MC68HC05T10D/H

8-Bit Microcomputers

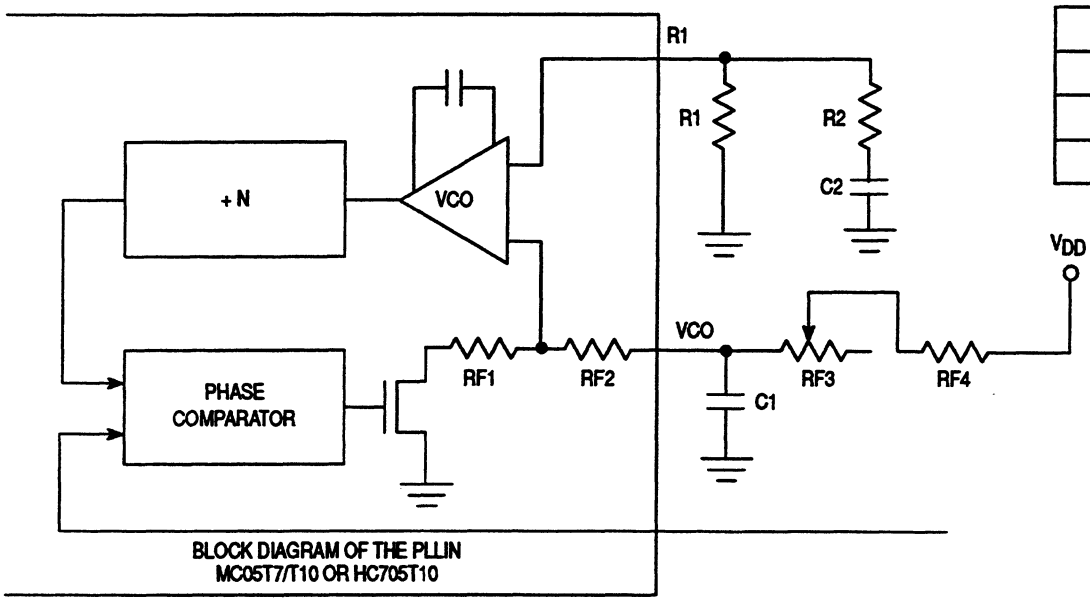
MC68HC05T10
MC68HC05T7
MC68HC705T10

TECHNICAL
DATA

External Components Required for PLL in MC68HC05T7/T10 and MC68HC705T10

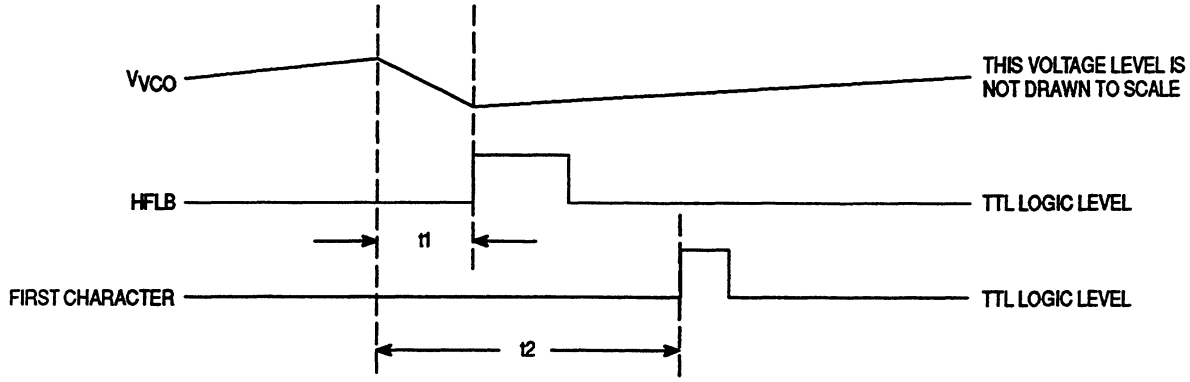
A/P MCU Technical Marketing, FEB. 21, 1992

The following figure shows the external components required for the operation of the PLL in the MC68HC05T7/T10 and MC68HC705T10. Furthermore, owing to the different technology used in fabricating the MC68HC05T7/T10 (ROM parts) and MC68HC705T10 (EPROM/OTP parts), the characteristics of the on-chip PLL circuitry differ slightly. Therefore, users have to adjust RF3 and R in order to obtain a stable On Screen Display when changing between these two kinds of devices.



R2	510 Ω
RF4	10 kΩ
C1	6800 pF
C2	0.01 µF

BLOCK DIAGRAM OF THE PLLIN
MC05T7/T10 OR HC705T10



t1 is the phase comparator pulse width which is proportional to $\left(\frac{V_{DD} - V_{VCO}}{RF3} \Big/ \frac{V_{VCO}}{RF1 + RF2}\right)$ where V_{VCO} is the DC level at the VCO pin.

- t2 is the timing controlled by the Horizontal Position Register.

- Effective horizontal position is (t2–t1) which is primarily controlled by the Horizontal Position Register, and can be fine-tuned by adjusting RF3.

The following is the procedure used in determining R1 and RF3:

1. With a fixed HFLB frequency, choose a R1 value so that $V_{VCO} \approx 1.2\ V_{DD}$.
2. Then choose a RF3 value so that the center value of RF3 makes t1 ≈ 2 µs. t1 can be measured from the VCO pin.
3. Program the horizontal position register so that the display is roughly at screen center.
4. Adjust RF3 so that the display is precisely at screen center.
5. Horizontal position shift caused by (RF1 + RF2) due to process variation should be able to adjust through RF3. Since the tolerance of RF1 and RF2 is around 10% only, its effect on the horizontal delay of the OSD screen is at most 0.2 µs, this amount to less than 1% of the screen width. Therefore, RF3 can also be fixed in applications where this shift in horizontal position is acceptable.

The following shows the typical values for R1 in an NTSC system

	ROM PART	EPROM PART
R1	12–15k	4k

RF3 is a 470k variable resistor, and should be adjusted when changing between ROM part and EPROM/OTP part.

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

MC68HC05T10
MC68HC05T7
MC68HC705T10

HCMOS MICROCONTROLLER UNIT

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION 1
INTRODUCTION

1.1 GENERAL

The MC68HC05T10 HCMOS microcomputer is a member of the MC68HC05 family, specially suitable for TV or VCR controller. This 8-bit microcomputer unit (MCU) contains an on-chip oscillator, CPU, RAM, ROM, I/O, Timer system, Pulse Length Modulated Outputs, On Screen Display (OSD), ADC and M-Bus Interface.

For the automatic TV system detection feature of OSD, this MCU is to be operated at not less than 3.6MHz.

The MC68HC05T7 MCU device is a cost reduced version of the MC68HC05T10. It is 100% compatible with the M68HC05T10 with the exception that it has only 8k bytes of ROM instead of 12k bytes. Information throughout this document pertaining to the MC68HC05T10 MCU is also applicable to the MC68HC05T7 MCU. Difference information applicable to the MC68HC05T7 MCU is provided in Appendix A of this document.

The MC68HC705T10 MCU device is the EPROM version of the MC68HC05T10. It has 12k bytes of user EPROM instead of 12k bytes of user ROM and bootstrap programming firmware instead of self-check firmware. Information throughout this document pertaining to the MC68HC05T10 MCU is also applicable to the MC68HC705T10 MCU. Difference information applicable to the MC68HC705T10 MCU is provided in Appendix B of this document.

There is no corresponding EPROM device for the MC68HC05T7, however, the MC68HC705T10 can be used to fully emulate the MC68HC05T7 by simpily ignoring the extra 4k bytes of EPROM in the MC68HC705T10.

Table 1-1 Summary of TVMCU Parts

Part Number	Pin Number	User Program Memory
MC68HC05T7	56	7904 bytes ROM
MC68HC05T10	56	12000 bytes ROM
MC68HC705T10	56	12032 bytes EPROM

1.2 FEATURES

The following are some of the hardware and software highlights of the MC68HC05T10.

HARDWARE FEATURES

* HCMOS Technology
* 8-bit Architecture
* CPU core as MC68HC05C4
* Power saving Wait and Standby modes
* 320 bytes of on-chip RAM
* 12000 bytes of on-chip ROM
* 20 Bidirectional I/O lines, 4 input lines (including A/D input)
* 2.1 MHz Internal Operating Frequency at 5 Volts
* 16 bit programmable timer and one 50/60 Hz 5 bits real time clock timer
* Self-Check mode
* 8 channels 6-Bit DAC Port and one channel 14-bit DAC
* 1 channel 8-Bit ADC
* On Screen Display System
    * Automatic Adjusted Multi-System display
    * Programmable character display of 10 rows by 18 columns
    * Double Height, Double Width character size independently selectable
    * 4 character colors per row selectable out of 8 colors
    * Character Black-Edge (4-sided) feature
    * Continuous horizontal/vertical lines for linear scale
    * Programmable window feature with 8 colors selectable
    * 64 character set
    * 8 x 11 character dot matrix for PAL system and 8 x 13 for NTSC system
    * Half-dot shift for hardware character rounding

* M-Bus (I^2C) Interface System (handling Multi-Master protocol)
* Keyboard interrupt
* External interrupt enable independently
* Master Reset and Power-On Reset
* POR bit allowing the user to distinguish Power-On from Master Reset
* Single 5 Volts±10% supply
* On-chip crystal oscillator
* 56-pin plastic SDIP Package

SOFTWARE FEATURES

* Similar to MC6800
* 8 x 8 Unsigned Multiply Instruction
* Efficient Use of Program Space
* Versatile Interrupt Handling
* Software Programmable External Interrupt Options
* True Bit Manipulation
* Addressing Modes with Indexed Addressing for Tables
* Efficient Instruction Set
* Memory Mapped I/O
* Upward Software Compatible with the MC146805 CMOS Family

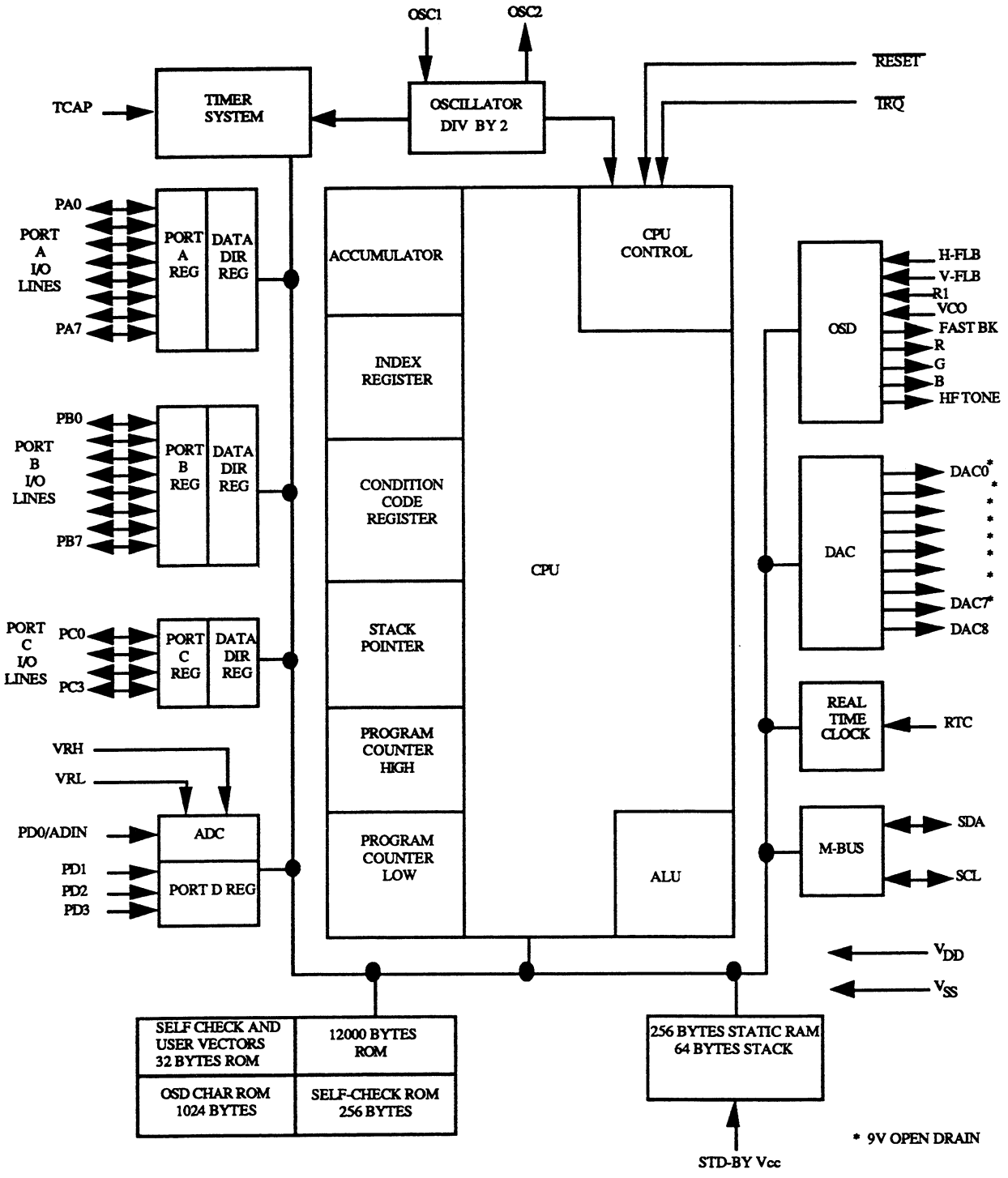


Fig 1-1 MC68HC05T10 Microcomputer Block Diagram

SECTION 2
FUNCTIONAL PIN DESCRIPTION, INPUT/OUTPUT PROGRAMMING, MEMORY, CPU REGISTER, AND SELF-CHECK

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

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 IRQ- (External Interrupt Request)

Two different choices of interrupt triggering sensitivity are software programmable by INTN Bit in MISC Register. 1) negative edge-sensitive triggering only, or 2) both negative edge-sensitive and level-sensitive triggering. It defaults to the latter case in which either type of input to the IRQ- pin will produce the interrupt. The MCU completes the current instruction before it responds to the interrupt request. When a valid edge has been sensed, it is latched internally to signify an interrupt has been requested. When the MCU completes its current instruction, the interrupt latch is tested. If the interrupt latch is set, the external interrupt enable bit (INTE) is set, and the interrupt mask bit (I-bit) in the condition code register is clear, then the MCU begins the interrupt sequence. And the I bit is then set, this masks further interrupt until present one is serviced.

2.1.3 RESET-

The external RESET- is active low which reset the MCU internal state and provide an orderly software startup procedure.

The MC68HC05T10 has two reset modes: an active low external reset pin (RESET-) and a power-on reset function.

2.1.4 OSC1, OSC2

OSC1 and OSC2 are the oscillator input pins of the MC68HC05T10. The internal clocks are derived by a divide-by-two of the external oscillator frequency (f_{OSC}).

2.1.5 PA0-PA7, PB0-PB7, PCO-PC3

These twenty I/O lines comprise port A, port B, and port C. The state of any pin is software programmable and all port A, B and C I/O lines are configured as input during power on or reset.

The PA0-PA4 lines are implemented with keyboard interrupt, it is enabled by bit 3 in MISC register. It defaults to disable. Refer to Section 3.2.7 for a detailed description of Keyboard Interrupt.

2.1.6 PD0 - PD3

The PD0-PD3 lines are fixed input lines. The state of any pin is software readable. The PD0 is also the 8-bit A/D input.

2.1.7 DAC0 - DAC7

These eight open drain output lines comprise the Digital to Analog output port. Refer to Section 7 for a detailed description of Pulse Length D/A converters. During power up, it defaults to zero.

2.1.8 DAC8

It is a 14-bit DAC output pin. Refer to Section 7 for a detailed description of Pulse Length D/A converters, with Binary Rate Multiplier. During power up, it defaults to zero.

2.1.9 TCAP

The TCAP input controls the input capture feature for the on-chip programmable 16-bit timer system. Refer to INPUT CAPTURE REGISTER in Section 4 for additional information.

2.1.10 SDA

M-Bus serial data input and output pin. Refer to Section 5 for a detailed description of M-Bus.

2.1.11 SCL

M-Bus serial clock input and output pin. Refer to Section 5 for a detailed description of M-Bus.

2.1.12 V-FLB, H-FLB

OSD input of TV vertical, horizontal flyback pulses. See Section 6 for more information on OSD.

2.1.13 VCO

Control voltage input of PLL oscillator in OSD block. See figure 2.5 for components required for proper operation.

2.1.14 R1

Connection to external timing elements of VCO in OSD block. See figure 2.5 for components required for proper operation.

2.1.15 R.G.B.

R.B.G. color outputs of OSD. See Section 6 for more information on OSD.

2.1.16 Fast-blanking

An active high video blanking output signal for fast blanking of the original TV display. See Section 6 for more information on OSD.

2.1.17 Half-tone

An active high output signal signifying window only area of OSD. See Section 6 for more information on OSD.

2.1.18 RTC

External clock input of Real Time Clock timer. See Section 4.7 for more information on RTC.

2.1.19 STBY

RAM standby power supply during CPU off power.

2.1.20 V_{RH}

V_{RH} pin is the positive reference voltage for the A/D converter.

2.1.21 V_{RL}

V_{RL} pin is the negative reference voltage for the A/D converter.

2.1.22 NC

Reserved for V_{pp} in MC68HC705T10, tie to V_{dd} for compatibility with MC68HC705T10.

2.2 INPUT/OUTPUT PROGRAMMING

2.2.1 Parallel Ports

Ports A#, B and C may be programmed as input or 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, or port C pin is configured as an output if its corresponding DDR bit is set to a logic one. A pin is configured as an input if its corresponding DDR bit is cleared to a logic zero. At power-on or reset, all DDRs are cleared, which configure all port A, B, and D pins as inputs. The data direction registers are capable of being written to or read by the processor. Refer to Figure 2-1 and 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 3 of $1C) is set and port A is configured as an input, there will be 250 Kohm pull up resistors associated with pins 0-4 of port A.

Table 2-1 I/O Pin Functions

R/$\overline{W}$*	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.

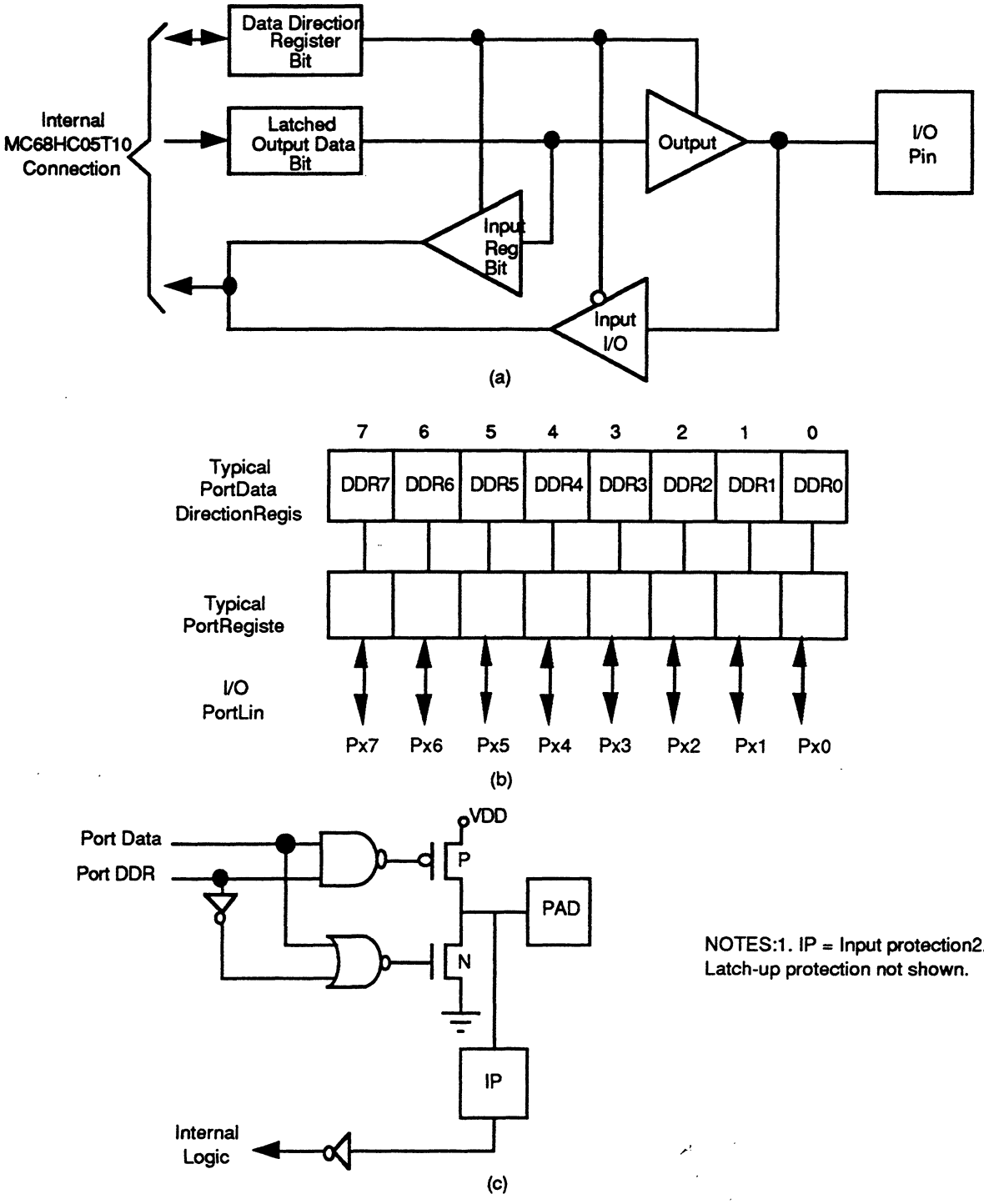
*R/$\overline{W}$ is an internal signal.

2.2.2 Fixed Port

Port D (PD0-PD3) is a 4-bit fixed input that continually monitors the external pins.

PD0 also reads the analogue input to the A/D converter, when it is 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).



NOTES:1. IP = Input protection2.
Latch-up protection not shown.

Figure 2-1 Parallel Port I/O Circuitry

2.3 MEMORY

As shown in Figure 2-2, the first 256 bytes of memory (page zero) include : 64 bytes of I/O registers such as data ports, the DDRs, timer, M-Bus, on screen display, and 192 bytes of RAM inclusive of 64 byte stack. The balanced 128 bytes RAM of total 256 bytes user RAM reside in 1st page. The user ROM is 12000 bytes. The 16 highest bytes in user vector contain the user defined reset and the interrupt vectors.

2.4 I/O AND CONTROL REGISTERS BIT ASSIGNMENT

The bit assignment for those first 62 bytes I/O and control registers are shown in Table 2-2.

