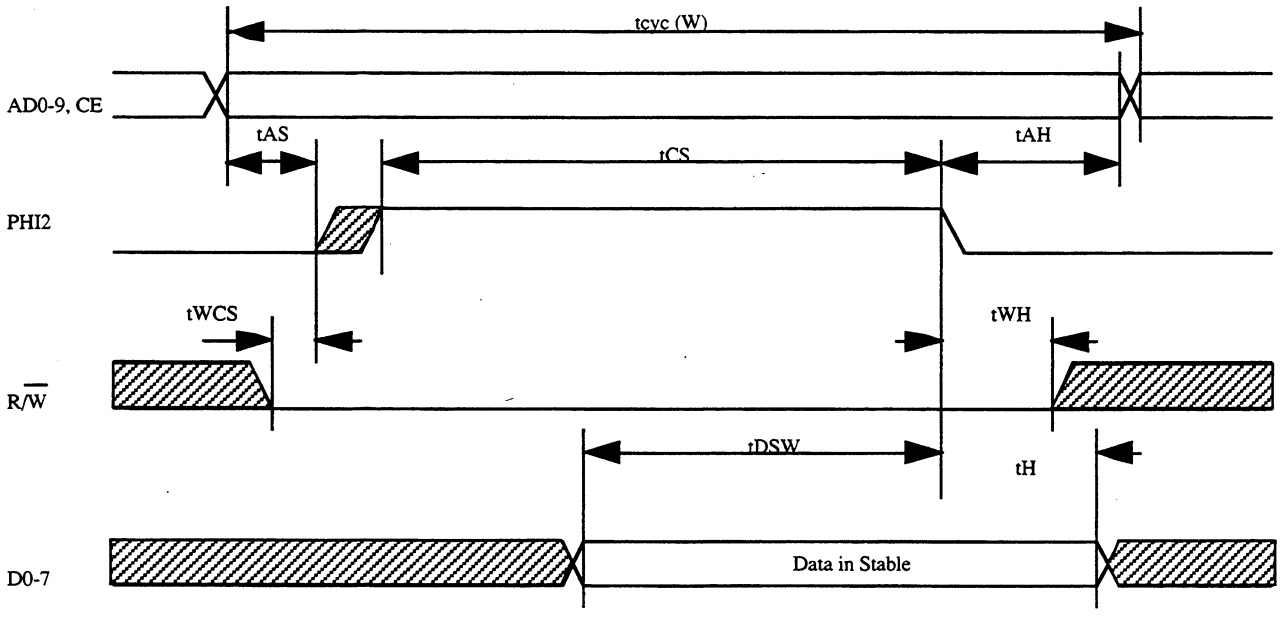


FIGURE 1 - WRITE CYCLE TIMING

READ CYCLE (VCC = 5.0 V ± 5%, VSS = 0)

Characteristics	Symbol	Min	Max	Unit
READ Cycle Time	tcyc (R)	400	-	ns
Address set up time	tas	100	-	ns
Data Delay Time (Read)	tDDR	-	150	ns
Output Hold Time	tH	10	-	ns

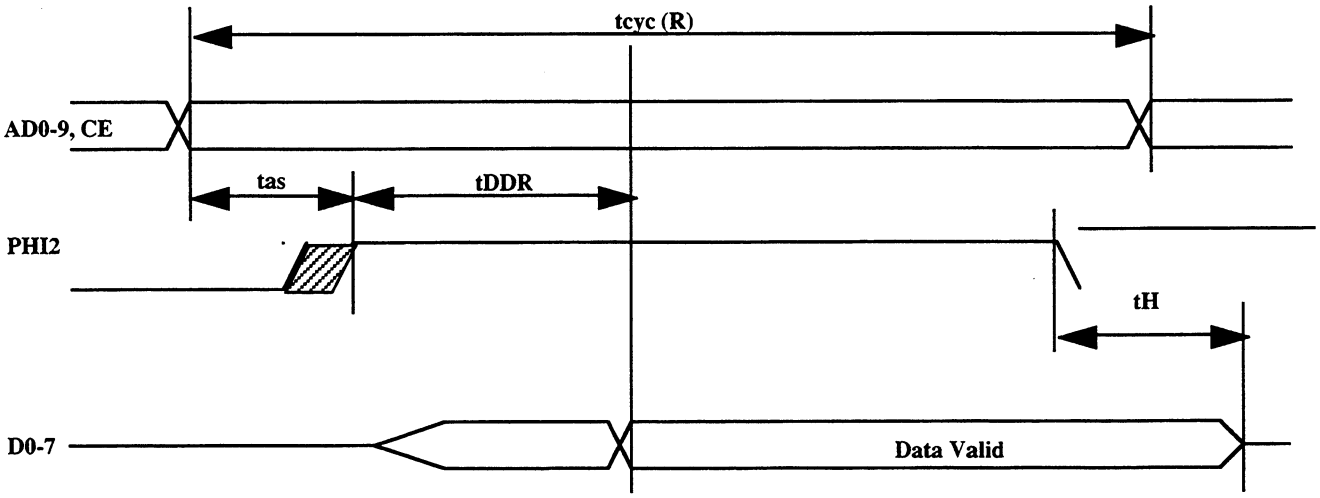


FIGURE 2 - READ CYCLE TIME

PIN DESCRIPTION

VDD and VSS
DC power is supplied to the part on these two pins. VDD is power and VSS is ground.

VLCD
This supply pin provides the voltage level for the segment driver and is connected to the Vout pin of MC68HC05L10.

VSEGL, VSEGH
These input pins are connected to V2 and V3 (Fig.3) of external voltage divider.

D0-D7
Eight bit wide bidirectional data bus which are connected to D0-D7 of MC68HC05L10.

A0-A9
Ten bit wide address bus for addressing the display RAM and are connected to A0-A9 of MC68HC05L10.

