INTELLIGENT CRT TERMINAL

by

Osman Serdar Abraz
B.S. in E.E. , Boğaziçi University,1981

Submitted to the Institute for Graduate Studies in Science and Engineering in partial fulfillment of the requirements for the degree of

Master of Science

in

Electrical Engineering

Boğaziçi University
1984

To my grandfather

ACKNOWLEDGEMENTS

I am grateful to Dr. Ömer Cerid for his guidance and helps till the beginning of the thesis, and his cooperation in our obstinate work to realize the system.

I would like to appreciate the patience of my family during the design and realization of the thesis.

I also would like to express my thanks to Gülşen Karaca for her work in drawings and Mukaddes Salcan for the typing of the manuscript.

ABSTRACT

The purpose of the thesis is to design and realize the hardware of an intelligent terminal which will also have graphic display possibilities.

The system is based on the MC 6809 MOTOROLA's high performance 8-bit microprocessor which supports modern programming techniques and has major architectural improvements other than MC 6800 family microprocessors, such as additional registers, instructions and addressing modes.

The terminal hardware has expandable memory capacity up to 58 K bytes, software dependable character set generation, one RS-232 interface with software selectable baud rate generation, four parallel output ports for keyboard and printer interfaces and four outputs for expansion cards such as a floppy disk controller unit.

The design is not only an intelligent terminal but with its hardware features and a powerful basic interpreter software that should be written, it can be used as a self contained microcomputer system.

ÖZETÇE

Bu tezin amacı grafik görüntüleme özelliğine sahip bir terminalin tasarımı ve gerçekleştirilmesidir.

Sistemde kullanılan mikroişlemci ünitesi MOTOROLA firmasının modern yazılım tekniklerine uygun ve diğer MC 6800 ailesi mikroişlemcilerden farklı olarak bazı önemli donanım özelliklerine sahip, MC 6809, 8 bitlik mikroişlemcisidir.

Terminalin 58 K byte'a kadar artırılabilen bellek kapasitesi, yazılıma bağlı olarak değiştirilebilen karakter seti, bir RS-232 seri çıkış kapısı ve klavye ve yazıcı üniteleri için paralel çıkış kapıları mevcuttur.

Sistem donanımının kuvvetli bir yazılım desteği ile sadece bir terminal olarak değil de başlıbaşına bir mikrobilgisayar ünitesi olarak kullanılabileceği düşünülerek tasarımlanmış ve gerçekleştirilmiştir.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

Simply stated, computer graphics is the technique of visual communication from computer to man. Interactive graphics started with the attempts to use the Cathode Ray Tube - CRT - as a computer output device. The Whirwind 1 in 1950 and Sketchpad in 1973 are examples of early attempts at computer graphics systems. Since that time, two distinct classes of CRT based devices have been developed for use in computer graphics: vector graphics (calligraphic) and raster scan technique that the ordinary television does.

Calligraphic displays produce images by drawing vectors using end point information. Even the most complex images can be created as a collection of vectors. Because of the short persistence of the CRT phosphorous required for a calligraphi display, once a vector is drawn it will disappear quickly. Thus the entire display must be continously refreshed. This requirement give rise to a structure called a display list. The processor repeatedly scans this list to send vector drawing information to the display electronic unit. The area of vector graphics has been used for several years since it does not need large memories. However, this is not necessarily true for raster scan devices.

Until recently, raster scan technology has not been economically feasible. But now with the decrease in hardware costs and especially for memory the technology have facilitated the trend toward raster scan displays.

In this thesis a microprocessor based intelligent terminal design is presented. In Chapter 2 a theory of raster scan technology and the block diagram of the intelligent terminal are given with all necessary information about the main components of the system. Chapter 3 consists of the detailed system description. A general discussion about the thesis and the results of the hardware realisation are given in Chapters 5 and 6.

CHAPTER 2
SYSTEM LAYOUT

2.1. THEORY

If we display a sequence of images that change only slightly from one to the next, and do it fast enough, the eye will not be able to separate them: persistance of vision will cause the separate images to fuse into a moving picture. In order to transmit such a sequence of images electronically, each image must be dissected into a series of dots that may be transmitted one at a time. The television camera does this by rapidly scanning the image in a series of horizontal lines which form a raster. When this rapidly changing signal is picked-up by a television receiver, it is converted back into a visible raster on the screen of the picture-tube (CRT). The neck of the picture-tube contains an electron gun that projects a beam of electrons onto a thin layer of phosphor on the inside of the screen. Wherever the electron beam strikes the phosphor it produces a spot of light whose brightness depends on the intensity of the signal being received. If the electron beam is swept across the screen so that the spot of light is always in the same relative position as the scanning dot in the camera, the picture will be recreated on the screen. The circuits in the television set controlling the position of the beam must be able to keep in step with the camera, so the picture information is interrupted for a short time at the end of each line and for a longer time at the end of each frame. During these intervals the signal is changed to an intensity level that is never used for picture information, thus creating synchronization pulses that the television circuits can distinguish from the picture signal.

Before, the repetition rate for the picture scanning process was set at 50 scans per second so that interference from 50 Hz AC power line will be synchronized: that is, any visible interference effect will stand still on the screen and be less noticeable than it would be if it were moving. Scanning the entire picture 50 times per second amounts to a lot of information per unit of time, and thus requires a very wide bandwith. Now, this bandwith requirement is cut in half by making the camera scan every other line during alternate scanning cycles called fields. Two successive fields cover all the lines in the raster 25 times each second, to make a frame. This technique is called interlaced scanning.

The vertical-retrace interval provides time for the television circuits to return the scanning dot to the top of the screen after each field has been completed. Since no picture information should be viewed during this time, the electron beam must be turned off or blanked: so, this time is also called the vertical blanking interval.

Television in Europe, excluding France and England, and in Turkey uses a total of 625 lines per frame. At 25 frames per second, 625 lines per frame is equivalent to 15625 lines per second or 64 μsec per line. Since all the lines are scanned in the same direction, the scanning dot must be returnedacross the screen between the end of one line and the start of the next. This is called horizontal retrace and takes 12 μsec Therefore the visual period of the scanning dot takes 64-12 : 52 μsec

For a computer to produce a display on a television set or on a video monitor, it must generate the video signal and the synchronization pulses - horizontal synchronization and vertical synchronization. Generating the horizontal and vertical synchronizing pulses is relatively easy, since they just repeat over and over in a fixed relationship. While synchronization is easy, generating a video signal with a computer is a little more difficult. First of all, a television picture must be continously regeneretad by repeating the entire scanning process. This continual regeneration of the display is called video refresh and it requires a stream of data at a rate much too fast for a computer to keep up with if the system had to compute a new data for every scan. Instead, all of the data that will appear on the screen is stored in a memory with suitable capacity. This reserved memory is called the video refresh memory. Circuits designed especially for video displaying read data from the refresh memory in step with the video synchronizing pulses, and transform the data into the video signal which is than displayed.

Using part of the computers own memory for video refresh is not the general rule. Most large computer systems include video terminals that are independa of the main computer and contain their own refresh memory. In other world a small personel computer is a hybrid: part computer, part terminal. Combinin functions in this way helps to keep down the cost of personal computing.

There are several different methods of transforming the data stored in the refresh memory into an effective display. The most straight-forward method is to take the data just as it is read from the refresh memory and transmit it to the display one bit at a time. Each bit which is 1 in this serial bit stream appears on the screen as a spot of light, and each 0 bit as darkness. The size of the refresh memory is matched to the picture scan so that for each bit in the refresh memory there is one spot on the display screen. A one - to - one correspondance of this kind is called a map, and this technique for generating computer video displays is called bit-mapping. Since we can program the computer so that data bits are stored into the refresh memory in any pattern we desire, this kind of display can have all the versatility we want, especially when used for graphic purpose

When displaying only letters and numbers the bit-mapping method becomes inefficient since it requires large memory and complicated software. For example a letter that occupies nine rows of eight dots requires nine bytes of memory in the bit-mapped display, but we can encode the same letter in ASCII (American Standart Code for Information Interchange) and reduce the size of the refresh memory by a factor of nine. This means that instead of sending the data bits directly to the display, it is necessary to decode each stored character and generate the appropriate video informatio To do this, the refresh circuits send the character code to another circuit called a charactor generator. The character generator is little more than a read-only memory that contains the video bit patterns for each of the characters we want to display.

The most important problem in the design of the Intelligent CRT Terminal is in the timing of the system operation.

As given befor the scanning time of one line is 64 μsec, 12 μsec of it beeing for the horizontal retrace. For a system with 80 characters per row - which is almost a standart - this requires 0.65 μsec or 1.53846 MHz for the character rate and 1.53846 x 8 : 12.30768 MHz for the dot clack rate. But with this character rate all of the CRT screen area will be the active scan area and at the edges of the screen (at the right most and left most

ends) there will be some characters and the reading will be very difficult. Therefore at the right and left of the screen there must be an area which is blanked. This means that the character rate and the dot clock rate must be faster to be able to display 80 characters per line in a shorter time. If we decrease the character time from 0.65 μsec to 0.60 μsec we have 1.666 MHz for the character rate and 13.333 MHz for the dot clock rate (which makes 0.075 μsec for dot time) For 80 charaters per line this makes:

80 x 0.6 μsec : 48 μsec

Therefore a total of 4 μsec (52 - 48 : 4 μsec) will be blanked. The illustration of the CRT screen format for the character display is given in figure 1.

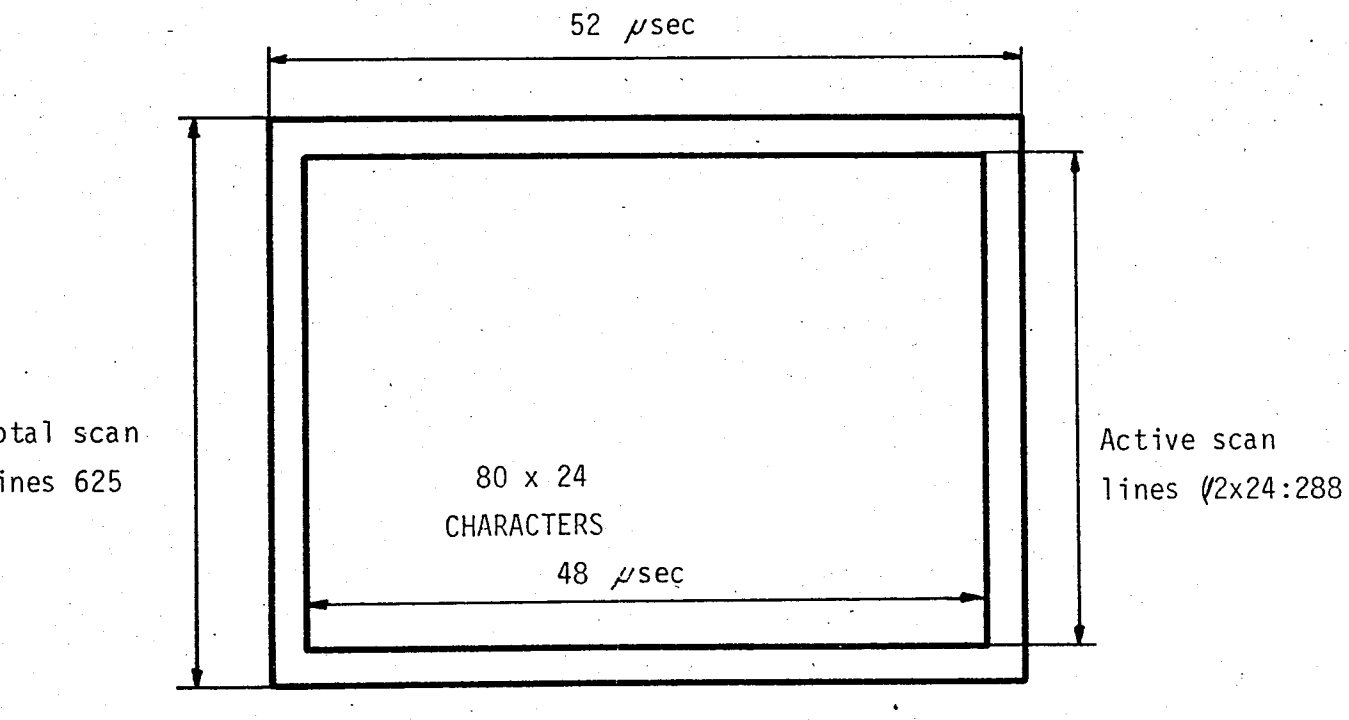


FIGURE 1
ILLUSTRATION OF THE CRT SCREEN FORMAT
FOR CHARACTER DISPLAY

If the video refresh memory is consisted of a 16 K-byte memory block, for the graphic display purposes this is equivalent to 256 x 512 dot matrix array. Therefore for the graphic display, the active scan lines will be equal to 256 and since the dot time is 0.075 μsec the active horizontal display time will be equal to : 512 x .075 : 38.4 μsec The illustration of the CRT screen format for graphic display is given in figure 2 .

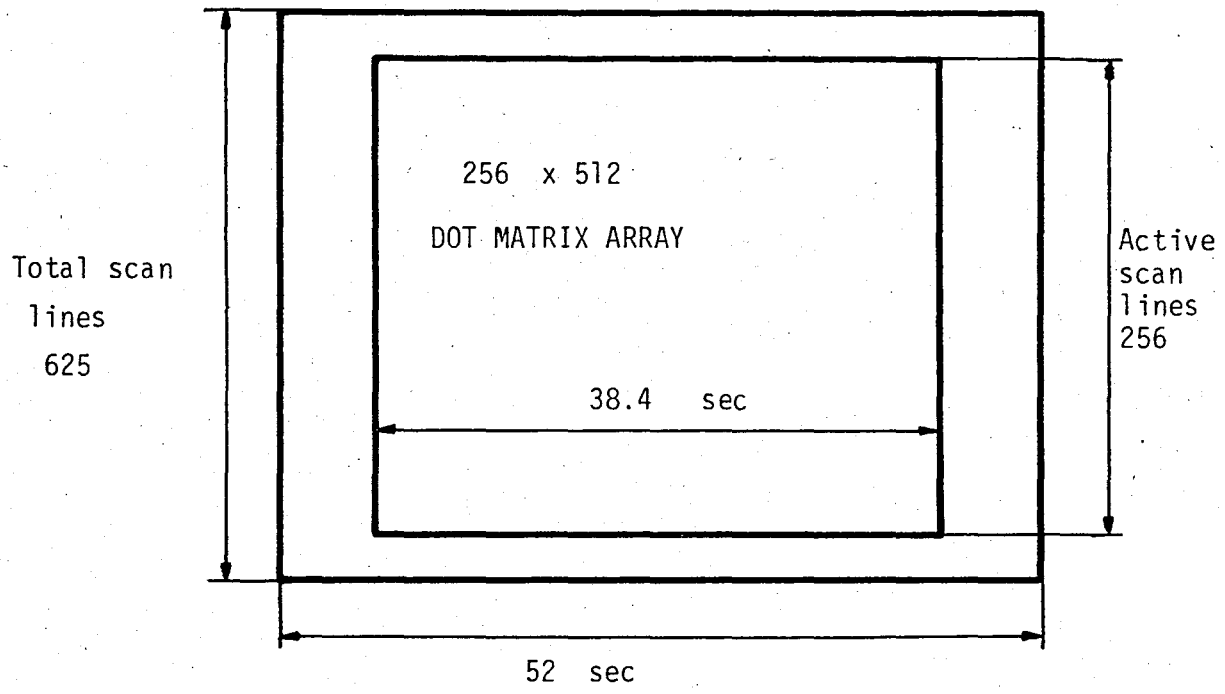


FIGURE 2

ILLUSTRATION OF THE CRT SCREEN FORMAT FOR GRAPHIC DISPLAY

The basic configuration of the microprossor controlled CRT terminal is shown in Figure 3.

The microprocessing unit - MPU - executes the CRT terminal executive software routine and jumps to driver subroutines when servicing the keyboard and the serial interface.