0 : MCU reads as zero
R : Read only
W : Write only
RW : Read and Write

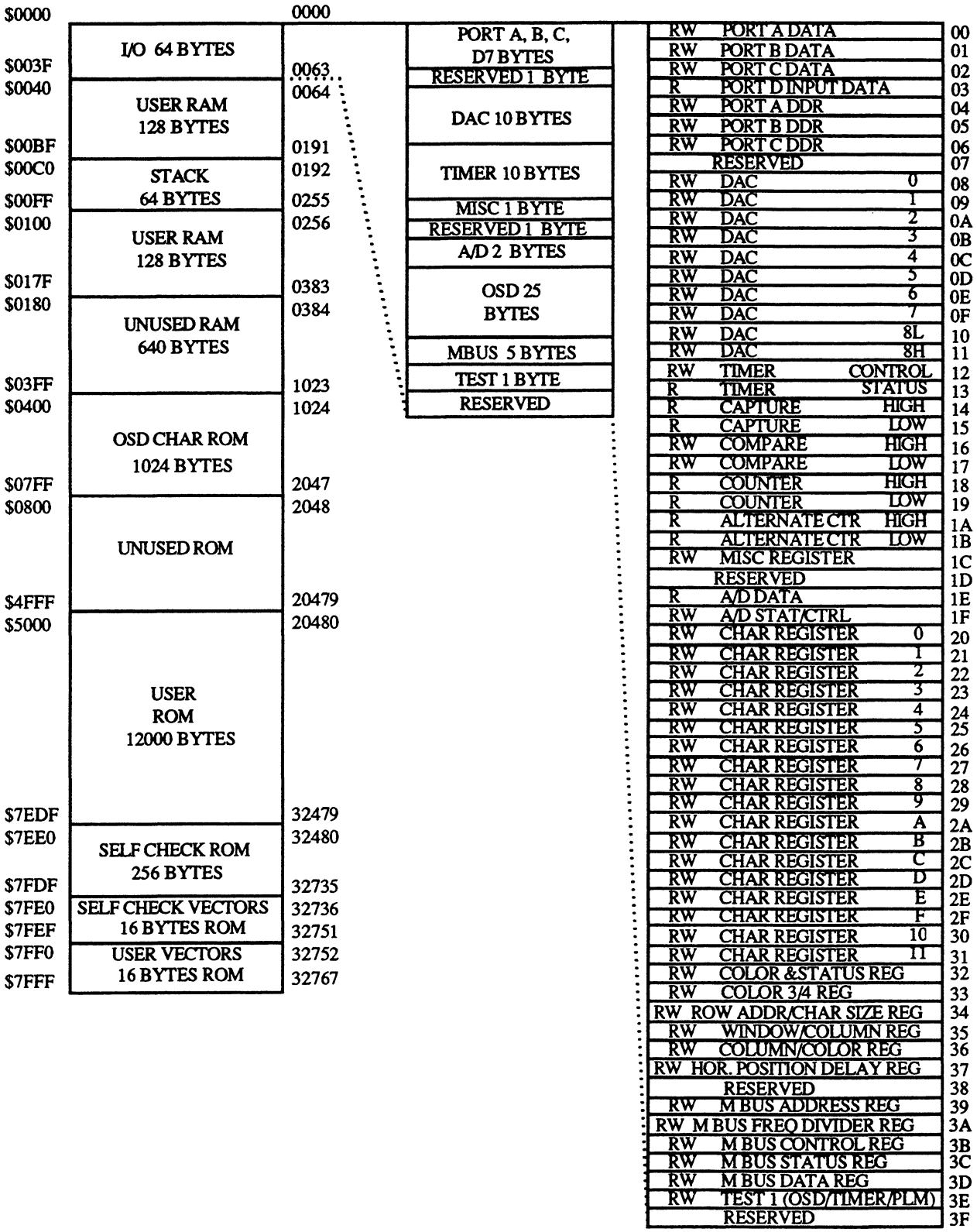


Figure 2-2 Memory Map MC68HC05T10

Table 2-2 I/O and Control Bit Assignment Table

ADDRESS 0000 TO 0063				DATA							
				7	6	5	4	3	2	1	0
00	RW	PORT A DATA		PDA7	PDA6	PDA5	PDA4	PDA3	PDA2	PDA1	PDA0
01	RW	PORT B DATA		PDB7	PDB6	PDB5	PDB4	PDB3	PDB2	PDB1	PDB0
02	RW	PORT C DATA		PDC7	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0
03	R	PORT D INPUT DATA		0	0	0	0	PDD3	PDD2	PDD1	PDD0
04	RW	PORT A DDR		PDRA7	PDRA6	PDRA5	PDRA4	PDRA3	PDRA2	PDRA1	PDRA0
05	RW	PORT B DDR		PDRB7	PDRB6	PDRB5	PDRB4	PDRB3	PDRB2	PDRB1	PDRB0
06	RW	PORT C DDR		PDRC7	PDRC6	PDRC5	PDRC4	PDRC3	PDRC2	PDRC1	PDRC0
07	RESERVE										
08	RW	DAC	0			0DA5	0DA4	0DA3	0DA2	0DA1	0DA0
09	RW	DAC	1			1DA5	1DA4	1DA3	1DA2	1DA1	1DA0
0A	RW	DAC	2			2DA5	2DA4	2DA3	2DA2	2DA1	2DA0
0B	RW	DAC	3			3DA5	3DA4	3DA3	3DA2	3DA1	3DA0
0C	RW	DAC	4			4DA5	4DA4	4DA3	4DA2	4DA1	4DA0
0D	RW	DAC	5			5DA5	5DA4	5DA3	5DA2	5DA1	5DA0
0E	RW	DAC	6			6DA5	6DA4	6DA3	6DA2	6DA1	6DA0
0F	RW	DAC	7			7DA5	7DA4	7DA3	7DA2	7DA1	7DA0
10	RW	DAC	8 L			8DA5	8DA4	8DA3	8DA2	8DA1	8DA0
11	RW	DAC	8 H	8DA13	8DA12	8DA11	8DA10	8DA9	8DA8	8DA7	8DA6
12	RW	TIMER	CONTROL	ICIE	OCIE	TOIE	TCAPS			IEDG	-
13	R	TIMER	STATUS	ICF	OCF	TOF	0	0	0	0	0
14	R	CAPTURE	HIGH	CTH7	CTH6	CTH5	CTH4	CTH3	CTH2	CTH1	CTH0
15	R	CAPTURE	LOW	CTL7	CTL6	CTL5	CTL4	CTL3	CTL2	CTL1	CTL0
16	RW	COMPARE	HIGH	CMH7	CMH6	CMH5	CMH4	CMH3	CMH2	CMH1	CMH0
17	RW	COMPARE	LOW	CML7	CML6	CML5	CML4	CML3	CML2	CML1	CML0
18	R	COUNTER	HIGH	CNH7	CNH6	CNH5	CNH4	CNH3	CNH2	CNH1	CNH0
19	R	COUNTER	LOW	CNL7	CNL6	CNL5	CNL4	CNL3	CNL2	CNL1	CNL0
1A	R	ALTERNATE CTR	HIGH	ACH7	ACH6	ACH5	ACH4	ACH3	ACH2	ACH1	ACH0
1B	R	ALTERNATE CTR	LOW	ACL7	ACL6	ACL5	ACL4	ACL3	ACL2	ACL1	ACL0
1C	RW	MISC REGISTER			RTCS	RTCE	KEYS	KEYE	POR	INTN	INTE
1D		RESERVE									
1E	R	A/D DATA		ADDR7	ADDR6	ADDR5	ADDR4	ADD3	ADD2	ADD1	ADD0
1F	RW	A/D STAT/CTRL								COCO	ADON
20	RW	CHAR REGISTER	0	0CS2	0CS1	0CHD5	0CHD4	0CHD3	0CHD2	0CHD1	0CHD0
21	RW	CHAR REGISTER	1	1CS2	1CS1	1CHD5	1CHD4	1CHD3	1CHD2	1CHD1	1CHD0
22	RW	CHAR REGISTER	2	2CS2	2CS1	2CHD5	2CHD4	2CHD3	2CHD2	2CHD1	2CHD0
23	RW	CHAR REGISTER	3	3CS2	3CS1	3CHD5	3CHD4	3CHD3	3CHD2	3CHD1	3CHD0
24	RW	CHAR REGISTER	4	4CS2	4CS1	4CHD5	4CHD4	4CHD3	4CHD2	4CHD1	4CHD0
25	RW	CHAR REGISTER	5	5CS2	5CS1	5CHD5	5CHD4	5CHD3	5CHD2	5CHD1	5CHD0
26	RW	CHAR REGISTER	6	6CS2	6CS1	6CHD5	6CHD4	6CHD3	6CHD2	6CHD1	6CHD0
27	RW	CHAR REGISTER	7	7CS2	7CS1	7CHD5	7CHD4	7CHD3	7CHD2	7CHD1	7CHD0
28	RW	CHAR REGISTER	8	8CS2	8CS1	8CHD5	8CHD4	8CHD3	8CHD2	8CHD1	8CHD0
29	RW	CHAR REGISTER	9	9CS2	9CS1	9CHD5	9CHD4	9CHD3	9CHD2	99CHD1	9CHD0
2A	RW	CHAR REGISTER	A	ACS2	ACS1	ACHD5	ACHD4	ACHD3	ACHD2	ACHD1	ACHD0
2B	RW	CHAR REGISTER	B	BCS2	BCS1	BCHD5	BCHD4	BCHD3	BCHD2	BCHD1	BCHD0
2C	RW	CHAR REGISTER	C	CCS2	CCS1	CCHD5	CCHD4	CCHD3	CCHD2	CCHD1	CCHD0
2D	RW	CHAR REGISTER	D	DCS2	DCS1	DCHD5	DCHD4	DCHD3	DCHD2	DCHD1	DCHD0
2E	RW	CHAR REGISTER	E	ECS2	ECS1	ECHD5	ECHD4	ECHD3	ECHD2	ECHD1	ECHD0
2F	RW	CHAR REGISTER	F	FCS2	FCS1	FCHD5	FCHD4	FCHD3	FCHD2	FCHD1	FCHD0
30	RW	CHAR REGISTER	10	10CS2	10CS1	10CHD5	10CHD4	10CHD3	10CHD2	10CHD1	10CHD0
31	RW	CHAR REGISTER	11	11CS2	11CS1	11CHD5	11CHD4	11CHD3	11CHD2	11CHD1	11CHD0
32	RW	COLOR & STATUS REG		BEEN/1	/SFG3	R2/1FL	G2/VERT	B2/HOR	R1/MODE	G1/SFG2	B1/SFG1
33	RW	COLOR 3/4 REG		H31D4	H15D4	R4	G4	B4	R3	G3	B3
34	RW	ROW ADDR/CHAR SIZE REG		CHWS	CHHS	RGBINV	OIEN	RWA3	RWA2	RWA1	RWA0
35	RW	WINDOW/COLUMN REG		WINE	OSDE	PLLEN	BCS4	BCS3	BCS2	BCS1	BCS0
36	RW	COLUMN/COLOR REG		R	G	B	BCSP4	BCSP3	BCSP2	BCSP1	BCSP0
37	RW	HOR. POSITION DELAY REG		H31D3	H31D2	H31D1	H31D0	H15D3	H15D2	H15D1	H15D0
38		RESERVE									
39	RW	M BUS ADDRESS REG		CADR7	CADR6	CADR5	CADR4	CADR3	CADR2	CADR1	CADR0
3A	RW	M BUS FREQ DIVIDER REG					MBC4	MBC3	MBC2	MBC1	MBC0
3B	RW	M BUS CONTROL REG		MEN	MIEN	MSTA	MTX	TXAK			
3C	RW	M BUS STATUS REG		MCF	MASS	MBB	MAL		SRW	MIF	RXAK
3D	RW	M BUS DATA REG		TRXD7	TRXD6	TRXD5	TRXD4	TRXD3	TRXD2	TRXD1	TRXD0

2.5 CPU REGISTERS

The MC68HC05T10 CPU contains five registers, as shown in the programming model of Figure 2-3. The interrupt stacking order is shown in Figure 2-4.

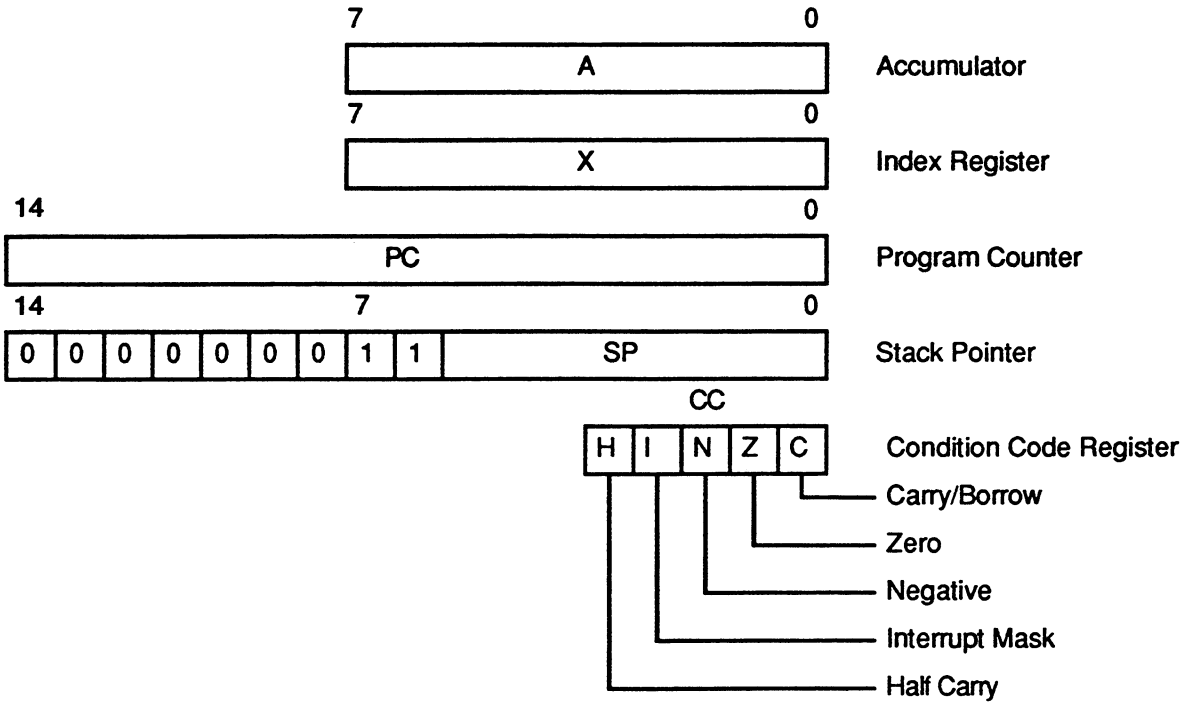
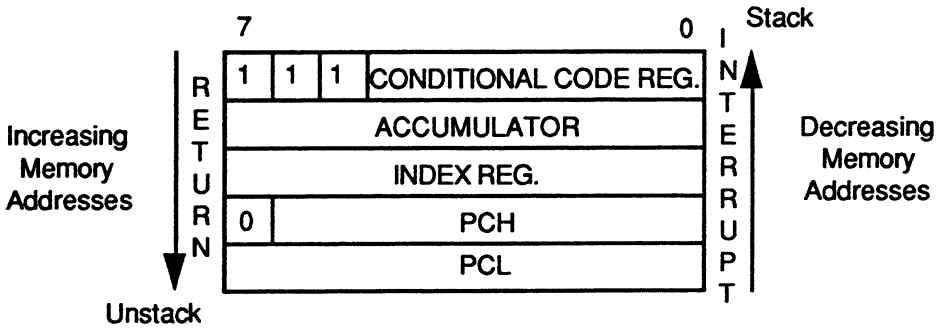


Figure 2-3 Programming Model



NOTE : Since the Stack Pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc. Pulling from the stack is in reverse order.

Figure 2-4 Stacking Order

2.5.1 Accumulator (A)

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

2.5.2 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.

2.5.3 Program Counter (PC)

The Program counter is 15-bit register that contains the address of next instruction to be executed by the processor.

2.5.4 Stack Pointer (SP)

The stack pointer is a 15-bit register containing the address of the next free locations on the push-down/pop-up stack. When accessing memory, the nine most significant bits are permanently configured to 000000011. 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.

2.5.5 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.

2.5.5.1 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.

2.5.5.2 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** for more information).

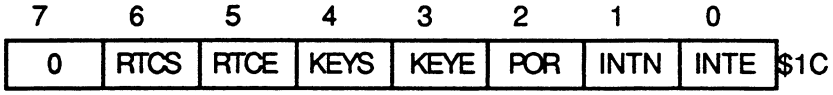
2.5.5.3 NEGATIVE (N). When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation is negative (bit 7 in the result is a logic one).

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

2.5.5.5 CARRY/BORROW (C). 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.

2.6 MISCELLANEOUS REGISTER

A description of the miscellaneous register is given below:



B6, RTCS — Real Time Clock input source selection
RTCS = 1 60 Hz input, select a set value of 30
RTCS = 0 50 Hz input, select a set value of 25

B5, RTCE — Real Time Clock interrupt enable
If this bit is set, the real time clock interrupt is enabled. Then, an interrupt will be generated whenever the internal clock counter value equals to the set value (2 Hz). Power on default to 0

B4, KEYS — Keyboard interrupt status flag. This flag is set when there is a pending keyboard interrupt. It should be cleared by software after a certain delay on servicing this interrupt so as to eliminate the false triggering due to the debounce of the key.

B3, KEYE — Keyboard interrupt enable bit. If this bit is set, the keyboard interrupt is enabled, and PA0 - PA4 are configured as input lines, regardless of the state of DDR. An internal pull up resistor of 250 Kohm is connected to each of these lines. Whenever one of PA0-PA4 sense a falling edge, an interrupt is generated. If this bit is cleared, keyboard PA0 - PA4 interrupt is disabled. Power on default to 0.

B2, POR — This bit is the Power-On Reset bit which is set each time the device is powered on, and is not affected by external reset. It allows the user to make a software distinction between a power-on and an external reset. Software can clear this bit by writing it to zero.

B1, INTN — When this bit is set, IRQ- pin responses to negative edge-sensitivity triggering only. If this bit is cleared, IRQ- responses to both negative edge-sensitivity and level-sensitivity. It is writable only while I bit is set and is cleared by power-on or external reset.

B0, INTE Interrupt enable bit
INTE = 1, IRQ- interrupt is enabled
INTE = 0, IRQ- interrupt is disabled. Power on default to 1.

2.7 SELF-CHECK

The self-check capability of the MC68HC05T10 MCU provides an internal check to determine if the device is functional. Self-check is performed using the circuit shown in the schematic diagram of Figure 2-5. As shown in the diagram, port A pins PA0-PA3 are monitored (light emitting diodes are shown but other devices could be used) for the self-check results.

To enter the self-check mode, apply 9V to the IRQ- pin (through a 4.7K resistor) and 5V to the TCAP pin (through a 10K resistor); then depress and release the reset switch. (See Table 2-3). The hold time on the IRQ- pin after the external Reset pin brought high is 2 clock cycles.

Table 2-3 Modes of Operation Selection

RESET- pin	IRQ- pin	TCAP pin	Mode
	VSS to VDD	VSS to VDD	Normal
	+9.0 Volts	VDD	Self Check
VSS	VSS to VDD	VSS to VDD	Reset Condition

After reset, the following tests are performed automatically:

I/O	- Test ports A, B, C and D
RAM	- Counter test for each RAM byte
ROM	- Exclusive OR with odd 1's parity result
TIMER	- Tracks counter register and checks OCF flag
M-BUS	- Transmit one byte of data ($55)
INTERRUPT	- Test TIMER, Keyboard, M-BUS, and external interrupts

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. Remark that SELF CHECK gets its own set of interrupt vectors apart from that of user mode.

Self-check results (using the LEDs as monitors) are shown in Table 2-4. The following subroutines are available to user programs and do not require any external hardware.

2.7.1 Timer Test Subroutine

This subroutine returns with the Z bit cleared if any error is detected; otherwise, the Z bit is set.

This subroutine is called at location $7EEE. the output compare register is first set to the current timer state. Because the timer is free running and has only a divide-by-four prescaler, each timer count cannot be tested. The test reads the timer once every 10 counts (40 cycles) and checks for correct counting. The test tracks the counter until the timer wraps around, triggering the output compare flag in the timer status register. RAM locations $0040 and $0041 are overwritten. Upon return to the user's program, X=40. If the test passed, A=0.

2.7.2 ROM Checksum Subroutine

This subroutine returns with the Z bit cleared if any error is detected; otherwise, the Z bit is set.

This subroutine is called at location $7F8C with RAM location $0043 equal to $01 and A=0. A short routine is set up and executed in RAM to compute a checksum of the entire ROM pattern. Upon return to the user's program, X=0. If the test passed, A=0. RAM locations $0040 through $0043 are overwritten.

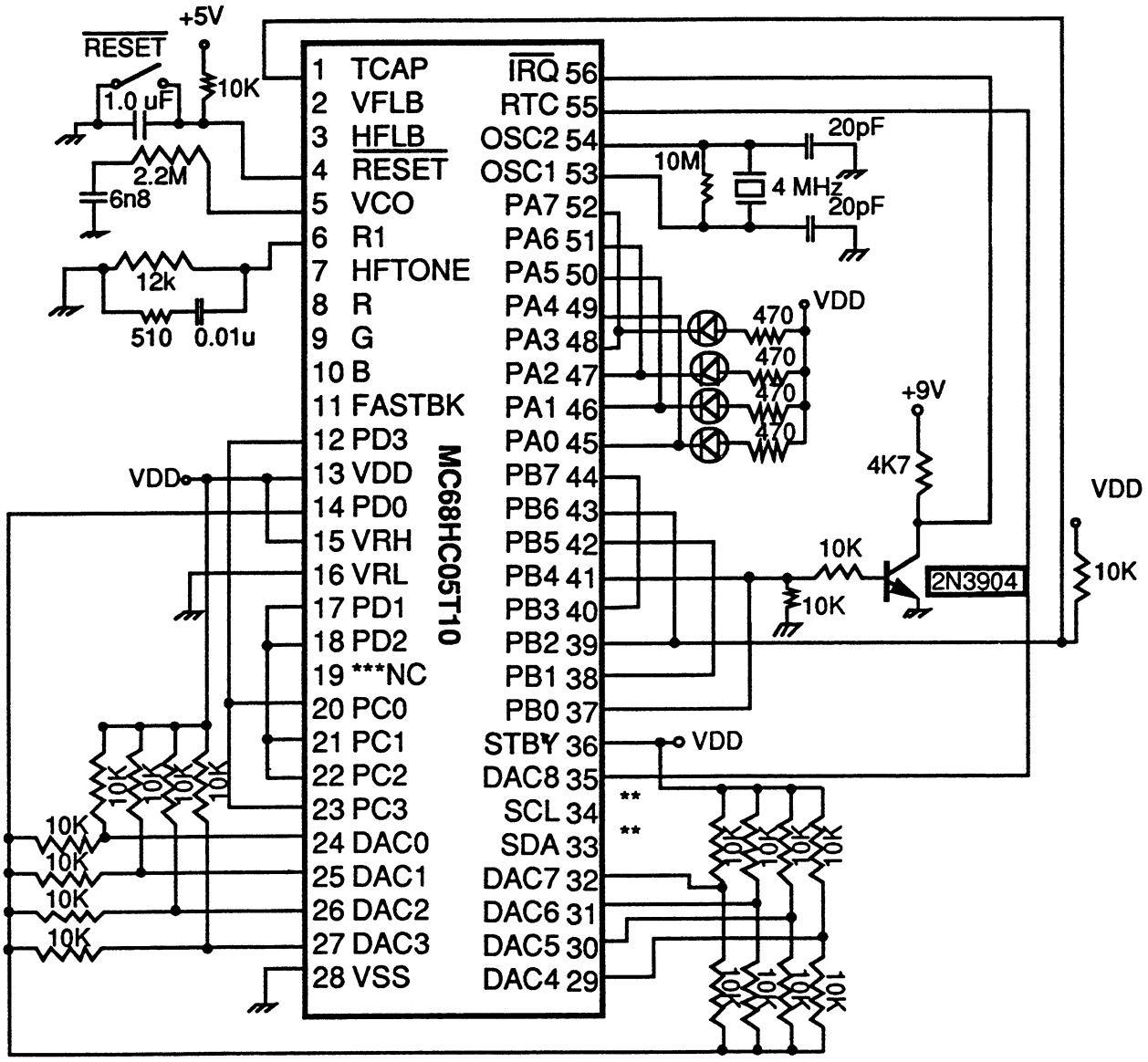
2.7.3 M-Bus Transmission Subroutine

This subroutine is called at location $7F19 to transmit one byte of data ($55), or called at location $7F1A to transmit the content of of Index Register. It first enables the M-Bus interface and then selects the master transmitter mode.

Table 2-4 Self-Check Results

PA3	PA2	PA1	PA0	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 ROM
1	1	0	1	Bad D/A or A/D
1	1	1	0	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.



NOTE ** : To be connected to a M-Bus slave receiver for M-bus testing
*** : To be connected to Vcc in other applications for compatibility with MC68HC705T10

Figure 2-5 Self-Check Circuit Schematic Diagram

SECTION 3
RESETS, INTERRUPTS AND LOW POWER MODES

3.1 RESETS

The MC68HC05T10 has 2 reset modes: an active low external reset pin (RESET-) and a power-on reset function (POR). Either of these two resets will cause the program to go to its starting address, specified by the contents of memory locations $7FFE and $7FFF, and cause the interrupt mask of the condition code register also to be set.

3.1.1 External Reset- (RESET- Pin)

The active low 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 time t_{RL}. The RESET- pin contains an internal Schmitt trigger as part of its input to improve noise immunity. When the reset pin goes high, the MCU will resume operation on the following cycle.

NOTE

The RESET- pin is pulled low during power-on reset in order to reset other circuits connected to RESET- pin.

3.1.2 Power on Reset (POR)

The power-on reset occurs when a positive transition is detected on V_{DD}. It 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.

NOTE

The user must ensure that the MCU remains in the reset state until the oscillator is stabilized. If there is doubt, a sufficiently large capacitor should be connected to the RESET- input .

Table 3-1 shows the actions of the two resets on internal circuits, but not necessarily in the order that they occur.

7	6	5	4	3	2	1	0	
0	RTCS	RTCE	KEYS	KEYE	POR	INTN	INTE	$1C

B2, POR — This bit is the Power-On Reset bit which is set each time the device is powered on, and is not affected by external reset. It allows the user to make a software distinction between a power-on and an external reset. Software can clear this bit by writing it to zero.

Table 3-1 Reset Action on Internal Circuit

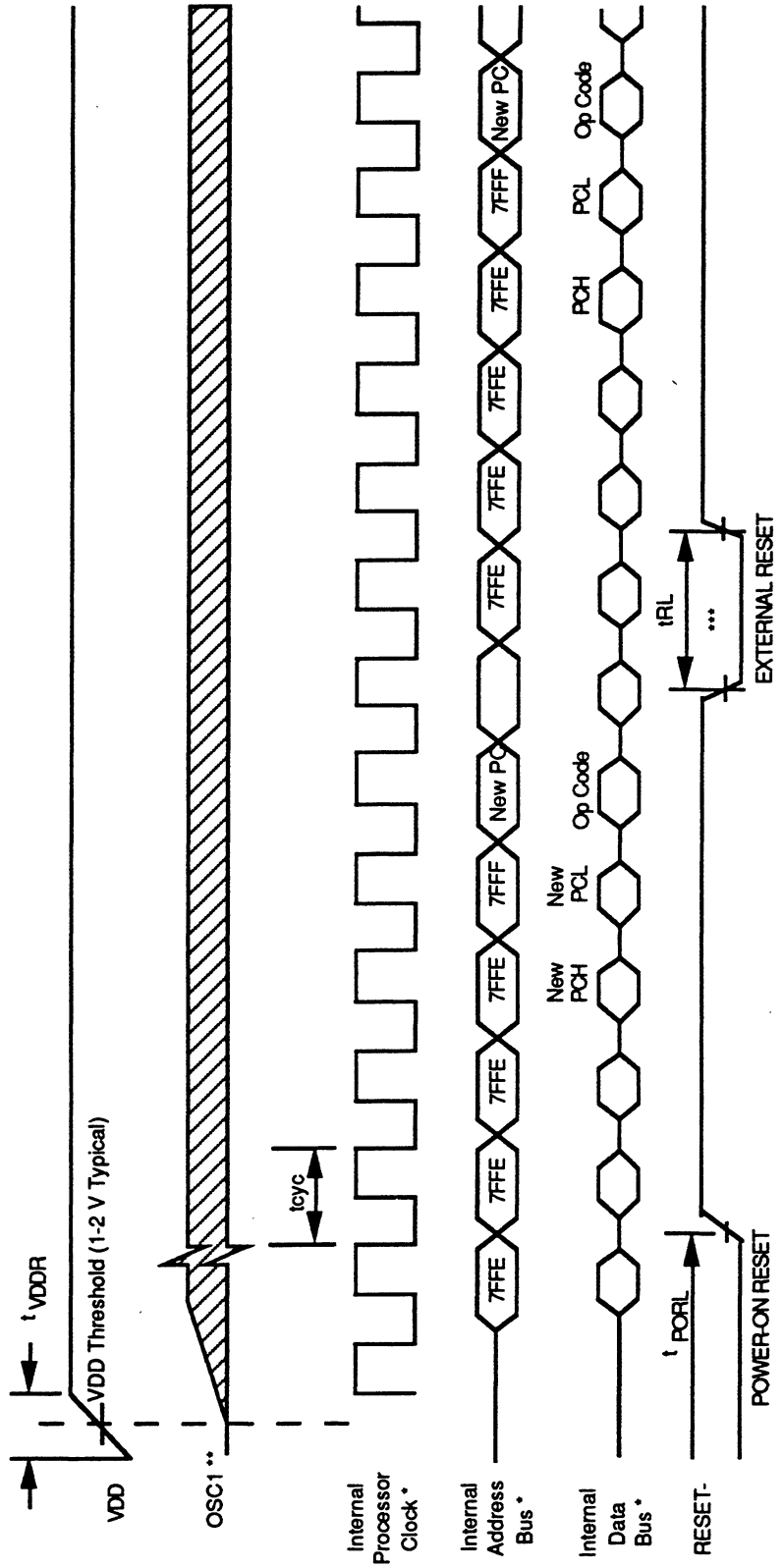
Condition	RESET Pin	Power-On Reset	WAIT MODE
Timer Prescaler reset to zero state	X	X	-
Timer counter configures to $FFFC	X	X	-
All timer interrupt enable bits cleared to disable timer interrupts	X	X	-
All data direction registers cleared to zero (input)	X	X	-
I bit in condition code register -> 1	X	X	-
I bit in condition code register -> 0	-	-	X
Configure stack pointer to $00FF	X	X	-
Force internal address bus to restart vector ($7FFE-$7FFF)	X	X	-
Interrupt enable bit (INTE) -> 1	X	X	-
OSD interrupt enable bit (OIEN) -> 0	X	X	-
Keyboard interrupt enable bit (KEYE) -> 0	X	X	-
M-Bus interrupt enable bit (MIEN) -> 0	X	X	-
M-Bus start/stop detection disable	-	X	-
OSD function disabled	X	X	-
OSD output enable bit (OSDE) -> 0	X	X	-
RTC interrupt enable bit (RTCE) -> 0	X	X	-
A/D enable bit (ADON) -> 0	X	X	-
DAC0-8 data bits -> 0	X	X	-
Power-On Reset bit (POR, Miscellaneous Register) -> 1	-	X	-
Clear external interrupt latch	X	X	-
Clear WAIT latch	X	X	-
Disable oscillator	-	X	-
Reset PHI2 circuit	-	X	-
Internal PHI1 -> 0	-	X	X
Internal PHI2 -> 0	-	X	X
Timer Clock -> 0	-	X	-

3.2 INTERRUPTS

Systems often require that normal processing be interrupted so that some external event may be serviced.

The MC68HC05T10 may be interrupted by one of six different methods: either one of six maskable hardware interrupts (IRQ, RTC, KEYBOARD, OSD, TIMER or M-BUS) and one non-maskable software interrupt (SWI). For OSD, TIMER and M-BUS, interrupt flags are located in their status registers respectively, whereas their corresponding enable bits are located in associated control register.

If the enable bit is in a logic zero, it blocks the interrupt from occurring but does not inhibit the flag from being set. Power-on or external reset clears all enable bits (but sets INTE-bit), to preclude interrupts during the reset procedure.



* Internal bus information not available externally.

** OSC1 line is not meant to represent frequency. It is only used to represent time.

*** The next rising edge of the processor clock following the rising edge of RESET- initiates the reset sequence.

Figure 3-1 Power-On Reset and Master Reset

The general sequence for clearing an interrupt is a software sequence of first accessing the status register while the interrupt flag is set, followed by a read or write of an associated register. When any of these interrupts occur, and if the enable bit is a logic one, normal processing is suspended at the end of the current instruction execution.

Interrupt causes the processor registers to be saved on the stack and the interrupt mask (I-BIT) is set to prevent additional interrupts. The appropriate interrupt vector then points to the starting address of the interrupt service routine.

Upon completion of the interrupt service routine, the RTI instruction (which is normally part of the service routine) causes the register contents to be restored from the stack followed by a return to normal processing.

3.2.1 Interrupt Priorities (Vectoring)

Due to the increased number of potential interrupt sources, it is desirable in real time applications to subdivide as finely as possible the vectoring associated with different sources to different addresses.

There are three different timer interrupt flags that will cause a timer interrupt whenever they are set and enabled. These three interrupt flags are found in the three most significant bits of the timer status register (location at $13) and all three vector to the same interrupt service routine ($7FF8 - $7FF9). **Table 3-2 Vector Address for Interrupts and Reset** gives the vectors associated with each interrupt as well as reset. Highest priority is RESET, followed by SWI, External Interrupt, TIMER, OSD, Keyboard, M-BUS, and Real Time Clock.

Table 3-2 Vector Address for Interrupts and Reset

Register	Flag Name	Interrupts	CPU Interrupt	Vector Address
X	X	Reset	RESET	$7FFE-$7FFF
X	X	Software	SWI	$7FFC-$7FFD
X	X	External Interrupt	IRQ	$7FFA-$7FFB
Timer Status	ICF	Input Capture	TIMER	$7FF8-$7FF9
	OCF	Output Compare		
	TOF	Timer Overflow		
Color & Status	IFL	OSD registers re-new	OSD	$7FF6-$7FF7
Miscellaneous	KEYS	Keyboard	KEYBOARD	$7FF4-$7FF5
M-Bus Status	MAL	Arbitraton Lost	M-BUS	$7FF2-$7FF3
	MAAS	Master Addressed as Slave		
M-Bus Control	MTX & MIF	Transmit Complete		
	MTX & MIF	Receive Complete		
X	X	Real Time Clock	RTC	$7FF0-$7FF1

3.2.2 Hardware Controlled Interrupt Sequence

The following functions (RESET and WAIT) are not in the strictest sense interrupts; however, they are acted upon in a similar manner. Flowcharts for hardware interrupts are shown in Figure 3-2, and for WAIT in Figure 3-3. A discussion is provided below.

(a) A low input on the RESET- input pin causes the program to vector to its starting address which is specified by the contents of memory locations $7FFE and $7FFF. The I-bit in the conditions code register is also set. Much of the MCU is configured to a known state during this type of reset as previously described in RESETS paragraph 3.1.

(b) WAIT - The WAIT instruction causes all processor clocks to stop, but leaves the TIMER, OSD, RTC and M-BUS clock running. This "rest" state of the processor can be cleared by reset, RTC, keyboard, external interrupt (IRQ-), TIMER interrupt, OSD interrupt or M-BUS interrupt.

3.2.3 Software Interrupt (SWI)

The software interrupt is an executable instruction. The action of the SWI instruction is similar to the hardware interrupt. The SWI is executed regardless of the state of the interrupt mask (I-bit) in the condition code register. The interrupt service routine address is specified by the content of memory location $7FFC and $7FFD.