BPSYNC
A periodic active low output from MC68HC05L10 to MC141511 for timing synchronization. This pin is connected to $\overline{FRM}$ of MC68HC05L10.

BPCLK
A 2.048KHZ 50% duty cycle signal which provides the required frame frequency for the segment driver. This pin is connected to BPCLK of MC68HC05L10.
In addition, this pin can be a 4.096KHZ 50% duty cycle signal of which the period of BPSYNC signal will be 1/64 sec.

PHI2
A bus clock input that is used for data bus timing synchronization. This pin is connected to P02 of MC68HC05L10.

SEG0 - SEG127
These 128 output lines provide the segment drive signal to the LCD panel. They are all grounded while display is turned off.

$\overline{CE}$
This is an active low input for chip enable. This pin is connected to either $\overline{CS1}$, $\overline{CS2}$, $\overline{CS3}$ or $\overline{CS4}$ of MC68HC05L10.

LRS
The left-right selection pin defines the direction of the segment driver display (Fig 5).
0 = SEG 0 - 127
1 = SEG 127 - 0

MS
This input pin selects how display RAM is addressed according to 1:32 or 1:41 multiplex ratio.
0 = 1:32 multiplex addressing
1 = 1:41 multiplex addressing

R/$\overline{W}$
R/$\overline{W}$ is an input that indicates which direction the data is to be passed over the data bus. When R/$\overline{W}$ is low, the LCD driver will be reading data from the data bus (D0-D7). When R/$\overline{W}$ is high, the LCD drive will be writing data to the data bus (D0-D7). This pin is connected to R/$\overline{W}$ of MC68HC05L10.

TEST
Allowing this pin float or connecting it to GND set the part in the normal mode.

OPERATION OF LCD DRIVER

INTRODUCTION

MC141511 is the LCD driver with1 : 32 or 1 : 41 multiplex addressing mode and consists of the following circuitry.

CONTROL LOGIC provides the control signals for display synchronisation.

DISPLAY RAM which stores the display data, and each bit of the display RAM is one to one corresponding to the pixel of the LCD. Display RAM is in vertical byte oriented format as shown in Figure 5 and the way display RAM is addressed depends on the multiplexing mode of the LCD (Fig. 7). With reference to fig 5, display RAM also contains 16 bytes of memory which is in horizontal format ($280-$28F). These display RAM is addressed when backplane reaches 41.

LEVEL SELECTOR consists of switching circuit to select appropriate voltage levels from external voltage divider (Fig 3).

SEGMENT DRIVER provides the segment driving signal to the LCD (Fig 6).

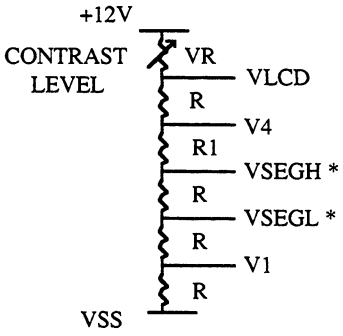
LCD driver clock is derived from the 2.048KHz BPCLK and frame frequency is 64 Hz for 1:32 multiplex and 50 Hz for 1:41 multiplex (Fig 4).

Generation of LCD bias levels (Fig 5)

In order to obtain optimum constrast for LCD panels, the bias levels should be selected such that

$$\text{BIAS} = R/(4R+R1) = 1/(\sqrt{MUX}+1)$$
$$V1/VLCD = 1/(\sqrt{MUX}+1)$$
$$V2/VLCD = 2/(\sqrt{MUX}+1)$$
$$V3/VLCD = (\sqrt{MUX}-1)/(\sqrt{MUX}+1)$$
$$V4/VLCD = \sqrt{MUX}/(\sqrt{MUX}+1)$$

Example : Mux = 41 ----- Bias = 1 : 7.4 ,
R = 10K, R1 = 3 3K, VR = 100K
Mux = 32 ----- Bias = 1 : 6.6 ,
R = 10k, R1 = 2 7K, VR = 100K



*Note : VSEGH is the same as V3 that shown in MCU data sheet, appendix A-2.
VSEGL is the same as V2 that shown in MCU data sheet, appendix A-2.