The CRT controller - CRTC - allows complete software control of the video display monitor. It provides horizontal synchronization, vertical synchronizatic and blanking to a video display monitor along with the address of the data to be displayed. The cursor output, thc display enable and the output

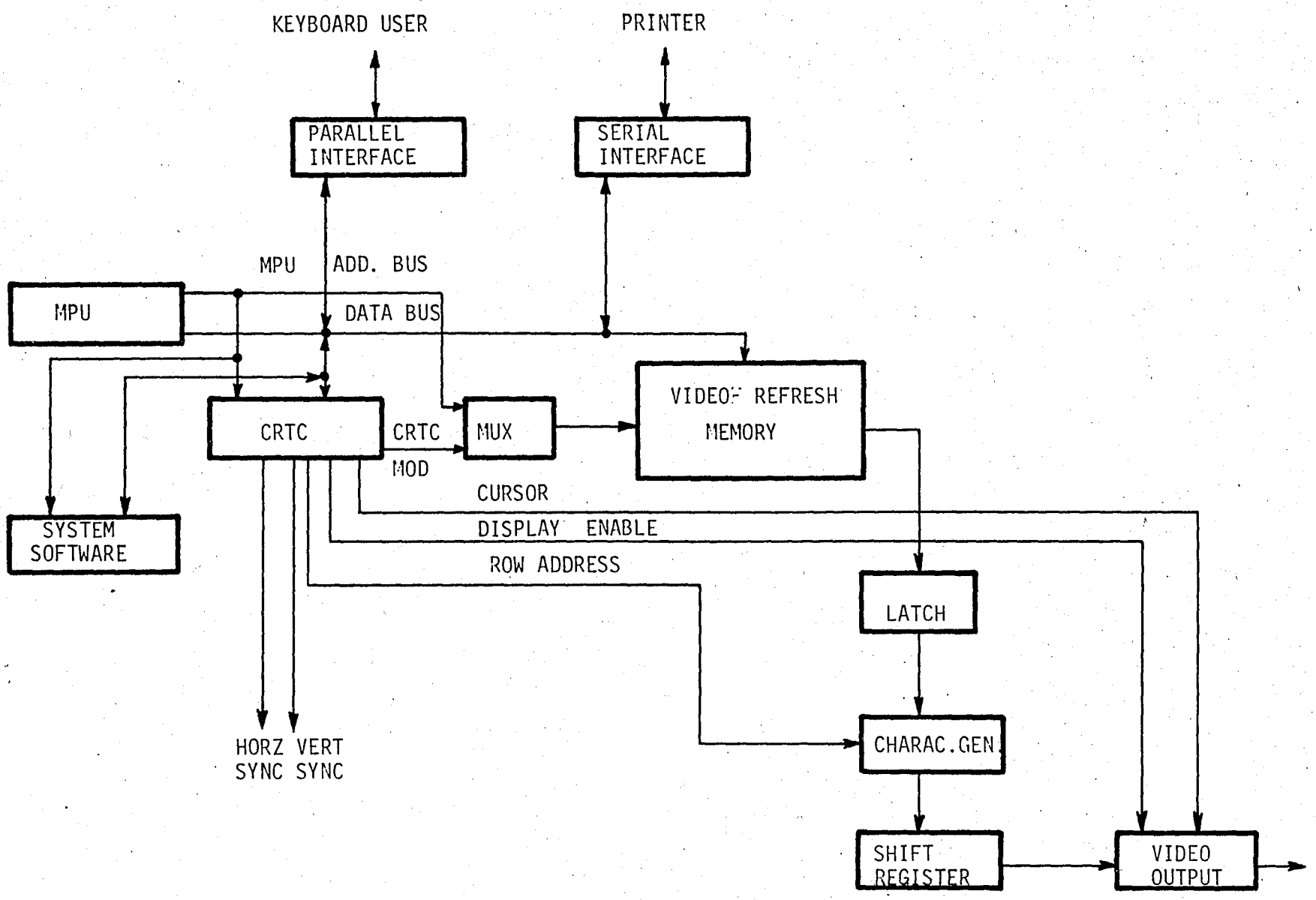
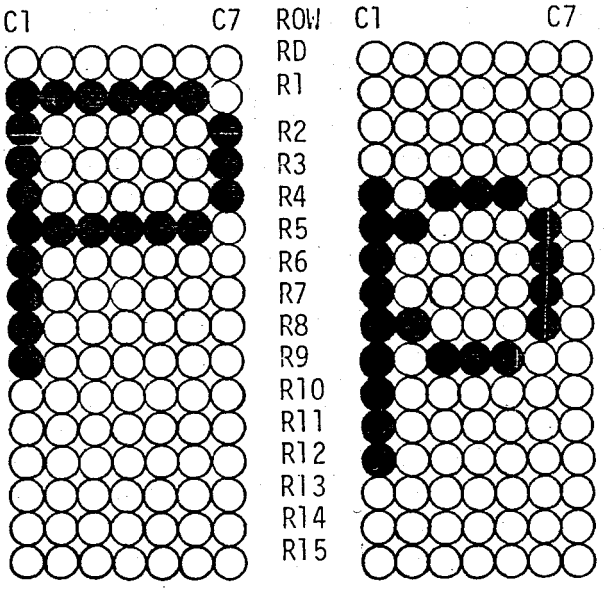


FIGURE 3

CRT TERMINAL CONFIGURATION

of the data output shift register form together the video output signal.

The device that determines which dots are to be illuminated to form a character is called a character generator. Actually, it is a read only memory - ROM containing a dot matrix that can be mask programmed or that can be preprogrammed for each character. 5x7 and 7x9 dot matrix are the most popular configurations with the 7x9 offering clearer definition. Some character generators are capable of automotically shifting descenders (such as g,j,p,q,y) Thus, while ayn one character is contained in a 7x9 matrix, a 7x12 matrix must be available on the CRT screen to contain both normal and descending characters



EXAMPLE TO UPPER AND LOWER CASE CHARACTER DISPLAY

2.2. THE MICROPROCESSOR UNIT - MC 6809

The MC 6809 is a high performance 8 bit microprocessor which supports modern programming techniques. The expanded block diagram is shown in figure 4.

The A and B registers are general purpose 8 bit accumulators which are used for arithmetic calculations and manipulation of data. Certain instructions join the A and B registers to form a single 16 bit accumulator. This is referred to as the D register and is formed with the A register as the most significant byte.

The direct page register serves to enhance the Direct Addressing Mode. The content of this register appears at the higher address outputs (A8-A15) during the direct addressing instruction execution. This allows the direct mode to be used at any place in memory, under program control.

The X and Y Index Registers are used in indexed mode of addressing. The 16 bit address in this register takes part in the calculation of effective addresses. This address may be used to point the data directly or may be modified by an optional constant or register offset. During some indexed modes the contents of the index register are incremented and decremented to point to the next item of tabular type data.

The hardware stack pointer S is used automatically by the processor during subroutine calls and interrupts. The user stack pointer U is controlled exclusively by the programmer thus allowing arguments to be passed to and from subroutines with ease. Both stack pointers have the same indexed mode addressing capabilities as the X and Y registers, but also support PUSH and PULL instructions.

The Program Counter - PC is used by the processor to point to the address of the next instruction to be executed by the processor. Relative Addressing is provided allowing the program counter to be used like an index register

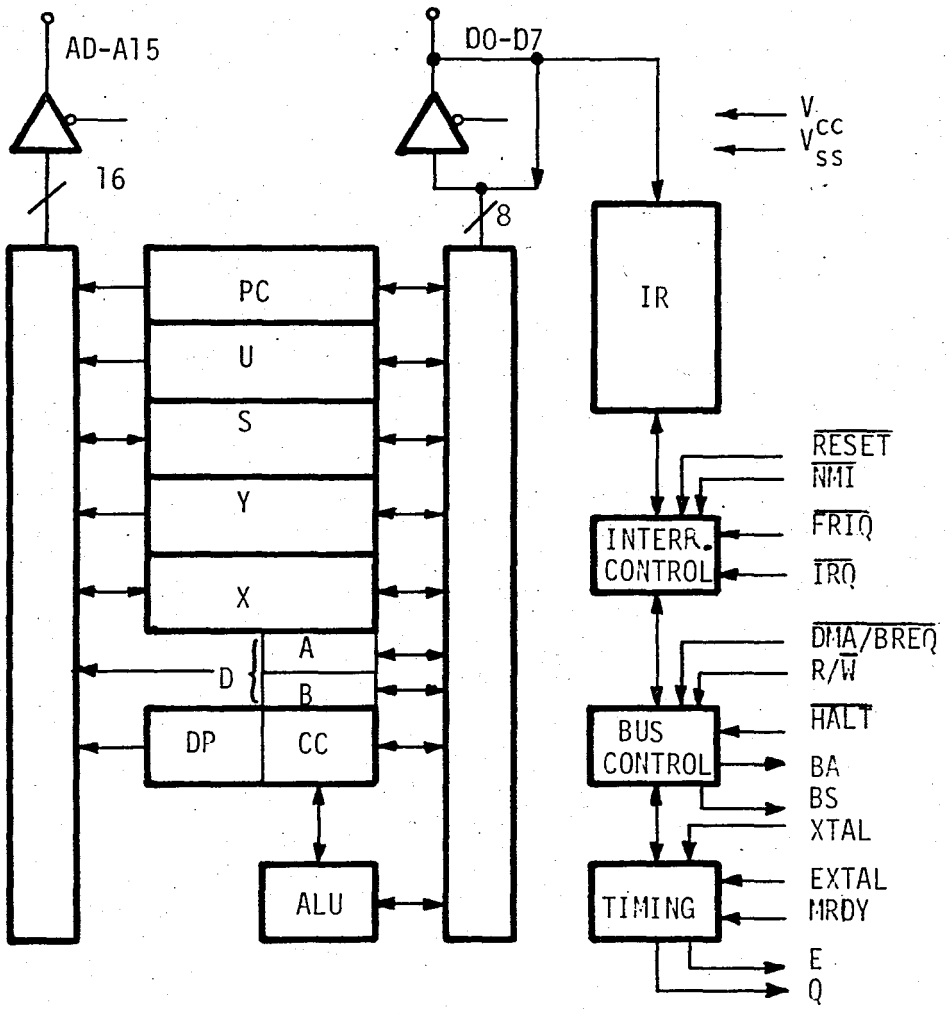
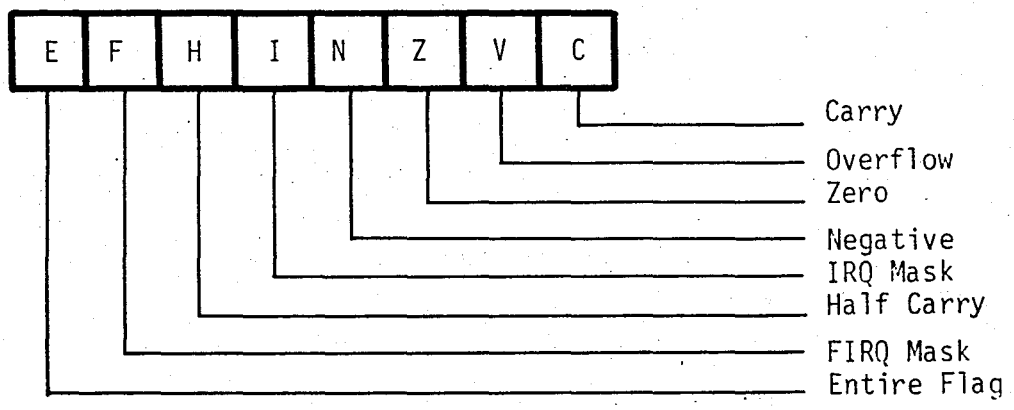


FIGURE 4

MC 6809 EXPANDED BLOCK DIAGRAM

in some situations.

The condition code register defines the state of the processor at any given time:



Bit 0 of the condition code register is the carry flag. It is usually the carry from the Arithmetic Logic Unit (ALU). It is also used to represent a barrow from substruct like instructions but here the carry flag is the complement of the carry from the binary alu.

Bit 1 (V) is the overflow flag and is set to "1" by an operation which causes a signed two's complement aritmetic overflow. This is detected when the carry from the MSB in the ALU does not match the carry from the MSB-1.

Bit 2 (z) is the zero flag and is set to one if the result of the previous operation was identically zero.

Bit 3 (N) is the negative flag which contains exactly the value of the MSB of the result of the preceeding operation. Therefore, a negative 2s complement result will set N to one.

Bit 4 (I) is the $\overline{IRQ}$ mask bit. The processor will not recognize interupts from the IRQ line if this bit is set to a one. $\overline{NMI}$, $\overline{FIRQ}$, $\overline{IRQ}$, $\overline{RESET}$ all set this bit to "1".

Bit 5(H) is the half carry bit, and it is used to indicate a carry from bit 3 in the Arithmetic Logic Unit as a result of an 8-bit addition only.

Bit 6(F) is the $\overline{FIRQ}$ mask bit. The processor will not recognize interrupts from the FIRQ line if this bit is "1".

Bit 7 (E) is the entire flag and when set to "1" indicates that the complete machine state (all the registers) was stacked.

2.2.1. MPU SIGNAL DESCRİPTION

POWER (V_{SS}, V_{CC})
Two pins are used to supply power to the part V_{SS} is ground or 0 volts, while V_{CC} is 5.0 V±5%

ADRESS BUS (A0-A15)
Sixteen pins are used to output address information from the MPU onto the Address Bus. When the processor does not require the bus for a data transfer it will output address FFFF , $R/\overline{W}$: 1, and BS : 0 Addresses are valid on the rising edge of Q (Figure 5 and 6). All address bus drivers are made high impedance when output Bus Available (BA) is high. Each pin will drive one Schottky TTL load and typically 90 pF.

DATA BUS (D0-D7)
These eight pins provide communication with the system bi-directional data bus. Each pin will drive one Schottky TTL load and typically 130 pF.

READ/WRITE ($R/\overline{W}$)
This signal indicates the direction of data transfer on the data bus. A low indicates that the MPU is writing data onto the data bus. $R/\overline{W}$ is made high impedance when BA is high. $R/\overline{W}$ is valid on the rising edge of Q. (Figure 5 and 6)

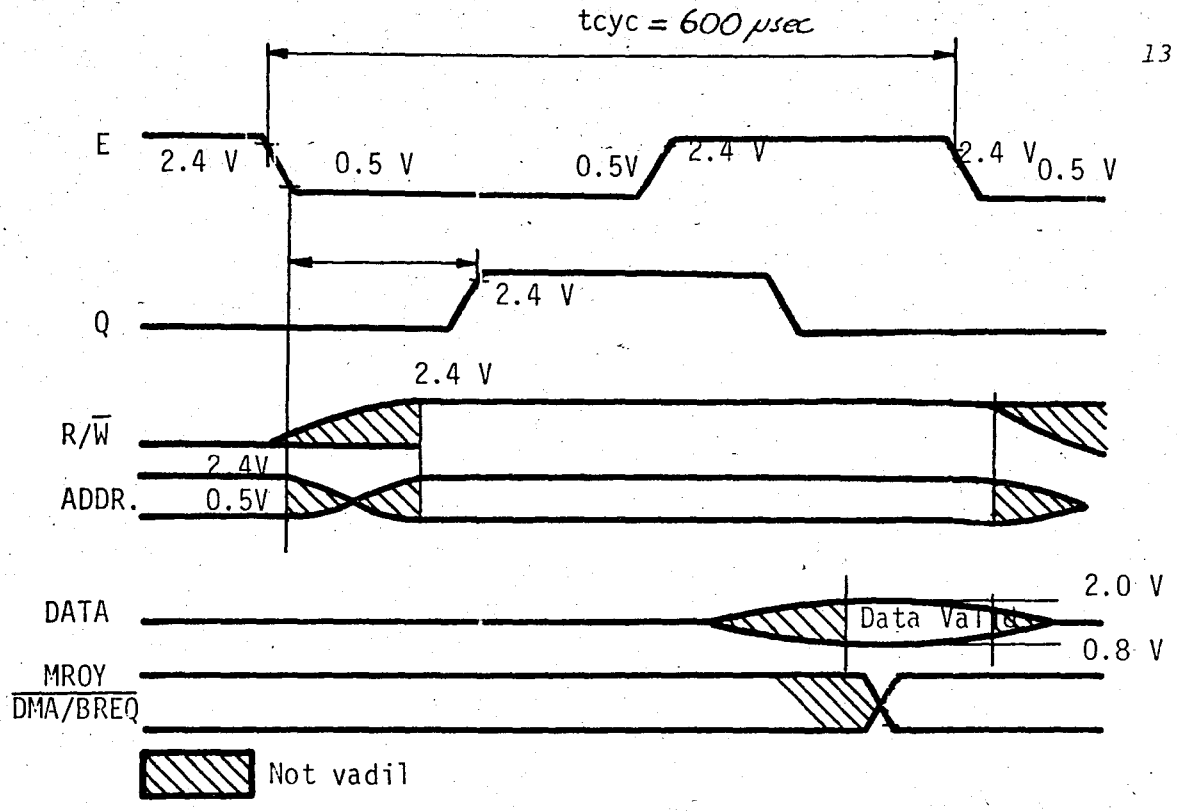


FIGURE 5 - READ DATA FROM MEMORY OR PERIPHERALS

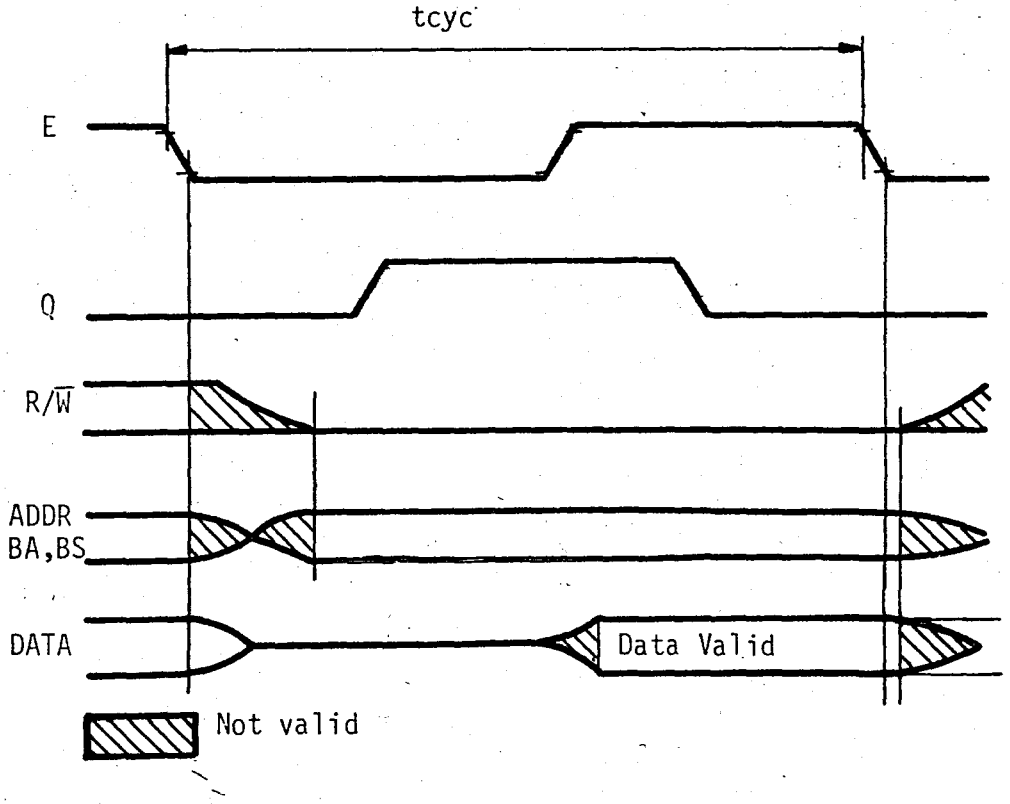


FIGURE 6 - WRITE DATA TO MEMORY OR PERIPHERALS

$\overline{RESET}$

A low level on this Schmitt-trigger input for greater than one bus cycle will reset the MPU as shown Figure 7. The Reset vectors are fetched from location FFFE and FFFF (Table 1) During initial power on the Reset line should be held low until the clock oscillator is fully operational.