NOTE

SWI sets the I-bit so as not to be interrupted when servicing the SWI interrupt routine. Upon the completion of the SWI service routine, the RTI instruction (which is normally part of the service routine) causes the register content to be recovered from the stack followed by a return to normal processing. The interrupt mask bit (I-BIT) will be cleared if and only if the corresponding bit stored in the stack is zero.

3.2.4 External Interrupt

If the interrupt mask (I-bit) of the condition code register has been cleared and the interrupt enable bit is set (INTE-BIT), and the signal of the external interrupt pin (IRQ-) satisfies the condition selected by the control bits (INTN), then the external interrupt is recognized. When the interrupt is recognized, the current state of the CPU is pushed onto the stack and the I-bit is set. This masks further interrupts until the present one is serviced.

The interrupt service routine address is specified by the content of memory locations $7FFA and $7FFB. Negative edge, or negative edge and low level-sensitive trigger are set by software on INTN of the miscellaneous register (bit 1 of address $1C).

In order to avoid any conflict and spurious interrupt, it is only possible to change the external interrupt options while the I-bit is set. Any attempt to change the external interrupt options while the I-bit is clear will be unsuccessful. If an external interrupt is pending, it will automatically be cleared when selecting a different interrupt option.

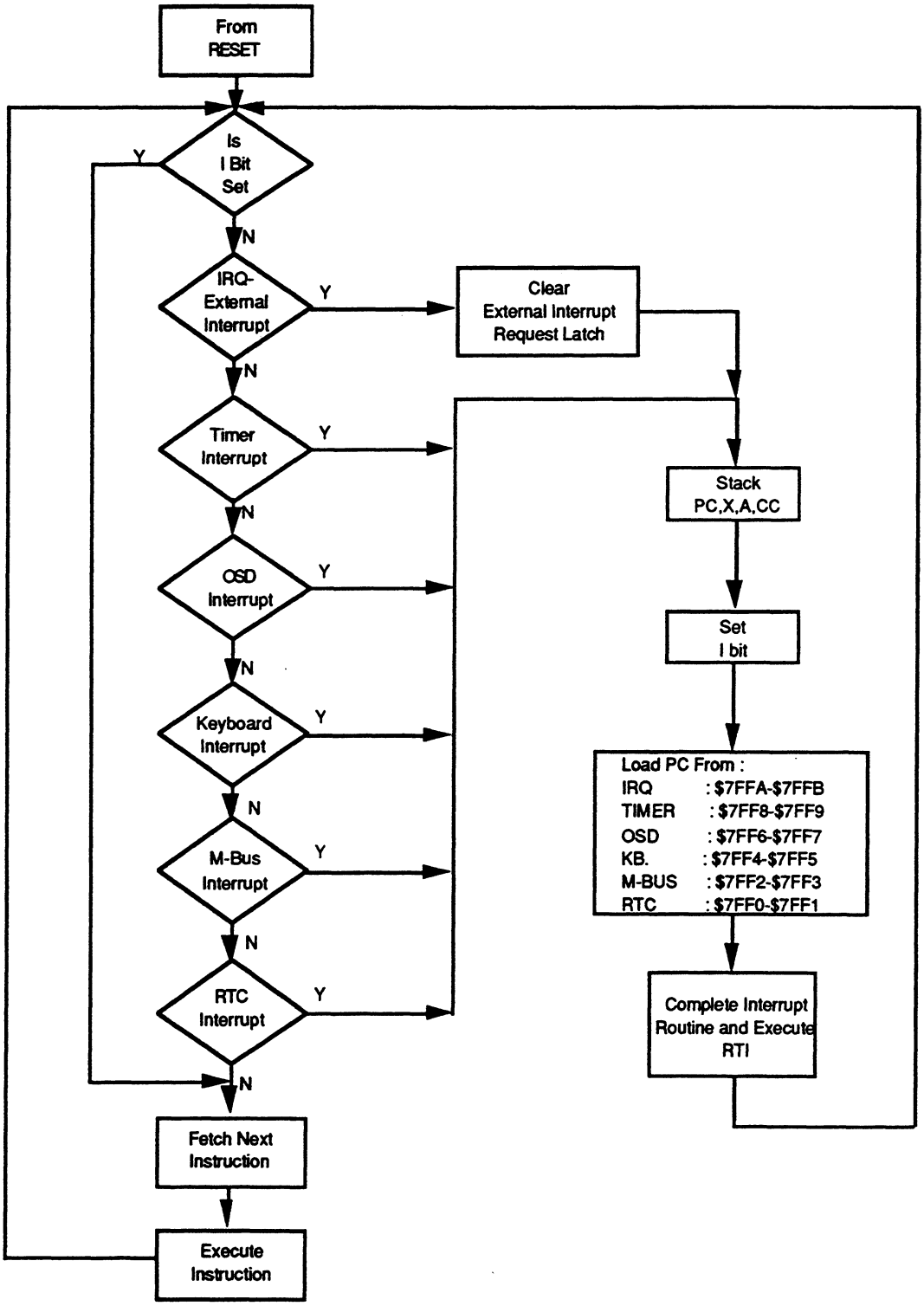


Figure 3-2 HARDWARE INTERRUPT FLOWCHART

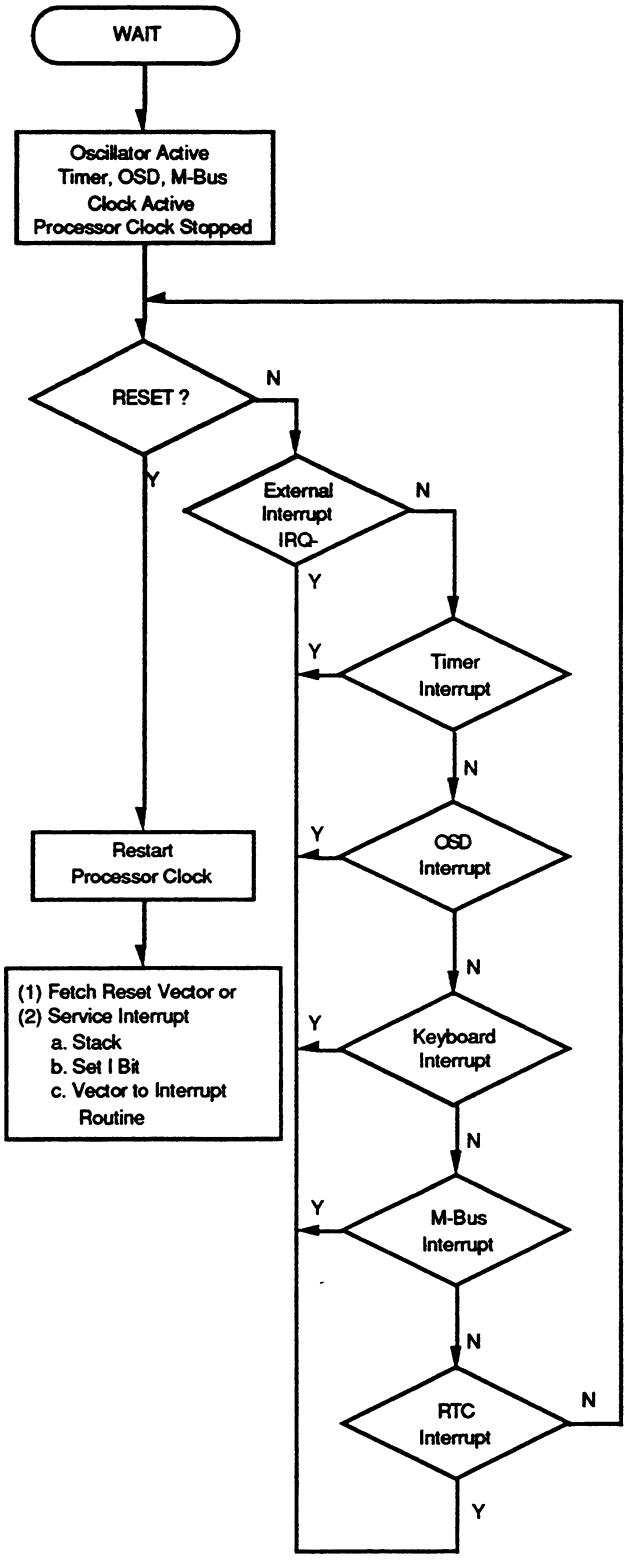
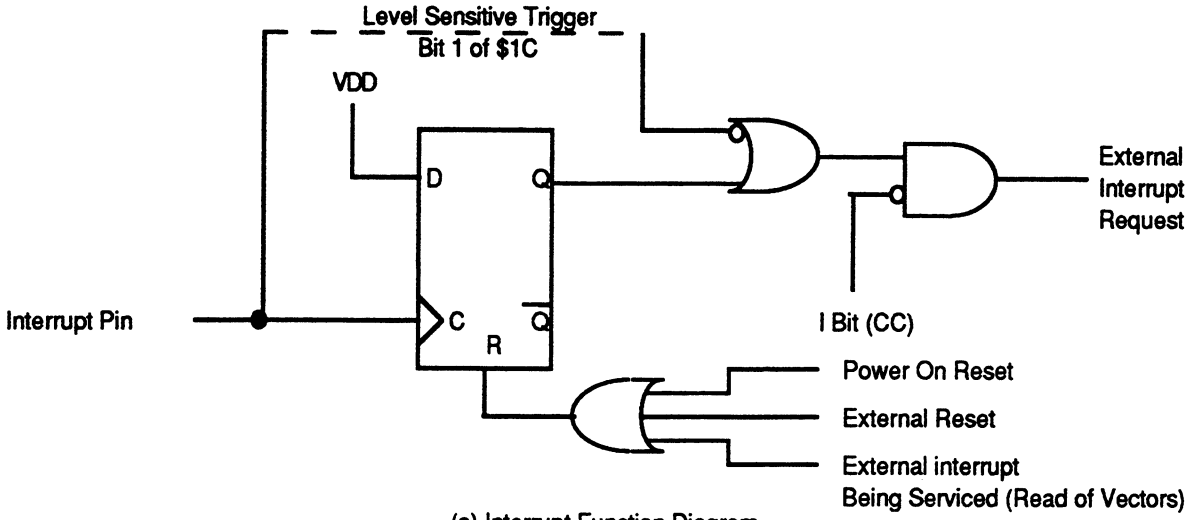
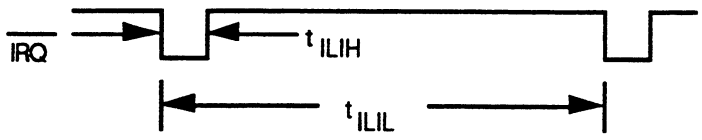


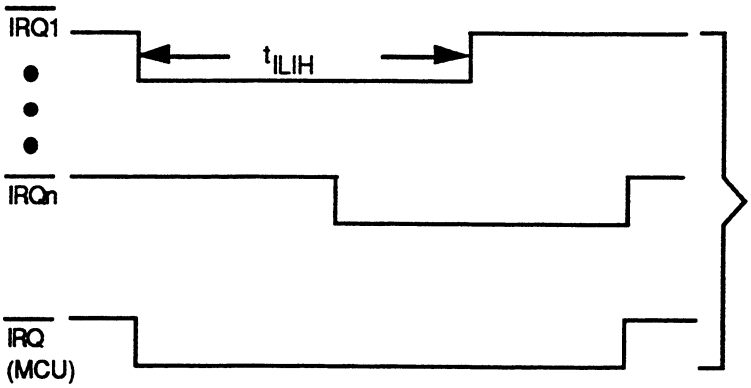
Figure 3-3 WAIT Flowcharts



(a) Interrupt Function Diagram



Edge-Sensitive Trigger Condition
The minimum pulse width (tILIH) is 125 ns (VDD = 5 V).
The period tILIL should not be less than the number of tcyc cycles it takes to execute the interrupt service routine plus 21 tcyc cycles.



Level-Sensitive Trigger Condition
If after servicing an interrupt the IRQ remains low, then the next interrupt is recognized.

Normally used with Wire-ORed connection

(b) Interrupt Mode Diagram

Figure 3-4 External Interrupt

Figure 3-4 shows both a functional and mode timing diagram for the interrupt line. The timing diagram shows two different treatments of the interrupt line (IRQ-) to the processor. The first method shows single 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). The second configuration shows several interrupt lines "wire-ORed" to form the interrupts at the processor. Thus, if after servicing one interrupt the interrupt line remains low, then the next interrupt is recognized.

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.

3.2.5 Timer Interrupt

There are three different timer interrupt flags that will cause a timer interrupt whenever they are set and enabled. These three Timer interrupt flags are found in the three most significant bits of the Timer Status Register (TSR, location $13). All three interrupts will vector to the same service routine location.

All interrupt flags have corresponding enable bits (ICIE, OCIE, and TOIE) found in the Timer Control Register (TCR, location $12). 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 also cleared. When the interrupt is recognized, 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 $7FF8 and $7FF9. Refer to **SECTION 4 PROGRAMMABLE TIMER** for additional information about the timer circuitry.

3.2.6 OSD Interrupt

As the single row registers architecture is employed in OSD, the CPU is interrupted every 1.4 ms (or 1.6 ms for PAL TV system) to update the contents of the single row registers in the worst case. The OSD interrupt flag (IFL, bit 5 of the Color & Status Register at $32) is read-only and is set once the present registers row is going to display. At that time, if the OSD interrupt enable flag (OIEN, bit 4 of the Row Addr/Char Size Register at $34) is set, an interrupt will be generated. The interrupt service routine address is specified by the contents of memory location $7FF6 and $7FF7.

On entering the OSD interrupt servicing routine, the OSD interrupt flag is cleared automatically. However, if the OSD interrupt enable flag is cleared, the software need to poll the state of OSD interrupt flag for normal interrupt servicing.

Reset disables the whole OSD block, thus preventing any undesirable display on the TV screen during and after the reset time period. Refer to **SECTION 6 ON SCREEN DISPLAY** for additional information about the OSD circuitry.

3.2.7 Keyboard Interrupt

Provided the interrupt mask bit of the condition code register is cleared, the keyboard interrupt is enabled by setting the keyboard interrupt enable flag, bit 3 of Miscellaneous Register (MISC, location $1C). This will force the five least significant I/O lines of port A (PA0-PA4) as input lines with an internal pull-up resistor of 250 Kohm respectively. Once a HIGH to LOW transition is sensed on any input lines of PA0-PA4, a keyboard interrupt is generated and the keyboard status flag, bit 4 of Miscellaneous Register (MISC, location $1C) is set. The interrupt service routine address is specified by the contents of memory location $7FF4 and $7FF5.

The keyboard interrupt status flag, bit 4 of Miscellaneous Register (MISC, location $1C) should be cleared by software in the interrupt servicing routine. To account for the debouncing effect, this flag is to be cleared only after a certain time delay.

3.2.8 M-Bus Interrupt

M-Bus interrupt is enabled when the M-Bus Interrupt Enable flag, bit 6 of M-Bus Control Register (MCR, location $3B) is set, provided the interrupt mask bit of the condition code register is cleared.

There are four different causes of M-Bus interrupt. First cause is arbitration lost which is signified by the Arbitration Lost flag, bit 4 of M-Bus Status Register (MSR, location $3C). The second cause is being addressed as slave which is indicated by the Master Address As Slave flag, bit 6 of M-Bus Status Register (MSR, location $3C).

The third and forth causes are complete transmission or reception of one byte of data. It depends on the original mode of the M-Bus interface which is determined by the Transmit/Receive flag, bit 4 of M-Bus Control Register (MCR, location $3B).

All these four possible interrupts are enabled by the M-Bus Interrupt Enable flag, bit 6 of M-Bus Control Register (MCR, location $3B). On entering the interrupt servicing routine, the M-Bus interrupt flag, bit 1 of M-Bus Status Register (MSR, location $3C) must be cleared by software. The interrupt service routine address is specified by the contents of memory location $7FF2 and $7FF3.

Reset disables the whole M-Bus block by clearing the M-Bus Control Register (MCR, location $3B). Refer to **SECTION 5 M-BUS SERIAL COMMUNICATION INTERFACE** for additional information about the M-Bus interface circuitry.

3.2.9 Real Time Clock Interrupt

A RTC timer interrupt is enabled when the RTC enable flag, bit 5 of Miscellaneous Register (MISC, location $1C) is set, provided the interrupt mask bit of the condition code register is cleared. The interrupt service routine address is specified by the contents of memory location $7FF0, $7FF1. There is no need to reset this interrupt by clearing or accessing any flag. Refer to **SECTION 4 PROGRAMMER TIMER** for additional information about the Real Time Clock.

3.3 WAIT MODE

The WAIT instruction places the MC68HC05T10 in a low power consumption mode. When the MCU enters the WAIT mode, the CPU clock is halted, CPU action is suspended; however the A/D, external interrupt, Timer, OSD, Keyboard interrupt, M-BUS interface, RTC, and PLM remain active. Any IRQ, Timer interrupt(overflow, input capture or output compare), OSD interrupt, M-BUS interrupt , RTC, keyboard interrupt in addition to a logic low on the RESET pin causes the processor to exit the WAIT mode. In WAIT mode, the A/D, PLM, Timer, OSD and M-BUS operate as they do in normal mode.

The WAIT mode power consumption depends on how many systems are active. The power consumption will be highest when all the systems are active. All other registers and memory remain unaltered and all parallel input/output lines remain unchanged.

If a non-reset exit from the WAIT mode is performed (e.g. Timer Overflow interrupt exit), the state of the remaining systems will be unchanged.

If a reset exit from the WAIT mode is performed, all the system reverts to the disabled reset state. See Table 3-1 for a list of the sections affected by the WAIT instruction.

The WAIT instruction has different effects on the programmable timer, the M-BUS, the OSD, the PLM, the A/D converter, external interrupt and the RTC. These different effects are discussed separately in the following sections.

3.3.1 Timer During Wait Mode

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

3.3.2 M-Bus During Wait Mode

The M-Bus system is not affected by the wait mode and continues regular operation. Any valid M-bus interrupt will wake the system up.

3.3.3 OSD During Wait Mode

Upon receiving interrupt request from the OSD, the MCU wakes up and updates the OSD display ram with new data.

3.3.4 DAC During Wait Mode

The DAC system is not affected by the WAIT mode and continues regular operation.

3.3.5 A/D Converter During Wait Mode

The A/D converter is not affected by WAIT mode and continues regular operation.

3.3.6 External Interrupt During Wait Mode

During the WAIT mode the I-bit in the condition code register is cleared to enable all interrupts. The INTE-bit is not affected by the WAIT mode. When any interrupt or reset is sensed, the program counter vectors to corresponding locations containing the starting address of the interrupt or reset service routine.

3.3.7 RTC During WAIT Mode

The RTC system is not affected by WAIT mode and continues regular operation. It will wake up the CPU for every 500 ms as usual.

3.4 STANDBY MODE

The standby V_{CC} (STBY) is a RAM protection supply voltage. Under normal operation, RAM is supplied by V_{CC}, 5V+/-10%. When V_{CC} drops down to CPU normal operation threshold voltage at 4 MHz crystal and under Hi-Low temperature condition, the CPU reset pin will be pulled low internally and stay down by internal low voltage detection circuit. This will prevent the CPU from writing its RAM, and the memory content is preserved as long as the standby V_{CC} is higher than the falling off V_{CC}. The minimum standby V_{CC} should be 2V.

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. Refer to Figure 4-1 for a timer block diagram.

Because the timer has a 16-bit architecture, each specific functional segment (capability) is represented by two registers. These registers contain the high and low byte of that functional segment. Generally, accessing the low byte of a specific timer function allows full control of that function; however, an access of the high byte inhibits that specific timer function until the low byte is also accessed.

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 below.

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, and
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 2.0 microseconds if the internal bus clock is 2.0 MHz. The counter is incremented during the low portion of the internal bus clock. Software can read the counter at any time without affecting its value.

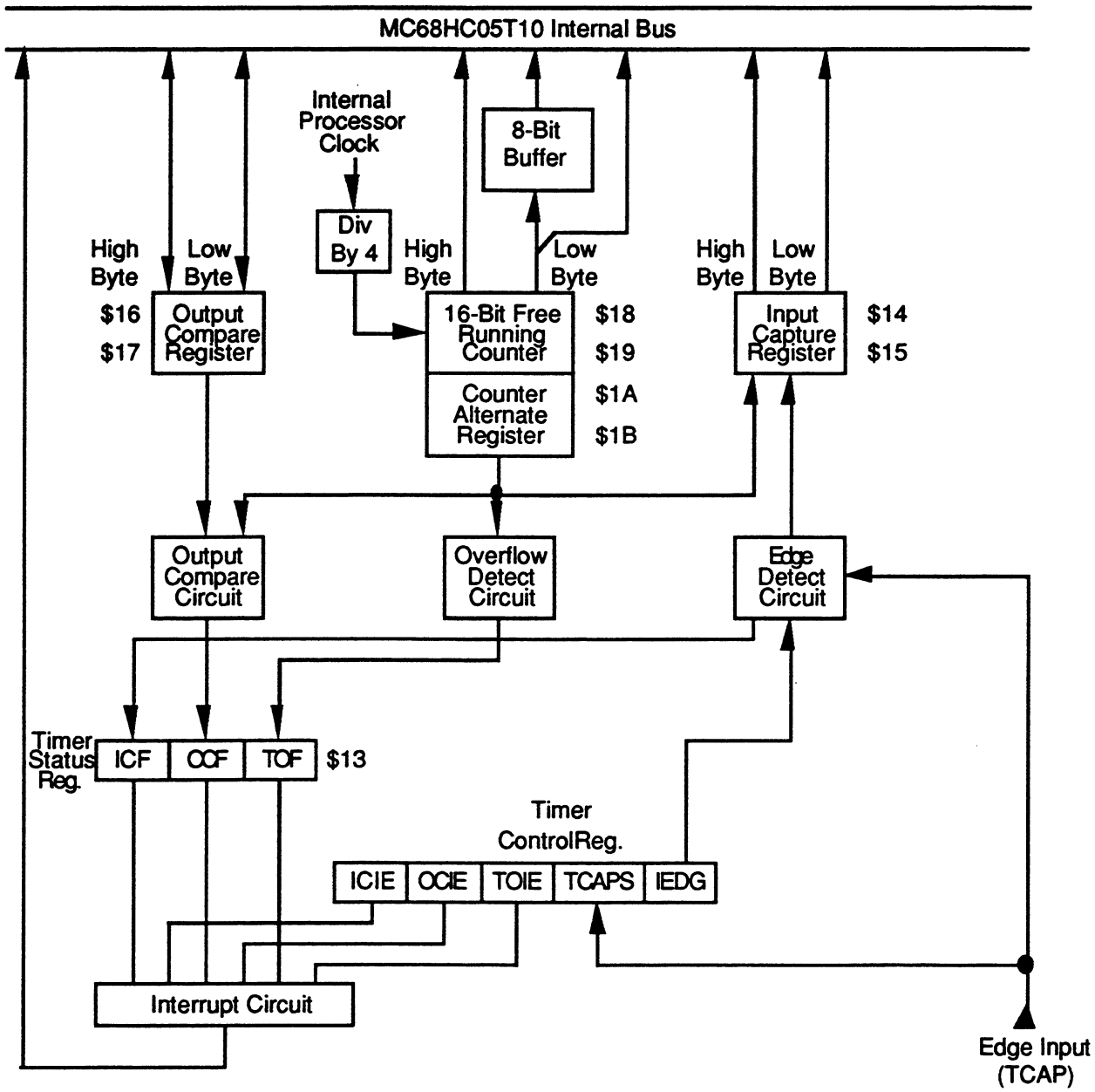
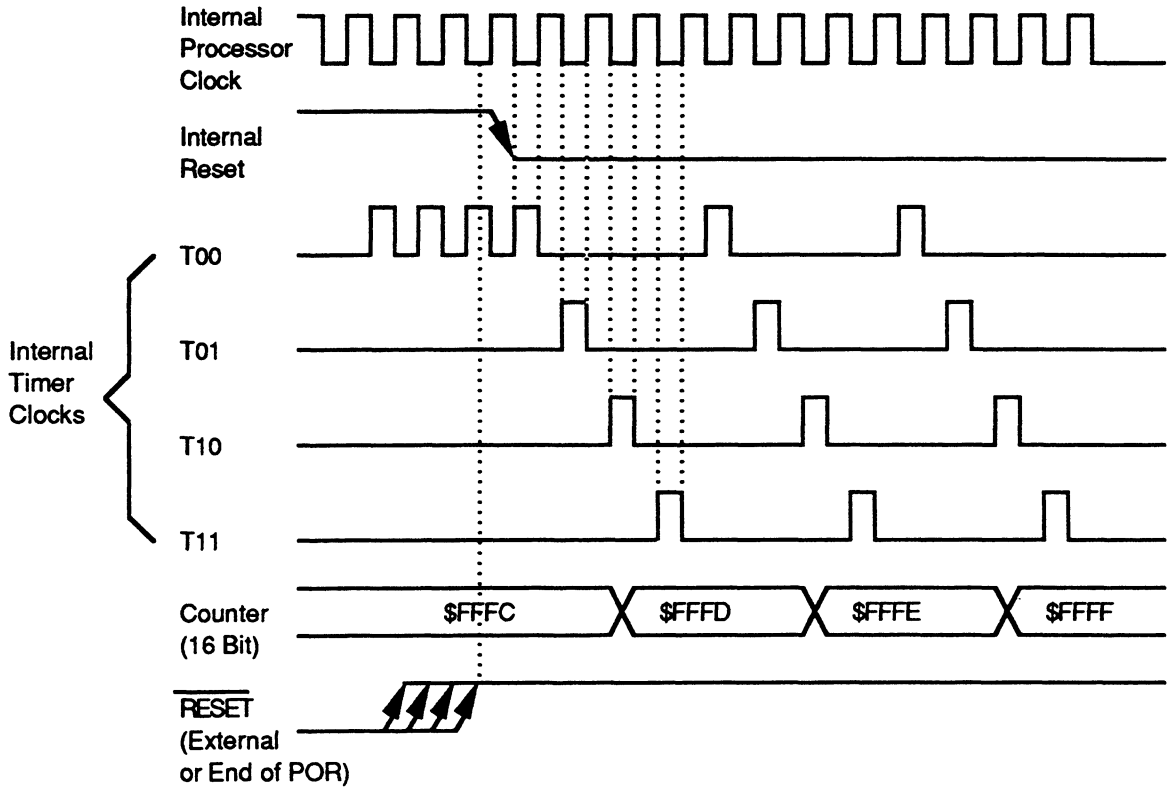
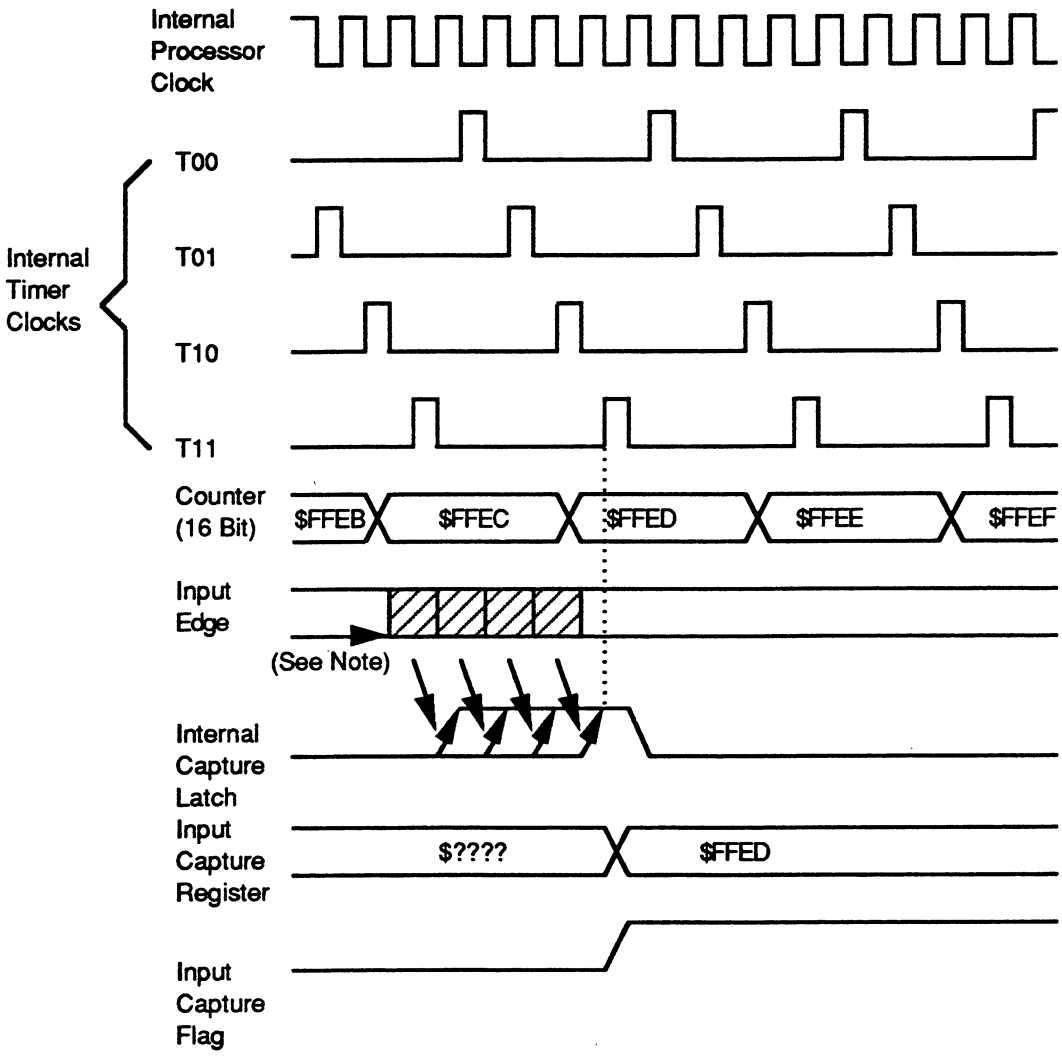