FIGURE 3 - External Voltage Divider

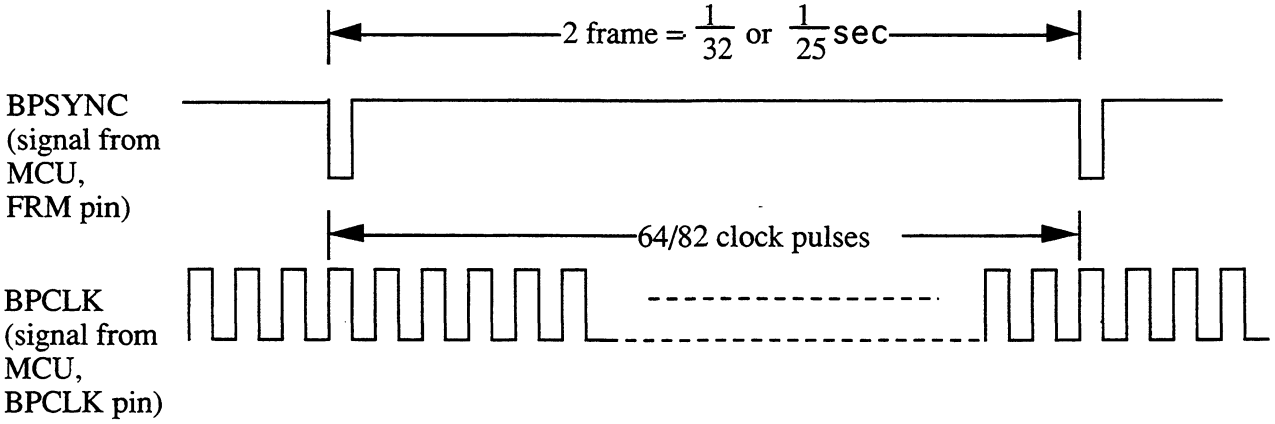


FIGURE 4 - Relationship between BPSYNC and BPCLK

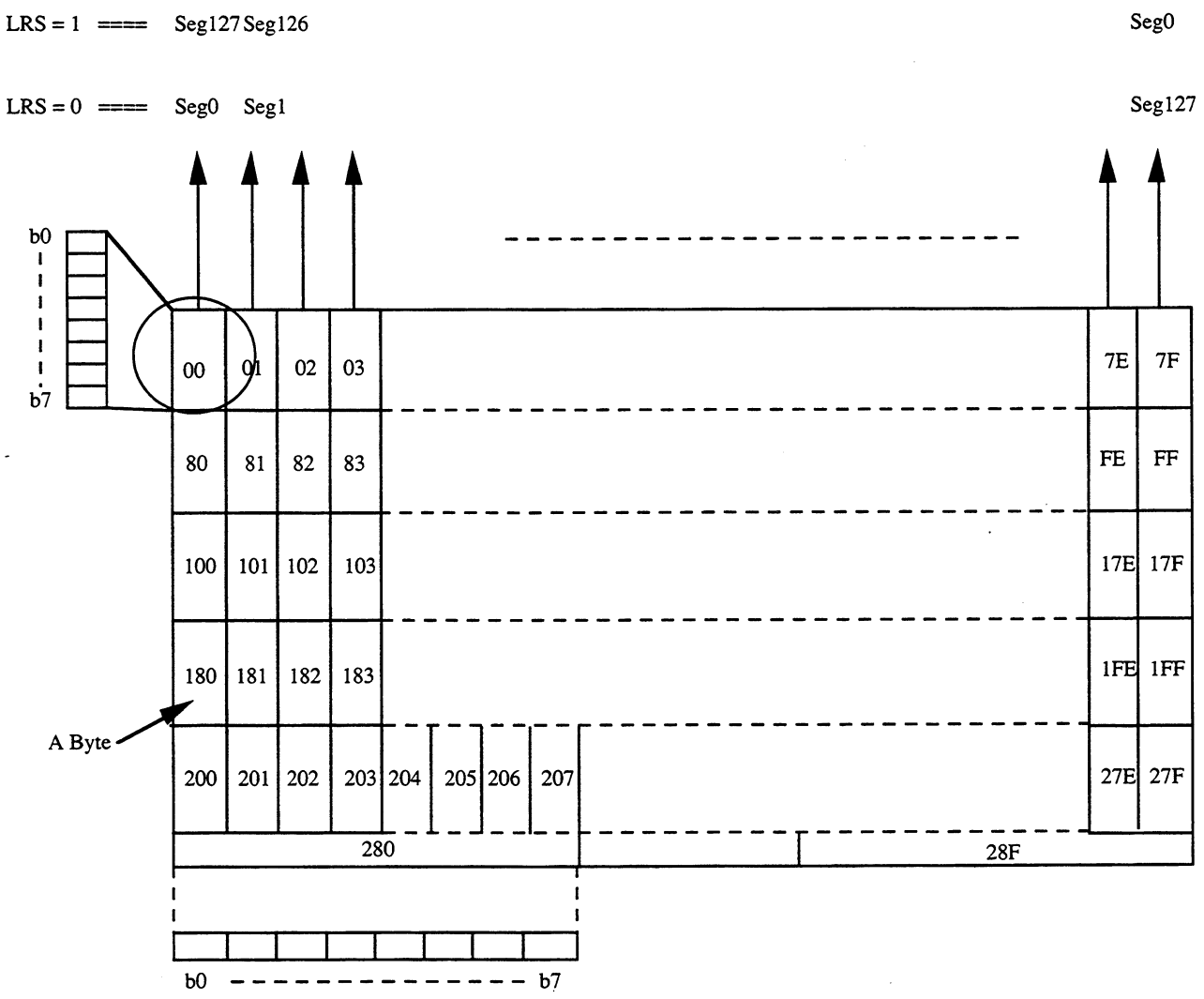