Because the MC6809 Reset pin has a Schmitt trigger input with a threshold voltage higher than that of standard peripherals, a simple R/C network may be used to reset the entire system. This higher threshold voltage insures that all peripherals are out of the reset state before the Processor.

$\overline{HALT}$

A low level on this input pin will cause the MPU to stop running at the end of the present instruction and remain halted indefinitely without loss of data. When Halted, the BA output is driven high indicating the buses are highimpedance BS is also high which indicates the processor is in the Halt or Bus Grant state. While halted the MPU will not respond to external real time requests ($\overline{FIRQ}$,$\overline{IRQ}$) although $\overline{DMA/BREQ}$ will always be accepted, and $\overline{NMI}$ or $\overline{RESET}$ will be latched for later response. During the, Halt state Q and E continue to run normally. If the MPU is not running ($\overline{RESET}$,$\overline{DMA/BREQ}$) a halted state (BA and BS : 1) can be achieved by pulling $\overline{HALT}$ low while $\overline{RESET}$ is still low. If $\overline{DMA/BREQ}$ and $\overline{HALT}$ are both pulled low, the processor will reach the last cycle of the instruction (by reverse cycle stealing) where the machine will then become halted. (Figure 8)

BUS AVAILABLE, BUS STATUS (BA,BS)

The Bus Available output is an indication of an internal control signal which makes the MOS buses of the MPU high impedance. This signal does not imply that the bus will be available for more than one cycle. When BA goes low, an additional dead cycle will elapse before the MPU acquires the bus.

The Bus Status output signal, when decoded with BA represents the MPU state. (valid with leading edge of Q)

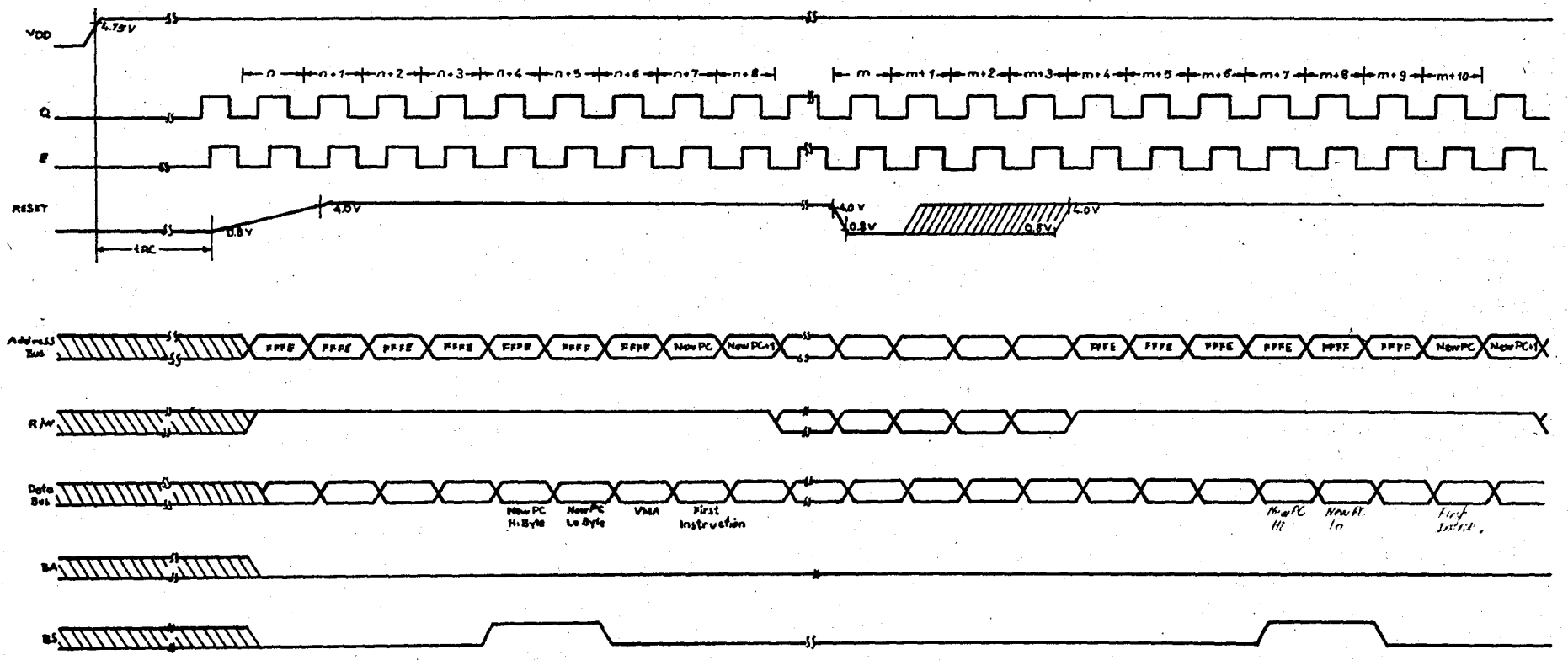


FIGURE 7

RESET TIMING

MPU *BA*	*State* *BS*	
0	*0*	*Normal (Running)*
0	*1*	*Interrupt Acknowledge*
1	*0*	*SYNC Acknowledge*
1	*1*	*HALT or Bus Grant*

TABLE 1 : MEMORY MAP FOR INTERRUPT VECTORS

MEMORY MAP FOR VECTOR LOCATION		INTERRUPT VECTOR DESCRIPTION
MS	LS	
FFFE	FFFF	$\overline{\text{RESET}}$
FFFC	FFFD	$\overline{\text{NMI}}$
FFFA	FFFB	SWI
FFF8	FFF9	$\overline{\text{IRQ}}$
FFF6	FFF7	$\overline{\text{FIRQ}}$
FFF4	FFF5	SWI2
FFF2	FFF3	SWI3
FFF0	FFF1	Reserwed

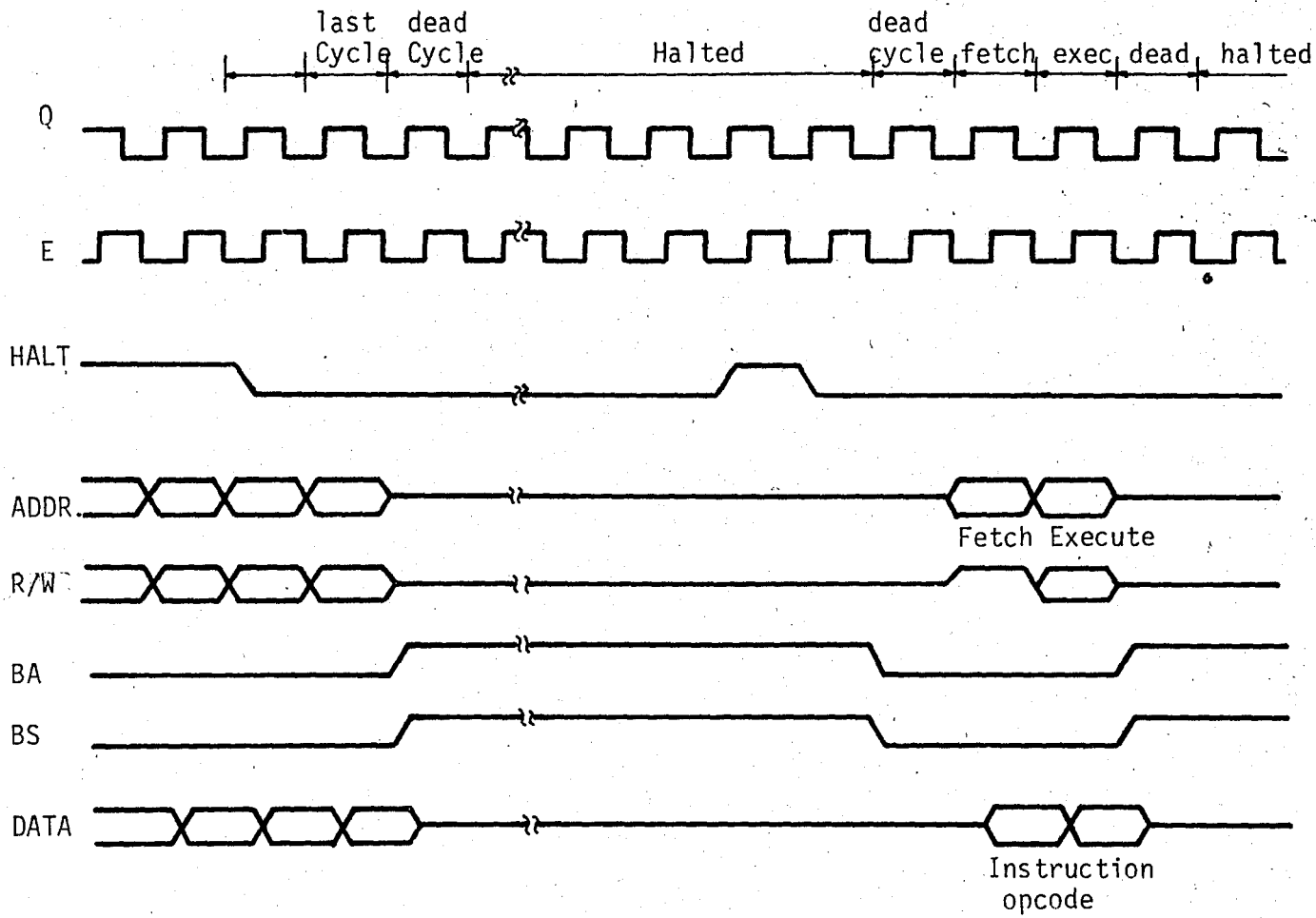


FIGURE 8

$\overline{\text{HALT}}$ AND SINGLE INSTRUCTION EXECUTION

Interrupt Acknowledge is indicated during both cycles of a hardware vector fetch ($\overline{RESET}$, $\overline{NMI}$, $\overline{FIRQ}$, $\overline{IRQ}$, SWI, SW12, SW13) This signal, plus decoding of the lower 4 address lines can provide the user with an indication of which interrupt level is being serviced and allow vectoring by device. (see Table 1)

Sync Acknowledge is indicated while the MPU is waiting for external synchronization on an interrupt line.

Halt/Bus grant is true when the MC6809 is in a Halt of Bus Grant condition.

NON MASKABLE INTERRUPT ($\overline{NMI}$)

A negative edge on this input requests that a nonmaskable interrupt sequence be generated.A nonmaskable interrupt cannot be inhibited by the program and also has a higher priority than $\overline{FIRQ}$, $\overline{IRQ}$ or software interrupts. During recognition of an $\overline{NMI}$, the entire machine state is saved on the hardware stack. After reset an $\overline{NMI}$ will not be recognized until the first program load of the Hardware Stack Pointer (S).The pulse width of $\overline{NMI}$ low must be at least one E cycle. If the $\overline{NMI}$ input does not meet the minimum set up with respect to Q, the interrupt will not be recognized until the next cycle. (Figure 9)

FAST INTERRUPT REQUEST ($\overline{FIRQ}$)

A low level on this input pin will initiate a fast interrupt sequence, provided its mask bit (F) in the CC is clear. This sequence has priority over the standard Interrupt Request ($\overline{IRQ}$), and is fast in the sense that it stacks only the contents of the condition code register and the program counter. The interrupt service routine should clear the source of the interrupt before doing an RTI (Figure 10)

INTERRUPT REQUEST ($\overline{IRQ}$)

A low level input on this pin will initiate an interrupt Request sequence provided the mask bit (I) in the CC is clear. Since $\overline{IRQ}$ stacks the entire machine state it provides a slower response to interrupts than $\overline{FIRQ}$, $\overline{IRQ}$ also has a lower priority than $\overline{FIRQ}$. Again, the interrupt service routine should clear the source of the interrupt before doing an RTI. (Figure 9)