Figure 4-1 Programmable Timer Block Diagram



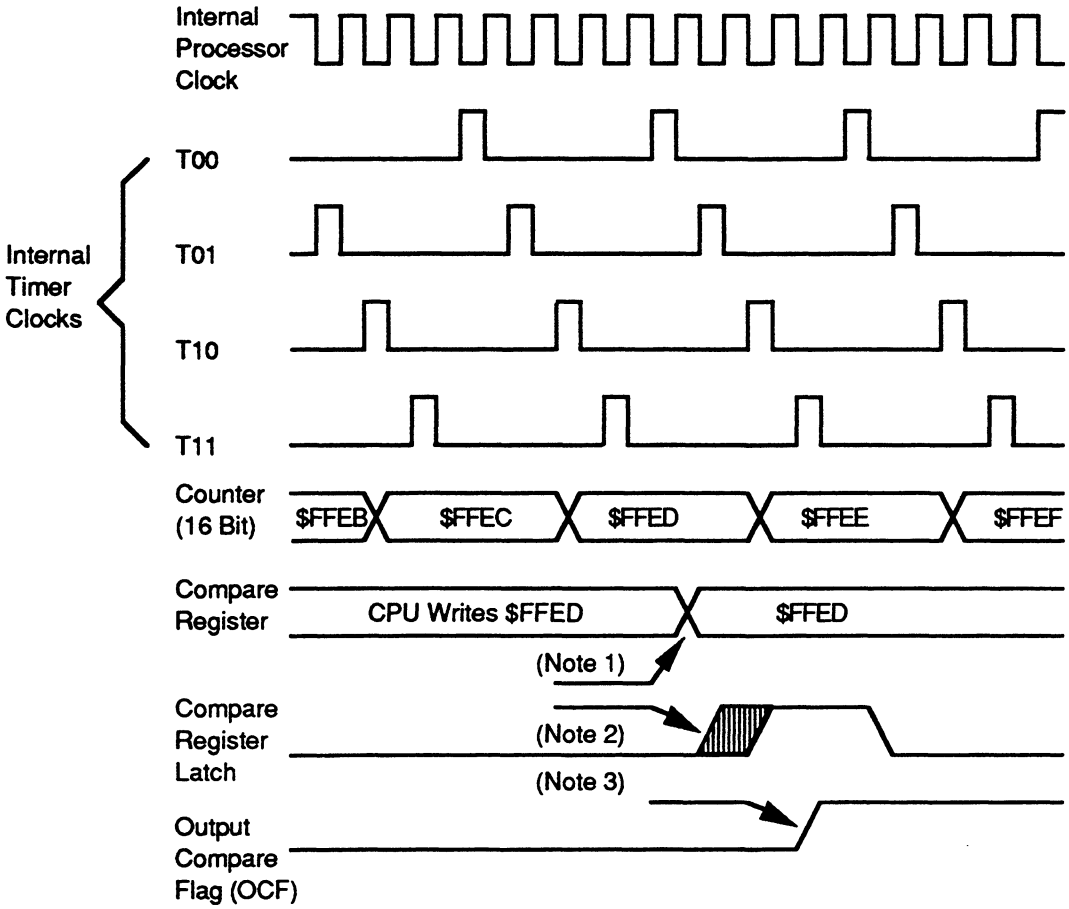
NOTE : The Counter Register and Timer Control Register are the only ones affected by $\overline{\text{RESET}}$

Figure 4-2 Timer State Timing Diagram for Reset



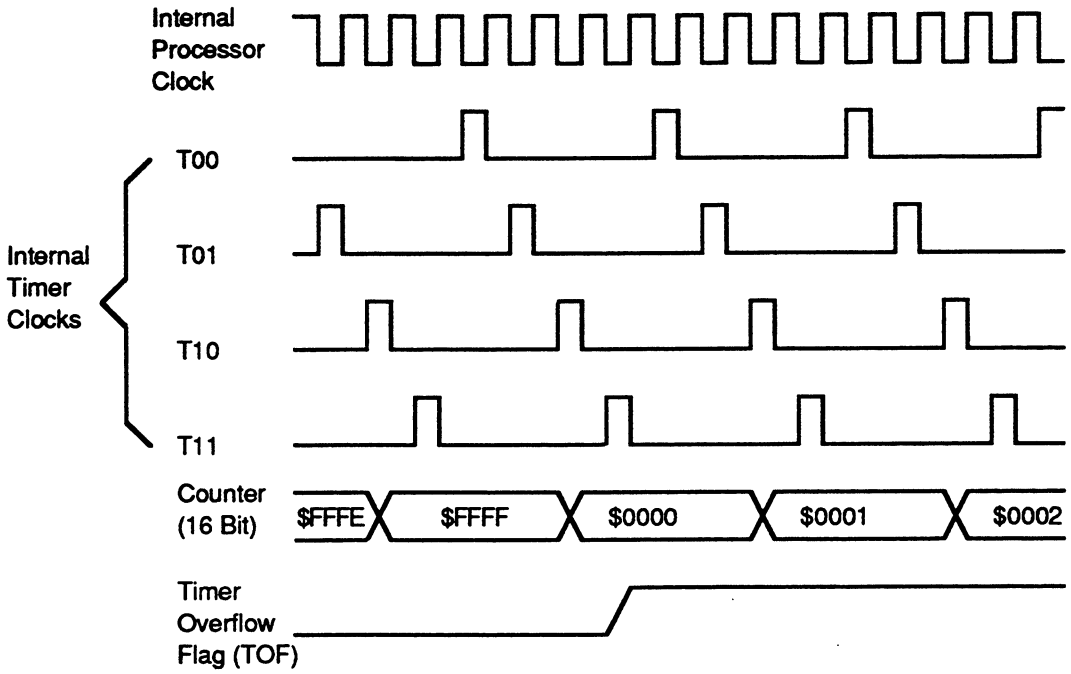
NOTE: If the input edge occurs in the shaded area from one timer state T10 to the other timer state T10 the input capture flag is set during the next state T11.

Figure 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).

Figure 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.

Figure 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 MSB 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. The output compare register values 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.

Because the output compare flag (OCF bit) is affected by reset, care must be exercised when initializing the output compare function with software. The following procedure is recommended:

(1) Write to the high byte of the output compare register to inhibit further compares until the low byte is written.
(2) Read the timer status register to arm the OCF if it is already set.
(3) Write to the low byte of the output compare register to enable the output compare function with the flag clear.

The advantage of this procedure is to prevent the OCF bit from being set between the time it is read and the write to the output compare register. A software example is shown below.

```
B7    16    STA   OCMPHI    INHIBIT OUTPUT COMPARE
B6    13    LDA   TSTAT     ARM OCF BIT IF SET
BF    17    STX   OCMPLD    READY FOR NEXT COMPARE
```

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 synchronization. Resolution is one count of the free-running counter, which is four internal bus clock cycles.

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

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

A read of the least significant byte ($15) of the input capture register does not inhibit the free running counter transfer. Again, minimum pulse periods are ones which allow software to read the least significant byte ($15) and perform needed operations. There is no conflict between the read of the internal processor clock.

4.5 TIMER CONTROL REGISTER (TCR)

The TCR is a read/write register containing five control bits and one status bit. 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 status bit reflects the state of the TCAP input pin signal. 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	TCAPS	0	0	IEDG	-	$12

B7, ICIE If the input capture interrupt enable (ICIE) bit is set, a timer interrupt is enable when the ICF status flag (in the timer status register) is set. If the ICIE bit is cleared, theinterrupt 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 enabledwhenever the TOF status flag (in the timer status register) is set. If the TOIE bit is cleared, the interrupt is inhibited. The TOIE bit is cleared by reset.

B4, TCAPS This bit directly echoes the logical state of the input signal at the TCAP pin.

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

0 = negative edge
1 = positive edge

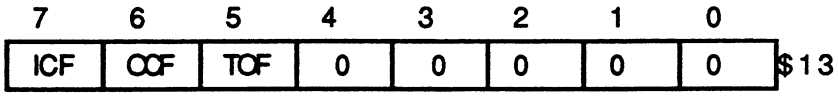
4.6 TIMER STATUS REGISTER (TSR)

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 a 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 WAIT instruction, 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.

4.7 REAL TIME CLOCK (RTC)

A 2 Hz clock interrupt is generated periodically if
1. a 50/60 Hz signal is fed to the internal counter via the RTC pin; and
2. the corresponding enable flag is set by software.

Either a 50 Hz or 60 Hz input frequency can be used provided the choice is specified by software.

Both enable flag and clock source selection flag are stored in MISC Register located at $1C as stated below,

7	6	5	4	3	2	1	0	
0	RTCS	RTCE	KEYS	KEYE	POR	INTN	INTE	$1C

B6, RTCS = 1 Select 60 Hz external clock source
= 0 Select 50 Hz external clock source

B5, RTCE = 1 RTC interrupt is generated at a rate of 2 Hz
= 0 Disable RTC interrupt

Once the RTC interrupt is serviced, the internal interrupt status flag will be cleared automatically.

SECTION 5
M-BUS SERIAL COMMUNICATION INTERFACE

5.1 INTRODUCTION

Motorola bus (in short: M-Bus) is a two wires, bidirectional serial bus which provides a simple, efficient way for data exchange between devices. It is fully compatible to I^2C bus from Philips. This two wire bus minimizes the interconnection between the devices and eliminates the need of address decoder, so that less PCB traces and economic hardware structure resulted.

This bus is suitable for the applications which need frequent, occasional communications in a short distance among a number of devices. It also provides a flexibility that allows additional devices to be "hanged" on it in further expansion for system developing.

The maximum data rate is limited on 100 Kbit/s, the maximum communication length and number of devices that can be connected are limited by a maximum bus capacitance of 400 pF.

M-Bus system is a true multi-master bus including collision detection and arbitration to prevent data corruption if two or more masters intend to control the bus simultaneously. This feature provides the capability for complex applications with multi-processor control. It may also be used for rapid testing and alignment of end products via external connections to an assembly-line computer.

5.2 M-BUS INTERFACE FEATURES

* Fully compatible to I^2C Bus standard
* Multi-master operation
* Software programmable for one of 32 different serial clock frequencies
* Software selectable acknowledge bit
* Interrupt driven byte by byte data transfer
* Arbitration lost driven interrupt with automatic mode switching from master to slave
* Calling address identification interrupt
* Generate/detect the start or stop signal
* Repeated START signal generation
* Generate/recognize the acknowledge bit
* Bus busy detection

5.3 M-BUS SYSTEM CONFIGURATION

M-bus system uses a serial data line (SDA) and a serial clock line (SCL) for data transfer. All devices connected on it must have open drain or open collector output, logic "and" function is exercised in both lines with two pulling up resistors.

5.4 M-BUS PROTOCOL

Normally, a standard communication is composed of four parts, START signal, Slave Address transmission, Data transfer, and STOP signal. They are described briefly in the following sections and illustrated in Figure 5-1.

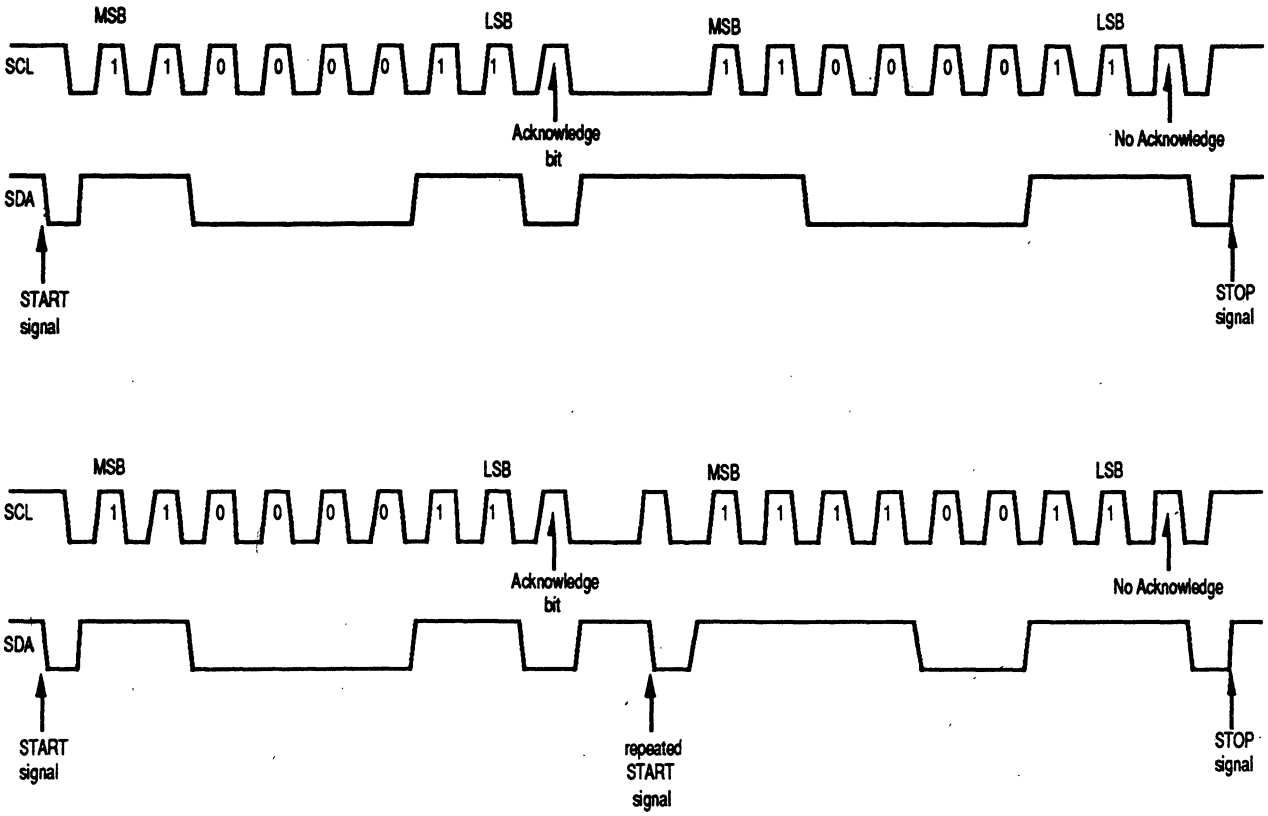


Figure 5-1 M-Bus Transmission Signal Diagram

5.4.1 START Signal

When the bus is free, i.e., no master device is engaging the bus (both SCL and SDA lines are at logical high), a master may initiate communication by sending a START signal. As shown in Figure 5-1, a START signal is defined as a high to low transition of SDA while SCL is high. This signal denotes the beginning of a new data transfer (each data transfer may contain several bytes of data) and wake up all slaves.

5.4.2 Slave Address Transmission

The first byte of data transfer immediately after the START signal is the slave address transmitted by the Master. This is a seven bits long calling address followed by a R/W- bit . The R/W- bit tells the slave the desired direction of data transfer.

Only the slave with matched address will respond by sending back an acknowledge bit which is signaled from the receiving device by pulling the SDA low at the 9th clock. (See Figure 5-1)

5.4.3 Data Transfer

Once a successful slave addressing is achieved, the data transfer can proceed byte by byte in a direction specified by the R/W- bit sent by the calling master.

Each data byte is 8 bit long. Data can be changed only during SCL is low and must be held stable during SCL is high as shown in Figure 5-1. One clock pulse is for one bit data transferring, MSB is transferred first. Each byte data has to be followed by an acknowledge bit, which is signaled from the receiving device by pulling the SDA low at the 9th clock. So one complete data byte transferring needs 9 clock pulses.

If the slave receiver does not acknowledge the master, the SDA line should be left high by the slave, the master can then generate a stop signal to abort the data transfer or a start signal (repeated start) to commence a new calling.

If the master receiver does not acknowledge the slave transmitter after one byte transmission, it means a 'end of data' to the slave, so that the slave shall release the SDA line for the master to generate 'stop' or 'start' signal.

5.4.4 Repeated START Signal

As shown in Figure 5-1, a repeated START signal is to generate a START signal without first generating a STOP signal to terminate the communication. This is used by the master to communicate with another slave or with the same slave in different mode (transmit/receive mode) without releasing the bus.

5.4.5 STOP Signal

The master can terminate the communication by generating a STOP signal to free the bus. However, the master may generate a START signal followed by a calling command without generating a STOP signal first. This is called repeat start. A STOP signal is defined as a low to high transition of SDA while SCL at logical high. (See Figure 5-1).

5.4.6 Arbitration Procedure

This interface circuit is a true multi-master system which allows more than one master to be connected on it. If two or more masters try to control the bus at the same time, a clock synchronization procedure determines the bus clock, for which the low period is equal to the longest clock low period and the high is equal to the shortest one among the masters. A data arbitration procedure determines the priority, the masters will lose arbitration if they transmits logic "1" while the other transmits logic "0", the losing masters will immediately switch over to slave receive mode and stop its data and clock outputs. The transition from master to slave mode will not generate a stop condition in this case. Meanwhile, a software bit will be set by hardware to indicate lost of arbitration.

5.4.7 Clock Synchronization

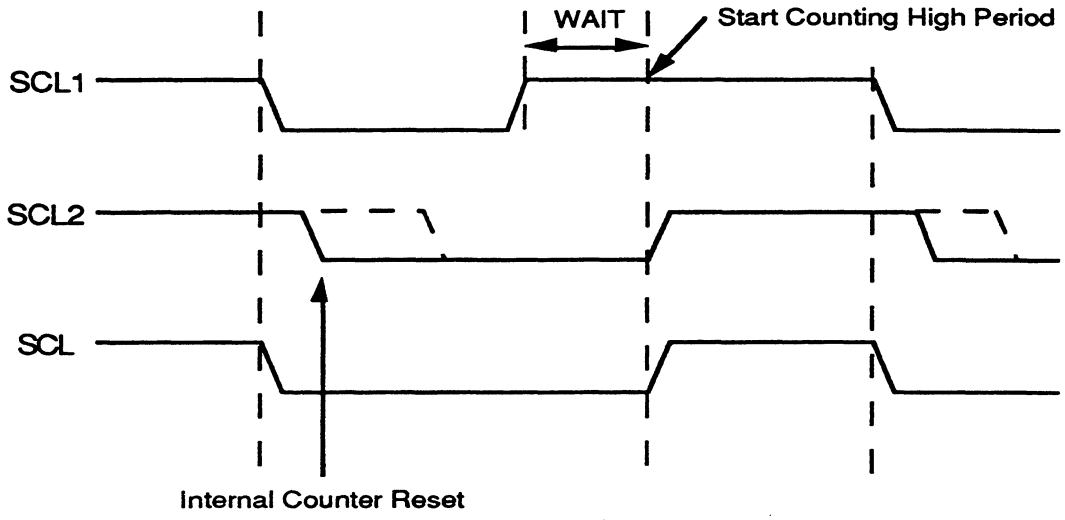


Figure 5-2 Clock Synchronization

Since wire-AND logic is performed on SCL line, a high to low transition on SCL line will affect the devices connected on the bus. The devices start counting their low period and once a device's clock has gone low, it will hold the SCL line low until the clock high state is reached. However, the change of low to high in this device clock may not change the state of the SCL line, if another device clock is still within its low period. Therefore synchronized clock SCL will be held low by the device with the longest low period. Devices with shorter low periods enter a high wait state during this time (See Figure 5-2). When all devices concerned have counted off their low period, the synchronized clock SCL line will be released and go high. There will then be no difference between the device clocks and the state of the SCL line and all of them will start counting their high periods. The first device to complete its high period will again pull the SCL line low.

5.4.8 Handshaking

The clock synchronization mechanism can be used as a handshake in data transfer. Slave device may hold the SCL low after completion of one byte transfer (9 bits). In such case, it will halt the bus clock and force the master clock in a wait state until the slave release the SCL line.

5.5 REGISTERS

There are five different registers used in the M-Bus interface and the internal configuration of these registers is discussed in the following paragraphs. A block diagram of the M-Bus system is shown in Figure 5-3.

5.5.1 M-Bus Address Register (MADR)

7	6	5	4	3	2	1	0	
ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	-	$39

Bit 1 to bit 7 contain its own specific slave address. This register is cleared upon reset.

5.5.2 M-Bus Frequency Divider Register (FDR)

7	6	5	4	3	2	1	0	
-	-	-	MBC4	MBC3	MBC2	MBC1	MBC0	$3A

Bit 0 to bit 4 are used for clock rate selection. The serial bit clock frequency is equal to the CPU clock divided by the divider shown in the following Table 5-1. This register is cleared upon reset.

Table 5-1 M-Bus Prescaler

MBC4, MBC3, MBC2, MBC1, MBC0	DIVIDER	MBC4, MBC3, MBC2, MBC1, MBC0	DIVIDER
00000	22	10000	352
00001	24	10001	384
00010	28	10010	448
00011	34	10011	544
00100	44	10100	704
00101	48	10101	768
00110	56	10110	896
00111	68	10111	4088
01000	88	11000	1408
01001	96	11001	1536
01010	112	11010	1792
01011	136	11011	2176
01100	176	11100	2816
01101	192	11101	3072
01110	224	11110	3584
01111	272	11111	4352

For a 4 MHz external crystal operation (2 MHz internal operating frequency), the serial bit clock frequency of M-Bus ranges from 460 Hz to 90,909 Hz.

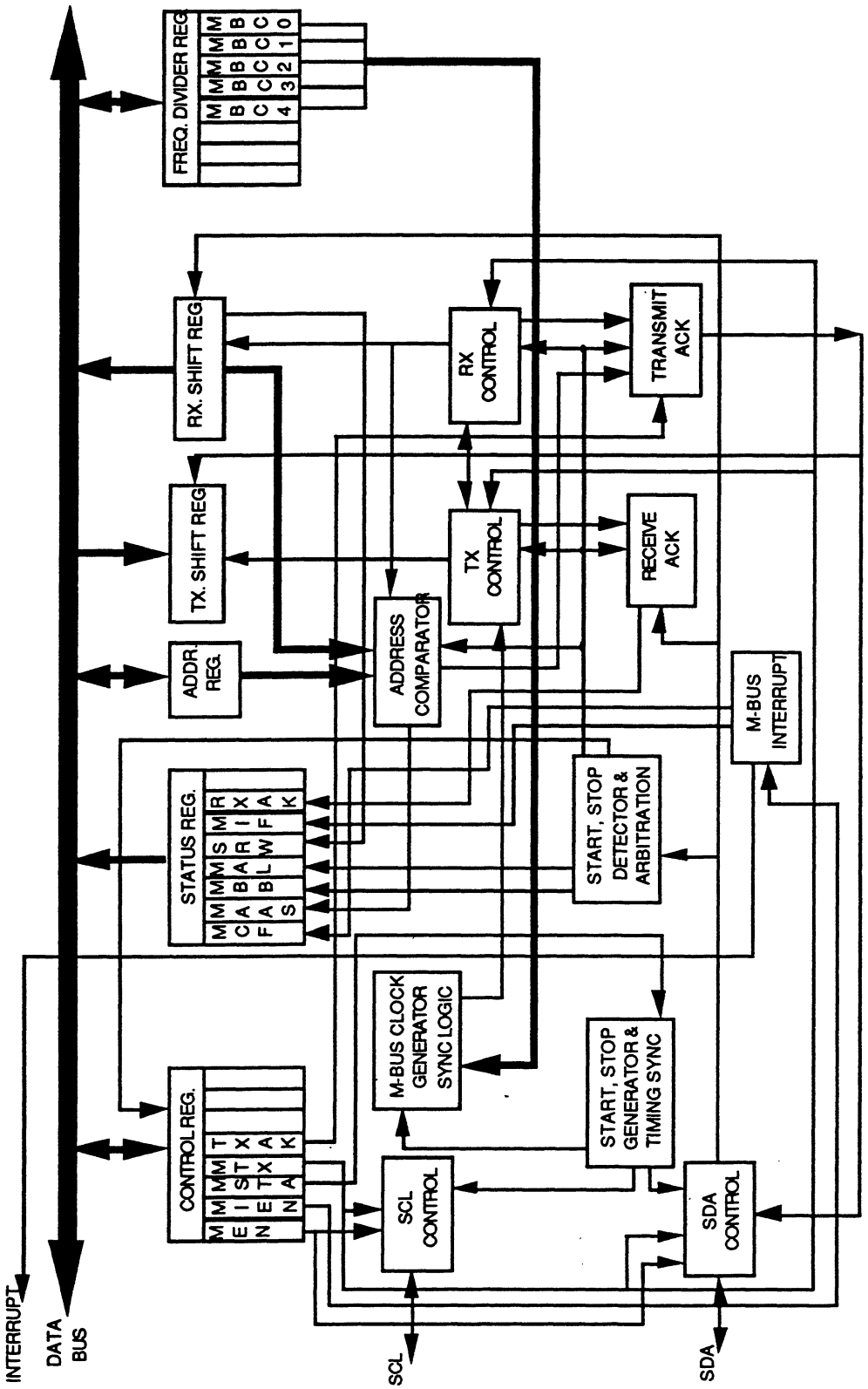


Figure 5-3 M-Bus Interface Block Diagram

5.5.3 M-Bus Control Register (MCR)

7	6	5	4	3	2	1	0	
MEN	MIEN	MSTA	MTX	TXAK	-	-	-	$3B

The MCR provides five control bits. MCR is cleared upon reset.

B7, MEN — If the M-Bus enable bit (MEN) is set, the M-Bus interface system is enabled. If MEN is cleared, the interface is reset and disabled.
The MEN bit must be set first before any bits of MCR being set.

B6, MIEN — If the M-Bus interrupt enable bit (MIEN) is set, the M-bus interrupt occurs provided the MIF in the status register is set. If MIEN is cleared, the M-Bus interrupt is disabled.

B5, MSTA — Master/slave mode select bit
1 = Master
0 = Slave
Upon reset, this bit is cleared. When this bit is changed from 0 to 1, a START signal is generated on the bus, and the master mode is selected. When this bit is changed from 1 to 0, a STOP signal is generated and the operation mode changes from master to slave.
In master mode, a bit clear immediately followed by a bit set of this bit generates a repeated START signal (see Figure 5-1) without generating a STOP signal.

B4, MTX — Transmit / Receive mode select bit
1 = Transmit
0 = Receive

B3, TXAK — If the transmit acknowledge enable bit (TXAK) is cleared, an acknowledge signal will be sent out to the bus at the 9th clock bit after receiving one byte data. When TXAK is set, no acknowledge signal response (i.e., acknowledge bit = 1).

5.5.4 M-Bus Status Register (MSR)

This status register is software readable only with exception of bit 1 (MIF) and bit 4 (MAL), which are software clearable. All bits are cleared upon reset except bit 7 (MCF) and bit 0 (RXAK).

7	6	5	4	3	2	1	0	
MCF	MAAS	MBB	MAL	-	SRW	MIF	RXAK	$3C

B7, MCF — Data transferring bit
When one byte of data is being transferred, this bit is cleared.

B6, MAAS — Addressed as a slave bit
1 = Addressed as a slave
0 = Not addressed
When its own specific address (M-Bus Address Register) is matched with the calling address, this bit is set. The CPU is interrupted provided the MIEN is set. Then CPU needs to check the SRW bit and set its TX/RX mode accordingly.
Writing to the M-Bus Control Register clears this bit.

B5, MBB — Bus busy bit
1 = bus busy
0 = bus idle
This bit indicates the status of the bus. When a START signal is detected, the MBB is set. If a STOP signal is detected, it is cleared.

B4, MAL — The arbitration lost bit (MAL) is set by hardware when the arbitration procedure is lost during a master transmission mode.
This bit must be cleared by software.

B2, SRW — When MAAS is set, the R/W- command bit of the calling address sent from master is latched into the R/W- command bit(SRW). Checking this bit, the CPU can select slave transmit/receive mode according to the command of master.

B1, MIF — The M-Bus interrupt bit (MIF) is set when there is a pending interrupt, which will cause a M-bus interrupt request provided MIEN is set. This bit is set when one of the following event occurs:

1. Complete one byte of data transfer, it is set at the falling edge of the 9th clock.
2. Receive a calling address which matches its own specific address in slave receive mode.
3. Arbitration lost.

This bit must be cleared by software in the interrupt routine.

B0, RXAK — If the received acknowledge bit (RXAK) is low, it indicates an acknowledge signal has been received after the completion of 8 bits data transmission on the bus. If RXAK is high, it means no acknowledge signal is detected at the 9th clock.
It is set upon reset.

5.5.5 M-Bus Data I/O Register (MDR)

7	6	5	4	3	2	1	0	
TRXD7	TRXD6	TRXD5	TRXD4	TRXD3	TRXD2	TRXD1	TRXD0	$3D

In master transmit mode, data writing into this register is sent to the bus automatically, most significant bit is sent first. In master receive mode, reading from this register initiates next byte data receiving. In slave mode, the same function is available after it is addressed.

5.6 PROGRAMMING CONSIDERATION

5.6.1 Initialization

Reset will put the M-bus Control Register to its default status. Before the interface can be used to transfer serial data, an initialization procedure must be carried out.