FIGURE 5 - Display RAM configuration

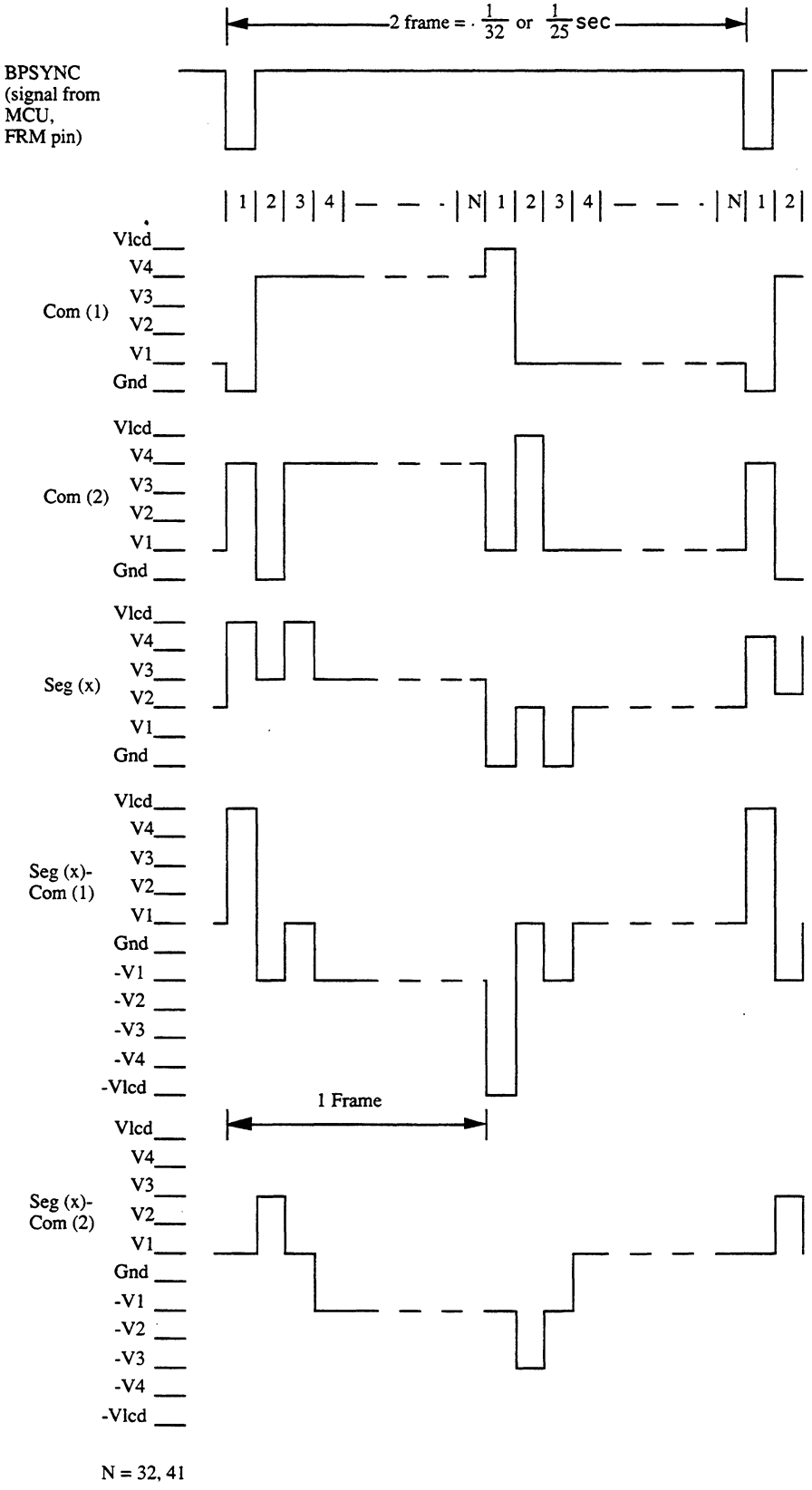


FIGURE 6 - Driving waveforms of 1:5 bias, 1:32 or 1:41 multiplex

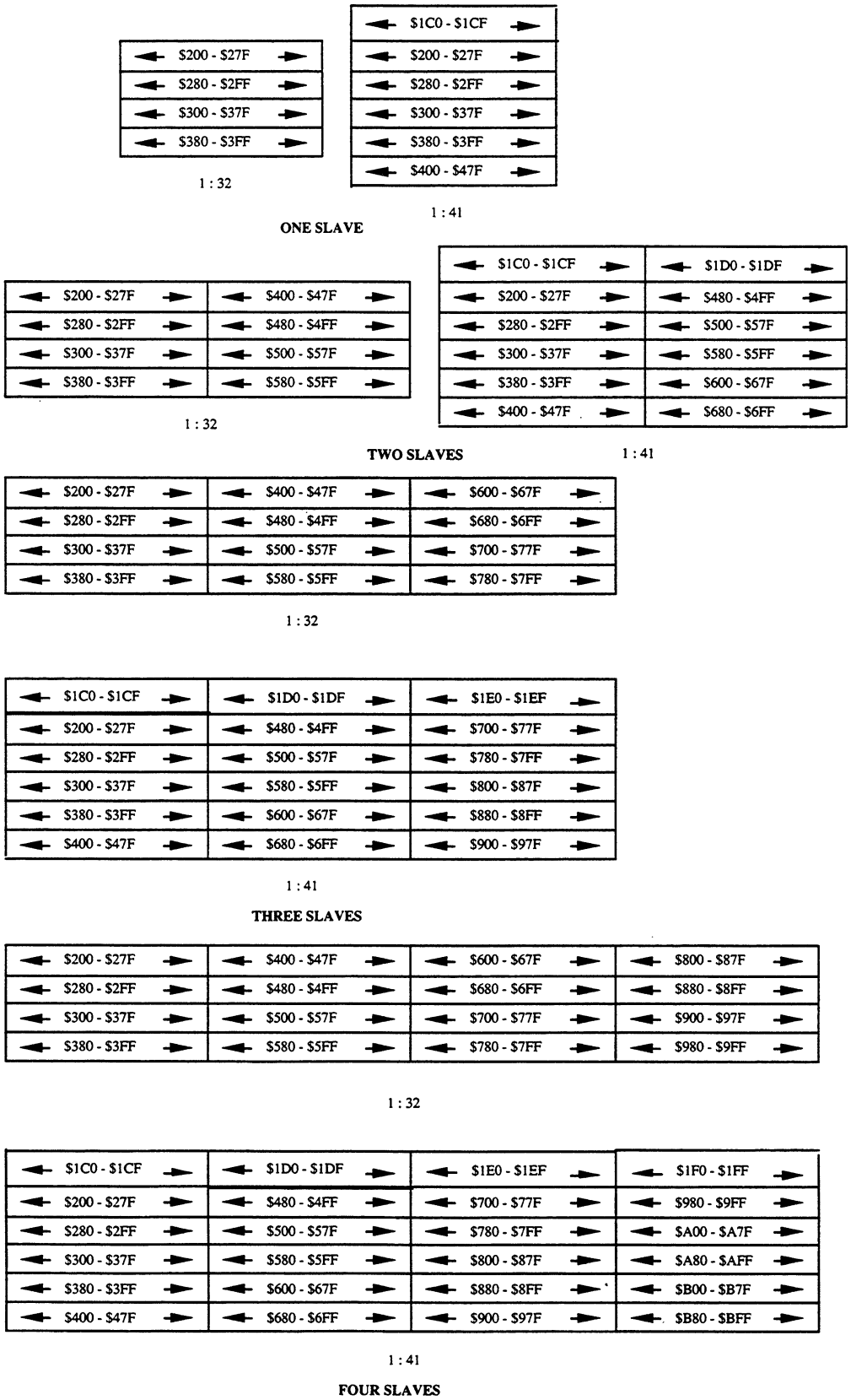