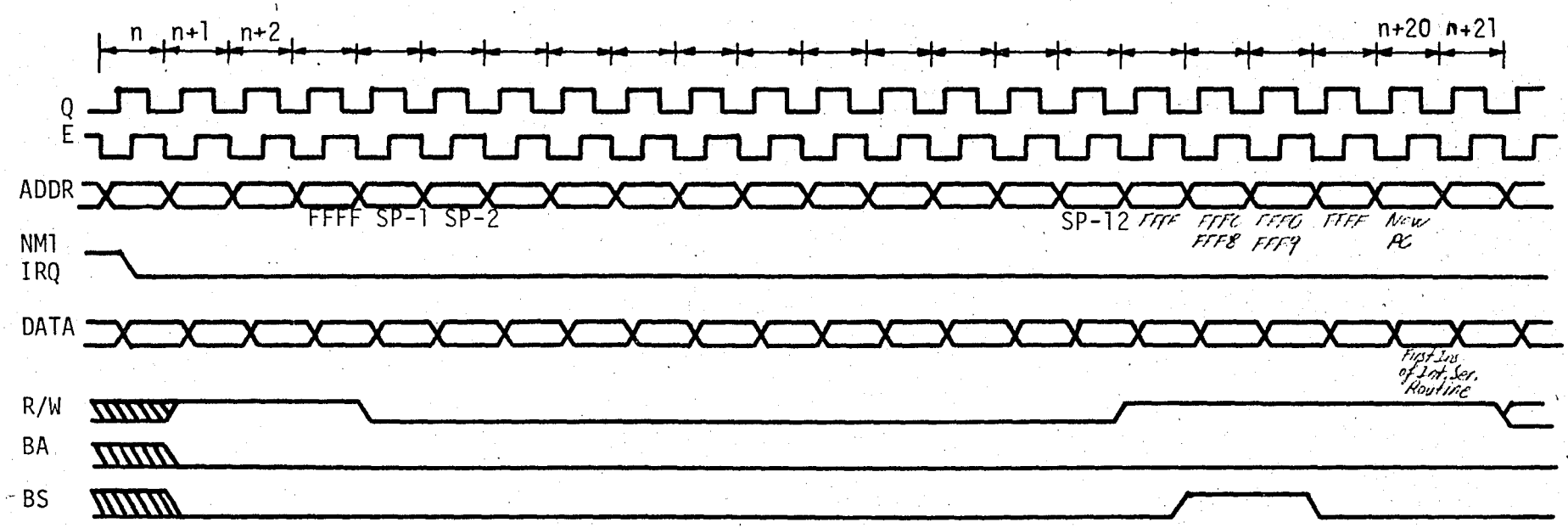


FIGURE 9 : $\overline{IRQ}$ AND $\overline{NMI}$ TIMING

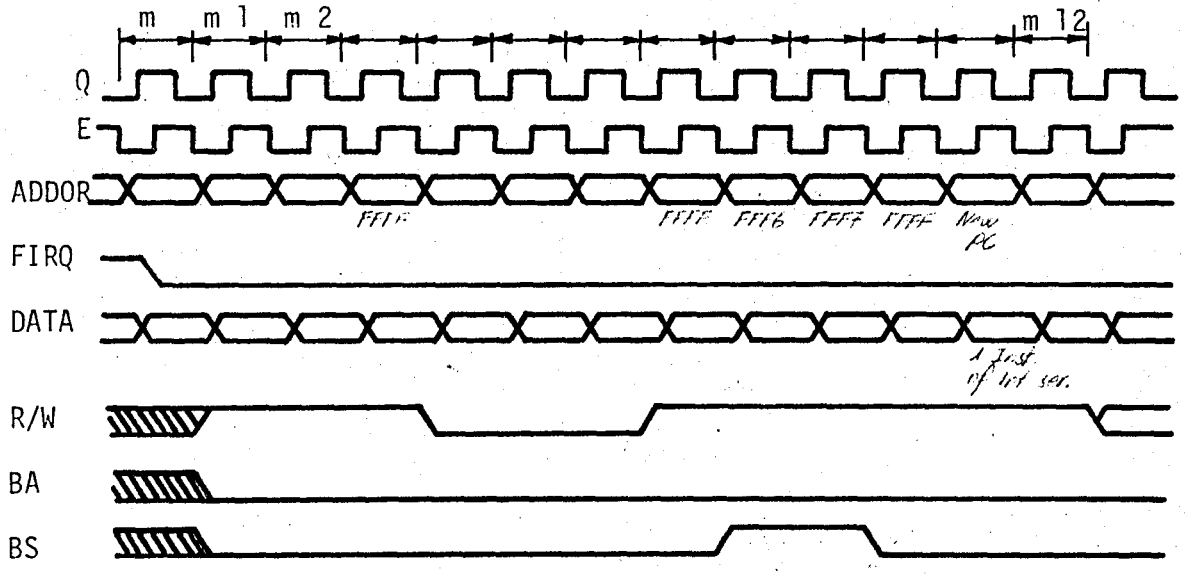


FIGURE 10 : $\overline{FIRQ}$ TIMING

XTAL EXTAL

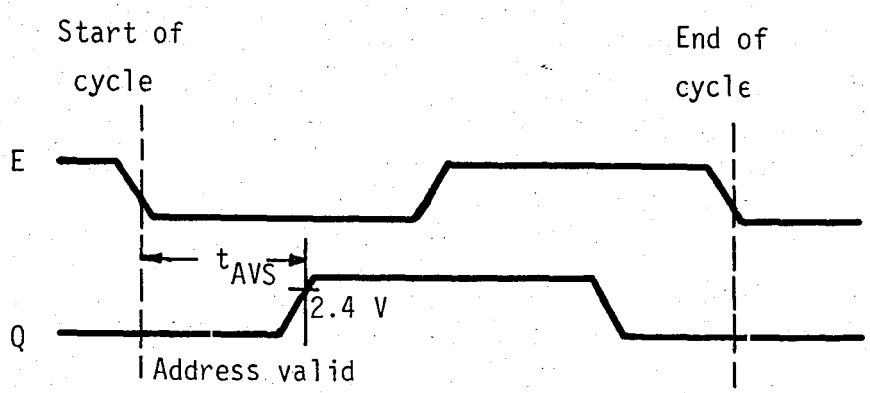
These input pins are used to connect the on-chip oscillator to an external parallel-resonant crystal. Alternately, thepin EXTAL may be used as a TTL level input for external timing by grounding XTAL. The crystal or external frequency is 4 times the bus frequency.

E,Q

E is similar to the MC6800 bus timing signal Ø2.Qis a quadrature clock signal which leads E. Addresses from the MPU will be valid with the leading egde of Q. Data is latched on the falling edge of E. Timing for E and Q is shown in Figure 11.

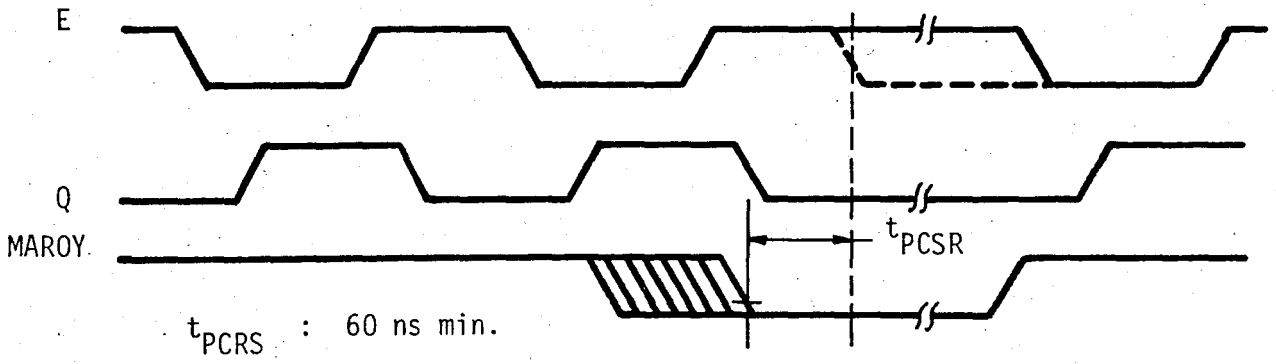
MRDY

This input control signal allows stretching of E to extend data-access time. When MRDY is high, E will be in normal operation. When MRDY is low, E may be stretched integral multiples of quarter (1/4) bus cycles, thus allowing interface to slow memories as shown in Figure 12.A maximum stretch is 10 microseconds.



t_{AVS} : 125 ns max. for MC 68B09

FIGURE 11 : E and Q RELATIONSHIP



t_{PCRS} : 60 ns min.

FIGURE 12 : MRDY TIMING

$\overline{DMA/BREQ}$

The $\overline{DMA/BREQ}$ input provides a method of suspending execution and acquiring the MPU bus for another use as shown in Figure 13. Typical uses include DMA and dynamic memory refresh.

Transition of $\overline{DMA/BREQ}$ should occur during Q. A low level on this pin will stop instruction execution at the end of the current cycle. The MPU will acknowledge $\overline{DMA/BREQ}$ by setting BA and BS to a one. The requesting device will now have up to 15 bus cycles before the MPU retrieves the bus for self-refresh. Self-refresh requires onebus cycle with a leading and trailing dead cycle, see Figure 14.

Typically, the DMA controller will request to use the bus by asserting the $\overline{DMA/BREQ}$ pin low on the leading edge of E. When the MPU replies with BA : BS : 1, that cycle will be a dead cycle used to transfer control to the DMA controller.

False memory accesses should be prevented during any dead cycles. When BA is cleared (either as a result of $\overline{DMA/BREQ}$: HIGH or MPU self-refresh) the DMA device should be taken off the bus.

Another dead cycle will elapse before the MPU is allowed a memory access to transfer control without contention.

2.2.2. MPU OPERATION

During normal operation, the MPU fetches an instruction from memory and then executes the requested function. This sequence begins at RESET and is repeate indenfinitely unless altered by a special instruction or hardware occurence. Sofatware instruction that alter normal MPU operation are SWI, SWI2, CWA1, RTI and SYNC. An interrupt, $\overline{HALT}$ or $\overline{DMA/BREQ}$ can also alter the normal executio of instruction. Figure 15 illustrates the flow chart for the MC6809. The lef half of the flowchart represents normal operation the righ-half represents th flow when an interrupt or special instruction occurs.

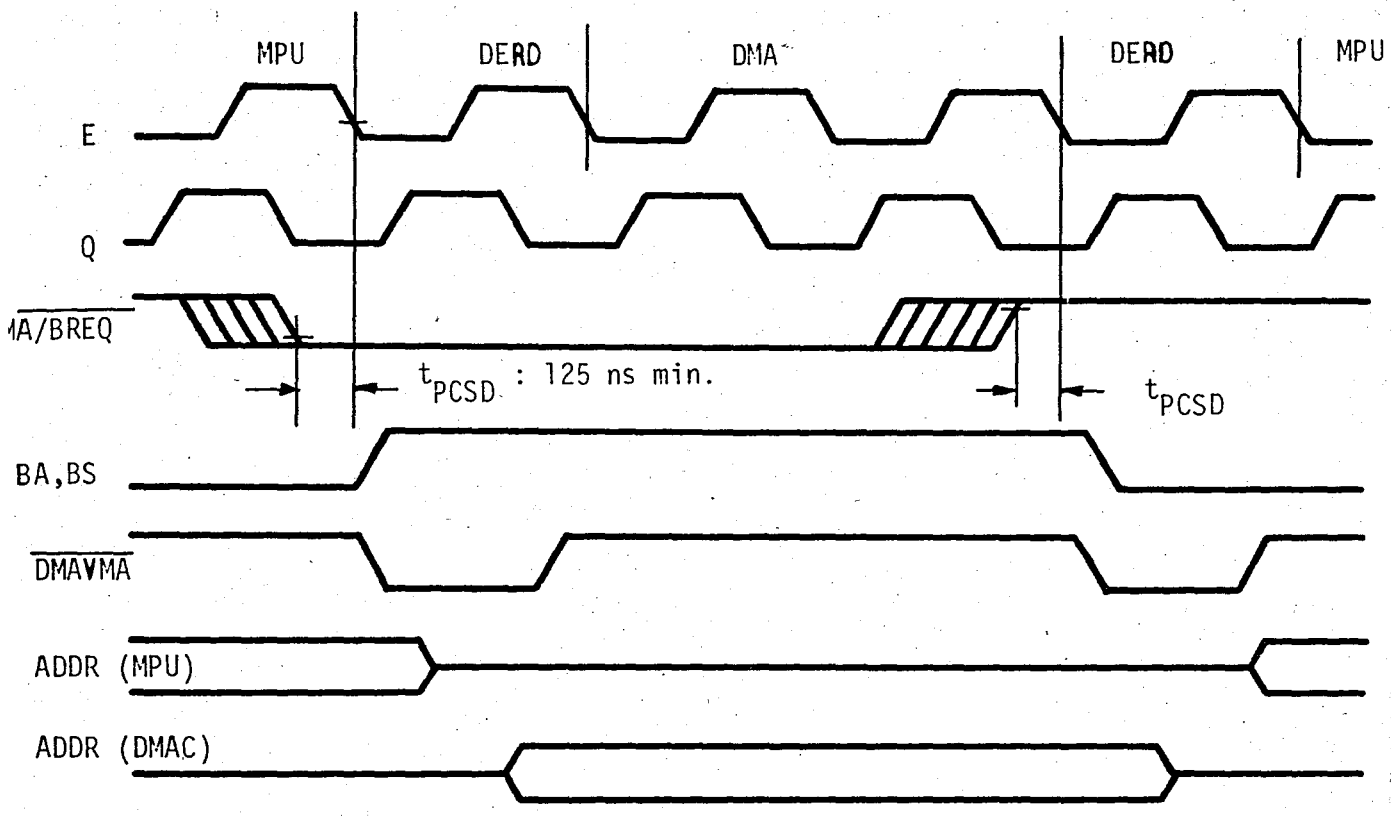


FIGURE 13 TYPICAL DMA TIMING

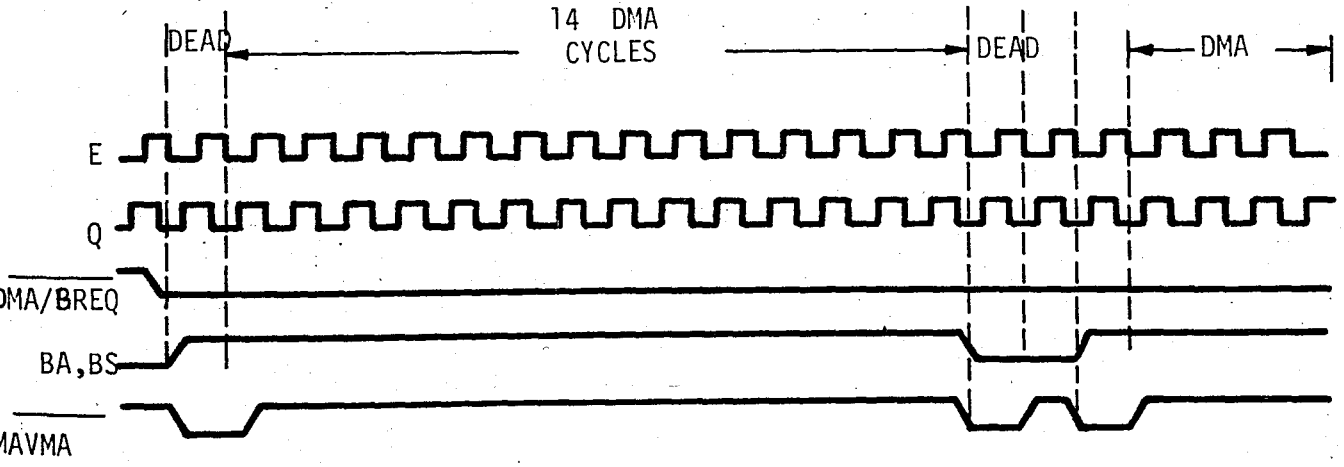
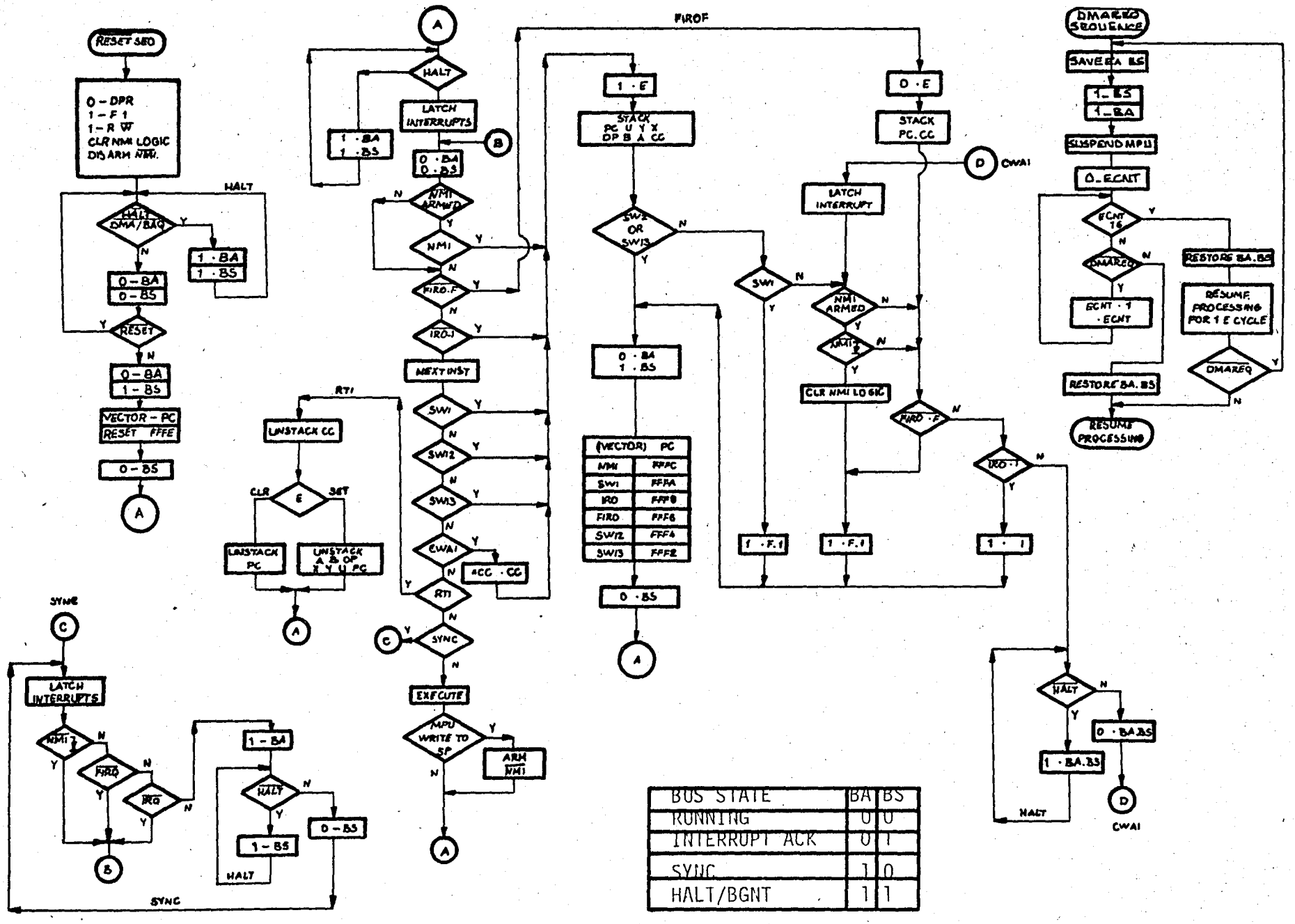


FIGURE 14 AUTO REFRESH DMA TIMING



BUS STATE	BA	BS
RUNNING	0	0
INTERRUPT ACK	0	1
SYNC	1	0
HALT/BGNT	1	1

FIGURE 15 : MPU OPERATION

2.2.3. ADDRESSING MODES

The basic instructions of any computer are greatly enhanced by the presence of powerful addressing modes. The MC6809 has the most complete set of addressing modes available on any microcomputer today. For example, the MC6809 has 59 basic instructions, however it recognizes 1464 different variations of instructions and addressing modes. The new addressing modes support modern programming techniques. The following addressing modes are available on the MC6809.