1. Update Frequency Divider Register (FDR) to select a SCL frequency.
2. Update M-Bus Address Register (MADR) to define its own slave address.
3. Set MEN bit of M-Bus Control Register (MCR) to enable the M-Bus interface system.
4. Modify the bits of M-Bus Control Register (MCR) to select Master/Slave mode, Transmit/Receive mode, interrupt enable or not.

5.6.2 Generation of a START Signal and the First Byte of Data Transfer

After completion of the initialization procedure, serial data can be transmitted by selecting the 'master transmitter' mode. If the device is connected to a multi-master bus system, the state of the M-Bus busy bit (MBB) must be tested to check whether the serial bus is free. If the bus is free (MBB=0), the start condition and the first byte (the slave address) can be sent. An example of a program which generates the START signal and transmits the first byte of data (slave address) is shown below:

```
            SEI                         ; DISABLE INTERRUPT
CHFALG      BRSET   5,MSR,CHFLAG        ; CHECK THE MBB BIT OF THE
                                        ; STATUS REGISTER. IF IT IS
                                        ; SET, WAIT  UNTIL IT IS CLEAR
TXSTART     BSET    4,MCR               ; SET TRANSMIT MODE
            BSET    5,MCR               ; SET MASTER MODE
                                        ; i.e. GENERATE START CONDITION
            LDA     #CALLING            ; GET THE CALLING ADDRESS
            STA     MDR                 ; TRANSMIT THE CALLING
                                        ; ADDRESS
            CLI                         ; ENABLE INTERRUPT
```

5.6.3 Software Responses after Transmission or Reception of a Byte

Transmission or reception of a byte will set the data transferring bit (MCF) to 1, which indicates one byte communication is finished. Also the M-Bus interrupt bit (MIF) is set to generate an M-bus interrupt if the interrupt function is enable in initialization. Software must clear the MIF bit in the interrupt routine first. The MCF bit will be cleared by reading from the M-Bus Data I/O Register (MDR) in receive mode or writing to MDR in transmit mode. Software may serves the M-bus I/O in the main program by monitoring the MIF bit if the interrupt function is disabled. The following is an example of a software response by a 'master transmitter' in the interrupt routine (See Figure 5-4).

```
ISR         BCLR    1,MSR               ; CLEAR THE MIF FLAG
            BRCLR   5,MCR,SLAVE         ; CHECK THE MSTA FLAG,
                                        ; BRANCH IF SLAVE MODE
            BRCLR   4,MCR,RECEIVE       ; CHECK THE MODE FLAG,
                                        ; BRANCH IF IN RECEIVE MODE
            BRSET   0,MSR,END           ; CHECK ACK FROM RECEIVER
                                        ; IF NO ACK, END OF
                                        ; TRANSMISSION
TRANSMIT    LDA     DATABUF             ; GET THE NEXT BYTE OF DATA
            STA     MDR                 ; TRANSMIT THE DATA
```

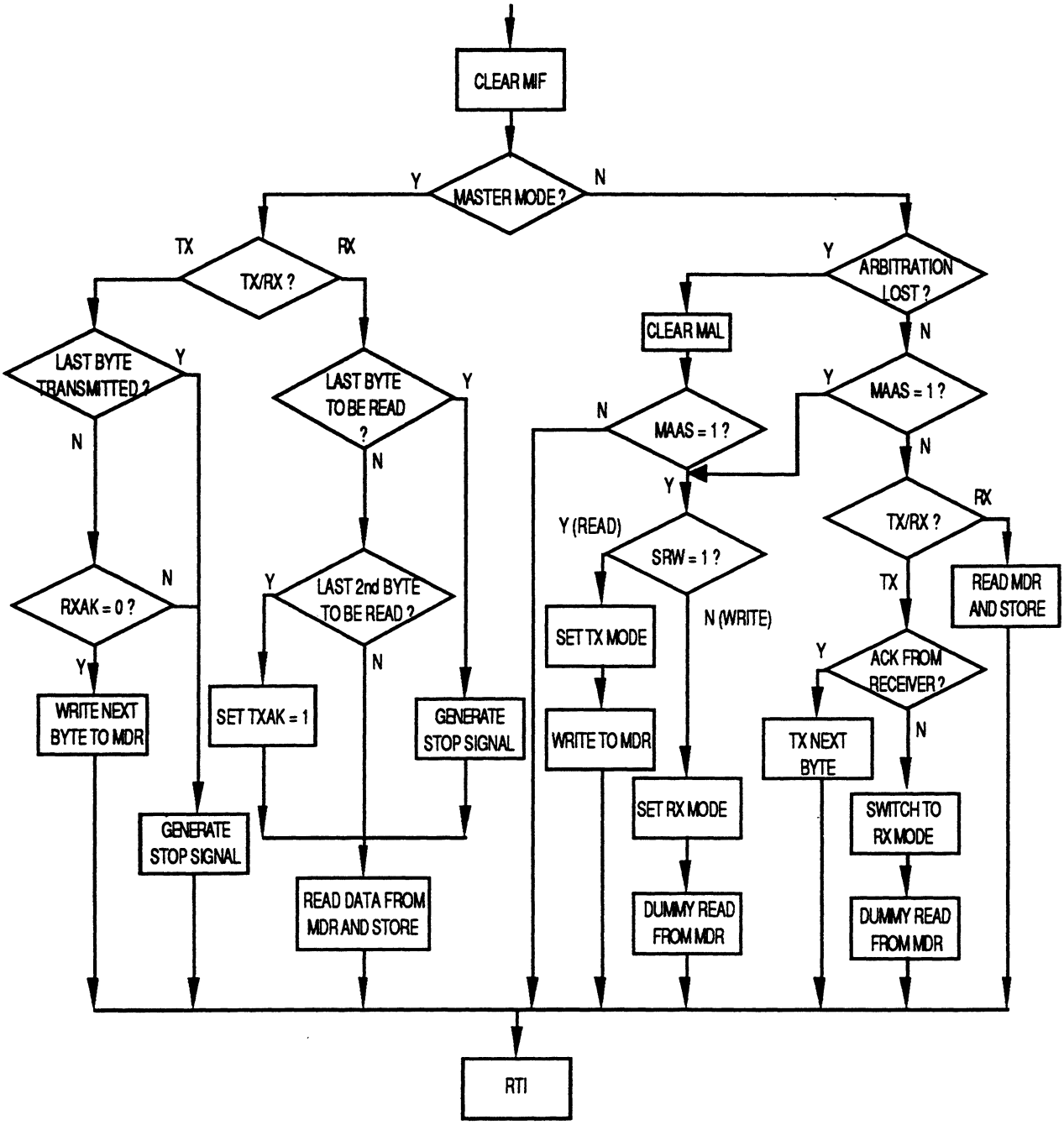


Figure 5-4 Flow-chart of M-Bus Interrupt Routine

5.6.4 Generation of the STOP Signal

A data transfer ends with a STOP signal generated by the 'master' device. A master transmitter can simply generate a STOP signal after all the data has been transmitted. The following is an example showing how a stop condition is generated by a master transmitter.

```
MASTX   BRSET   0,MSR,END       ; IF NO ACK, BRANCH TO END
        LDA     TXCNT           ; GET VALUE FROM THE
                                ; TRANSMITTING COUNTER
        BEQ     END             ; IF NO MORE DATA, BRANCH TO
                                ; END
        LDA     DATABUF         ; GET NEXT BYTE OF DATA
        STA     MDR             ; TRANSMIT THE DATA
        DEC     TXCNT           ; DECREASE THE TXCNT
        BRA     EMASTX          ; EXIT
END     BCLR    5,MCR           ; GENERATE A STOP CONDITION
EMASTX  RTI                     ; RETURN FROM INTERRUPT
```

If a master receiver wants to terminate a data transfer, it must inform the slave transmitter by not acknowledging the last byte of data which can be done by setting the transmit acknowledge bit (TXAK) before reading the 2nd last byte of data. Before reading the last byte of data, a STOP signal must be generated first. The following is an example showing how a STOP signal is generated by a master receiver.

```
MASR    DEC     RXCNT
        BEQ     ENMASR          ; LAST BYTE TO BE READ
        LDA     RXCNT
        DECA                    ; CHECK LAST SECOND BYTE
                                ; TO BE READ
        BNE     NXMAR           ; NOT LAST ONE OR LAST SECOND
LAMAR   BSET    3,MCR           ; LAST SECOND, DISABLE ACK
                                ; TRANSMITTING
        BRA     NXMAR
ENMASR  BCLR    5,MCR           ; LAST ONE, GENERATE 'STOP'
                                ; SIGNAL
NXMAR   LDA     MDR             ; READ DATA AND STORE
        STA     RXBUF
        RTI
```

5.6.5 Generation of a Repeated START Signal

At the end of data transfer, if the master still want to communicate on the bus, it can generate another START signal followed by another slave address without first generate a STOP signal. A program example is as shown.

```
RESTART BCLR    5,MCR           ; ANOTHER START (RESTART) IS
        BSET    5,MCR           ; GENERATED BY THESE TWO
                                ; CONSEQUENCE INSTRUCTION
        LDA     #CALLING        ; GET THE CALLING ADDRESS
        STA     MDR             ; TRANSMIT THE CALLING
                                ; ADDRESS
```

5.6.6 Slave Mode

In slave service routine, the master addressed as slave bit (MAAS) should be tested for checking if a calling of its own address has just received (See Figure 5-4). If MAAS is set, software should set the transmit/receive mode select bit (MTX bit of MCR) according to the R/W- command bit (SRW). Writing to the MCR clears the MAAS automatically. Then a data transfer may be initiated by writing information to MDR or dummy reading from MDR.

In slave transmitter routine, the received acknowledge bit (RXAK) must be tested before transmitting next byte of data. RXAK is set means 'end of data' signal from the master receiver, then it must switch from transmitter mode to receiver mode by software and a dummy read followed to release the SCL line so that the master can generate a STOP signal.

5.6.7 Arbitration Lost

If more than one master want to engage the bus simultaneously, only one master wins and the others lost arbitration. The arbitration lost devices immediately switch to slave receive mode by hardware. Their data output to the SDA line is stopped, but internal transmitting clock still run until the end of the byte transmitting. An interrupt occurs when this dummy 'byte' transmitting is accomplished with MAL=1 and MSTA=0. If one master attempt to start transmission while the bus is being engaged by another master, the hardware will inhibit the transmission; switch the MSTA bit from 1 to 0 without generating STOP condition; generate an interrupt to CPU and set the MAL to indicate that the attempt to engage the bus is failed. Consideration of these cases, the slave service routine should test the MAL first, software should clear the MAL bit if it is set.

SECTION 6
ON SCREEN DISPLAY

6.1 INTRODUCTION

The programmable On Screen Display (OSD) system displays 10 rows of 18 characters or graphic symbols in 7x9 dot matrix (maximum 8x11 or 8x13) over the full TV screen. The display color, position, size, and windowing are software programmable. The block diagram of OSD is illustrated in Figure 6.1.

The display timing is derived from a 14MHz on-chip oscillator which is phase locked with the input horizontal flyback pulse. The horizontal and vertical time bases are derived from this 14 MHz oscillator. For different TV system applications, e.g., 15.734KHz/60Hz (horizontal line frequency of 15.734KHz and vertical field frequency of 60 Hz), 15.625KHz/50Hz, 31.5KHz/60Hz, 31.25KHz/50Hz, 33.7KHz/60Hz, and 31.25KHz/100Hz, there are 4 software flags set by auto-detection circuit which counts the line number (number of horizontal flyback pulses) between two consecutive vertical flyback pulses and time the horizontal period. Then the time base is automatically adjusted to achieve a stable character display in any one of these TV systems.

A single row memory map architecture is adopted in this device to minimize the display RAM size. The CPU loads a row of display data into the 24 bytes OSD buffered registers through the internal data & address buses, and the time base starts to scan. If the row counter address is matched with the display row address stored in the ROW ADDR/CHAR SIZE Register (See **SECTION 6.9.4** for details of this register), the appropriate characters will then be fetched from character ROM according to the character address stored in CHAR Register $0 - $11. The character dot matrix data pass through a half dot shift circuit and feed to a parallel in/serial out shift register to obtain the luminous signal. An interrupt request will be generated immediately to the CPU at the beginning of display for the next row data preparation. The luminous signal, after passing through a color encoder, is converted into R, G, B signals. Video attributes and window are then added to the R, G, B output signals. Each row is refreshed constantly by software within a maximum of 1397µs for NTSC system (1664µs for PAL system). However, CPU consumes well less than 400µs for loading the 24 bytes display data into the OSD registers so that the CPU can do other tasks during the rest of time.

6.2 FEATURES

The OSD provides the following features :

* Auto-detection of TV system of 525 Line/60Hz (interlace/progressive), 625 Line/50Hz (interlace/progressive), 625 Line/100Hz and mode of single line/double line (15KHz/31KHz)
* Programmable character display of 10 rows by 18 columns
* Programmable horizontal positioning for display centering of 15 KHz and 31 KHz modes
* Character set of 64 characters
* Character ROM of 8 (width) by 14 (height) dot matrix
* Half dot shifting for hardware character rounding

* Continuous horizontal/vertical lines attainable for linear scale
* 4 colors for each row selectable out of 8 colors (including black & white)
* Character by character color selectable out of the 4 chosen row colors of each row
* 4 sizes of character selectable; standard, double width, double height, double width & double height
* Black-edge feature (software selectable on a row by row basis)
* One programmable window with 8 colors (including black window) per row
* Total 5 output signal; fast luminous blanking output (for blanking of TV video), half-tone output (for control of window intensity), and R, G, B color outputs with software programmable polarity

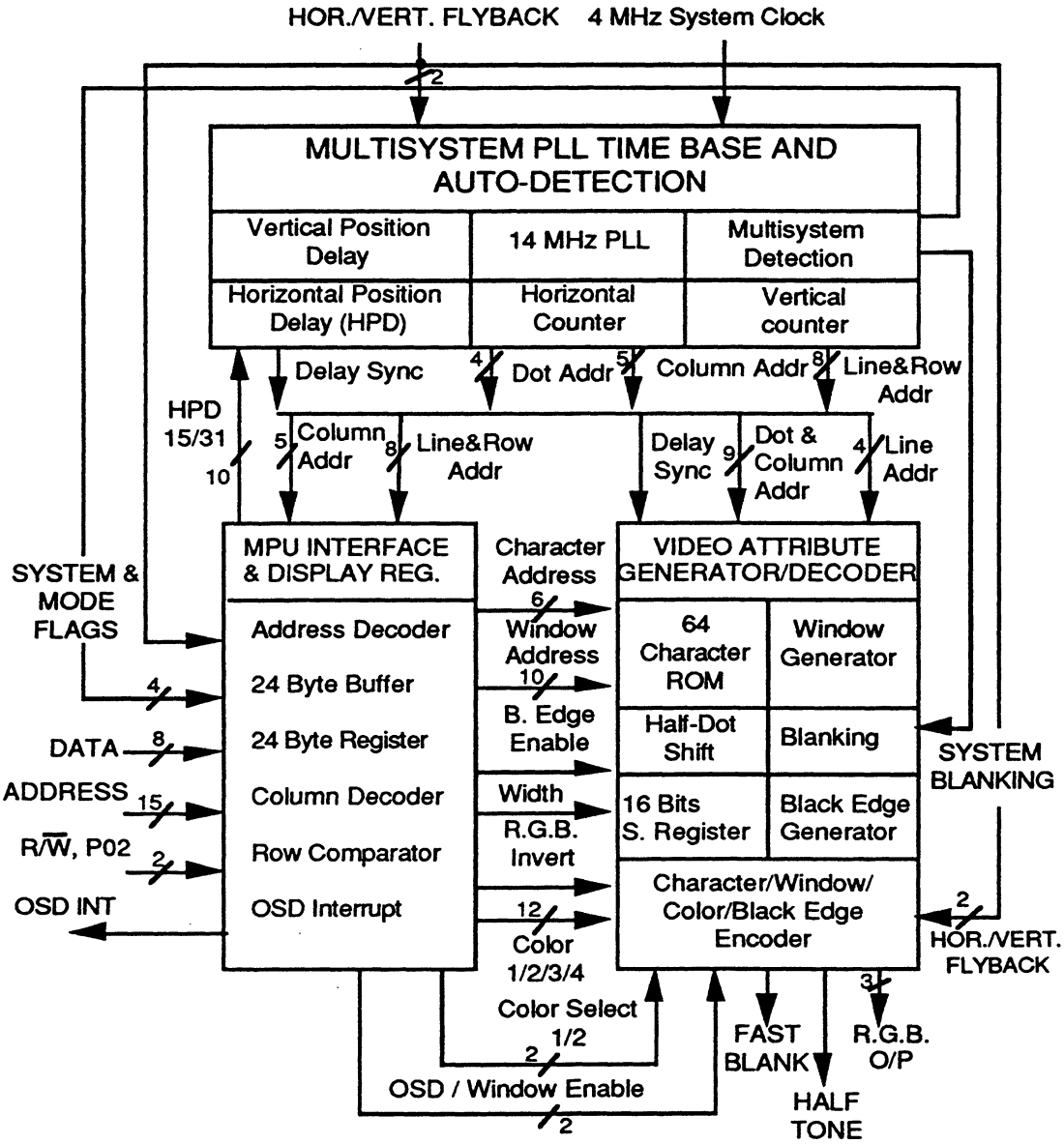


Figure 6-1 The OSD Block Diagram

6.3 MULTI-SYSTEM OSD

The OSD is capable of displaying a stable screen on a variety of TV system automatically. This is achieved by auto-detection of TV system and auto-adjustment of the internal PLL and related circuits.

The auto-detection of TV system is done by two measurements. The first one determines the frequency range of horizontal flyback pulse. A reference signal of frequency f_{ref} is obtained by counting down from the system clock of MCU. For a conventional 4 MHz crystal clock,

$$f_{ref} = 4\ MHz \div 192 = 20.83\ KHz$$

By comparing the input horizontal flyback signal with this reference signal, a software flag is set or cleared to exhibit the frequency range of it. (See **SECTION 6.9.2** for more details of this flag)

Table 6-1 Horizontal Flyback Frequency Classification

Frequency of Horizontal Flyback	Mode Bit
0 - f_{ref}	0
f_{ref} - 2f_{ref}	1

The second measurement is to count the line number between two consecutive vertical flyback pulses. When the counting process completed, the rising edge of the vertical flyback pulse clocks the deduced system flags into flip-flops. The falling edge of the vertical pulse is used to reset the counter and restart the counting cycle.

Table 6-2 Predefined Line Number Windows

SFG1	SFG3	Line Number	TV System
0	0	0-192	Abnormal line number, all outputs blanked
0	0	192-288	15734/60 Hz
1	0	288-416	15625/50 Hz or 31250/100 Hz
0	1	416-576	31500/60 Hz or 33750/60 Hz
1	1	576-704	31250/50 Hz

Combining these two measurements, most of the standard TV system can be determined as shown in Table 6-3.

NOTE

System flag 2 is simply the AND-product of mode flag and the complement of system flag 3. It is used to denote whether it is a European double system or not.

Table 6-3 TV System Classification

TV System	SF1	SF2	SF3	Mode Flag	$\overline{SF3}$
15K / 60Hz	0	0	0	0	1
15K / 50Hz	1	0	0	0	1
31.5K / 60Hz	0	0	1	1	0
31.25K / 50Hz	1	0	1	1	0
33.7K / 60Hz	0	0	1	1	0
31.25K / 100Hz	1	1	0	1	1

6.4 SINGLE ROW ARCHITECTURE

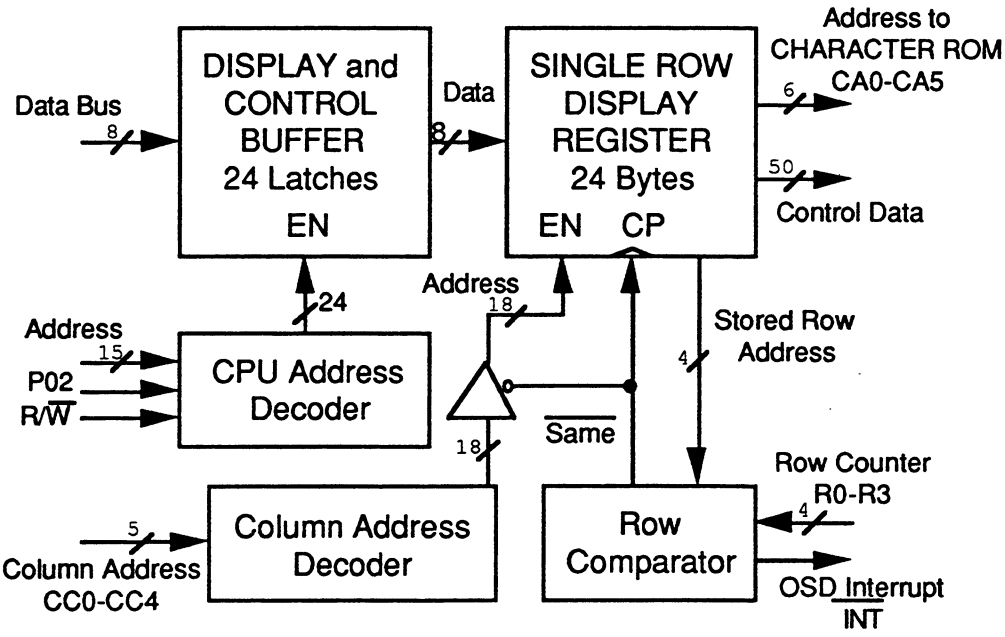


Figure 6-2 Single Row Architecture Block Diagram

The basic concept of single row architecture is to use one row of buffered registers. These registers always store the display data of the next row. See **SECTION 6.9** for the details of these display data. When this row is about to be displayed, a row comparator (See Figure 6-2) will generate a Same- signal. This Same- signal opens the gate feeding column address to enable those registers output sequentially. At the same time, an OSD interrupt occurs and the CPU start to serially transfer the next row of display data into the register buffers. At the end of display of this row, the Same- signal goes high and this clock the buffered data into their corresponding registers and the cycle repeats again.

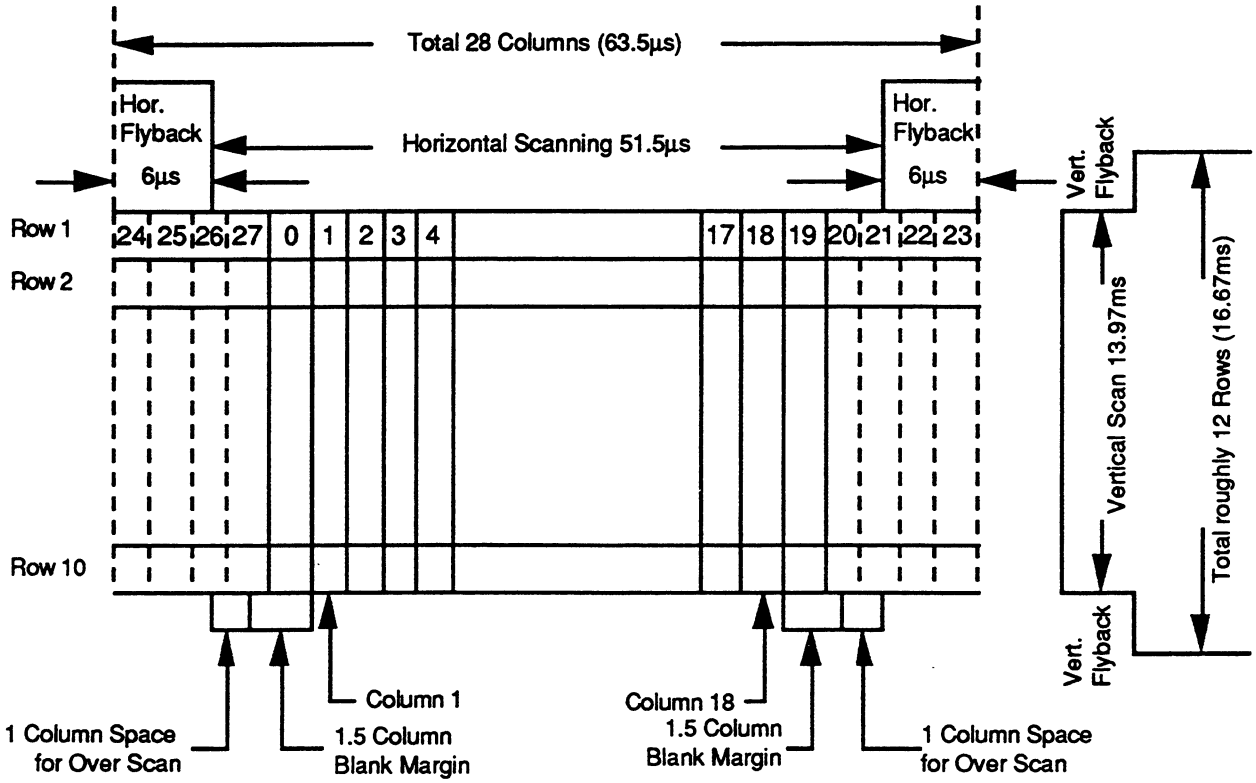
For the display of one row only, the CPU can leave the registers unchanged all the way.

For two adjacent rows of characters/graphic symbols to be displayed, the CPU must update the buffer within a period of one row display. This period is 1397µs (63.5 x 22 = 1397) for NTSC

system and 1664µs (64 x 26 = 1664) for PAL system. This software constraint must also be satisfied for a full screen display (10 rows).

Basically, the task of transferring data to 24 OSD registers consumes roughly 400µs.

6.5 FULL SCREEN DISPLAY TIMING



Remark:
1. 1 Row = 22 Scan Lines
2. Row 1 starts at 24 horizontal lines (1524µs) after the rising edge of the vertical flyback pulse

Figure 6-3 Full Screen (NTSC) Timing Diagram

The timing of the full screen OSD on NTSC system is illustrated in Figure 6-3. The vertical position of the 18 by 10 display area is hardwired so that Row 1 always starts at 24 horizontal lines (24x63.5 = 1524µs) after the rising edge of the vertical flyback pulse. The horizontal position of the display area is software programmable by a half column steps ranging from 0 to 31 steps. For more details about the programmable horizontal delay, see **SECTION 6.9.3 & 6.9.7.**

6.6 PROGRAMMABLE COLOR WINDOW (BACKGROUND)

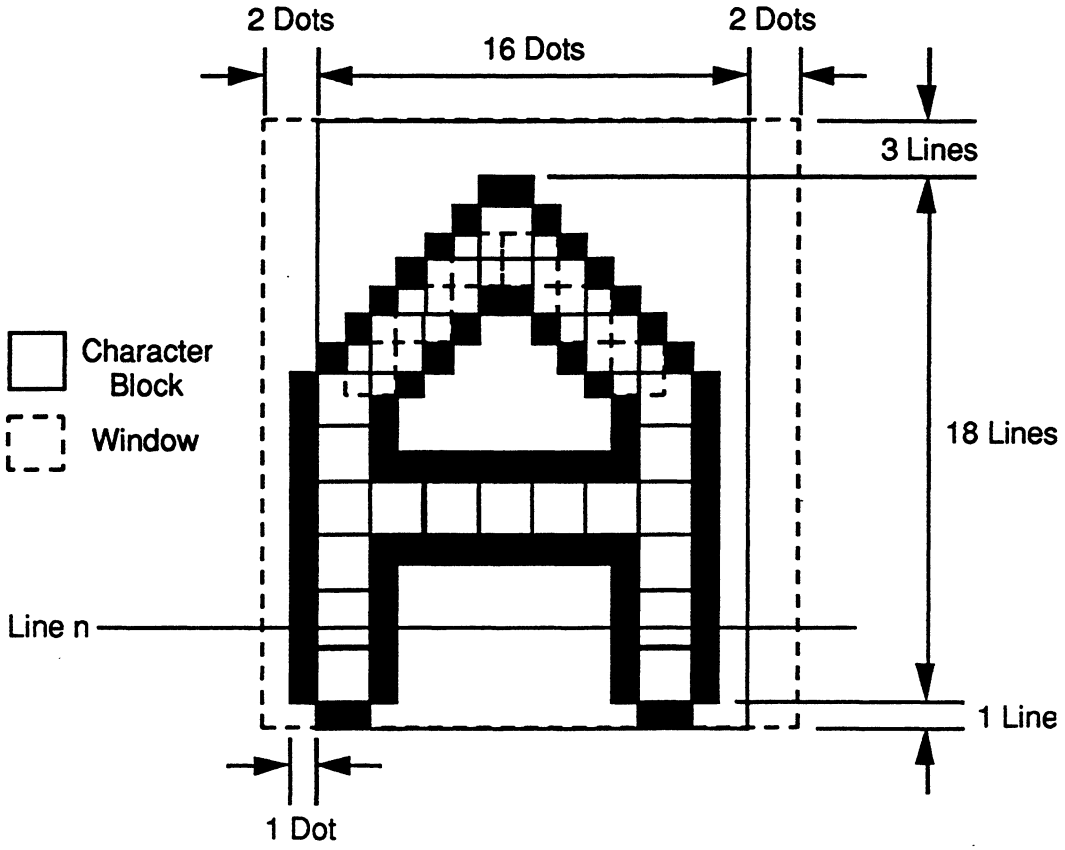


Figure 6-4A Black Edge Surrounded Character with Window Diagram (for NTSC System)

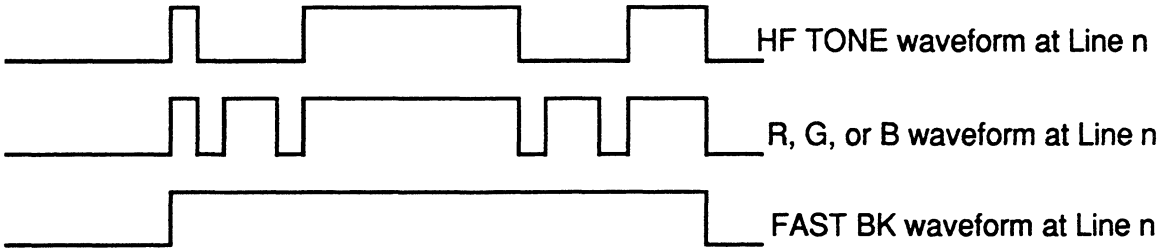


Figure 6-4B OSD Output Signal Waveforms

The timing diagram of window (background) for NTSC is illustrated in Figure 6-4A. This programmable window feature gives an easy way to highlight display characters/graphical symbols.