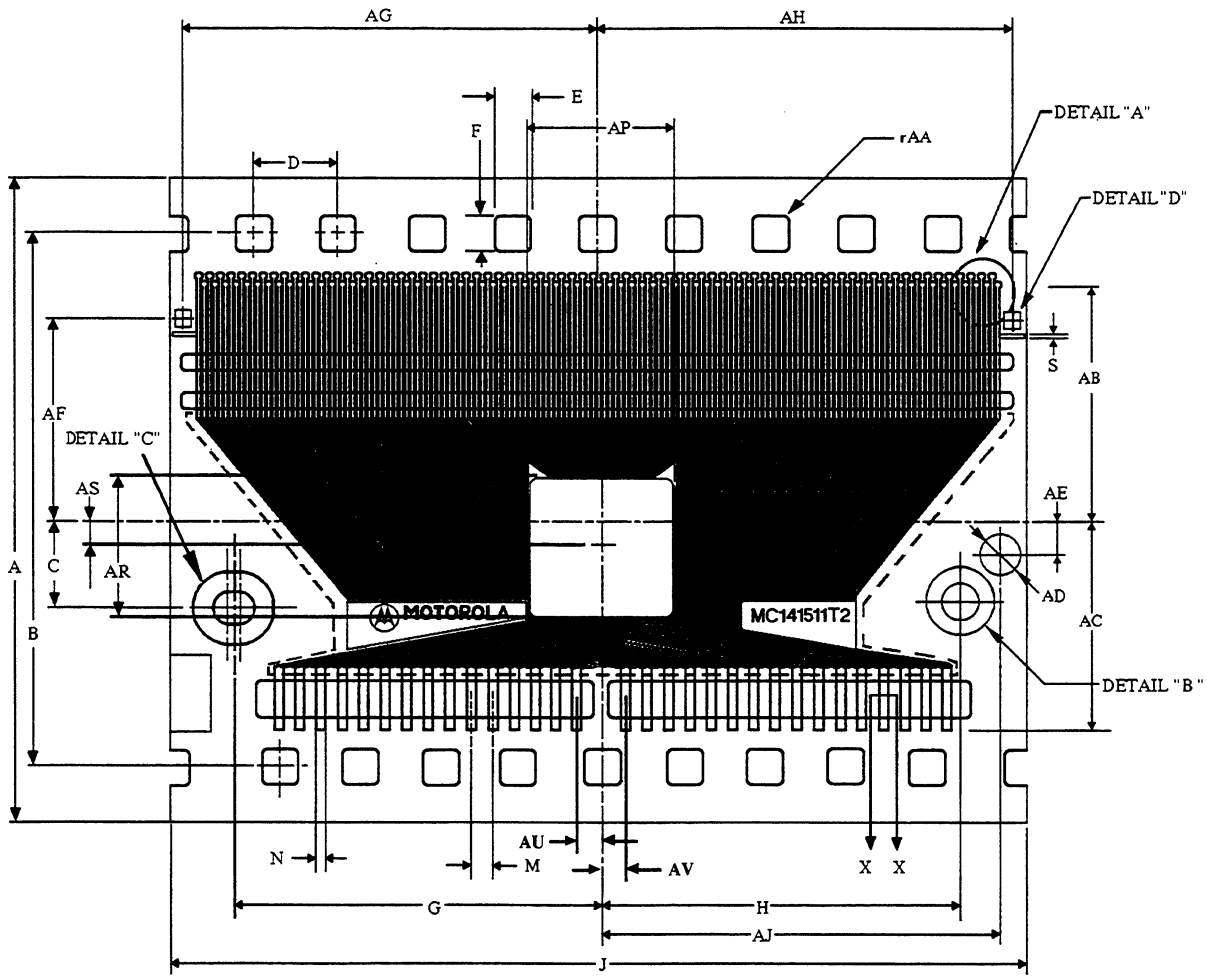
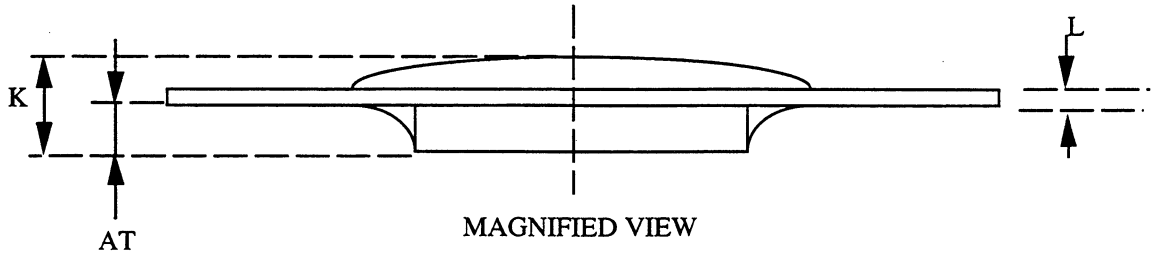


FIGURE 7 - Display RAM mapping for 1:32 and 1:41 multiplex ratio

MC141511T2 TAB PACKAGE DIMENSION (1 of 3)