- Inherent (Includes Accumulator)
- Immediate
- Extended

  Extended indirect
- Direct
- Register
- Indexed

  Zero-Offset

  Constant Offset

  Accumulator Offset

  Auto Increment Decrement

  Indexed Indirect
- Relative

  Short/Long Relative Branching

  Program Counter Relative Addressing

(i) INHERENT (INCLUDES ACCUMULATOR)

In this addressing mode, the opcode of the instruction contains all the address information necessary.

(ii) IMMEDIATE ADDRESSING

In Immediate Addressing the effective address of the data is the location immediately following the opcode: the data to be used in the instruction immediately follows the opcode of the instruction. The MC6809 uses both 8

and 16 bit immediate values depending on the size of argument specified by the opcode. Examples of instructions with immediate Addressing are:
LDA#$20
LDX#$ F000

(iii) EXTENDED ADDRESSING
In Extended Addressing the contents of the two bytes immediately following the opcode fully specify the 16 bit effective address used by the instruction. Note that the address generated by an extended instruction defines an absolute address and is not position independent

(iv) EXTENDED INDIRECT
As a special case of indexed addressing (discussed below) one level of indirection may be added to Extended Addressing In Extended indirect, the two bytes following the postbyte of an Indexed instruction contains the address of the address of the data.

(v) DIRECT ADDRESSING
Direct addressing is similar to extended addressing except that only one byte of address follows the opcode. This byte specifies the lower 8 bits of the address to be used. The upper 8 bits of the address are supplied by the direct page register. Since only one byte of address is required in direct addressing, this mode requires less memory and executes faster than extended addressing. Of course, only 256 locations (one page) can be accessed without redefining the contents of the DP register; the DP register is set to $00 on Reset. Indirection is not allowed in direct addressing.

(vi) REGISTER ADDRESSING
Some opcodes are followed by a byte that defines a register or set of registers to be used by the instruction, this is called a POSTBYTE . Some examples of register addressing are:

TFR	XY	Transfers X into Y
EXG	AB	Exchanges A with B
PSHS	ABXY	Push onto SY,X,B then A
PULU	X,Y,D	Pull from U D,X then Y

(vii) INDEXED ADDRESSING

In all indexed addressing one of the pointer registers (X,Y,U,S and sometimes PC) is used in a calculation of the effective address of the operand to be used by the instruction. Five basic types of indexing are available and are discussed below. The postbyte of an indexed instruction specifies the basic type and variation of the addressing mode as well as the pointer register to be used.

ZERO OFFSET INDEXED

In this mode, the selected pointer register contains the effective address of the data to be used by the instruction. This is the fastest indexing mode.

CONSTANT OFFSET INDEXED

In this mode a two's complement offset and the contents of one of the pointer registers are added to form the effective address of the operand. The pointer register's initial content is unchanged by the addition. Three sizes of offsets are available

± 4 bit (-16 to + 15)

± 7 bit (-128 to+127)

± 15 bit (-32768 to+32767)

The two's complement 5 bit offset is included in the postbyte and therefore is most efficient in use of bytes and cycles. The two's complement 8-bit offset is contained in a single byte following the postbyte. The two's complement 16 bit offset is in the two bytes following the postbyte.

Examples of constant- offset indexing are:

LDA 23,X

LDX 2,S

LDY 300,X

ACCUMULATOR OFFSET INDEXED

This mode is similar to constant offset indexed except that the two's complement value in one of the accumulators (A,B or D) and the content of one of the pointer registers are added to form the effective address of the operand. The contents of both the accumulator and the pointer register are unchanged by the addition. The postbyte specifies which accumulator to use as an offset and no additional bytes are required. The advantage of an accumulator offset is that the value of the offset can be calculated by program at run-time.

AUTO INCREMENT/DECREMENT INDEXED

In the auto increment addressing mode, the pointer register contains the address of the operand. Then, after the pointer register is used it is incremented by one or two. This addressing mode is useful in stepping through tables, moving data, or for the creation of sowtware stacks. In auto decrement, the pointer register is decremented prior to use as the address of the data. The use of auto decrement is similar to that of auto increment but the tables, etc. are scanned from the high to low addresses. The size of the increment/decrement can be either one or two to allow for tables of either 8 or 16 bit data to be accessed and is selectable by the programmer. The pre-decrement, post-increment nature of these modes allow them to be used to create additional software stacks that behave identically to the U and S stacks.

(viii) INDEXED INDIRECT

All of the indexing modes with the exception of auto increment/decrement by one, or a ± 4 bit offset may have an additional level of indirection specified. In indirect addressing, the effective address is contained at the location specified by the content of the index register plus any offset. In the example below, the A accumulator is loaded indirectly using an effective address calculated from the index register and an offset.

```
                    Before Execution
                    A : XX (don't care)
                        X : $F000
$0100   LDA     (10.X)  EA is now $F010

$F010   $F1             F150 is now the
$F011   $50             new EA

$F150   $AA           After Execution
                    A : $AA Actual Data  Loaded
```

All modes of indexed indirect are included except those which are meaningless (e.g. auto increment/decrement by 1 indirect)

(ix) RELATIVE ADDRESSING

The byte(s) following the branch opcode is (are) treated as a singned offset which is added to the program counter, if the branch condition is true then the calculated address (PC signed offset) is loaded into the program counter. Program execution continues at the new location as indicated by the PC, Short (1 byte offset) and long (2 bytes offset) relative addressing modes are available. All of memory can be reached in long relative addressing as an effective address is interpreted modulo 2^{16}.

(x) PROGRAM COUNTER RELATIVE

The PC can be used as the pointer register with 8 or 16 bit signed offsets. As in relative addressing the offset is added to the current PC to create the effective address. The effective address is then used as the address of the operand or data. Program Counter Relative Addressing is used for writing position independent programs. Tables related to a particular routine will maintain the same relationship after the routine is moved, If referenced relative to the Program Counter.

2.2.4. MC 6809 INSTRUCTION SET

(i) 8 BIT ACCUMULATOR AND MEMORY INSTRUCTIONS

Mnemonic(s)	Operation
ADCA, ADCB	Add memory to accumulator with carry
ADDA, ADDB	Add memory to accumulator
ANDA, ANDB	And memory with accumulator
ASL,ASLA,ASLB	Arithmetic shift of accumulator or memory left
ASR,ASRA,ASRB	Arithmetic shift of accumulator or memory right
BITA, BITB	Bit test memory with accumulator
CLR,CLRA,CLRB	Clear accumulator or memory location
CMPA,CMPB	Compare memory from accumulator
COM,COMA,COMB	Complement accumulator or memory location
DAA	Decimal adjust A accumulator
DEC,DECA,DECB	Decrement accumulator or memory location
EORA,EORB	Exclusive or memory with accumulator
EXG R1,R2	Exchange R1 with R2 (R1,R2 : A,B,CC,DP)
INC,INCA,INCB	INcrement accumulator or memory location
LDA,LDB	Load accumulator from memory
LSL,LSLA,LSLB	Logical shift left accumulator or memory location
LSR,LSRA,LSRB	Logical shift right accumulator or memory location
MUL	Unsigned multiply (AxB-D)
NEG,NEGA,NEGB	Negate accumulator or memory
ORA,ORB	Or memory with accumulator
ROL,ROLA,ROLB	Rotate accumulator or memory left
ROR,RORA,RORB	Rotate accumulator or memory right
SBCA,SBCB	Subtract memory from accumulator with borrow
STA,STB	Store accumulator to memory
SUBA,SUBB	Subtract memory from accumulator
TST,TSTA,TSTB	Test accumulator or memory location
TFR,R1,R2	Transfer R1 to R2 (R1,R2 : A,B,CC,DP)

NOTE: A,B,CC or DP may be pushed to (pulled from) either stack with PSHS,PSHU, (PULS,PULU) instructions.

(ii) 16-BIT ACCUMULATOR AND MEMORY INSTRUCTIONS

Mnemonic(s)	*Operation*
ADDD	*Add memory to D accumulator*
CMPD	*Compare memory from D accumulator*
EXG D,R	*Exchange D with X,Y,S,U or PC*
LDD	*Load D accumulator from memory*
SEX	*Sign Extend B accumulator into A accumulator*
STD	*Store D accumulator to memory*
SUBD	*Subtract memory from D accumulator*
TFR D,R	*Transfer D to X,Y,S,U or PC*
TFR R,D	*Transfer X,Y,S,U or PC to D*

(iii) INDEX REGISTER/STACK POINTER INSTRUCTIONS

Mnemonic(s)	*Operation*
CMPS,CMPU	*Compare memory from stack pointer*
CMPX,CMPY	*Compare memory from index register*
EXG R1,R2	*Exchange D,X,Y,S,U or PC with D,X,Y,S,U or PC*
LEAS, LEAU	*Load effective address into stack pointer*
LEAX,LEAY	*Load effective address into index register*
LDS,LDU	*Load stack pointer from memory*
LDX,LDY	*Load index register from memory*
PSHS	*Push any register(s) onto hardware stack (except S)*
PSHU	*Push any register(s) onto user stack (except U)*
PULS	*Pull any register(s) from hardware stack (except S)*
PULU	*Pull any register(s) from hardware stack (except U)*
STS,STU	*Store stack pointer to memory*
STX,STY	*Store index register to memory*
TFR,R1,R2	*Transfer D,X,Y,S,U or PC to D X,Y,S,U or PC*
ABX	*Add B accumulator to X (unsigned)*

(iv) BRANCH INSTRUCTIONS

Mnemonic(s)	*Operation*
BCC LBCC	*Branch if carry clear*
BCS LBCS	*Branch it carry set*
BEX LBEQ	*Branch if equal*
BGE,LBGE	*Branch if greater than or equal (signed)*
BGT LBGT	*Branch if greater (signed)*
BHL LBHI	*Branch if higher (unsigned)*
BHS,LBHS	*Branch if higher or same (unsigned)*
BLE,LBLE	*Branch if less than or equal (signed)*
BLO,LBLO	*Branch if lower (unsigned)*
BLS,LBLS	*Branch if lower or same (unsigned)*
BLT,LBLT	*Branch if less than (signed)*
BMI,LBMI	*Branch if minus*
BNE,LBNE	*Branch if not equal*
BPL,LBPL	*Branch if plus*
BRA,LBRA	*Branch always*
BRN,LBRN	*Branch never*
BSR,LBSA	*Branch to subroutine*
BVC,LBVC	*Branch if overflow clear*
BVS,LBVS	*Branch if overflow set*

(v) MISCELLANEOUS INSTRUCTIONS

Mnemonic(s)	*Operation*
ANDCC	*AND condition code register*
CWAI	*AND condition code register than wait for interrupt*
NOP	*No operation*
ORCC	*OR condition code register*
JMP	*Jump*
JSR	*Jump to subroutine*
RTI	*Return from interrupt*
RTS	*Return from subroutine*
SWI,SWI2,SWI3	*Software interrupt (absolute indirect)*
SYNC	*Synchronize with interrupt line*

2.3. THE CRT CONTROLLER MC 6845

The MC6845 CRT Controller performs the interface to raster scan CRT displays. It is intended for use in processor-based controllers for CRT terminals in stand-alone or cluster configurations.

The CRTC is optimized for hardware/software balance in order to achieve integration of all key functions and maintain flexibility. For instance, all keyboard functions, R/W, cursor movements, and editing are under processor control: whereas the CRTC provides video timing and Refresh Memory Addressing.

The primary function of the CRTC is to generate refresh addresses (MA0-MA13) row selects (RA0-RA4), and video monitor timing (HSYNC, VSYNC) and Display Enable. Other functions include an internal cursor register which generates a Cursor output when its contents compare to the current Refresh Address. A light-pen strobe input signal allows capture of Refresh Address in an internal light pen register.

All timing in the CRTC is derived from the Clk input. In alphanumeric terminals, this signal is the character rate.

The processor communicates with the CRTC through a buffered 8-bit Data Bus by reading/writing into the 18-register file of the CRTC.

The Refresh Memory address is multiplexed between the Processor and CRTC. Data appears on a Secondary Bus which is buffered from the processor Primary Bus. A number of approaches are possible for solving contentions for the Refresh Memory.

1. Processor always gets priority
2. Processor gets priority access anytime, but can be synchronized by an interrupt to perform accesses only during horizontal and vertical retrace times.
3. Synchronize processor by memory wait cycles.

4. Synchronize processor to character rate. The 6800 MPU family lends itself to this configuration because it has constant cycle lengths. This method provides zero burden on the processor because there is never a contention for memory. All accesses are "transparent".

The secondary data bus concept in no way precludes using the Refresh RAM for other purposes. It looks like any other RAM to the Processor. For example, using Approach 4, a 64K byte RAM Refresh Memory could perform refresh and program storage functions transparently.

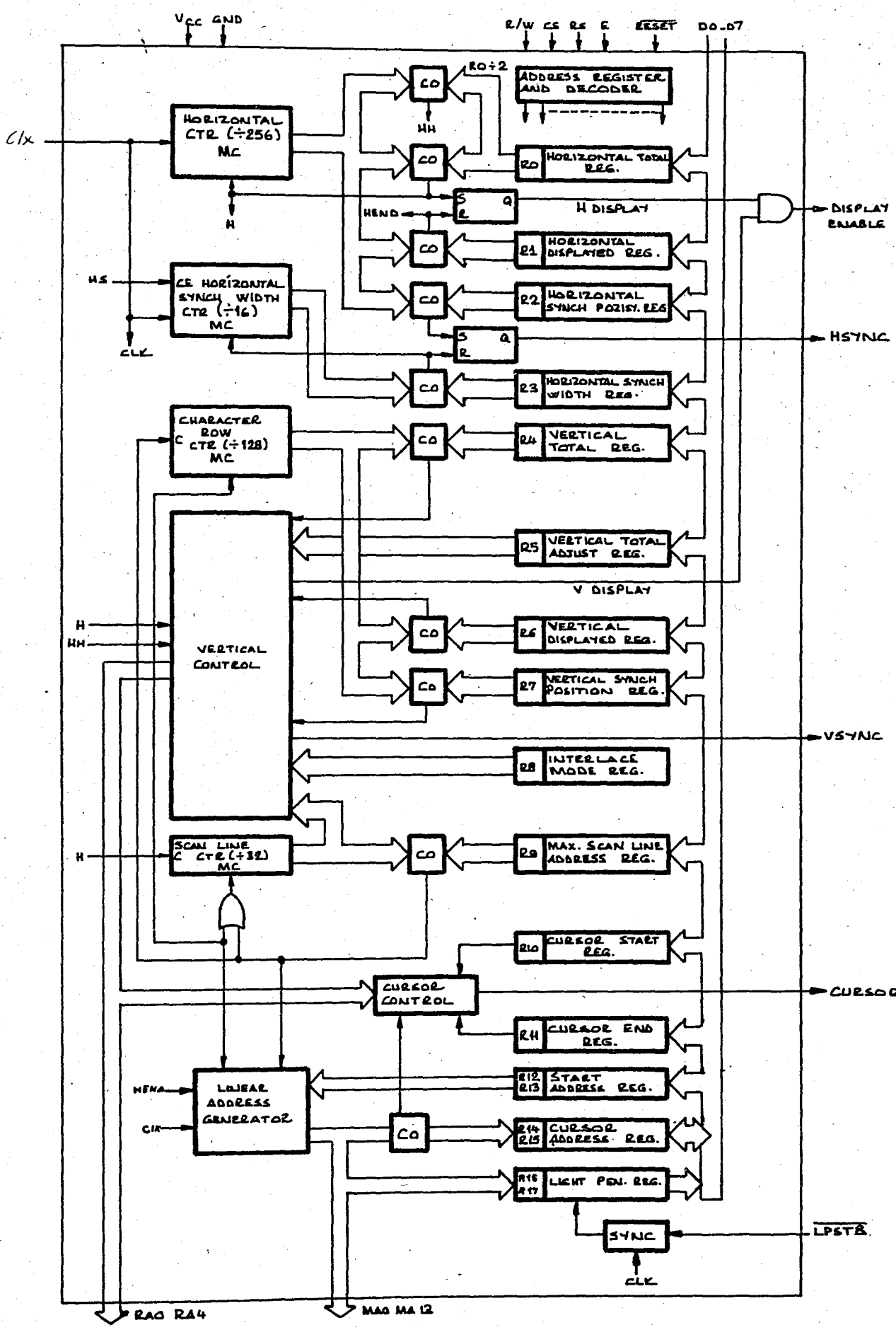
The CRTC consists of programmable horizontal and vertical timing generators, programmable linear address register, programmable cursor logic, light pen capture register, and control circuitry for interface to a processor bus (Figure 16).

All CRTC timing is derived from CLK, usually the output of an external dot rate counter. Coincidence (CO) circuits continuously compare counter contents to the contents of the programmable register file, R0-R17. For horizontal timing generation, comparisons result in: 1)Horizontal sync pulse (HS) of a frequency, position and width determined by the registers 2) Horizontal Display Signal of a frequency,position, and duration determined by the registers.

The Horizontal counter produces H clock which drives the Scan Line Counter and Vertical Control. The contents of the Raster Counter are continuously compared to the Max Scan Line Address Register. A coincidence resets the Raster Counter and clocks the Vertical Counter.