1. Multi-window. Each row of display can have one independent window. Combining them, a big window resulted.
2. Variable size. The start and stop columns are programmable by software. See **SECTION 6.9.5 & 6.9.6** for more details.
3. Color selectable. The color of window is selectable out of 8 colors (including black and white) by software. See **SECTION 6.9.6** for more details.

6.7 CHARACTER PRESENTATION

The crude dot matrix data stored in Character ROM undergoes several operations before come to the final display outlook. The following paragraphs describes these operations on a 525 system separately.

6.7.1 Character ROM

There is an internal Character ROM of totally 64 characters/graphical symbols. Each of them is composed of an 8 dots (width) by 14 dots (height) dot matrix. For the ease of addressing, each character is considered as 16 bytes data instead of 14 bytes. Hence, there are totally 16x64=1024 bytes situated at address locations $0400 to $07FF of the memory (see Figure 2-2 for the memory map).

An example for 525 system is shown in Figure 6-5. Basically, for a 525 system, only the Line0 to Line10 (11 lines) are displayable and the Line11 is used as a look ahead line (see **SECTION 6.7.3** for definition of look ahead line). For a 625 system, Line 0 to Line12 (13 lines) are displayable and the Line13 is look ahead line.

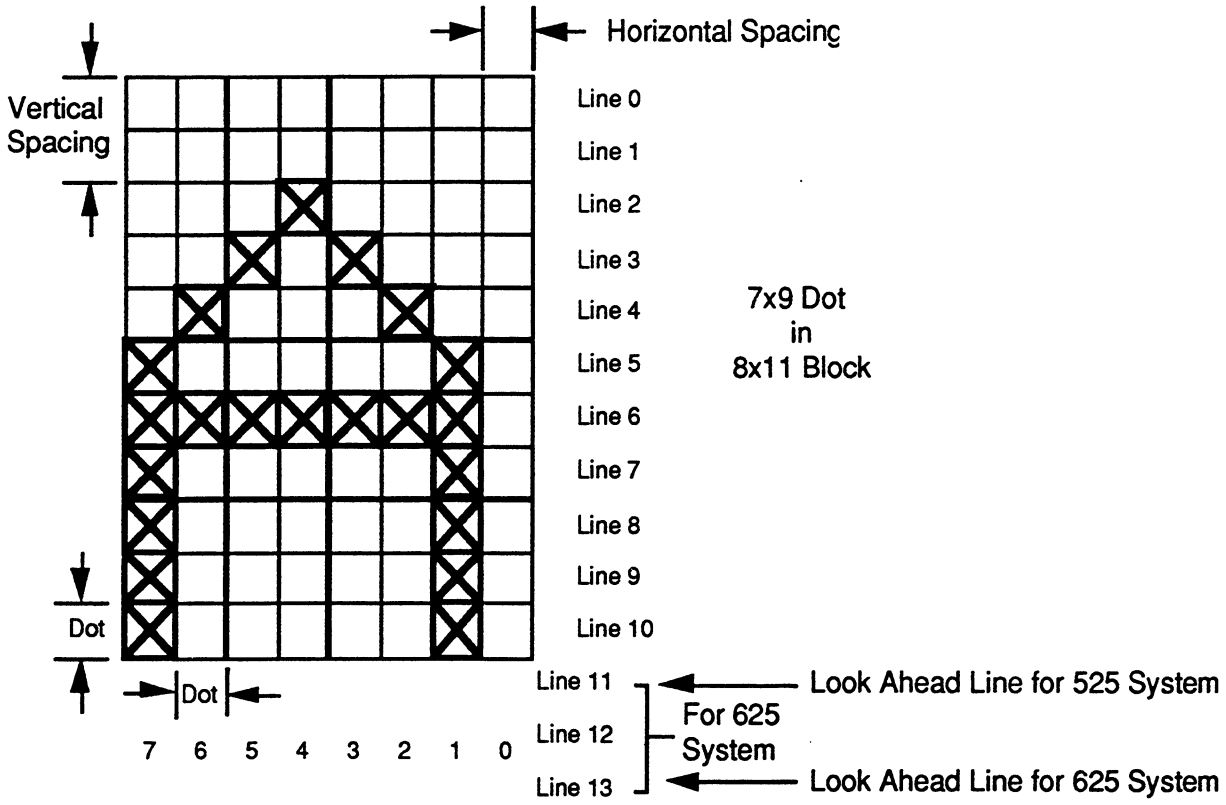


Figure 6-5 Character Dot Matrix and Row Spacing (NTSC System)

For vertically/horizontally continuous characters, the full character block and the look ahead line should be used. However, for a normal characters, the vertical/horizontal spacing is determined by the way the Character ROM is programmed. As shown in Figure 6-5, vertical space of 2/11 and horizontal space of 1/8 resulted.

6.7.2 Half Dot Shifting

Half dot shifting is a hardware mechanism to round a dot matrix to roughly double resolution. This gives a much better outlook. As shown in Figure 6-6, half dot shifting converts a 7x9 dot matrix to a 14x18 dot matrix by adding those dotted line squares.

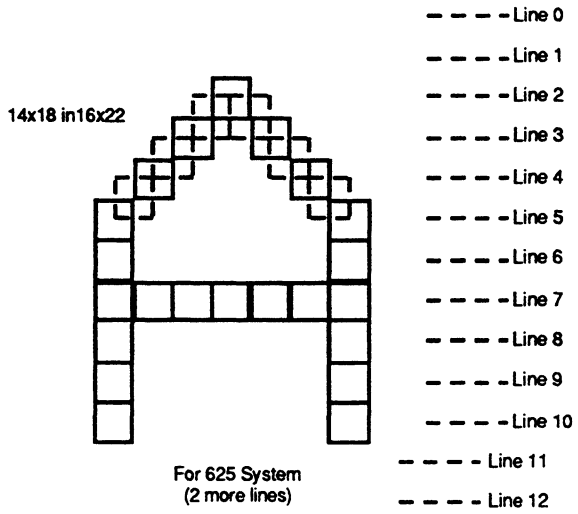


Figure 6-6 Half Dot Shifting

6.7.3 Look Ahead Line

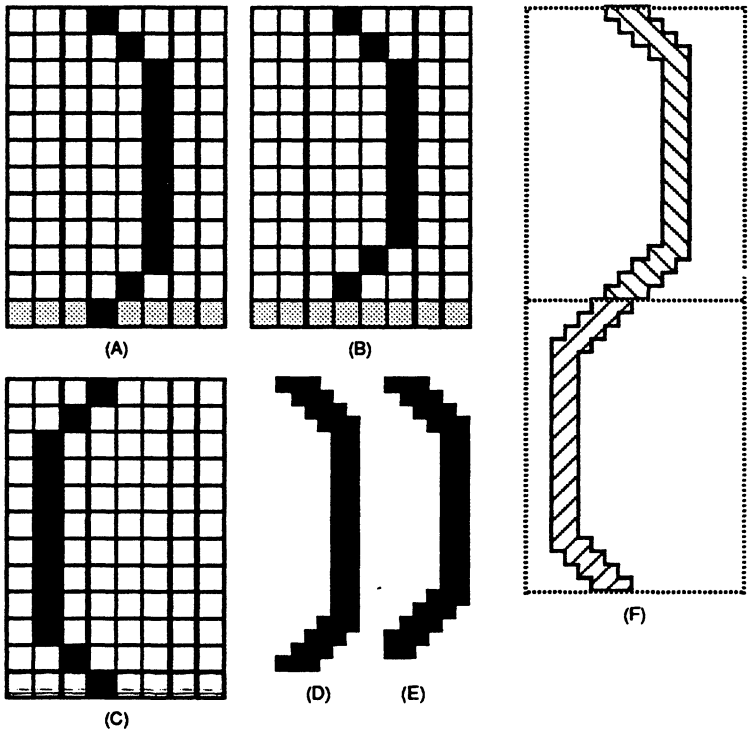


Figure 6-7 Use of Look Ahead Line

The basic purpose of using look ahead line is to ensure the vertical continuity of graphical symbols.

Due to the algorithm of the half-dot shifting, all graphical symbols displayed on TV will be shifted upward by one line (out of 22 lines). Then, the base of the graphical symbols cannot have oblique lines and continuity in vertical direction. These can be remedied by look ahead line.

Figure 6-7A & Figure 6-7B show a graphical symbol with and without look ahead lines respectively. As Figure 6-7A is supposed to be displayed on top of Figure 6-7C, the look ahead line of it is actually the replica of the uppermost line of Figure 6-7C. After half dot shifting, Figure 6-7A becomes Figure 6-7D which exhibit vertical continuity with Figure 6-7C as shown in Figure 6-7F. On the other hand, Figure 6-7B is half dot shifted to become Figure 6-7E which has 1 line spacing at the bottom.

6.7.4 Black Edge Surround

The 4-sided black edge (or black edge surround) character is illustrated in Figure 6-4A. The black edge is 1 dot thick (out of 16 dots) horizontally and 1 line thick (out of 22 lines) vertically. For a double width character, the horizontal thickness of black edge is also doubled. Similarly, the vertical thickness of black edge of a double height character is also doubled.

The black edge feature is a software programmable. It can be switched on/off by a software bit, BEEN of Color & Status Register. See **SECTION 6.9.2** for more details.

6.7.5 Timing of Variable Size Characters

There are totally four possible sizes of characters/graphical symbols; standard height/standard width, standard height/double width, double height/standard width, and double height/double width. These sizes are software selectable on a row by row basis (see **SECTION 6.9.4** for more details). The timing for them are illustrated in Figure 6-8 and Table 6-4.

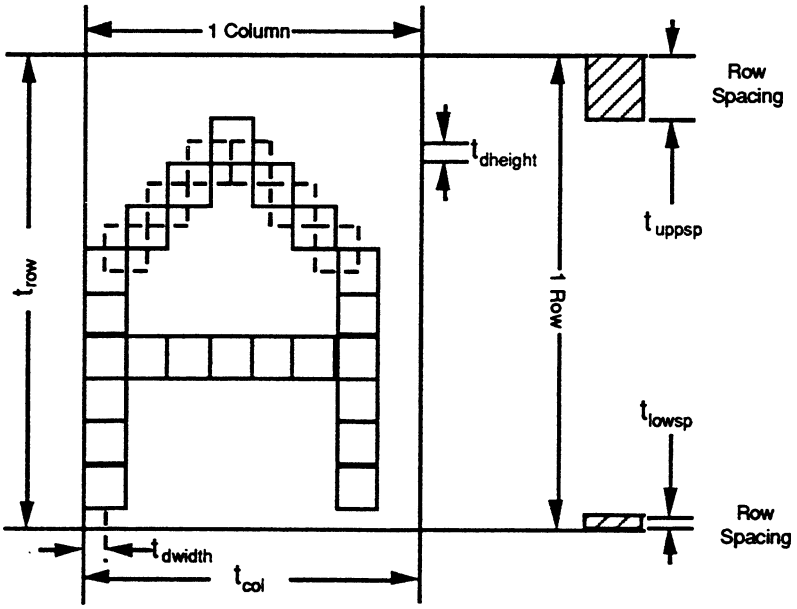


Figure 6-8 Character Timing

Table 6-4 Character Timing of Variable Sizes

	Standared W Standard H	Standard W Double H	Double W Standard H	Double W Double H
t_{dwidth}	t_{dot}	t_{dot}	$2xt_{dot}$	$2xt_{dot}$
t_{col}	$16xt_{dot}$	$16xt_{dot}$	$32xt_{dot}$	$32xt_{dot}$
$t_{dheight}$	t_{hor}	$2xt_{hor}$	t_{hor}	$2xt_{hor}$
t_{uppsp}	$3xt_{hor}$	$6xt_{hor}$	$3xt_{hor}$	$6xt_{hor}$
t_{lowsp}	t_{hor}	$2xt_{hor}$	t_{hor}	$2xt_{hor}$
t_{row}	$22xt_{hor}$	$44xt_{hor}$	$22xt_{hor}$	$44xt_{hor}$

t_{hor} : Period of Horizontal Scan Line (63.5 µs for NTSC)

t_{dot} : Period of Dot Clock (0.14 µs for NTSC)

6.8 OSD OUTPUTS

There are totally five output signals from OSD, i.e., R, G, B, HF TONE and FAST BK. The R, G, B are digital signal with programmable polarity for driving the R, G, B inputs of the video amplifier. By defaults, they are active high signal.

The FAST BK signal is an active high signal which is used to drive the fast blanking input of the video amplifier to mute the original display video in order to obtain the result of character overlay on the analog video signal. See Figure 6-4B.

The HF TONE signal is an active high signal which is used to half the gain of an external op-amp for R, G, B so that the window color (background) gives half intensity. See Figure 6-4B.

6.9 REGISTERS

There are totally 24 OSD control status registers. All of them are buffered for the sake of single row architecture so that when the current row of characters/graphic symbols is being displayed, the data of the next row can be updated at the same time.

6.9.1 Character Registers

There are totally 18 one-byte character registers ($20-$31)

7	6	5	4	3	2	1	0	
nCS2	nCS1	nCHD5	nCHD4	nCHD3	nCHD2	nCHD1	nCHD0	$2n

B7-B6 Character color select bits

B7	B6	
0	0	Color 0
0	1	Color 1
1	0	Color 2
1	1	Color 3

B5-B0 Character address bits
They are used to select one of the 64 characters in character ROM

6.9.2 Color & Status Register

WRITE mode

7	6	5	4	3	2	1	0	
BEEN	-	R2	G2	B2	R1	G1	B1	$32

B7, BEEN Black edge enable bit
This bit is used to enable the black edge feature when set and disable it when clear.
B5-B3 Color 2 select bits
These bits are used to select the color 2 out of 8 possible colors
B2-B0 Color 1 select bits
These bits are used to select the color 1 out of 8 possible colors

READ mode

7	6	5	4	3	2	1	0	
1	SFG3	IFL	VERT	HOR	MODE	SFG2	SFG1	$32

B6, SFG3 System Flag 3
B5, IFL Interrupt Status bit
When the character row address in ROW ADDR/CHAR SIZE Register (location $34) is matched with character row running counter, this flag is set. If the interrupt enable bit of OSD, OIEN, in ROW ADDR/CHAR SIZE Register (location $34), is set, an OSD interrupt will be generated, and this flag will be cleared automatically by hardware once the OSD interrupt is serviced.
B4, VERT Vertical flyback status bit
This bit reflects the status of the vertical flyback input signal
B3 Horizontal flyback status bit
This bit reflects the status of the horizontal flyback input signal
B2, MODE TV mode bit
This bit specifies the frequency of the horizontal flyback pulse input signal (f_H).
When it is clear, f_H is lower than f_{ref}.
When it is set, f_H lies between fref and $2xf_{ref}$.
The reference frequency, f_{ref}, is calculated by dividing the external crystal frequency by 192. E.g., for a 4 MHz crystal operation, f_{ref} = 20.83 KHz.
B1, SFG2 System Flag 2
This bit is actually the AND-product of MODE and the complement of SFG3. Together with SFG1, it can specify the type of TV system,

B1	B0	Vert. field freq./ Hz	Hor. freq./KHz
0	0	60	15.75
0	1	50	15.625
1	0	120	31.5
1	1	100	31.25

B0, SFG1 System Flag 1
System flag 1 & 3 specify the number of horizontal scan lines per field.

B6	B0	No. of horizontal scan lines per field
0	0	192-288
0	1	288-416
1	0	416-576
1	1	576-704

6.9.3 Color 3/4 Register

7	6	5	4	3	2	1	0	
H31D4	H15D4	R4	G4	B4	R3	G3	B3	$33

B7, H31D4 Horizontal delay bit 4 for 31 KHz mode (See **SECTION 6.9.7** for more details)
B6, H15D4 Horizontal delay bit 4 for 15 KHz mode (See **SECTION 6.9.7** for more details)
B5-B3 Color 4 select bits
B2-B0 Color 3 select bits

6.9.4 Row Address/Character Size Register

7	6	5	4	3	2	1	0	
CHWS	CHHS	RGBINV	OIEN	RWA3	RWA2	RWA1	RWA0	$34

B7, CHWS Character width select bit
When it is set, the whole row of characters/graphic symbols are of double width.
When it is clear, the whole row of characters/graphic symbols are of standard width.
B6, CHHS Character height select bit
When it is set, the whole row of characters/graphic symbols are of double height.
When it is clear, the whole row of characters/graphic symbols are of standard height.
B5, RGBINV RGB inversion bit
This bit is used to invert the polarity of the output R, G, B signals when set.
B4, OIEN OSD interrupt enable bit
When it is set, the OSD interrupt is enabled. Then if the OSD internal row counter address is matched with the row address bits(described later in this section), an OSD interrupt will be generated.
B3-B0 Row address bits
They specify any one row of characters/graphic symbols (row number 1 to 10 continuously) up to a total of 10 rows/field displayed on the TV screen from top to bottom.
In case one row of them is of double height, only 9 rows can be displayed and they are still addressed consecutively from 1 to 9.

6.9.5 Window/Column Register

7	6	5	4	3	2	1	0	
WINE	OSDE	PLLEN	BCS4	BCS3	BCS2	BCS1	BCS0	$35

B7, WINE Window enable bit
When it is set, the window (background) feature is enabled.

B6, OSDE OSD enable bit
When it is set, the OSD output signal are enabled. When it is cleared, the OSD output signal become high impedance.

B5, PLLEN Phase Lock Loop enable bit
When it is clear, the PLL is switched off to save power.

B4-B0 Window start column bits
They specify the column number of the start (leftmost) column of the window (background) ranging from column 1 to column 18.

6.9.6 Column/Color Register

7	6	5	4	3	2	1	0	
R	G	B	BCSP4	BCSP3	BCSP2	BCSP1	BCSP0	$36

B7-B5 Window color select bits
They select the color of the window (background) out of the 8 possible colors.

B4-B0 Window stop column bits
They specify the column number of the stop (rightmost) column of the window (background) ranging from column 1 to column 18.
In case the window stop column number is smaller than window start column number, an erroneous display resulted.

6.9.7 Horizontal Position Delay Register

7	6	5	4	3	2	1	0	
H31D3	H31D2	H31D1	H31D0	H15D3	H15D2	H15D1	H15D0	$37

B7-B4 Horizontal delay bits for 31 KHz operation
B3-B0 Horizontal delay bits for 15 KHz operation

15KHz Operation

The 4 low order bits of Horizontal Position Delay Register (location $37) together with bit 6 of Color 3/4 Register (location $33), H15D0-H15D4, specify the horizontal positioning of total 32 steps of half column width.

31KHz Operation

The 4 high order bits of Horizontal Position Delay Register (location $37) together with bit 7 of Color 3/4 Register (location $33), H31D0-H31D4, specify the horizontal positioning of total 32 steps of half column width.

SECTION 7
PULSE LENGTH D/A CONVERTERS

7.1 6-BIT DAC

The pulse length D/A converter (PLM) system works in conjunction with the 16-bit free running Timer to implement eight channels of 6-bit D/A PLM conversion. It has eight data registers associated with it: DAC0-DAC7 located at $08-$0F respectively. Each one has the same bits structure as shown,

7	6	5	4	3	2	1	0	
0	0	0DA5	0DA4	0DA3	0DA2	0DA1	0DA0	$08

Bit 0-5, 0DA0 - 0DA5
6 data bits of DAC channel 0

A system block diagram of the PLM system is shown in Figure 7-1.

This is a 6-bit resolution D/A converter. The output is pulse length modulated signal whose duty cycle may be modified. These signals can be used directly as PLMs, or filtered average value for general purpose analog outputs. Some examples of PLM output waveforms are shown in Figure 7-2.

The repetition rate speed is 256 times the Programmable Timer clock period (the repetition rate frequency for a 4.00 MHz crystal is 1953 Hz). A value of $00 loaded into these registers results in a continuously low output on the corresponding D/A output pin. A value of $20 results in a 50% duty cycle output, and so on, to the maximum value, $3F which corresponds to an output which is at state "1" for 63/64 of the cycle. When the MCU makes a write to register DAC n, the new value will only be picked up by the D/A converter n at the end of a complete cycle of conversion. This results in a monotonous change of the DC component of the output without overshoots or vicious starts (a vicious start is an output which gives totally erroneous PLM during the initial period following an update of the PLM D/A register). This feature is achieved by double buffering of the PLM D/A registers.

Note: Since the PLM system uses the timer counter, PLM results will be affected while resetting the timer counter.

WAIT mode does not affect the output waveform of the D/A converters.

The eight D/A registers are reset to $00 during power-on or external reset.

There is a 15-20 ns gate delay between each DAC to avoid simultaneously starting on 8 DACs.

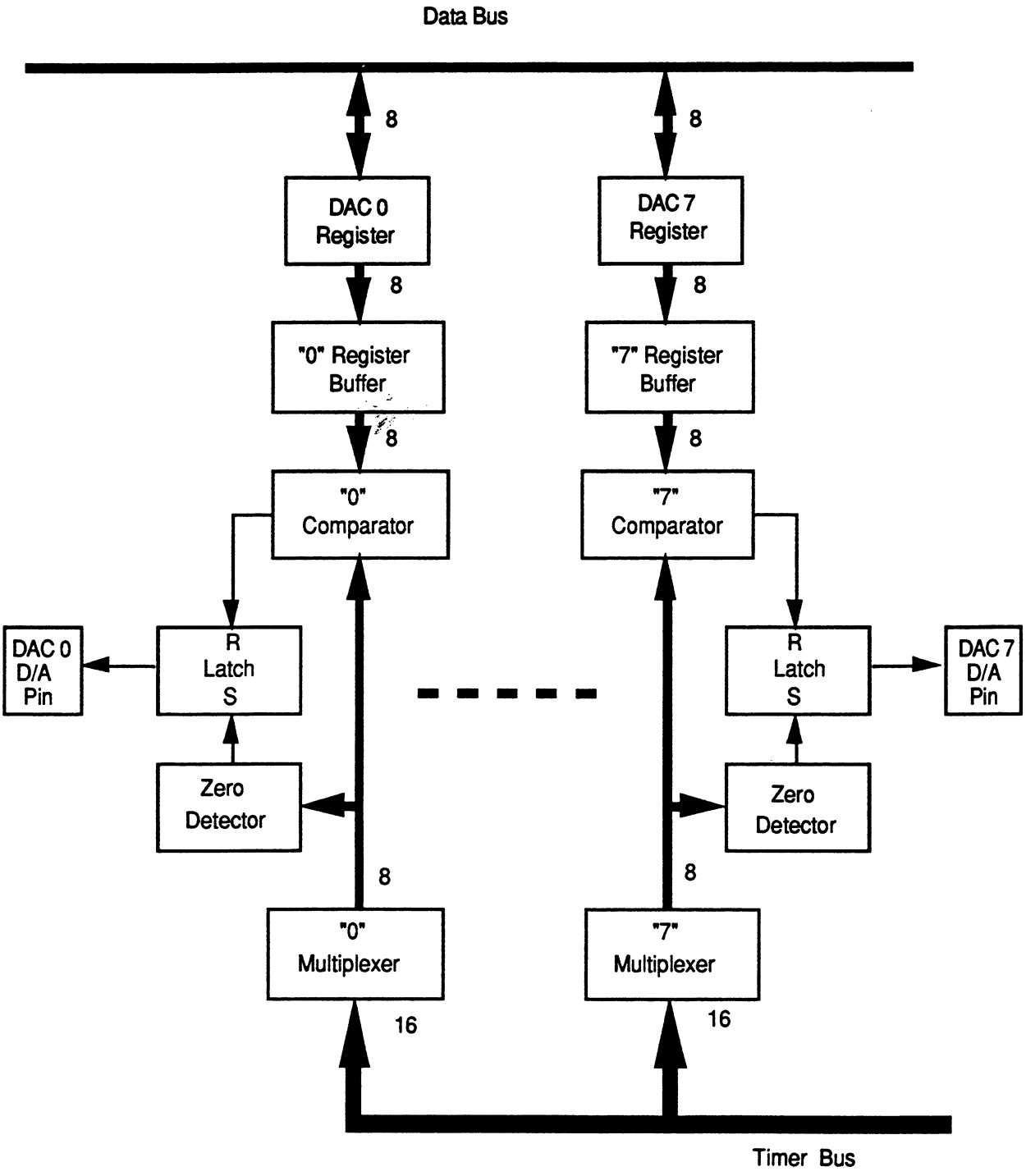
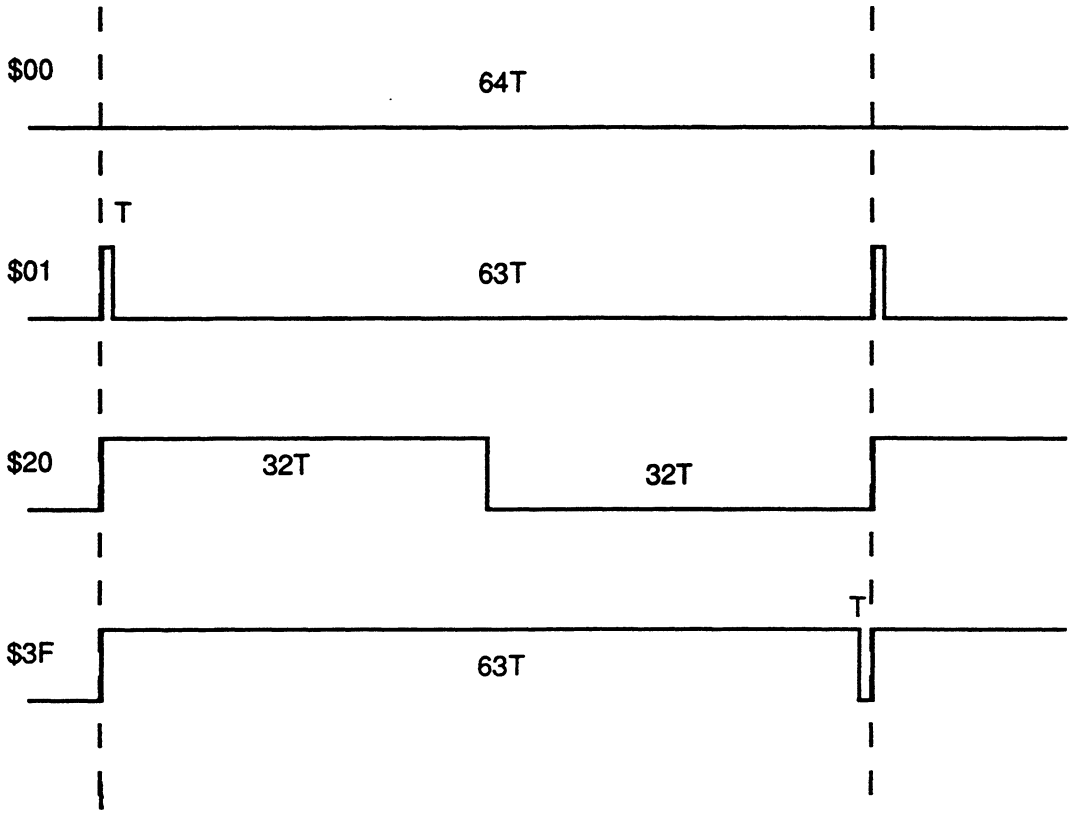


Figure 7-1 PLM System Block Diagram



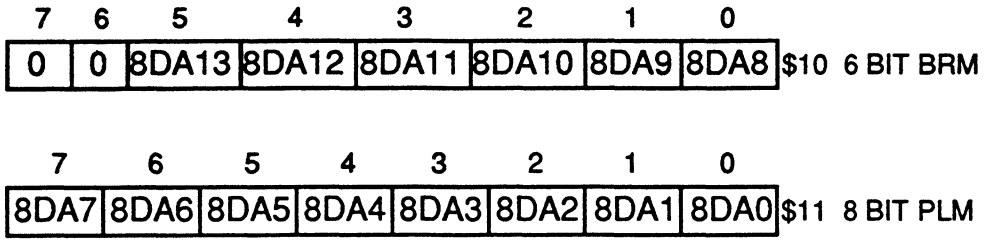
T = 16 CPU Clock = 8.0 us

$$\frac{1}{64T} = 1953Hz$$

Figure 7-2 6-Bit PLM Output Waveform Examples

7.2 14-BIT DAC

The 14-bit DAC in HC05T10 is composed of one 8-bit PLM in high order byte and a 6-bit binary rate multiplier (BRM) in low order byte. The value programmed in the 8-bit high order register will determine the pulse length of the output. The input to the 8-bit PLM is a 2 MHz clock and the repetition rate of the output is about 7.8 KHz. The 6-bit BRM of narrow pulses which are equally distributed among 6 64-PLM-CYCLE frame. The total expective duty cycle at the output will be (M+N/64)/256, where M is the value programmed in 8-bit high order register and N is the value in 6 bit low order register.