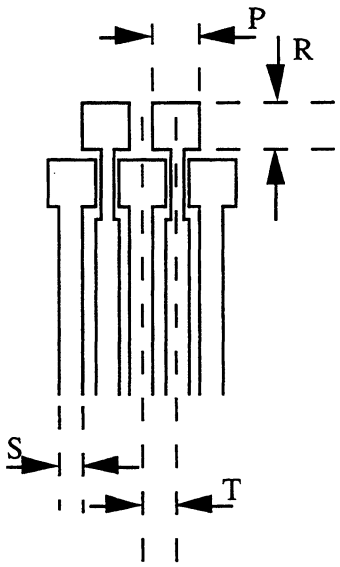
DRAWING NOT TO SCALE



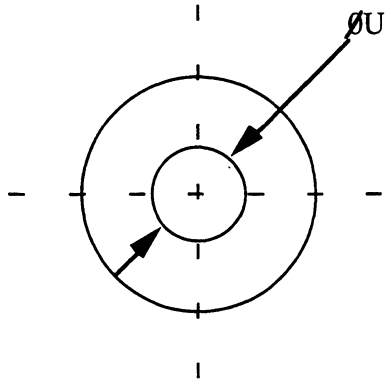


MC141511T2 TAB PACKAGE DIMENSION (2 of 3)

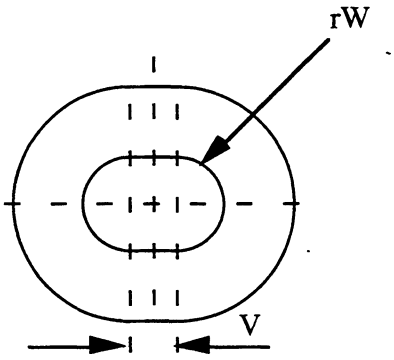
DRAWING NOT TO SCALE



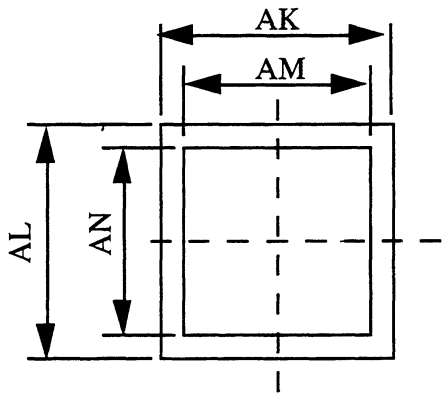
DETAIL "A"



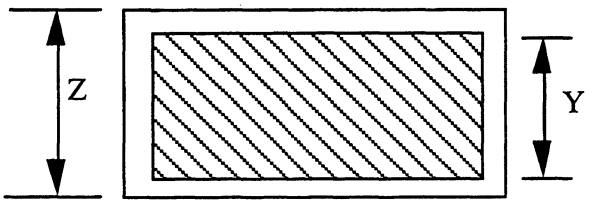
DETAIL "B"



DETAIL "C"



DETAIL "D"



SECTION X-X

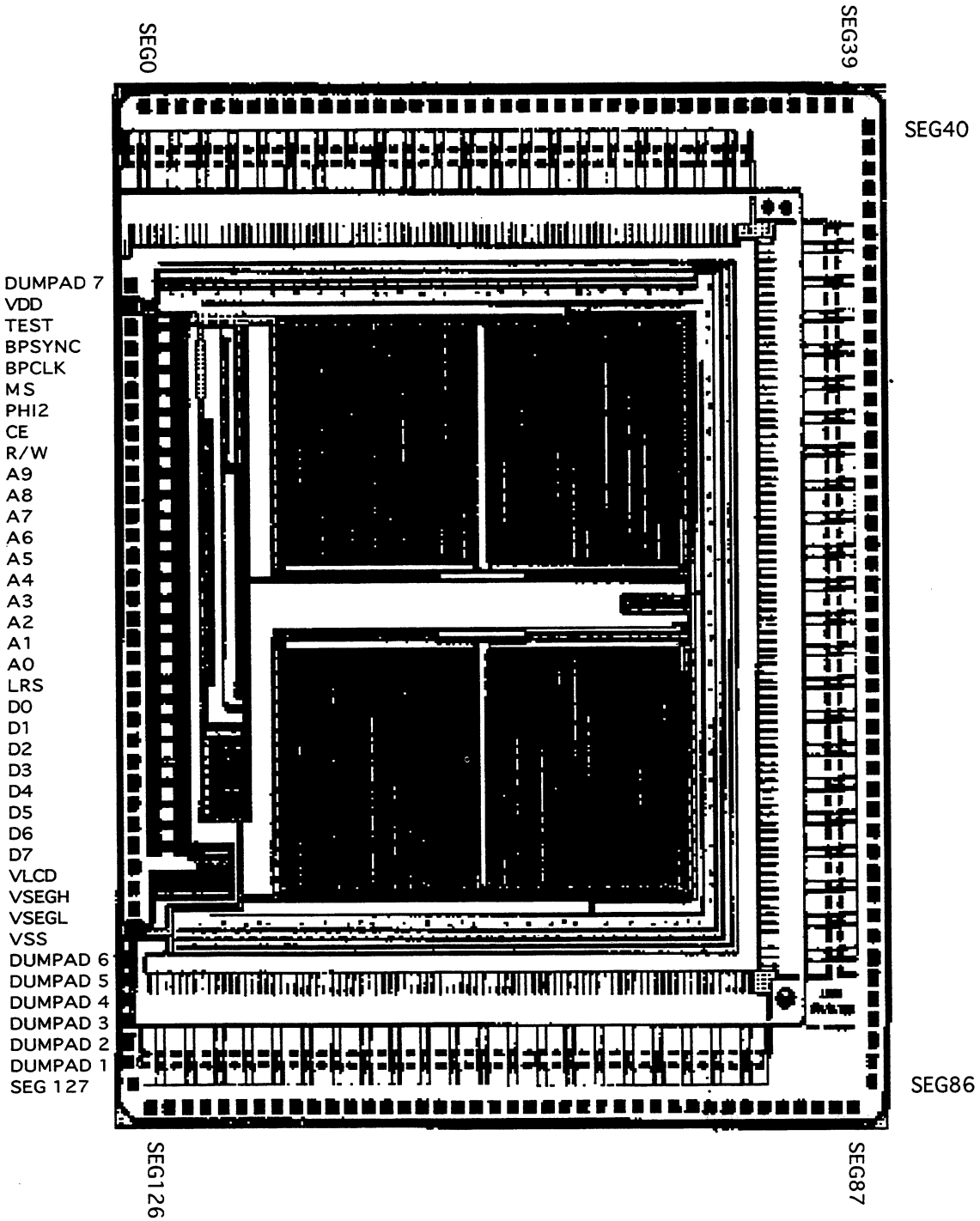
MC141511T2 TAB PACKAGE DIMENSION (3 of 3)