Comparisons of Vertical Counter contents and Vertical Registers result in: 1) Vertical sync pulse (VS) of a frequency and position determined by the registers - the width is fixed at 16 raster lines in the vertical control section and is not programmable 2) Vertical Display of a frequency and position determined by the registers.

FIGURE 16 CRTC BLOCK DIAGRAM



The Vertical Control Logic has other functions.

1. Generate row selects, RA0-RA4, from the Raster Count for the corresponding interlace or non-interlace modes.
2. Extend the number of scan lines in the vertical total by the amount programmed in the Vertical Total Adjust Register.

The Linear Address Generator is driven by CLK and locates therelative positions of characters in memory with their positions on the screen. Fourteen lines, MA0-MA13, are available for addressing up to four pages of 4K characters, 8 pages of 2K characters, etc. Using the Start Address Register, hardware scrolling through 16K characters is possible. The Linear Address Generator repeats the same sequence of addresses for each scan line of a character row.

The cursor logic determines the cursor location, size, and blinking rate on the screen. All are programmable.

The light pen strobe going high causes the current contents of the Address Counter to be latched in the Light Pen Register. The contents of the Light Pen Register are subsequently read by the Processor.

Internal CRTC registers are programmed by the processor through the data bus, D0-D7, and the control signals $R/\overline{W}$, $\overline{CS}$, RS and E.

2.3.1. CRT CONTROLLER PIN DESCRIPTION

(i) PROCESSOR INTERFACE

The CRTC interfaces to a processor bus, on the bidirectional data bus (D0-D7) using $\overline{CS}$, RS, E and R/$\overline{W}$ for control signals.

Data Bus (D0-D7) - The bidirectional datalines (D0-D7) allow data transfers between the CRTC internal Register File and the processor. Data bus output drivers are 3-state buffers which remain in the high impedance state except when the processor performs a CRTC read operation. A high level on a data pin is a logical "1".

Enable (E) - The Enable signal is a high impedance TTL/MOS compatible input which enables the data bus input/output buffers and clocks data to and from the CRTC. This signal is usually derived from the processor clock and the high to low transition is the active edge.

Chip Select ($\overline{CS}$) - The $\overline{CS}$ line is a high impedance TTL/MOS compatible input which selects the CRTC when low to read or write the internal Register File. This signal should only be active when there is a valid stable address being decoded from the processor.

Register Select (RS) The RS line is a high impedance TTL/MOS compatible input which selects either the Address Register (RS: "0") or one of the Data Registers (RS: "1") of the internal Register File.

Read/Write (R/$\overline{W}$) - The R/$\overline{W}$ line is a high impedance TTL/MOS compatible input which determines whether the internal Register File gets written or read. A write is active low ("0")

(ii) CRT CONTROL

The CRTC provides horizontal sync (HS), vertical sync (VS), and Display Enable signals.

Vertical Sync (V SYNC) - This TTL compatible output is an active high signal which drives the monitor directly or is fed to Video Processing Logic for composite generation. This signal determines the vertical position of the displayed text.

Horizontal Sync (H SYNC) - This TTL compatible output is an active high signal which drives the monitor directly or is fed to Video Processing Logic for composite generation. This signal determines the horizontal position of the displayed text.

Display Enable - This TTL compatible output is an active high signal which indicates the CRTC is providing addressing in the active Display Area.

(iv) REFRESH MEMORY/CHARACTER GENERATOR ADDRESSING

The CRTC provides Memory Addresses (MAO-MA 13) to scan the Refresh RAM. Also provided are Raster Addresses (RAO-RA4) for the character ROM.

Refresh Memory Addresses (MAO-MA13) These 14 outputs are used to refresh the CRT screen with pages of data located within a 16K block of refresh memory. These outputs drive a TTL load and 30pF. A high level on MAO-MA13 is a logical "1".

Raster Addresses (RAO-RA4) These 5 outputs from the internal Raster Counter addresses the Character ROM for the row of a character. These outputs drive a TTL load and 30pF. A high level (on RAO-RA4) is a logical "1".

(v) OTHER PINS

Cursor - This TTL compatible output indicates Cursor Display to extarnal Video Processing Logic. Active high signal.

Clock (CLK) - The CLK TTL/MOS compatible input is used to synchronize all CRT control signals. An external dot counter is used to derive this signal which is usually the character rate in an alphanumeric CRT. The active transition is high to low.

Light Pen Strobe (LPSTR) This high impedance TTL/MOS compatible input latches the current Refresh Addresses in the Register File. Latching is on the low to high edge and is synchronized internally to character clock.

$\overline{RES}$ — The $\overline{RES}$ input is used to Reset the CRTC. An input low level on $\overline{RES}$ forces CRTC into following status:

(A) All the counters in CRTC are cleared and the device stops the display operation.

(B) All the outputs go down to low level

(C) Control registers in CRTC are not affected and remain unchanged.

This signal is different from other M6800 family in the following functions:

(A) $\overline{RES}$ signal has capabilty of reset function only when LPSTB is at low level

(B) After $\overline{RES}$ has gone down to low level, output signals of MAO-MA13 and RAO-RA4, synchronizing with CLK low level, goes down to low level. (At least 1 cycle CLK signal is necessary for reset)

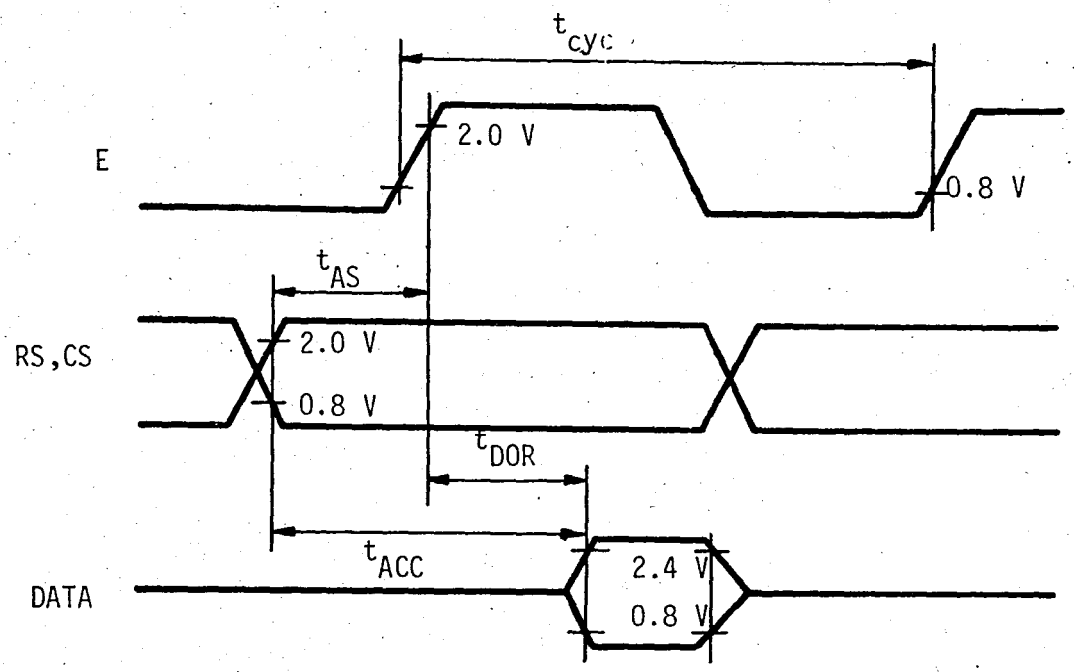
(C) The CRTC starts the Display operation immediately after the release of $\overline{RES}$ signal.

TABLE 2 CRTC OPERATING MODE

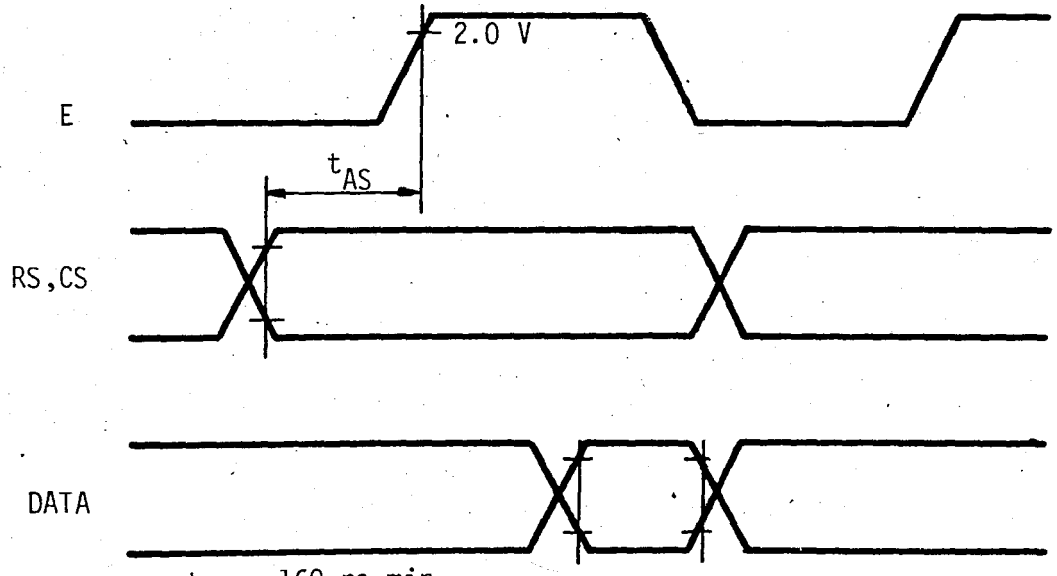
RES	*LPSTB*	*OPERATING MODE*
0	*0*	*Reset*
0	*1*	*Test Mode*
1	*0*	*Normal Mode*
1	*1*	*Normal Mode*

2.3.2. CRTC TIMING

BUS READ TIMING (READ FROM CRTC)



BUS WRITE TIMING (WRITE INTO CRTC)



t_{AS} : 160 ns min.
t_{DOR} : 320 ns max.
t_{ACC} X 480 ns max.

2.3.3. CRT CONTROLLER REGISTER FILE DESCRIPTION

Nineteen registers in the CRTC can be accessed by means of the data bus.

(i) ADDRESS REGISTER

The Address Register is a 5 bit write-only register used as an "indirect" or "pointer" register. Its contents are the address of one of the other 18 registers in the file. When RS and CS are low the Address Register itself is addressed. When RS is high, the Register File is accessed.

(ii) HORIZONTAL TIMING REGISTERS RO, R1, R2, and R3

Figure 17 shows the visible display area of a typical CRT monitor giving the point of reference for horizontal registers as the left most displayed character position. Horizontal registers are programmed in "character time" units with respect to the reference.

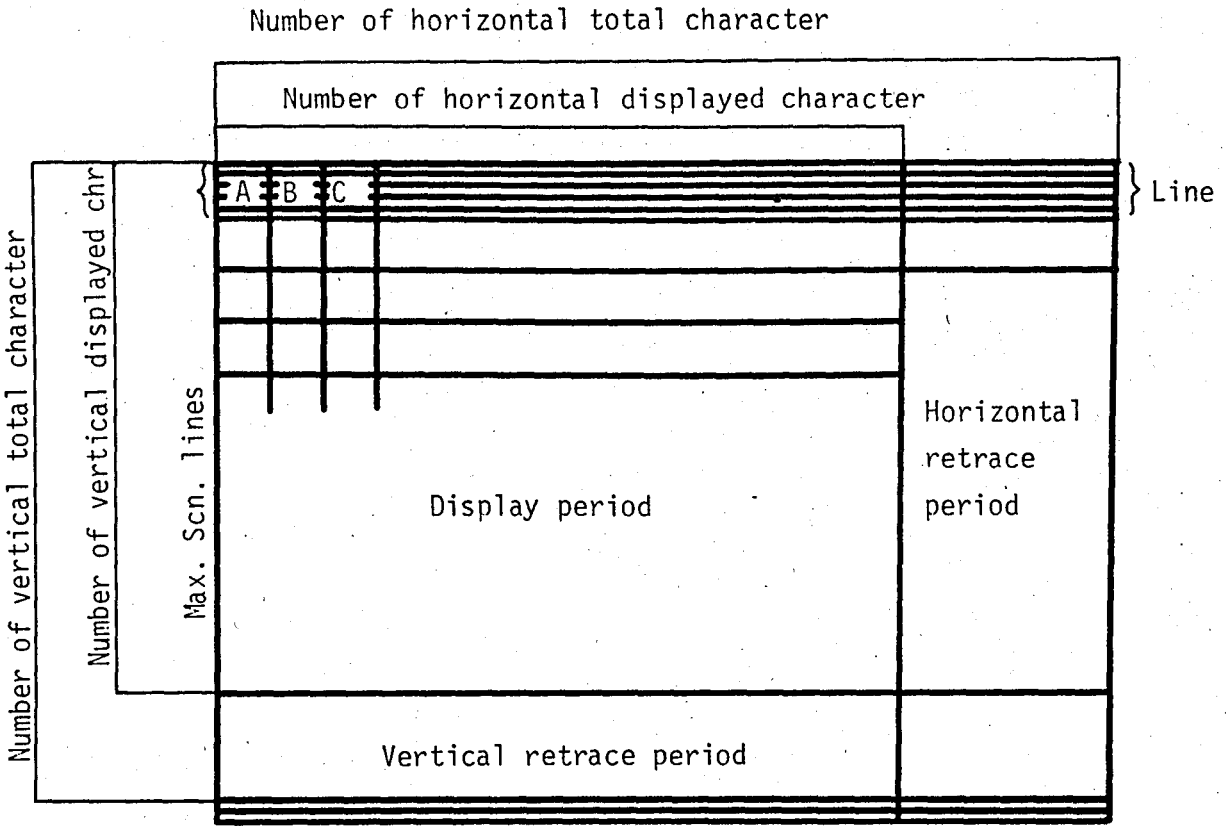
a) Horizontal Total Register (RO) - This 8 bit write only register determines the horizontal frequency of HS. It is the total of displayed plus non-displayed character time units minus one.

b) Horizontal Displayed Register (R1) - This 8 bit write only register determines the number of displayed characters per horizontal line.

c) Horizontal Sync Position Register (R2) - This 8 bit write-only register determines the horizontal sync position on the horizontal line.

d) Horizontal Sync Width Register (R3) - This 4 bit write-only register determines the width of the HS pulse. It may not be apparent why this width needs to be programmed. However, consider that all timing widths must be programmed as multiples of the character clock period which varies. If HS width were fixed as an integral number of character times, it would vary with character rate and be out of tolerance for certain monitors. The rate programmable feature allows compensating HS width.

FIGURE 17 : ILLUSTRATION OF THE CRT SCREEN FORMAT



(iii) VERTICAL TIMING REGISTERS R4, R5, R6, R7, R8 and R9

The point of reference for vertical registers is the top character position displayed. Vertical registers are programmed in character row times or scan line times.

a) Vertical total register (R4) and Vertical Total Adjust Register (R5) - The vertical frequency of VS is determined by both R4 and R5. The calculated number of character line times is usually an integer plus a fraction to get exactly a 50 or 60 Hz vertical refresh rate. The integer number of character line times minus one is programmed in the 7 bit write-only Vertical Total Register the fraction is programmed in the 5 bit write-only Vertical Scan Adjust Register as a number of scan line times.

b) Vertical Displayed Register (R6) - This 7 bit write-only register determines the number of displayed character rows on the CRT screen and is programmed in character row times.

c) Vertical Sync Position (R7) - This 7 bit write - only register determines the vertical sync position with respect to the reference. It is programmed in character row times.

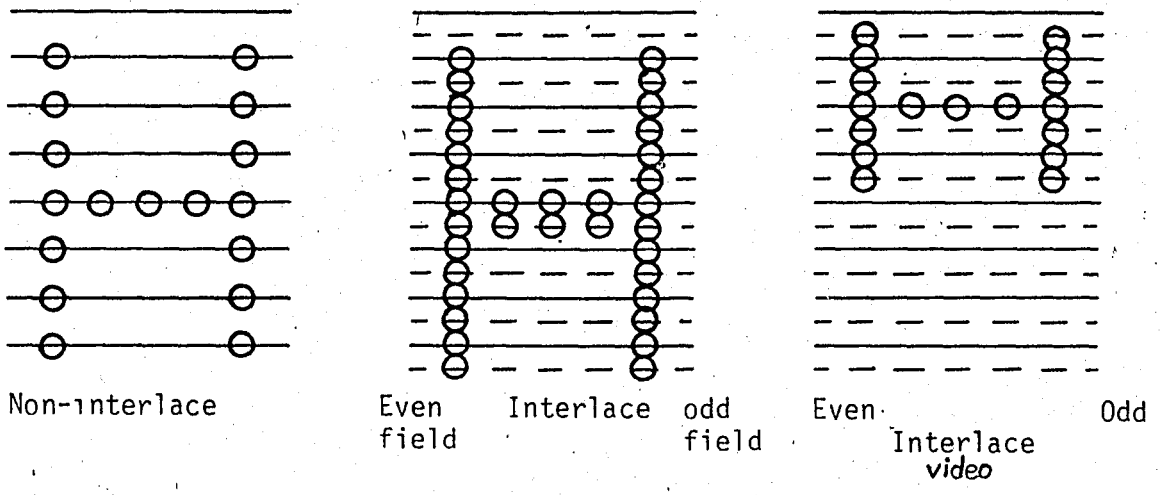
d) Interlace Mode Register (R8) - This 2 bit write only register controls the raster scan mode (see Figure 18) When bit 0 and bit 1 are reset, or bit 0 is reset and bit 1 set, the non-interlace raster scan mode is selected. Two interlace modes are available. Both are interlaced 2 fields per frame. When bit 0 is set and bit 1 is reset, the interlace sync raster scan mode is selected. Also when bit 0 and bit 1 are set, the interlace sync and video raster scan mode is selected.

e) Maximum Scan Line Address Register (R9) - This 5 bit write-only register determines the number of scan lines per character row including spacing. The programmed value is a max address and is one less than the number of scan lines.