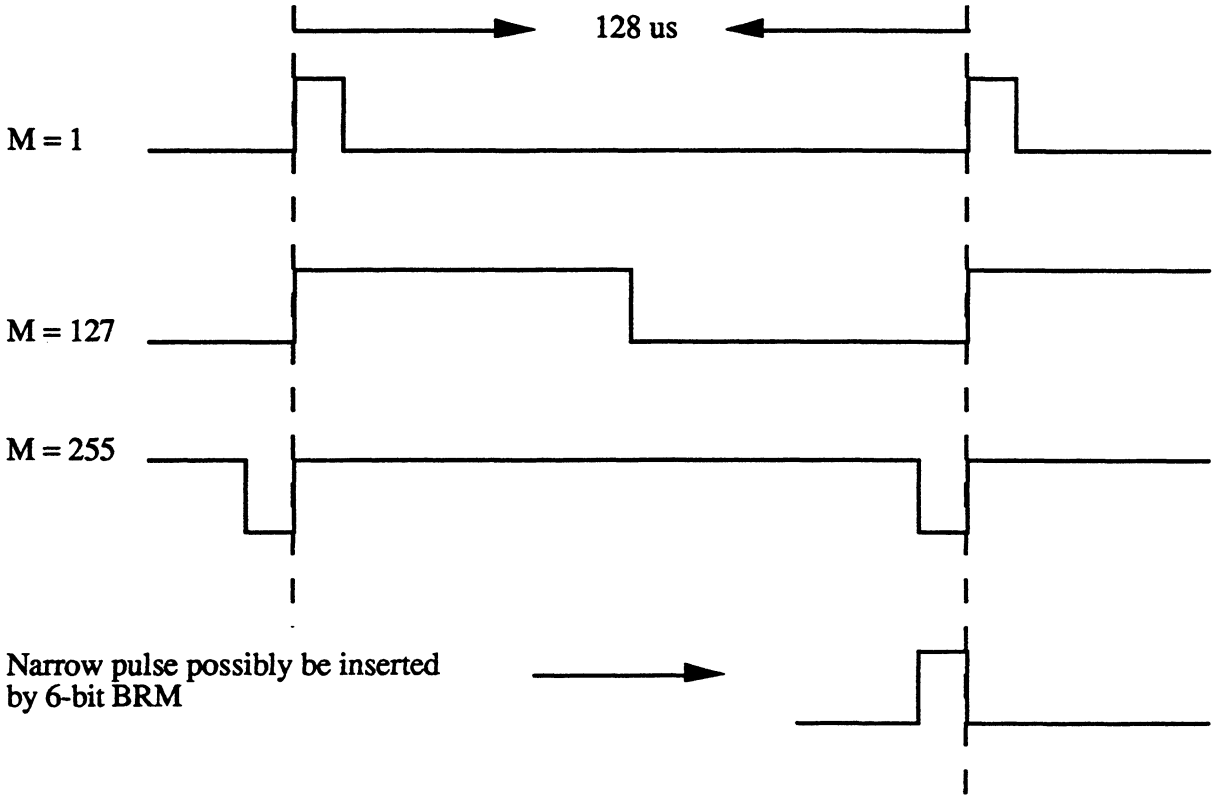


In order to prevent transient noise at the 14-bit DAC output during MPU write to the 8-bit PLM and 6-bit BRM registers, double buffering is used. Programming of the 2 registers must follow the sequence as shown below:

```
LDA BRM_VALUE
STA $10 (data put in 6-bit BRM buffer)

LDA PLM_VALUE
STA $11 (load 6-bit BRM and 8-bit PLM register)
```

The instruction STA $10 simply puts the 6-bit BRM data in buffer. Output is not affected at this time. The instruction STA $11, then, puts the total 14-bit data to BRM and PLM registers at the same time. Output wave form will change accordingly starting from the beginning of the coming PLM cycle.



N	PLM CYCLES TO WHICH NARROW PULSE ARE INSERTED IN A 64-CYCLE FRAME
00XXXXX1	32
00XXXX1X	16 , 48
00XXX1XX	8 , 24 , 40 , 56
00XX1XXX	4 , 12 , 20 , 28 , 36 , 44 , 52 , 60
00X1XXXX	2 , 6 , 10 , 14 , 18 , 22 , 26 , 30 34 , 38 , 42 , 46 , 50 , 54 , 58 , 62
001XXXXX	1 , 3 , 5 , 7 , ... 61 , 63

Figure 7-3 14-bit PLM D/A Output Waveform Examples

SECTION 8
A/D CONVERTER

8.1 A/D INTRODUCTION

The Analog to Digital converter system consists of a single 8-bit successive approximation converter. There is one 8-bit result data register (address $1E) and one 8-bit status/control register (address $1F).

The reference supply for the converter uses two dedicated pins rather than being driven by the system power supply lines, because the voltage drops in the bonding wires of those heavily loaded pins would degrade the accuracy of the A/D conversion.

The analog signal input is at PD0.

The ADON-bit allows the user to disconnect the A/D when not used, in order to save power. This is particularly useful to reduce consumption when going into the WAIT mode.

The A/D is ratiometric and two dedicated pins supply the reference voltage (V_{RH} and V_{RL}). An input voltage equal to or greater than V_{RH} converts to $FF (full scale) with no overflow indication (if greater). An input voltage equal to V_{RL} converts to $00. For ratiometric conversions, the source of analog input should use V_{RH} as the supply voltage and be referenced to V_{RL}.

8.2 A/D PRINCIPLE

The A/D reference inputs are applied to precision internal digital to analog converter. Control logic drives this D/A and the analog output is successively compared to analog input which was sampled at the beginning of the conversion time. The conversion is monotonous with no missing codes.

8.3 A/D OPERATION

The MC68HC05T10 has an 8-bit S.A.R. type A/D converter, with continuous conversion. The result of a conversion is loaded into the read-only Result Data Register ($1E), and a conversion complete flag COCO is set in the A/D status/control register($1F).

Any write to the A/D status/control register will abort the current conversion, reset the conversion complete flag and start a new conversion.

At power-on or external reset, the ADON bit will be cleared. Thus the A/D is disabled.

8.4 A/D STATUS/CONTROL REGISTER

7	6	5	4	3	2	1	0	
0	0	0	0	0	0	COCO	ADON	$1F

B1, COCO — The COCO flag is set each time a conversion is complete, allowing the new result to be read from the A/D register ($1E). The converter immediately starts a new conversion. COCO is cleared by reading the data register or writing to the status/control register, or by power-on or external reset.

B0, ADON — The ADON-bit allows the user to enable/disable the A/D converter. When the ADON-bit is cleared the A/D is disabled. When the ADON-bit is set the A/D is enabled.

When the A/D converter is turned on, it takes a time t_{ADON} (less than 1ms) for the current sources to stabilize. During that t_{ADON} time, A/D conversions result may be inaccurate. Power-on or external reset will clear the ADON-bit.

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 : Cleared

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 $0A, which is read only, serial communications status register ($10), timer status register ($13), and timer input capture register ($14-$15). 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 MC68HC05T10 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 initialize 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 specifications and associated timing information for the MC68HC05T10.

10.2 MAXIMUM RATINGS
(Voltages Referenced to VSS)

Ratings	Symbol	Value	Unit
Supply Voltage	V_{DD}	-0.5 to +7.0	V
Input Voltage	V_{in}	V_{SS}-0.5 to V_{DD} +0.5	V
Self-Check Mode ($\overline{IRQ}$ Pin Only)	V_{in}	V_{SS}-0.5 to 2xV_{DD} + 0.5	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 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 V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{DD}$. Reliability of operation is enhanced if unused input are connected to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

10.3 DC ELECTRICAL CHARACTERISTICS

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

Characteristics	Symbol	Min	Typ	Max	Unit
Output Voltage, Iload ≤ 10.0 uA	V_{OL} V_{OH}	- V_{DD} - 0.1	- -	0.1 -	V
Output High Voltage (Iload =0.8 mA) PA0-PA7, PB0-PB7,PC0-PC7, TCMP, R, G, B, FAST BK, HF TONE	V_{OH}	V_{DD} - 0.8	-	-	V
Output Low Voltage (Iload = 1.6 mA) PA0-PA7, PB0-PB7,PC0-PC7, TCMP, R, G, B, FAST BK, HF TONE, SDA, SCL	V_{OL}	-	-	0.4	V
Input High Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD3, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1,SDA, SCL, VFLB, HFLB	V_{IH}	0.7xV_{DD}	-	V_{DD}	V
Input Low Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD3, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1,SDA, SCL, VFLB, HFLB	V_{IL}	V_{SS} V_{SS} V_{SS}	- - -	0.2x V_{DD} 0.2x V_{DD} 0.3xV_{DD}	V
Supply Current Run Wait	I_{DD}	- -	6.0 1.5	8.0 2.0	mA
I/O Ports Hi-Z Leakage Current PA0-PA7, PB0-PB7, PC0-PC7,PD0-PD3, RESET, SDA,	I_{IL}	-	-	±10	uA
Input Current TCAP, $\overline{IRQ}$, $\overline{RESET}$, PD0, OSC1	I_{in}	-	-	±1	uA
Capacitance Ports (Input or Output), $\overline{RESET}$, SDA, SCL, $\overline{IRQ}$, TCAP, OSC1, PD0	C_{out} C_{in}	- -	- -	12 12	pF

NOTES : 1. OSC1 is a square wave with V_{IL} = 0.2V, V_{IH} = V_{DD} - 0.2V, f_{OSC} = 2 MHz, 45-55% duty cycle, C_L = 20 pF on OSC2, no load on TCMP.

2. I_{DD} : all ports configured as inputs V_{IL} = 0.2V, V_{IH} = V_{DD} - 0.2V.

3. Wait I_{DD} is affected linearly with OSC2 capacitance.

10.4 A/D CONVERTER CHARACTERISTICS

(V_{DD} = 5.0 V_{dc} ± 10%, V_{SS} = 0 V_{dc})

Characteristic	Parameter	Min	Max	Unit
Resolution	Number of bits resolved by the A/D	8	-	Bit
Non-Linearity	Max deviation from the best straight line Through the A/D transfer characteristics (V_{RH} = V_{DD} and V_{RL} = 0V)	-	±1/2	LSB
Quantization Error	Uncertainly due to converter resolution	-	±1/2	LSB
Absolute Accuracy	Difference between the actual input voltage and the full -scale equivalent of the binary code output code for all errors	-	±1	LSB
Conversion Range	Analog input voltage range	V_{RL}	V_{RH}	V
V_{RH}	Maximum analog reference voltage	V_{RL}	V_{DD} + 0.1	V
V_{RL}	Minimum analog reference voltage	V_{SS} - 0.1	V_{RH}	V
Conversion Time	Total time to perform a single analog to digital conversion	-	32	T_{cyc}
Mononticity	Conversion result never decreases with an increase in input voltage and has no missing codes	GUARANTEED		
Zero Input Reading	Conversion result when V_{in} = V_{RL}	00	-	Hex
Full Scale Reading	Conversion result when V_{in} = V_{RH}	-	FF	Hex
Sample Acquisition Time (see Note)	Analog input acquisition sampling	-	12	T_{cyc}
Sample/Hold Capacitance	Input capacitance on PD0	-	12	pF
Input Leakage	Input leakage on A/D pin PD0, V_{RL}, V_{RH}	- -	10 1	µA µA

NOTES

The external system error caused by input leakage current is approximately equal to the product of R souce and input current.

NOTE : 1. The external system error caused by input leakage current is approximately equal to the product of R source and input current.

10.5 Open Drain PLM D/A Electrical Specifications*

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

Characteristics	Symbol	Minimum	Maximum	Unit
Leakage Current (Output ties to 8V)	I_{OL}	—	10	uA
DC Output low voltage (V_{DD} pull-up with1 kohm resistor)	V_{OLDC}	—	0.4	V
Output low voltage (V_{DD} pull-up with 1 kohm resistor)	V_{OL}	—	0.2 x V_{DD}	V
Output high voltage (V_{DD} pull-up with 1 kohm resistor)	V_{OH}	V_{DD} - 0.8	—	V

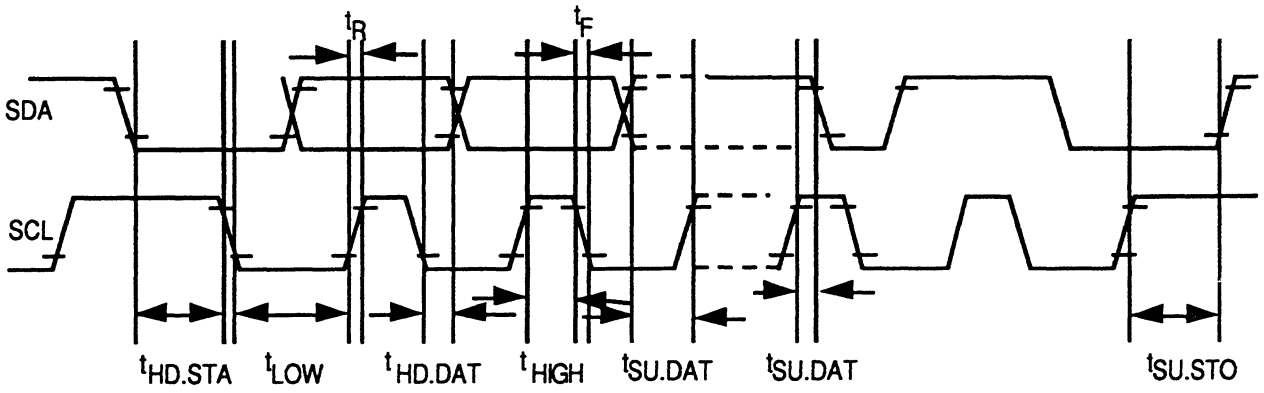
*Applies to DAC0 - DAC7n.b. DAC8 has the same electrical characteristics as other I/O ports (eg. port A)

10.6 M-BUS TIMING

10.6.1 M-Bus Interface Input Signal Timing

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

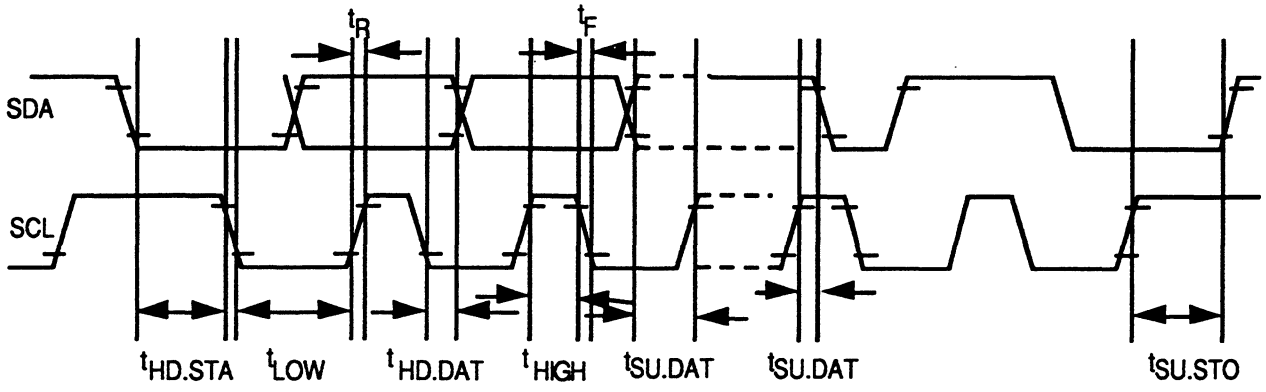
PARAMETER	SYMBOL	MIN	TYPICAL	MAX	UNIT
Start condition hold time	$t_{HD.STA}$	2	-	-	t_{cyc}
Clock low period	t_{LOW}	4.7	-	-	t_{cyc}
SDA/SCL rise time	t_R	-	-	1	µs
Data hold time	$t_{HD.DAT}$	0	-	-	t_{cyc}
SDA/SCL fall time	t_F	-	-	300	ns
Clock high period	t_{HIGH}	4	-	-	t_{cyc}
Data set up time	$t_{SU.DAT}$	250	-	-	ns
Start condition set up time (for repeated start condition only)	$t_{SU.STA}$	2	-	-	t_{cyc}
Stop condition set up time	$t_{SU.STO}$	2	-	-	t_{cyc}



10.6.2 M-Bus Interface Output Signal Timing

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

PARAMETER	SYMBOL	MIN	TYPICAL	MAX	UNIT
Start condition hold time	$t_{HD.STA}$	8	-	-	t_{cyc}
Clock low period	t_{LOW}	11	-	-	t_{cyc}
SDA/SCL rise time	t_R	-	-	1	µs
Data hold time	$t_{HD.DAT}$	0	-	-	t_{cyc}
SDA/SCL fall time	t_F	-	-	300	ns
Clock high period	t_{HIGH}	11	-	-	t_{cyc}
Data set up time	$t_{SU.DAT}$	t_{LOW} - t_{cyc}	-	-	µs
Start condition set up time (for repeated start condition only)	$t_{SU.STA}$	10	-	-	t_{cyc}
Stop condition set up time	$t_{SU.STO}$	10	-	-	t_{cyc}



10.7 CONTROL TIMING

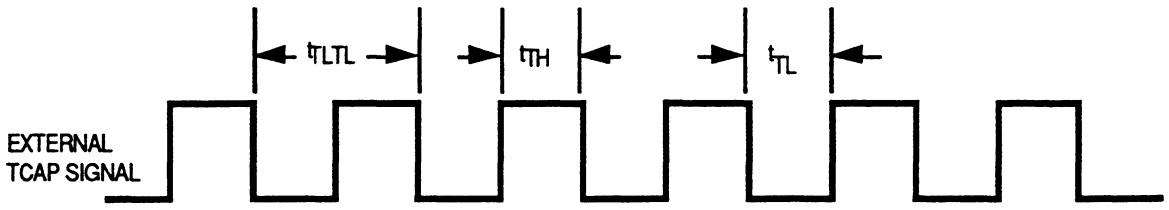
(V_{DD} = 5.0 V_{dc} ± 10 %, V_{SS} = 0 V_{dc}, $T_A = T_L$ to T_H)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Crystal Option External Clock Option	 fosc fosc	 3.0 3.0	 4.2 4.2	 MHz MHz
Internal Operation Frequency Crystal (fosc / 2) External Clock (fosc / 2)	 fop fop	 1.5 1.5	 2.1 2.1	 MHz MHz
Cycle Time	tcyc	480	-	ns
Crystal Oscillator Startup Time	t_{OXOV}	-	100	ms
External RESET Input Pulse Width	t_{RL}	1.5	-	tcyc
Timer Resolution ** Input Capture Pulse Width Input Capture Pulse Period	t_{RESL} t_{TH} t_{TL} t_{TLTL}	4.0 125 ***	- - -	tcyc ns tcyc
Interrupt Pulse Width	t_{ILIH}	125	-	ns
Interrupt Pulse Period	t_{ILIL}	*	-	tcyc
OSC1 Pulse Width	t_{OH} t_{OL}	90	-	ns

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

** Since a 2-bit prescaler in the timer must count for internal cycles (t_{cyc}) this is the limiting minimum factor in determining the timer resolution.

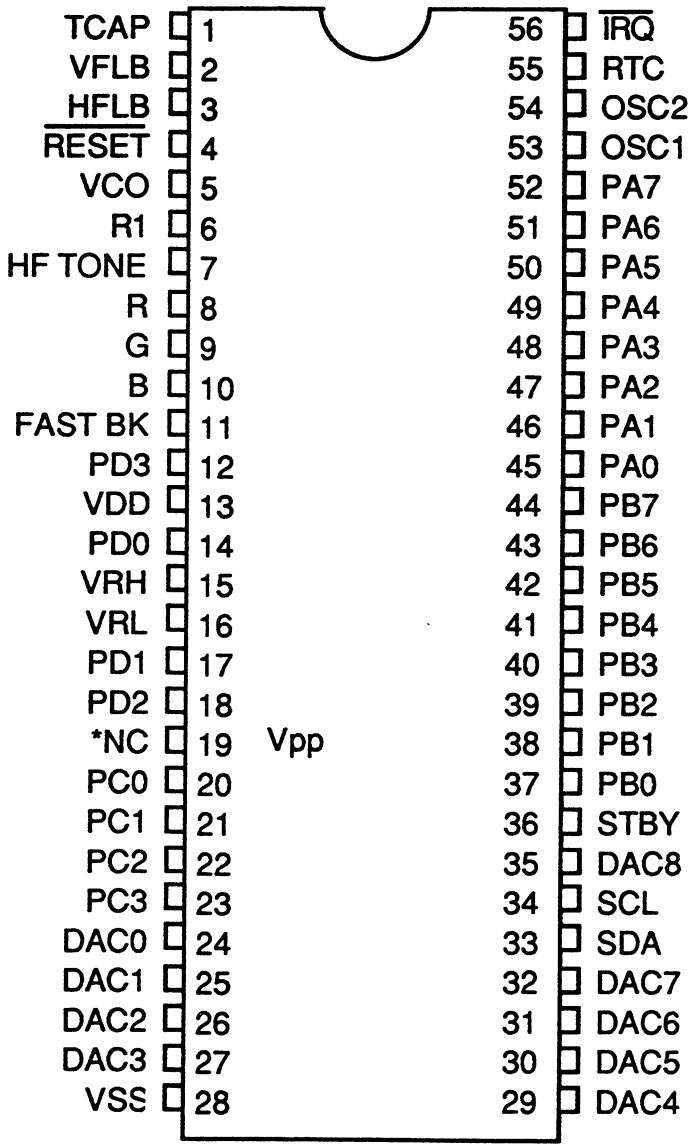
*** The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}.



SECTION 11
MECHANICAL DATA

This section contains the pin assignment and mechanical dimensions for the MC68HC05T10 microcomputer.

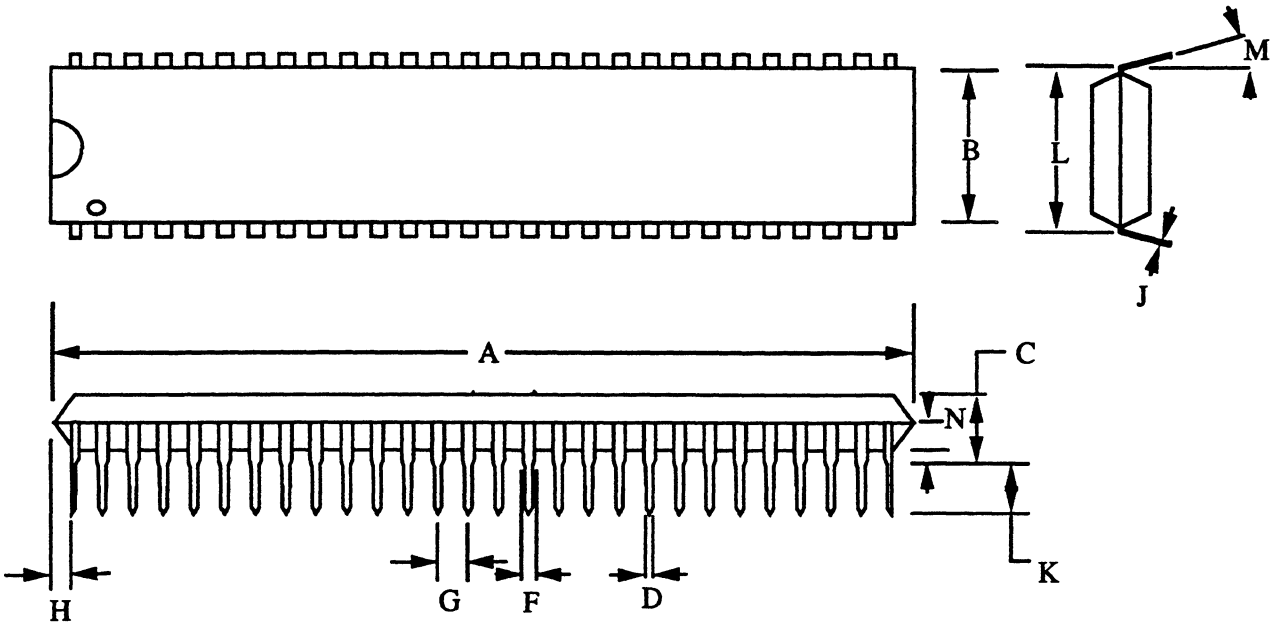
11.1 PIN ASSIGNMENT



* Reserved for Vpp on 705T10
Tied to VDD for user's applications

Figure 11-1 MC68HC05T10 Pin Assignment

11.2 Mechanical Dimensions



56 PIN SHRINK DIP

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	51.69	52.45	2.035	2.065
B	13.72	14.22	0.540	0.560
C	3.94	5.08	0.155	0.200
D	0.36	0.56	0.014	0.022
F	0.81	1.17	0.032	0.046
G	1.778 BSC		0.070BSC	
H	1.78	2.29	0.070	0.090
J	0.20	0.38	0.008	0.015
K	2.92	3.43	0.115	0.135
L	15.24 BSC		0.600 BSC	
M	O deg	15 deg	0 deg	15 deg
N	0.51	1.02	0.020	0.040

Figure 11-2 MC68HC05T10 Mechanical Dimensions

APPENDIX A
MC68HC05T7

The MC68HC05T7 microcomputer unit (MCU) device is a cost reduced version of the MC68HC05T10 MCU device. The main difference between MC68HC05T7 and MC68HC05T10 is that MC68HC05T7 has only 7904 bytes instead of 12000 bytes of user ROM as in MC68HC05T10. The entire data sheet of the MC68HC05T10 MCU device applies to the MC68HC05T7 MCU device with the exceptions provided in this appendix.

A.1 INTRODUCTION

Information contained in **SECTION 1 INTRODUCTION** of this document applies to the MC68HC05T7 MCU device except for the areas described in the following paragraphs.

A.1.1 Features

The features of the MC68HC05T7 MCU are the same as that of MC68HC05T10 MCU except that:

* 7904 bytes of user ROM.

A.1.2 Block Diagram

Figure A-1 illustrates the MC68HC05T7 MCU device block diagram.

A.2 FUNCTIONAL PIN DESCRIPTION, INPUT/OUTPUT PROGRAMMING, MEMORY, CPU REGISTERS, AND SELF-CHECK

Information contained in **SECTION 2 FUNCTIONAL PIN DESCRIPTION, INPUT/OUTPUT PROGRAMMING, MEMORY, CPU REGISTERS, AND SELF-CHECK** of this document applies to the MC68HC05T7 MCU device except for the areas described in the following paragraphs.

A.2.1 MEMORY

As shown in Figure A-2, the page zero 256 bytes of memory (page zero) include : 64 bytes of I/O features such as data ports, the DDRs, timer, M-Bus, on screen display, and 192 bytes of RAM exclusive of 64 byte stack. The balanced 128 bytes RAM of total 256 bytes user RAM reside in 1st page. The user ROM is 7936 bytes. The 16 highest bytes in user vector contain the user defined reset and the interrupt vectors.