Notes:

1. Dimensioning and tolerancing per ANSI Y14.5M, 1982.
2. Controlling dimension: millimeter.
3. Recommended excise area J x (AB + AC)

	MILLIMETERS		INCHES			MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX	DIM	MIN	MAX	MIN	MAX
A	34.775	35.175	1.3691	1.3848	Y	0.032	0.038	0.0013	0.0015
B	28.927	29.027	1.1389	1.1428	Z	0.032	0.038	0.0013	0.0015
C	4.950	5.050	0.1949	0.1988	AA	---	0.20	---	0.0079
D	4.720	4.780	0.1858	0.1882	AB	10.9	11.9	0.4291	0.4685
E	1.951	2.011	0.0768	0.0792	AC	11.9	12.9	0.4685	0.5079
F	1.951	2.011	0.0768	0.0792	AD	1.950	2.050	0.0768	0.0807
G	20.450	20.550	0.8051	0.8091	AE	1.500	2.500	0.0591	0.0984
H	20.450	20.550	0.8051	0.8091	AF	9.779	9.879	0.3850	0.3889
J	47.000	48.000	1.8504	1.8898	AG	23.155	23.255	0.9116	0.9156
K	0.686	0.838	0.027	0.033	AH	23.155	23.255	0.9116	0.9156
L	0.0675	0.0825	0.0027	0.0032	AJ	21.750	22.750	0.8563	0.8957
M	1.260	1.280	0.0496	0.0504	AK	0.750	0.850	0.0295	0.0335
N	0.480	0.520	0.0189	0.0205	AL	0.750	0.850	0.0295	0.0335
P	0.350	0.450	0.0138	0.0177	AM	0.600	0.700	0.0236	0.0276
R	0.350	0.450	0.0138	0.0177	AN	0.600	0.700	0.0236	0.0276
S	0.150	0.190	0.0059	0.0075	AP	---	9.600	---	0.3780
T	0.340	0.360	0.0134	0.0142	AR	---	8.229	---	0.3240
U	1.750	1.850	0.0689	0.0728	AS	1.450	1.550	0.0571	0.0610
V	0.450	0.550	0.0177	0.0217	AT	0.5794	0.6294	0.0228	0.0248
W	0.850	0.950	0.0335	0.0374	AU	1.220	1.320	0.048	0.052
					AV	1.220	1.320	0.048	0.052

MCC141511 PAD ASSIGNMENT



MC141511 PAD COORDINATES : (UNIT : μm)