FIGURE 18 - INTERLACE CONTROL

interlace mode reg.

B1	B0	Mode
0	0	Non-interlace
1	0	
0	1	Interlace
1	1	Interlace and video



(iv) OTHER REGISTERS

a) Cursor Start Register (R10) - This 7 bit write-only register controls the cursor format (see Figure 19). Bit 5 is the blink timing control. When bit 5 is low, the blink frequency is 1/16 of the vertical field rate, and when bit 5 is high, the blink frequency is 1/32 of the vertical field rate. Bit 6 is used to enable a blink. The cursor start scan line is set by the lower 5 bits.

b) Cursor End Register (R11) - This 5 bit write-only register sets the cursor end scan line.

c) Start Address Register (H L) (R12,R13) - Start Address Register is a 14 bit write-only register which determines the first address put out as a refresh address after vertical blanking. It consists of an 8 bit lower register, and a 6 bit higher register.

d) Light Pen Register (H & L) (R16, R17) - This 14 bit read-only register is used to store the contents of the Address Register (H & L) when the LPSTB input pulses high. This register consists of an 8 bit lower and 6 bit higher register.

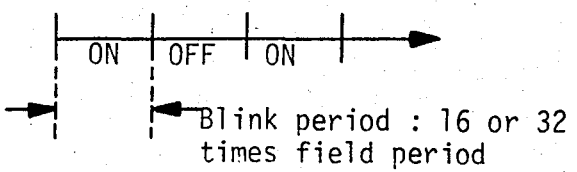
e) Cursor Register (H & L) (R14, R15) - This 14 bit read write register stores the cursor location. This register consists of an 8 bit lower and 6 bit higher register.

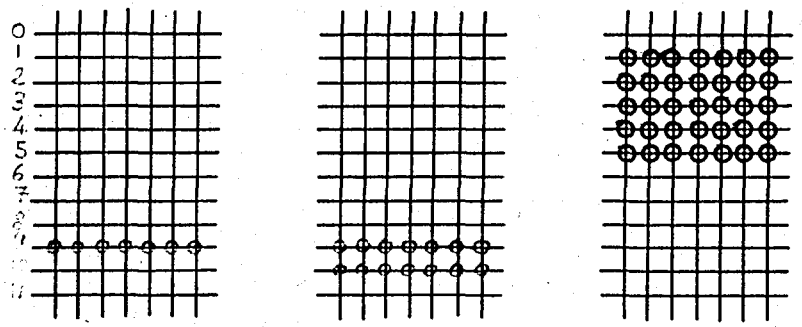
FIGURE 19 CUSROR CONTROL

Cursor start register

B P

Bit 6	Bit 5	Cursor display
0	0	Non-blink
0	1	Cursor non-
1	0	Blink 1/16
1	1	Blink 1/32





Cursor start add:9
Cursor end add:9

start:9
end :10

start:1
end:5

2.4. PROGRAMMABLE TIMER MODULE MC 6840

The MC6840 is a programmable subystem component of the M6800 family designed to provide variable system time intervals.

The MC6840 has three 16-bit binary counters, three corresponding control registers and a status register. These counters are under software control and may be used to cause system interrupts and/or generate output signals. The MC6840 may be utilized for such tasks as frequency measurements, event counting, interval measuring and similar tasks. The device may be used for square wave generation, gated delay signals, single pulses of controlled duration, and pulse width modulation as well as system interrupts.

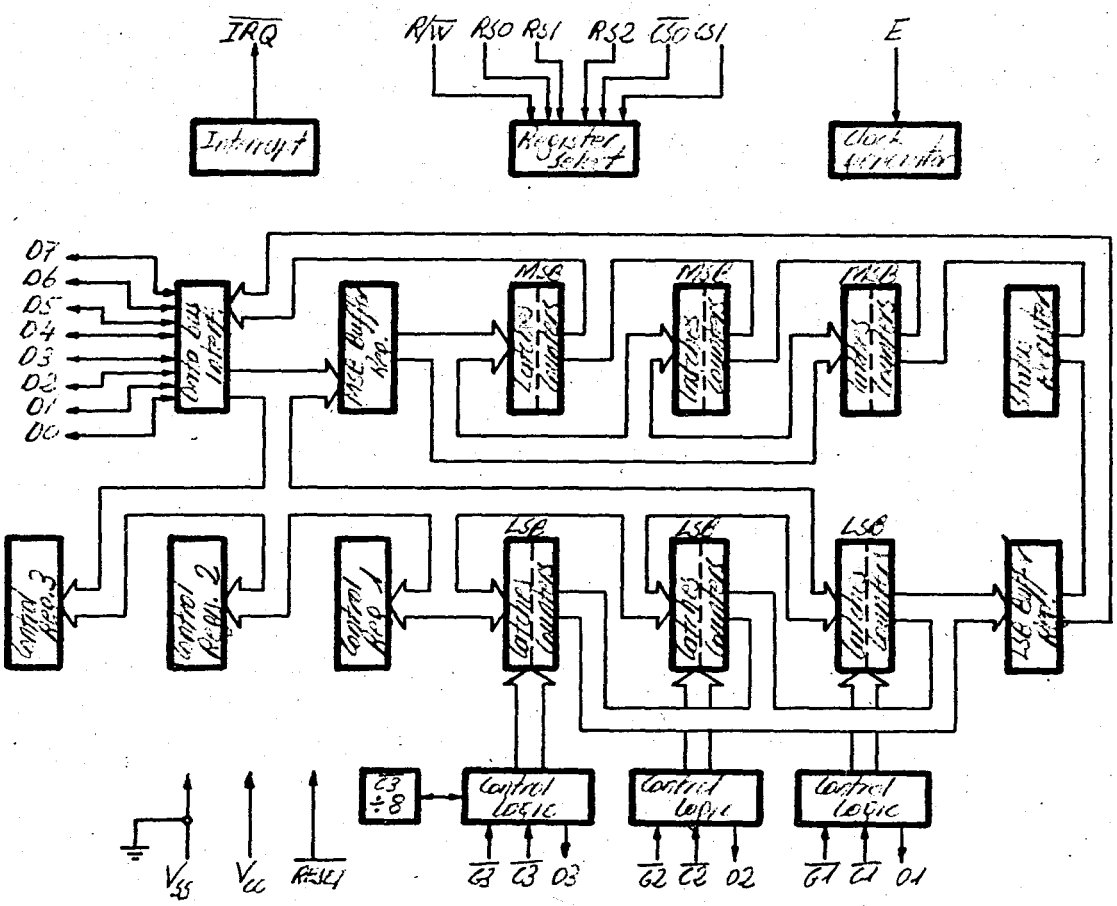


FIGURE 20 PTM BLOCK DIAGRAM

2.4.1. PTM OPERATION

The three timers in the MC6840 may be independently programmed to operate in modes which fit a wide variety of applicatinos. The device is fully bus compatible with M6800 systems and is accessed by load and store operations from the MPU in much the same manner as a memory device. In a typical application, a Timer will be loaded by first storing two bytes of data into an associates Counter Latch. This data is then transferred into the counter via a Counter Initialization cycle. The counter decrements on each subsequent clock period which may be an external clock or Enable (System Ø2) until one of several predetermined conditions causes it to halt or recycle. The timers are thus programmable, cyclic in nature, controllable by external inputs or the MPU program, and accessible by the MPU at any time.

(i) BUS INTERFACE

The Progrrambel Timer Module (PTM) interfaces to the M6800 Bus with an eight bit bidirectional data bus, two Chip Select lines, a Read/Write line, an Enabl (System Ø2) line, an Interrupt Request line, an external Reset Line, and three Register Select Lines.

a) Bidirectional Data (D0-D7) -The bidirectional data lines (D0-D7) allow the transfer of data between the MPU and PTM. The data bus output drivers are three-state devices which remain in the high-impedance (off) state except when the MPU performs a PTM read operation (Read/$\overline{\text{Write}}$ and Enable lines high and PTM Chip Selects activated).

b) Chip Select ($\overline{CS0}$,CS1) - These two signals are used to activate the Data Bus interface and allow transfer of data from the PTM. With $\overline{CS0}$: 0 and CS1 : 1, the device is selected and data transfer will occur.

c) Read/$\overline{\text{Write}}$ (R/$\overline{W}$) -This signal is generated by the MPU to control the direction of data transfer on the Data Bus: with the PTM selected, a low state on the PTM R/$\overline{W}$ line enables the input buffers and data is transferred from the MPU to the PTM on the trailing edge of the Enable (System Ø2) signal. Alternately, (under the same conditions) R/$\overline{W}$: 1 and Enable high allows data in the PTM to be read by the MPU.

d) Enable (System Ø2) - THis signal synchronizes data transfer between the MPU and the PTM. It also performs an equivalent synchronization function on the external clock, reset, and gate inputs of the PTM.

e) $\overline{\text{Interrupt Request}}$ ($\overline{IRQ}$) - The active low interrupt Request signal is normally tied directly (or through priority interrupt circuitry) to the $\overline{IRQ}$ input of the MPU. This is an "open drain" output (no load device on the chip) which permits other similar interrupt request lines to be tied together in a wire OR configuration.

The $\overline{IRQ}$ line is activated if, and only if, the Composite Interrupt Flag (Bit 7 of the Internal Status Register) is asserted. The conditions under which the $\overline{IRQ}$ line is activated are discussed in conjunction with the Status Register.

f) External $\overline{\text{Reset}}$ - A low level at this input is clocked into the PTM by the Enable (System Ø2) input. Two Enable pulses are required to synchronize and process the signal. The PTM then recognizes the active "low" or inactive "high" on the third Enable pulse. If the Reset signal is asynchronous, an additional Enable period is required if setup times are not met.

Recognition of a low level at this input by the PTM causes the following action to occur:

1. *All counter latches are preset to their maximal count values.*
2. *All Control Register bits are cleared with the exception of CR10 (internal reset bit) which is set.*
3. *All counters are preset to the contents of the latches.*
4. *All counter outputs are reset and all counter clocks are disabled.*
5. *All Status Register bits (interrupt flags) are cleared.*

g) Register Select Lines (RS0, RS1, RS2) These inputs are used in conjunctio with the R/W line to select the internal registers, counters and latches as shown in Table 4.

It has been previously stated that the PTM is accessed via MPU Load and Store operations in much the same manner as a memory device. The instructions available with the M6800 family of MPUs which perform operations directly on memory should not be used when the PTM is accessed. These instructions actuall fetch a byte from memory, perform an operation, then restore it to the same address location. Since the PTM used the R/$\overline{W}$ line as an additional register select input, the modified data may not be restored to the same register if these instructions are used.

TABLE 4 REGISTER SELECTION

Register Select Inputs			Operations	
RS2	RS1	RS0	R/$\overline{W}$: 0	R/$\overline{W}$: 1
0	0	0	CR20 : 0Write Control Register CR20 : 1 Write Control Register	No operation
0	0	1	Write control register 2	Read status register
0	1	0	Write MSB Buffer register	Read timer 1 counter
0	1	1	Write timer 1 latches	Read LSB buffer regist
1	0	0	Write MSB Buffer register	Read timer 2 counter
1	0	1	Write timer 2 latches	Read LSB Buffer regist
1	1	0	Write MSB buffer register	Read timer 3 counter
1	1	1	Write timer 3 latches	Read LSB buffer regist

(ii) CONTROL REGISTER

Three Write-Only registers in the MC6840 are used to modify timer operation to suit a variety of applications. Control Register 2 has a unique address spa (RSO : 1, RSI : 0, RS2 : 0) and therefore may be written into at any time. The remaining Control Registers (1 and 3) share the Address Space selected by a logic zero on all Register Select inputs. The least significant bit of Control Register 2 (CR20) is used as an additional addressing bit for Control Registers 1 and 3. Thus, with all Register Selects and R/$\overline{W}$ inputs at logic zero, Control Register 1 will be written into if CR20 is a logic one. Under the same conditions, Control Register 3 will be written into if CR20 is a logic zero. Control Register 3 can also be written into after a Reset low condition has occured, since all control register bits (except CR10) are cleared. Therefore one may write in the sequence CR3, CR2, CR1, The least significant bit of Control Register 1 is used as an Internal Reset bit. When this bit is a logic zero, all timers are allowed to operate in the modes prescribed by the remaining bits of the control registers. Writing a 'one' into CR10 causes all counters to be preset with the contents of the corresponding counter latches, all counter clocks to be disabled, and the timer outputs and interrupt flags (Status, Register) to be reset. Counter Latches and Control Registers are undisturbed by an Internal Reset and may be written into regardless of the state of CR10.

The least significant bit of Control Register 3 is used as a selector for a ÷8 prescaler which is available with Timer 3 only. The prescaler, if select is effectively placed between the clock input circuitry and the input to Counte 3. It can therefore be used with either the internal clock (Enable) or an external clock source.

The functions depicted in the foregoing discussions are tabulated on the first row in Table 5 for ease of reference.

TABLE 5 CONTROL REGISTER BITS

CR 10 Internal Reset Bit	CR20 Control Register Address Bit	CR 30 Timer 3 Clock Control
0 All timers allowed to operate	0 CR 3 may be written	0 T3 Clock is not prescaled
1 All timers held in preset state	1 CR 1 may be written	1 T3 Clock is prescaled by 8

CRX1 [x]	Timer X Clock Source
0	TX uses external clock source on CXinput
1	TX uses Enable clock
CRX2	Timer X Counting Mode Control
0	TX configured for normal (16 bit) counting mode
1	TX configured for dual 8-bit counting mode
CRX3 CRX4 CRX5	Timer X Counter Mode and Interrupt Control (See Table 6)
CRX6	Timer X interrupt Enable
0	Interrupt Flag masked on IRQ
1	Interrupt Flag enabled to IRQ
CRX7	Timer X Counter Output Enable
0	TX Output masked on output OX
1	TX Output enabled on output OX

[x] Control Register for Timer 1,2, or 3, Bit 1.

Control Register Bits CR10 CR20 and CR30 are unique in that each selects a different function. The remaining bits (1 through 7) of each Control Register select common functions with a particular Control Register affecting only its corresponding timer. For example, Bit 1 of Control Register 1 (CR11) selects whether an internal or external clock source is to be used with Timer 1. Similarly, CR21 selects the clock source for Timer 2, and CR31 performs this function for Timer 3. The function of each bit of Control Register "X" can therefore be defined as the remaining section of Table 5.

Control Register Bit 2 selects whether the binary information contained in the Counter Latches (and subsequently loaded into the Counter) is to be treated as a single 16 bit word or two 8 bit bytes. In the single 16 bit Counter Mode (CRX2 : 0) the counter will decrement to zero after N + 1 enabled ($\overline{G}$: 0) clock periods, where N is defined as the 16 bit number in the Counter Latches. With CRX2 : 1, a similar Time Out will occur after (L + 1)·(M + 1) enabled clock periods, where L and M respectively, refer to the LSB and MSB bytes in the Counter Latches.

Control Register Bits 3,4 and 5 are explained in detail in the Timer Operating Mode section. Bit 6 is an interrupt mask bit which will be explained more fully in conjunction with the Status Register, and bit 7 is used to enable the corresponding Timer Output. A summary of the control register programming modes is shown in Table 6.

(iii) STATUS REGISTER/INTERRUPT FLAGS

The MC6840 has an internal Read-Only Status Register which contains four interrupt Flags. (The remaining four bits of the register are not used, and default to zeros when being read) Bits 0,1 and 2 are assigned to Timers 1, 2 and 3 respectively as individual flag bits, while Bit 7 is a Composite INterrupt Flag. This flag bit will be asserted if any of the individual flag bits is set while Bit 6 of the corresponding Control Register is at a logic one. The conditions for asserting the Composite Interrupt Flag bit can therefore be expressed as:

INT : I1·CR16 + I2·CR 26 + I3. CR36

where INT : Composite Interrupt Flag (Bit 7)

I1 : Timer 1 Interrupt Flag (Bit 0)

I2 : Timer 2 INterrupt Flag (Bit 1)

I3 : Timer 3 Interrupt Flag (Bit 2)

An interrupt flag is cleared by a Timer Reset condition, i.e. External Reset : 0 or Internal Reset Bit (CR10) : 1. It will also be cleared by a Read Timer Conuter Command provided that the Status Register has previously been read while the interrupt flag was set. This condition on the Read Status Register - Read Timer Counter (RS-RT) sequence is designed to prevent missing interrupts which might occur after the status register is read, but prior to reading the Timer Counter.