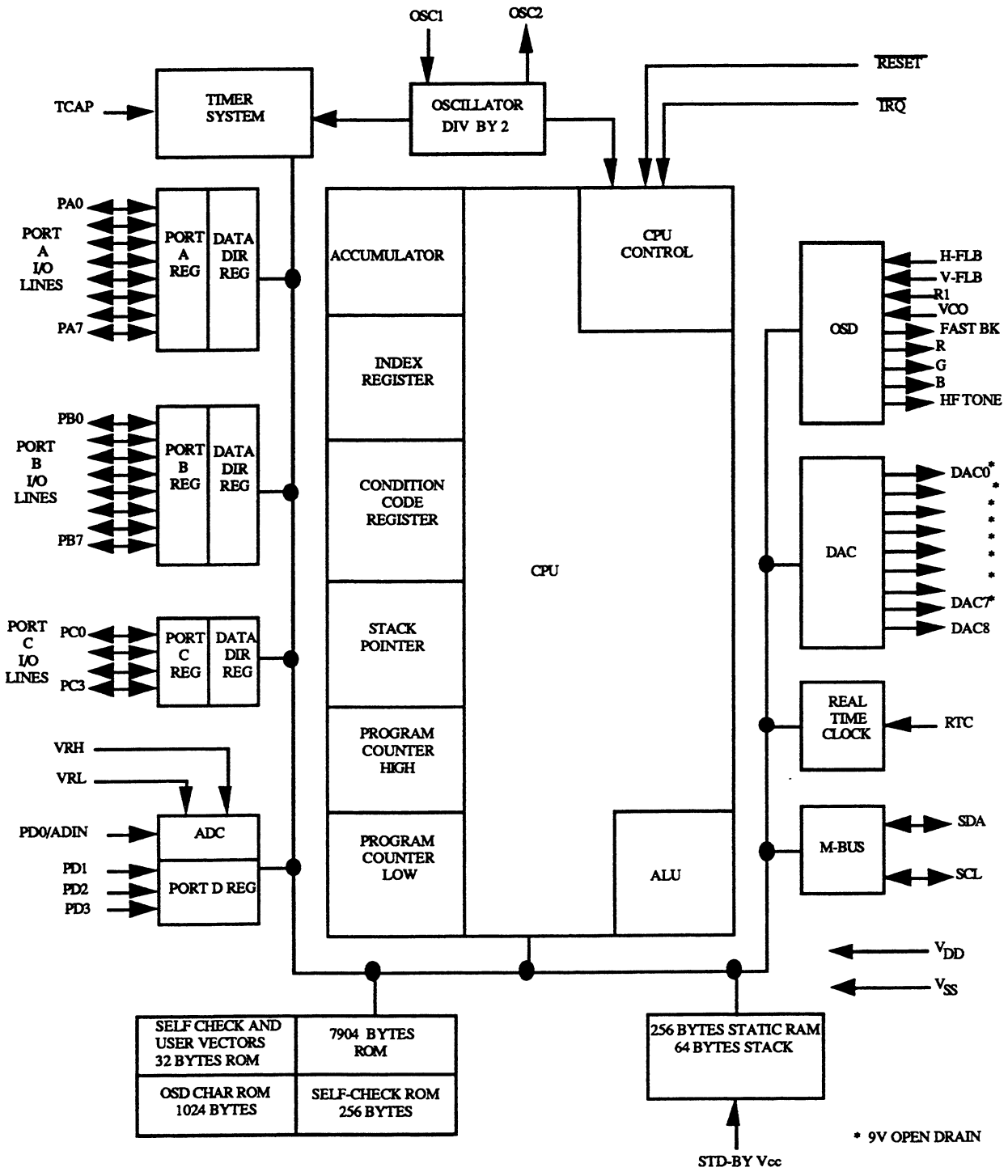


Figure A-1 MC68HC05T7 Microcomputer Block Diagram

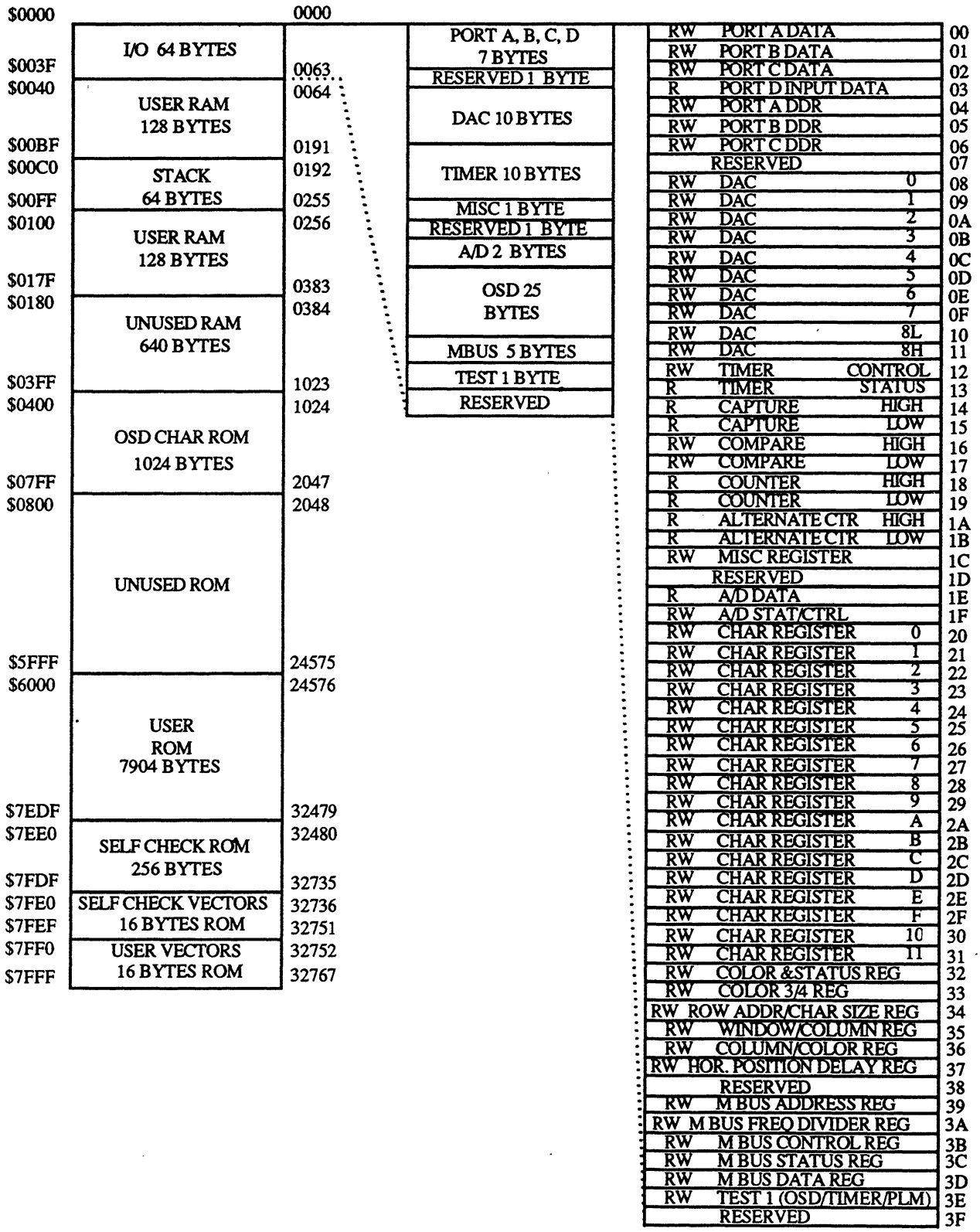


Figure A-2 Memory map of MC68HC05T7

APPENDIX B
MC68HC705T10

The MC68HC705T10 microcomputer unit (MCU) device is similar to the MC68HC05T10 MCU device with the exception of the EPROM feature. This feature of the MC68HC705T10 MCU enables the user to emulate the MC68HC05T10/T7 MCU device. Information pertaining to the EPROM emulation feature is contained in **APPENDIX C MC68HC05T10/T7 EMULATION**. The entire data sheet of the MC68HC05T10 MCU applies to the MC68HC705T10 MCU with the exceptions provided in this appendix.

B.1 INTRODUCTION

Information contained in **SECTION 1 INTRODUCTION** (general information, features, and block diagram) of this document applies to the MC68HC705T10 MCU device except for the areas described in the following paragraphs.

B.1.1 Features

* Emulation of MC68HC705T10/T7
* 12032 bytes EPROM
* On-chip Bootstrap Firmware for programming use
* Self-check Mode replaced by Bootstrap capability
* Security Mechanism for protection of EPROM content
* Ceramic SDIP package (UV erasable) or plastic SDIP package (one time programmable)

B.1.2 Block Diagram

Figure B-1 illustrates the MC68HC705T10 MCU device block diagram.

B.2 FUNCTIONAL PIN DESCRIPTION, INPUT/OUTPUT PROGRAMMING, MEMORY, CPU REGISTERS, AND SELF-CHECK

Information contained in **SECTION 2 FUNCTIONAL PIN DESCRIPTION, INPUT/OUTPUT PROGRAMMING, MEMORY, CPU REGISTERS, AND SELF-CHECK** of this document applies to the MC68HC705T10 MCU device except for the areas described in the following paragraphs.

B.2.1 V_{PP}

The V_{PP} pin (pin 19) is used when programming the EPROM. By applying the programming voltage to this pin, one of the requirements is met for programming the EPROM. In normal operation, this pin is connected to V_{DD}.

CAUTION : Do not connect V_{PP} pin to voltage less than V_{DD}, otherwise, damage to the MCU will result.

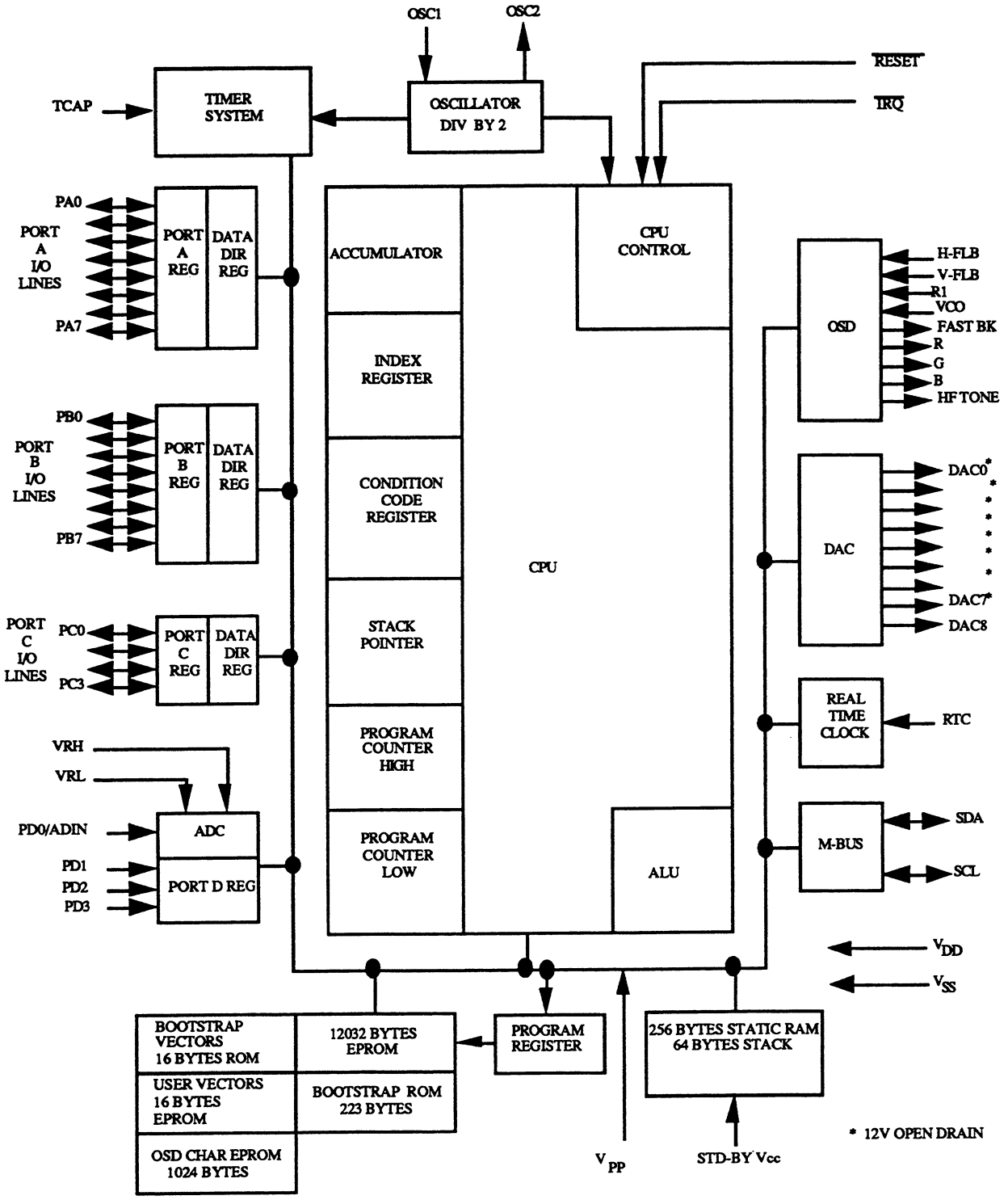


Figure B-1 MC68HC705T10 Microcomputer Block Diagram.

B 2.3 Memory

Figure B-2 illustrates the MC68HC705T10 MCU device address map.

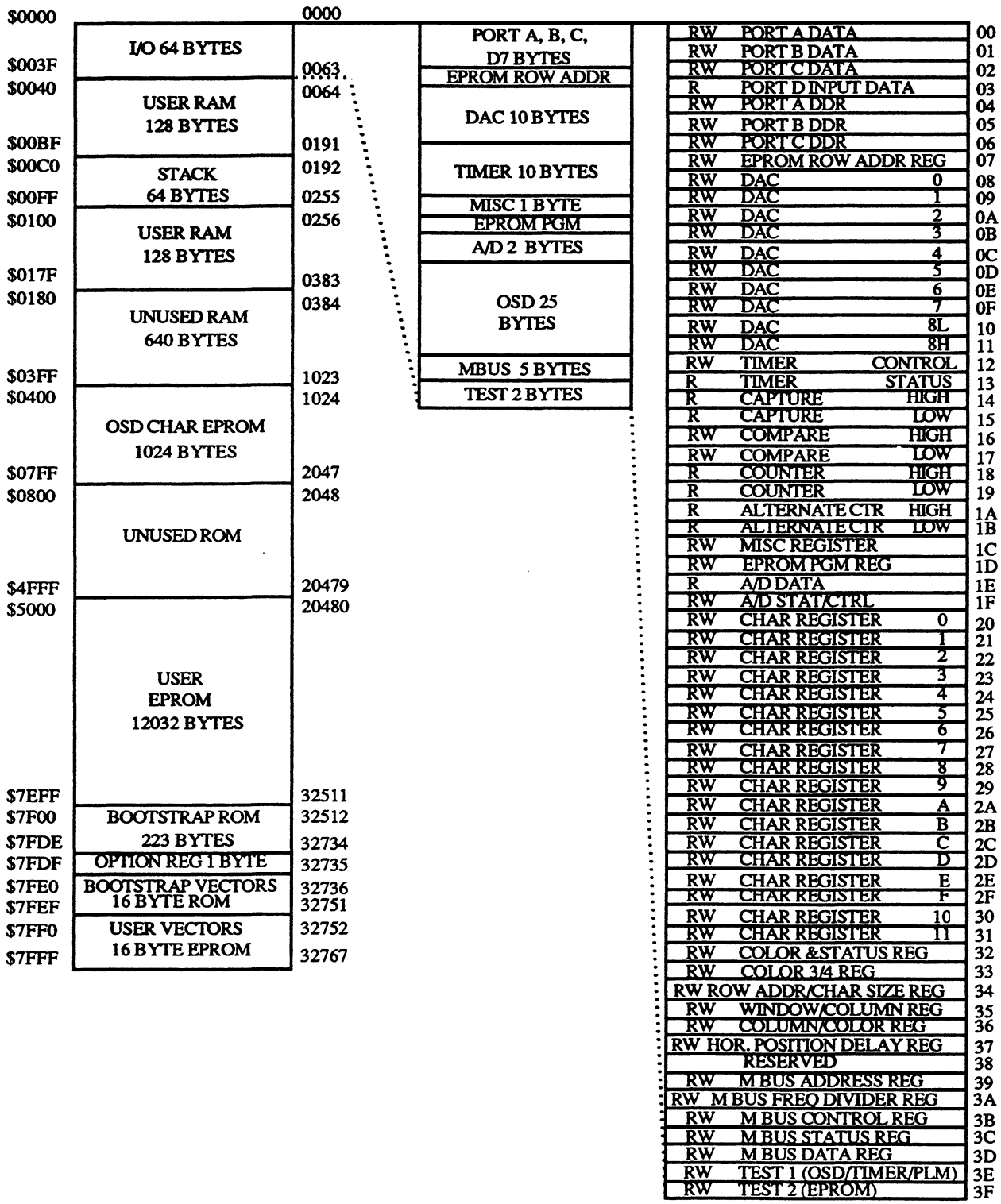


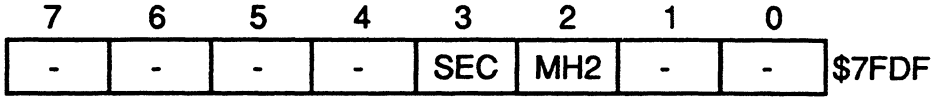
Figure B-2 Memory Map of MC68HC705T10

B.2.4 Self-Check

The self-check ROM is replaced with the bootstrap ROM, therefore the self-check capability is not applicable for the MC68HC705T10 MCU device.

B.3 OPTION REGISTER

The option register is used to enable EPROM security and select the EPROM sense AMP gain or speed.

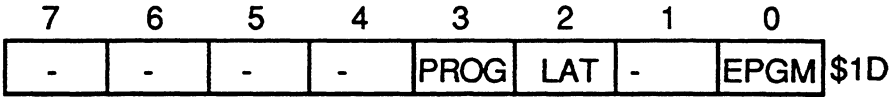


B3, SEC The security (SEC) bit is implemented as a EPROM bit. During EPROM programming, the SEC bit is set to enable the security feature (to disable the bootloader) and enables the MCU to operate in the single chip mode only. For an EPROM device, clearing of SEC bit is accomplished by exposing the EPROM to UV light.

B2, MH2 The MH2 bit is used to set the EPROM AMP speed or gain. If MH2 is erased, it is set to high speed (4 MHz). If MH2 is set, it is set to high gain (2 MHz), which is useful during marginal programming.

B.4 PROGRAM REGISTER

The program register is used to perform EPROM programming.



B3,PROG This bit is used for testing.

B2,LAT Prior to a EPROM write operation, the latch bit (LAT) must be set. This enables the EPROM data and address buses to be latched for programming on the next byte write cycle. Reset clears the LAT bit. When the LAT bit is cleared, EPROM data and address buses are unlatched for normal CPU operations. This bit is both readable and writable. This bit must be cleared, to allow EPROM read operations.

B0,EPGM When the EPROM program (EPGM) bit is set, V_{PP} power is applied to the EPROM for programming mode of operation. Reset clears the EPGM bit. This bit is readable, but only writable when the LAT bit is set. If LAT bit is cleared, EPGM bit cannot be set.

B.5 EPROM PROGRAMMING

Figure B-3 illustrates the write cycle sequences of the MC68HC705T10 EPROM programming operation.

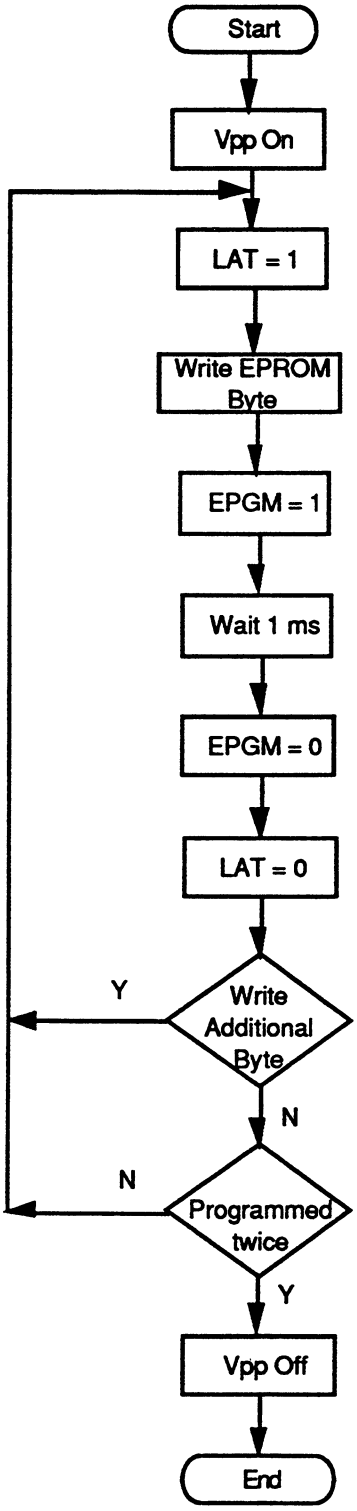


Figure B-3 Eprom Programming Sequence

B.6 ELECTRICAL CHARACTERISTICS

The Electrical Characteristics for the MC68HC705T10 are the same as the MC68HC05T10, apart from the following.

Maximum Ratings
(Voltages Referenced to V_{SS})

Ratings	Symbol	Value	Unit
Bootstrap Mode (IRQ Pin Only)	V_{IN}	V_{SS} - 0.3 to 2 x V_{DD} + 0.3	V
Programming Voltage	V_{pp}	V_{DD} to 15.0	
Operation Temperature Range		T_L to T_H	
MC68HC705T10 (std)	T_A	0 to 70	°C

DC Electrical Characteristics
(V_{DD} = 5.0 V_{dc} +/- 10%, V_{SS} = 0 V , T = T_L to T_H unless otherwise noted)

Characteristics	Symbol	Min.	Typ.	Max.	Unit
Supply Current					
Run	I_{DD}	-	5.0	10.0	mA
Wait	I_{DD}	-	1.6	5.0	mA

APPENDIX C
MC68HC05T10/T7 EMULATION

C.1 INTRODUCTION

The EPROM feature of the MC68HC705T10 MCU enables the user to emulate the MC68HC05T10/T7 MCU devices. This appendix describes the multi-byte EPROM programming technique used to program the MC68HC705T10/7 MCU internal EPROM to emulate the MC68HC05T10/T7. The multi-byte EPROM programming technique uses a bootstrap program contained in ROM to program the MC68HC705T10 MCU internal EPROM. This appendix also provides a schematic diagram (Figure C-1) which illustrates a typical MC68HC705T10 MCU EPROM programming board/circuitry used in conjunction with the multi-byte EPROM programming technique described in this appendix.

The MC68HC705T10 EPROM MCU devices are erased by exposure to a high-intensity ultraviolet (UV) light with a wavelength of 2537 Å (Angstrom). The recommended dose (UV intensity x exposure time) is 15 Ws/sq.cm. UV lamps should be used without short-wave filters, and the EPROM MCU device should be positioned about one inch from the UV lamps.

C.2 PROGRAMMING TECHNIQUE

The EPROM programming technique is used to load a user program into MC68HC705T10 MCU EPROM. The type of programming is accomplished via a bootstrap mode operation. The user program contained in an external memory device is copied into the internal EPROM of the MC68HC705T10 device.

The MC68HC705T10 device is inserted into the programming board/circuitry as illustrated in Figure C-1. Programming routine is selected via mode switches S1 through S4, and +5 Volt and V_{PP} power is applied to the programming circuitry. The MCU is removed from the reset state and placed in the run mode via switch S5, and MCU control is transferred to the bootstrap ROM. The selected programming routine is then executed.

Programming sequence is as follows :

1. Place switch S5 to RESET position.
2. Select programming routine via switches S1-S4.
3. Connect +5 V and V_{PP} (12-14V) power to programming circuitry.
   (Vpp must be applied before or at the same time when +5V is applied)
4. Place switch S5 to RUN position.
5. Programming routine is executed to completion.
6. Place switch S5 to RESET position.
7. Remove V_{PP} and +5 Volt power, or select and run new routine.

Once the bootstrap mode is entered, mode switch settings are scanned to establish the routine to be executed. The routines are as follows :

Program and Verify EPROM
Verify EPROM Contents
Load Program in RAM and Execute
Secure and Verify EPROM
Execute Program in RAM

C.2.1 Program and Verify EPROM

In the program and verify EPROM routine, the contents of an external 32K EPROM are copied into the EPROM areas of the MC68HC705T10 device. There is a direct correspondence of addresses between the two devices. Non-EPROM addresses are ignored so data contained in those areas are not accessed. Unprogrammed EPROM address locations should contain $00 to speed up the programming operation. During the programming routine the PROGRAMMING LED DS2 is illuminated, and the EPROM will be programmed twice for better data retention. . At the end of the programming routine, the verification routine is entered. If the contents of the EPROM and external EPROM exactly match, then the VERIFIED LED DS1 is illuminated and the DS2 is turned off. This program and verify procedure should take no more than two minutes. If the verified LED (DS1) does not illuminate within two minutes, a discrepancy between the on-chip EPROM and external memory device has been detected.

C.2.2 Verify EPROM Contents

The verify EPROM contents routine is normally entered automatically after the EPROM is programmed. Direct entry of this mode will cause the EPROM contents to compared to external memory contents residing at the same address locations. Both DS1 and DS2 LEDs are turned off at this time until verification is completed. Upon completion of the verification routine (every location verified) the VERIFIED LED DS1 is illuminated. Since the verification procedure should not take more than 10 sec, so if DS1 does not illuminate after 10 seconds, a discrepancy has been detected and the error address location will be placed on the external memory address bus.

C.2.3 Load Program in RAM and Execute

In the load program in RAM and execute routine, user programs are loaded into MCU RAM via the M-Bus interface system, and then executed. Data is loaded sequentially, starting at RAM location $40, until the last byte is loaded. Program control is then transferred to the RAM program starting at location $41. The first byte loaded is the count of the total number of bytes in the program plus the count byte. The program starts at the second byte in RAM. During the firmware initialization stage the M-Bus interface is configured as a slave receiver (see **SECTION 5 M-BUS SERIAL COMMUNICATION INTERFACE** for a detailed description of M-Bus interface). The slave address of the MCU is $C6.

If immediate execution is not desired after loading the RAM program, it is possible to hold off execution. This is accomplished by the setting of the byte count to a value that is greater than the overall length of the loaded data. When the last byte is loaded, the firmware will halt operations expecting additional data to arrive. At this point, switch S5 is placed in the RESET position which will reset the MCU but keep the RAM program intact. All other routines (modes) can now be entered from this state, including the one which will execute the program in RAM, once switch S5 is placed in the RUN position.

C.2.4 Secure and Verify EPROM

This routine is used after the EPROM is successfully programmed. Only the security (SEC) bit of the Option Register ($7FDF) is programmed (see **APPENDIX B.3** for a detailed description of the Option Register). Then a normal verify EPROM contents is performed as described in **SECTION C.2.2.**

To ensure that security is properly enabled, attempt to perform another program and verify, or verify only routine. If the proper LED does not illuminate, the EPROM has been properly secured.

C.2.5 Execute Program in RAM

Using this routine, the MCU will transfer control to a program that has been previously loaded into RAM. This program will be executed once the bootstrap mode has been entered, if mode switch S1 is activated. No firmware initialization will take place. The program must start at RAM location $0041 to be compatible with the load program in RAM routine.

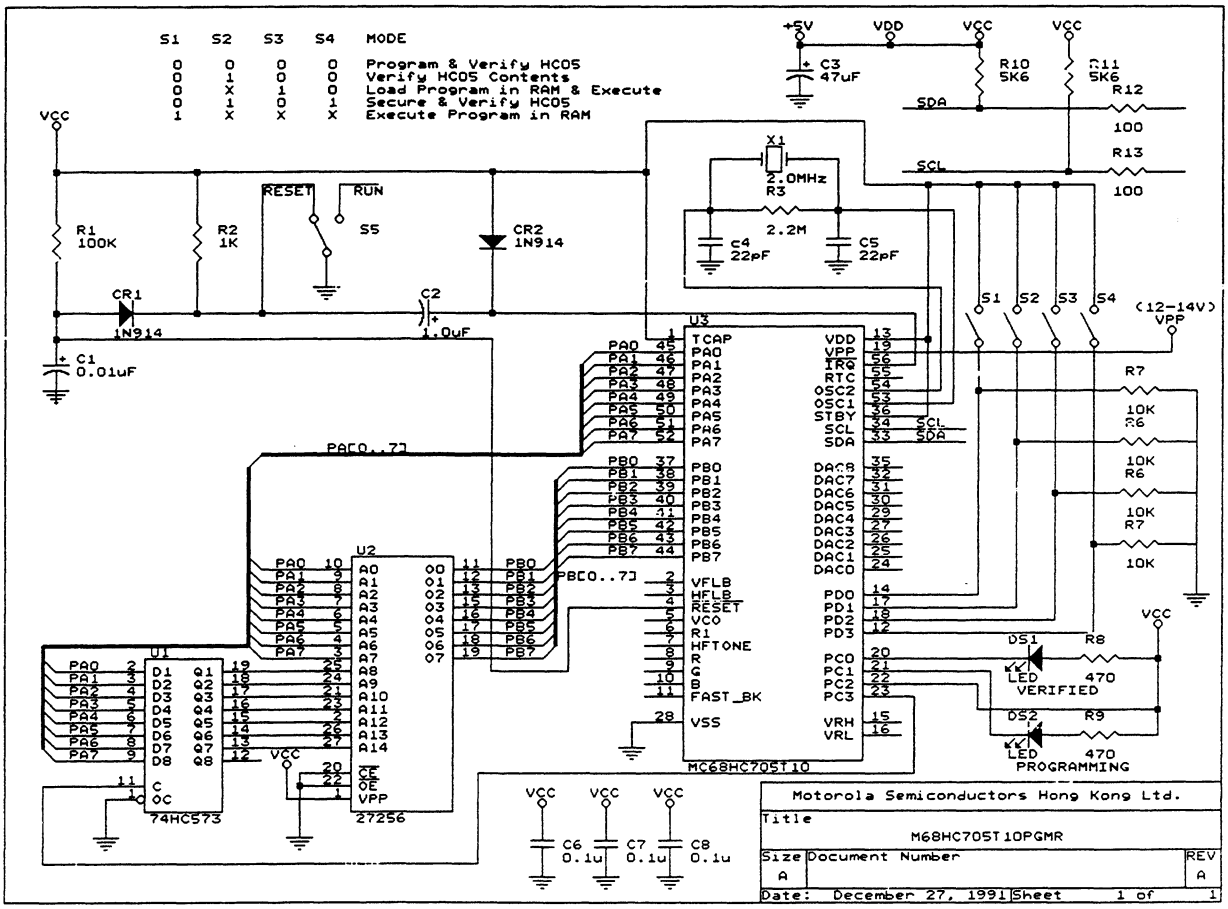


Figure C-1 MC68HC705T10 MCU EPROM Programming Circuit

Freescale Semiconductor, Inc.

MC68HC05T10/T7 MCU ORDERING FORM

Date_____ Customer PO Number_____

Customer Company_____

Address_____

City_____ State_____ Zip_____

Country_____

Phone_____ Extention_____

Customer Contact Person _____

Customer Part Number _____
(12 Charaters Maximum - If Applicable)

Customer Application _____

Device

- ☐ MC68HC05T10
- ☐ MC68HC05T7

Package type

- ☐ 56 PIN Shink DIP

Temperature Range

- ☐ 0 to +70 deg.C

Special Electrical Provisions : _____

(Customer specifications required.)

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

_____

(SIGNATURE)

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

_____

(SIGNATURE)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

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

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

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