PIN NAME	X	Y	PIN NAME	X	Y	PIN NAME	X	Y	PIN NAME	X	Y
Dumpad 7	2616.90	-1998.92	SEG126	2490.62	3082.20	SEG86	-2616.90	2917.42	SEG39	-2458.50	-3082.20
VDD	2616.90	-1870.44	SEG125	2363.02	3082.20	SEG85	-2616.90	2789.82	SEG38	-2330.90	-3082.20
TEST	2616.90	-1741.96	SEG124	2235.42	3082.20	SEG84	-2616.90	2662.22	SEG37	-2203.30	-3082.20
BPSYNC	2616.90	-1613.48	SEG123	2107.82	3082.20	SEG83	-2616.90	2534.62	SEG36	-2075.70	-3082.20
BPCLK	2616.90	-1485.00	SEG122	1980.22	3082.20	SEG82	-2616.90	2407.02	SEG35	-1948.10	-3082.20
MS	2616.90	-1356.52	SEG121	1852.62	3082.20	SEG81	-2616.90	2279.42	SEG34	-1820.50	-3082.20
PHI2	2616.90	-1228.04	SEG120	1725.02	3082.20	SEG80	-2616.90	2151.82	SEG33	-1692.90	-3082.20
CE	2616.90	-1099.56	SEG119	1597.42	3082.20	SEG79	-2616.90	2024.22	SEG32	-1565.30	-3082.20
R/W	2616.90	-971.08	SEG118	1469.82	3082.20	SEG78	-2616.90	1896.62	SEG31	-1437.70	-3082.20
A9	2616.90	-842.60	SEG117	1342.22	3082.20	SEG77	-2616.90	1769.02	SEG30	-1310.10	-3082.20
A8	2616.90	-714.12	SEG116	1214.62	3082.20	SEG76	-2616.90	1641.42	SEG29	-1182.50	-3082.20
A7	2616.90	-585.64	SEG115	1087.02	3082.20	SEG75	-2616.90	1513.82	SEG28	-1054.90	-3082.20
A6	2616.90	-457.16	SEG114	959.42	3082.20	SEG74	-2616.90	1386.22	SEG27	-927.30	-3082.20
A5	2616.90	-328.68	SEG113	831.82	3082.20	SEG73	-2616.90	1258.62	SEG26	-799.70	-3082.20
A4	2616.90	-200.20	SEG112	704.22	3082.20	SEG72	-2616.90	1131.02	SEG25	-672.10	-3082.20
A3	2616.90	-71.72	SEG111	576.62	3082.20	SEG71	-2616.90	1003.42	SEG24	-544.50	-3082.20
A2	2616.90	56.76	SEG110	449.02	3082.20	SEG70	-2616.90	875.82	SEG23	-416.90	-3082.20
A1	2616.90	185.24	SEG109	321.42	3082.20	SEG69	-2616.90	748.22	SEG22	-289.30	-3082.20
A0	2616.90	313.72	SEG108	193.82	3082.20	SEG68	-2616.90	620.62	SEG21	-161.70	-3082.20
LRS	2616.90	442.20	SEG107	66.22	3082.20	SEG67	-2616.90	493.02	SEG20	-34.10	-3082.20
D0	2616.90	570.68	SEG106	-61.38	3082.20	SEG66	-2616.90	365.42	SEG19	93.50	-3082.20
D1	2616.90	699.16	SEG105	-188.98	3082.20	SEG65	-2616.90	237.82	SEG18	221.10	-3082.20
D2	2616.90	827.64	SEG104	-316.58	3082.20	SEG64	-2616.90	110.22	SEG17	348.70	-3082.20
D3	2616.90	956.12	SEG103	-444.18	3082.20	SEG63	-2616.90	-17.38	SEG16	476.30	-3082.20
D4	2616.90	1084.60	SEG102	-571.78	3082.20	SEG62	-2616.90	-144.98	SEG15	603.90	-3082.20
D5	2616.90	1213.08	SEG101	-699.38	3082.20	SEG61	-2616.90	-272.58	SEG14	731.50	-3082.20
D6	2616.90	1341.56	SEG100	-826.98	3082.20	SEG60	-2616.90	-400.18	SEG13	859.10	-3082.20
D7	2616.90	1470.04	SEG99	-954.58	3082.20	SEG59	-2616.90	-527.78	SEG12	986.70	-3082.20
VLCD	2616.90	1598.52	SEG98	-1082.18	3082.20	SEG58	-2616.90	-655.38	SEG11	1114.30	-3082.20
VSEGH	2616.90	1727.00	SEG97	-1209.78	3082.20	SEG57	-2616.90	-782.98	SEG10	1241.90	-3082.20
VSEGL	2616.90	1855.48	SEG96	-1337.38	3082.20	SEG56	-2616.90	-910.58	SEG9	1369.50	-3082.20
VSS	2616.90	1983.96	SEG95	-1464.98	3082.20	SEG55	-2616.90	-1038.18	SEG8	1497.10	-3082.20
Dumpad 6	2651.66	2150.72	SEG94	-1592.58	3082.20	SEG54	-2616.90	-1165.78	SEG7	1624.70	-3082.20
Dumpad 5	2651.66	2279.20	SEG93	-1720.18	3082.20	SEG53	-2616.90	-1293.38	SEG6	1752.30	-3082.20
Dumpad 4	2651.66	2407.68	SEG92	-1847.78	3082.20	SEG52	-2616.90	-1420.98	SEG5	1879.90	-3082.20
Dumpad 3	2651.66	2536.16	SEG91	-1975.38	3082.20	SEG51	-2616.90	-1548.58	SEG4	2007.50	-3082.20
Dumpad 2	2651.66	2664.64	SEG90	-2102.98	3082.20	SEG50	-2616.90	-1676.18	SEG3	2135.10	-3082.20
Dumpad 1	2651.66	2793.12	SEG89	-2230.58	3082.20	SEG49	-2616.90	-1803.78	SEG2	2262.70	-3082.20
SEG127	2616.90	2939.20	SEG88	-2358.18	3082.20	SEG48	-2616.90	-1931.38	SEG1	2390.30	-3082.20
			SEG87	-2485.8	3082.20	SEG47	-2616.90	-2058.98	SEG0	2517.90	-3082.20
						SEG46	-2616.90	-2186.58			
						SEG45	-2616.90	-2314.18			
						SEG44	-2616.90	-2441.78			
						SEG43	-2616.90	-2569.38			
						SEG42	-2616.90	-2696.98			
						SEG41	-2616.90	-2824.58			
						SEG40	-2616.90	-2952.18			

Note : 1mil ~ 25.4μm

MC68HC05L10 MCU ORDERING FORM

Date _____ Customer PO Number _____

Customer Company _____

Address _____

City _____ State _____ Zip_____

Country _____

Phone _____ Extension _____

Customer Contact Person _____

Customer Part Number _____
(12 characters maximum - if applicable)

Application _____ End Equipment _____

Temperature Range
- ☐ 0° to +70° C for package
- ☐ -40° to +85° C for package
- ☐ +25° C for dice

Package Type
- ☐ 128-pin QFP
- ☐ Die

Operation Bus Frequency
- ☐ 307 KHz
- ☐ 1.22 MHz
- ☐ 2.45 MHz
- ☐ 3.68 MHz

Voltage Supply (±10%)
- ☐ +5.0 V
- ☐ +3.0 V*

On-Chip Function Used
- ☐ Timer
- ☐ SPI
- ☐ SCI
- ☐ RTC

Mask Options

1. Crystal Oscillator in 'STOP' Mode
   - ☐ Running (RTC option)
   - ☐ Stop

2. POR & STOP Recovery Cycles
   - ☐ 16**
   - ☐ 4064

* 3.0V is not guaranteed at 2.45 & 3.68 MHz bus frequency operation.

** Longer delay is needed through external R-C reset circuit for the crystal OSC to become stable from POR. Mask option of OSC stops in 'STOP' mode is not recommended if this option is chosen

Special Electrical Provisions : _____
(Customer specifications required)

_____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purpose only.

_____
(SIGNATURE)

Device to be tested to customer specifications. (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.