An Individual Interrupt. Flag is also cleared by a Write Timer Latches (W) command or a Counter Initialization (C1) sequence, provided that W or C1 affects the Timer corresponding to the individual interrupt Flag.

(iv) COUNTER LATCH INITIALIZATION

Each of the three independent timers consists of a 16 - bit addressable counter and 16 bits of addressable latches. The counters arepreset to the binary numbers stored in the latches. Counter initialization results in the transfer of the latch contents to the counter. See notes in Table 8 regarding the binary number N,L or M placed into the Latches and their relationship to the output waveforms and counter Time-Outs.

Since the PTM data bus is 8 bits wide and the counters are 16 bits wide, a temporary register (MSB Buffer Register) is provided. This "write only"register is for the Most Significant Byte of the desired latch data. Three addresses are provided for the MSB Buffer Register (as indicated in Table 4) but they all lead to the same Buffer Data from the MSB Buffer will automatically be transferred into the Most Signicicant Byte of Timer X when a Write Timer X Latches Command is performed. So it can be seen that the MC6840 has been designed to allow transfer of two bytes of data into the counter latches provided that the MSB is transferred first.

In many applications, the source of the datawill be an MC6800 MPU. It should be noted that the 16 bitstore operations of the M6800 family microprocessors (STS and STX) transfer data in the order required by the PTM. A store index register instruction, for example, results in the MSB of the X register being transferred to the selected address, then the LSB of the X register being written into the next higher location. Thus, either the index register or stack pointer may be transferred directly into a selected counter latch with a single instruction.

A logic zero at the Reset input also initializes the counter latches. In this case, all latches will assume a maximum count of 65.536. It is important to note that an Internal Reset (Bit zero of Control Register 1 set) has no effect on the counter latches.

(v) COUNTER INITIALIZATION

Counter Initialization is defined as the transfer of data from the latches to the counter with subsequent clearing of the Individual Interrupt Flag associated with the counter. Counter Initialization always occurs when a reset condition (Reset : 0 or CR10 : 1) is recognized, It can also occur - depending on Timer Mode - with a Write Timer Latches command or recognition of a negative transition of the Gate input.

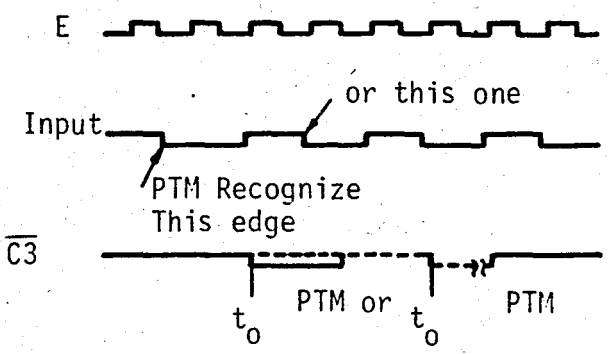
Counter recycling or re-initialization occurs when a negative transition of the clock input is recognized after the counter has reached an all-zero state. In this case, data is transferred from the Latches to the Counter.

(vi) ASYNCHRONOUS INPUT/OUTPUT LINES

Each of the three timers within the PTM has external clock and gate inputs as well as a counter output line. The inputs arehigh impedance. TTL compatible lines and outputs are capable of driving two standard TTL loads.

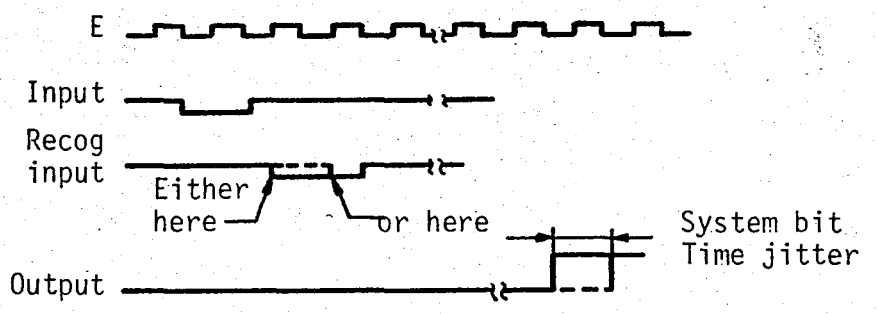
a) Clock Inputs ($\overline{C1}$,$\overline{C2}$ and $\overline{C3}$) Input pins $\overline{CL}$,$\overline{C2}$,and $\overline{C3}$ will accept asynchronous TTL voltage level signals to decrement Timers 1,2 and 3, respectively. The high and low levels of the external clocks must each be stable for at least one system clock period plus the sum of the setup and hold times for the inputs. The asynchronous clock rate can vary from dc to the limit imposed by Enable (System ϕ2) Setup, and Hold time.

The external clock inputs are clocked in by Enable (System ϕ2) pulses. Three Enable periods are used to synchronize and process the external clock. The fourth Enable pulse decrements the internal counter. This does not affect the input frequency, it merely creates a delay between a clock input transition and internal recognition of that transition by the PTM. All references to C inputs in this report relate to internal recognition of the input transition. Note that a clock high or low level which does not meet setup and hold time specifications may require an additional Enable pulse for recognition. When observing recurring events a lack of synchronization will result in "jitter" being observed on the output of the PTM when using asynchronous cloks and gate input signals. There are two types of jitter. "System jitter" is the result of the input signals being out of synchronization with the Enable (System ϕ2). permitting signals with marginal setup and hold time to be recognized by either the bit time nearest the input transition or the subsequent bit time.



"Input jitter" can be as great as the time between input signal negative going transitions plus the system jitter, if the first transition is recognized during one system cycle, and not recognized the next cycle, or vice versa.

External clock input C3 represents a special case when Timer 3 is programmed to utilize its optional ÷8 prescaler mode. The maximum input frequency and



allowable duty cycles for this case are specified under the AC Operating Characteristics. The output of the ÷8 prescaler is treated in the same manner as the previously discussed clock inputs. That is it is clocked into the counter by Enable pulses, is recognized on the fourth Enable pulse (provided setup and hold time requirements are met), and must produce an output pulse at least as wide as the sum of an Enable period, setup and hold times.

b) Gate Inputs ($\overline{G1}$,$\overline{G2}$,$\overline{G3}$) Input pins $\overline{G1}$, $\overline{G2}$ and $\overline{G3}$ accept asynchronous TTL compatible signals which are used as triggers or clock gating functions to Timers 1, 2 and 3 respectively. The gating inputs are clocked into the PTM by the Enable (System ϕ2) signal in the same manner as the previously discussed clock inputs. That is, a Gate transition is recognized by the PTM on the fourth Enable pulse (provided setup and hold time requirements are met), and the high or low levels of the Gate input must be stable for at least one system clock period plus the sum of setup and hold times. All references to G transition in this document relate to internal recognition of the input transition.

The Gate inputs of all timers directly affect the internal 16 bit counter. The operation of $\overline{G3}$ is therefore independent of the ÷8 prescaler selection.

c) Timer Outputs (01,02,C3) - Timer outputs 01,02 and 03 are capable of driving up to two TTL loads and produce a defined output waveform for either Continuous or Single - Shot Timer modes. Output waveform definition is accomplished by selecting either Single 16 - bit or Dual 8 - bit operating modes. The single 16 bit mode will produce a square wave output in the continous timer mode and will produce a single pulse in the Single-Shot Timer mode. The Dual 8-bit mode will produce a variable duty cycle pulse in both the continuous and single shot Timer modes. Onebit of each Control Register (CRX7) is used to enable the corresponding output. If this bit is cleared, the output will remain low (V_{OL}) regardless of the operating mode.

(vii) TIMER OPERATING MODES

The MC6840 has been designed to operate effectively in a wide variety of applicatinos. This is accomplished by using three bits of each control register (CRX3, CRX4 and CRX5) to defined different operating modes of the Timers. These modes are outlined in Table 7.

TABLE 7 : OPERATING MODES

CONTROL REGISTER			Timer Operating Mode
CRX3	CRX4	CRX5	
0	—	0	Continous
0	—	1	Single shot
1	0	—	Frequency comparison
1	1	—	Pulse width comparison

In addition to the four timer modes in Table 7 the remaining control register bit is used to modify counter initialization and enabling or interrupt conditions.

a) Continuous Operating Mode (Table 8)— Any of the timers in the PTM may be programmed to operate in a continuous mode by writing zeroes into bits 3 and 5 of the corresponding control register. Assuming that the timer output is enabled (CRX7 : 1), either a square wave or a variable duty cycle waveform will be generated at the Timer Output, OX. The type of output is selected via Control Register Bit 2.

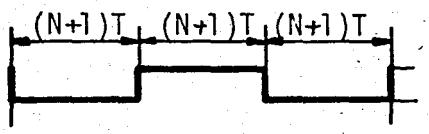
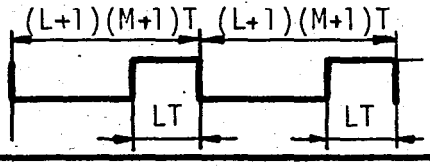
Either a Timer Reset (CR10 : 1 or EXternal Reset : 0) condition or internal recognition of a negative transition of the Gate input results in Counter Initialization. A Write Timer Latches command can be selected as a Counter Initialization signal by clearing CRX4.

The counter is enabled by an absence of a Timer Reset condition and a logic zero at the Gate input. The counter will then decrement on the first clock signal recognized during or after the counter initialization cycle. It continues to decrement on each clock signal so long as G remains low and no reset condition exists. A Counter Time Out (the first clock after all counter bits : 0) results in the Individual Interrupt Flag being set and re-initialization of the counter.

A special condition exists for the dual 8-bit mode (CRX2 : 1) if L : 0. In this case, the counter will revert to a mode similar to the single 16 -bit mode, except Time Out occurs after M+1 clock pulses. The output, if enabled, goes low during the Counter Initialization cycle and reverses state at each Time Out. The counter remains cyclical (is re-initialized at each Time Out) and the Individual Interrupt Flag is set when Time Out occurs. If M : L : 0, the internal counters do not change, but the output toggles at a rate of 1/2 the clock frequency.

In the dual 8 - bit mode (CRX2 : 1) the MSB decrements once for every full countdown of the LSB+1. When the LSB : 0 the MSB is unchanged: on the next clock pulse the LSB is reset to the count in the LSB Latches and the

TABLE 8 CONTINUOUS OPERATING MODES

CONTINUOUS MODE (CRX3 : 0 CRX5 : 0)			
Control Reg.		Initialization / Output	
CRX2	CRX4	Initialization	Timer outputs OX (CRX7 : 1)
0	0	$\overline{G}\downarrow$+W+R	
0	1	$\overline{G}\downarrow$+R	
1	0	$\overline{G}\downarrow$ + W + R	
1	1	$\overline{G}\downarrow$ + R	

$\overline{G}\downarrow$: Negative transition of GATE input
W : Write timer latches command
R : Timer reset (CR10 : 1 or Ext. $\overline{RESET}$: 0)
N : 16 bit number in counter latch
L : 8 bit number in LSB counter latch
M : 8 bit number in MSB counter latch
T : Clock input negative transitions to counter

MSB is decremented by 1 (one). The output, if enabled, remains low during and after initialization and will remain low until the counter MSB is all zeroes. The output will go high at the beginning of the next clock pulse. The output remains high until both the LSB and MSB of the counter are all zeroes. At the beginning of the next clock pulse the defined Time Out (TO) will occur and the output will go low.

The discussion of the Continuous Mode has assumed that the application requires an output signal. It should be noted that the Timer operates in the same manner with the output disabled (CRX7 : 0). A Read Timer Counter command is valid regardless of the state of CRX7.

b) Single-Shot Timer Mode - This mode is identical to the Continuous Mode with three exceptions. The first of these is obvious from the name - the output returns to a low level after the initial Time Out and remains low until another Counter Initialization cycle occurs. The wave-forms available are shown in Table 9.

As indicated in Table 9 the internal counting mechanism remains cyclical in the Single-Shot Mode. Each Time Out of the counter results in the setting of an Individual Interrupt Flag and re-initialization of the counter.

The second major difference between the Single-Shot and Continuous modes is that the internal counter enable is not dependent on the Gate input level remaining in the low state for the Single-Shot mode.

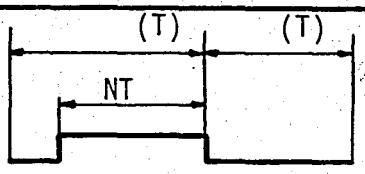
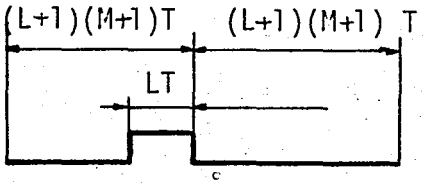
Another special condition is introduced in the Single-Shot mode. If L:M:0 (Dual 8-bit) or N : 0 (Single 16-bit), the output goes low on the first clock received during or after Counter Initialization. The output remains low until the Operating Mode is changed or nonzero data is written into the Counter Latches. Time Outs continue to occur at the end of each clock period.

The three differences between Single-Shot and Continuous Timer Modes can be summarized as attributes of the Single-Shot mode:

1. *Output is enabled for only one pulse until it is reinitialized.*
2. *Counter Enable is independent of Gate.*
3. *L : M : 0 or N : 0 disables output.*

Aside from these differences, the two modes are identical.

TABLE 9 SINGLE SHOT OPERATING MODES

SINGLE SHOT MODE CRX3 : 0, CRX7:1,CRX5: 1			
Control Reg.		Initialization/Output waveforms	
CRX2	CRX4	Conter Init.	Timer output (OX)
0	0	$\overline{G}\downarrow$ + W + R	
0	1	$\overline{G}\downarrow$ + R	
1	0	$\overline{G}\downarrow$ + W + R	
1	1	$\overline{G}\downarrow$ + R	

c) Time Interval Modes - The Time Interval Modes are provided for those applications which require more flexibility of interrupt generation and Counter Initialization. Individual Interrupt Flags are set in these modes as a function of both Counter Time Out and transitions of the Gate input. Counter Initialization is also affected by Interrupt Flag status.

Refer to the Programmable Timer Fundamentals and Applications manual for a discussion of the output signals in these modes. The counter does operate in either Single 16-bit or Dual 8-bit modes as programmed by CRX2. Other features of the Time Interval Modes are outlined in Table 10.

d) Frequency Comparison or Period Measurement Mode (CRX3 : 1, CRX4 : 0) The Frequency Comparison Mode with CRX5 : 1 is straightfordward. If Time Out occurs prior to the first negative transition of the Gate input after a Counter Initialization cycle an Individual Interrupt Flag is set. The counter is disabled and a Counter initialization cycle cannot begin until the interrupt flag is cleared and a negative transition on G is detected.

If CRX5 : 0, as shown in Table 10 and Table 11 an interrupt is generated if Gate input returns low prior to a Time Out. If Counter Time-Out occurs first, the counter is recycled and continues to decrement. A bit is set within the timer on the initial Time Out which precludes further individual interrupt generation until a new Counter Initialization cycle has been completed

When this internal bit is set a negative transition of the Gate input starts a new Counter Initialization cycle. (The condition of $\overline{G}\downarrow \cdot \overline{1} \cdot TO$ *is satisfied, since a Time Out has occurred and no individual Interrupt has been generated)*

TABLE 10 – TIME INTERVAL MODES

CRX3 = 1			
CRX4	CRX5	Application	Condition for Setting Individual Interrupt Flag
0	0	Frequency Comparison	Interrupt Generated if $\overline{\text{Gate}}$ Input Period (1/F) is less than Counter Time Out (TO)
0	1	Frequency Comparison	Interrupt Generated if $\overline{\text{Gate}}$ Input Period (1/F) is greater than Counter Time Out (TO)
1	0	Pulse Width Comparison	Interrupt Generated if $\overline{\text{Gate}}$ Input "Down Time" is less than Counter Time Out (TO)
1	1	Pulse Width Comparison	Interrupt Generated if $\overline{\text{Gate}}$ Input "Down Time" is greater than Counter Time Out (TO)

Any of the timers within the PTM may be programmed to compare the period of a pulse (giving the frequency after calculations) at the Gate input with the time period required for Counter Time-Out. A negative transition of the Gate input enables the counter and starts a Counter Initialization cycle-provided that other conditions as noted in Table 11 are satisfied. The counter decrements on each clock signal recognized during or after Counter Initialization until an Interrupt is generated, a Write Timer Latches command is issued, or a Timer Reset condition occurs. It can be seen from Table 11 that an interrupt condition will be generated if CRX5 : 0 and the period of the pulse (single pulse or measured separetely repetitive pulses) at the Gate input is less than the Counter Time Out period. If CRX5 : 1, an interrupt is generated if the reverse is true.

Assume now with CRX5 : 1 that a Counter Initialization has occured and that the Gate input has returned low prior to Counter Time Out. Since there is now Individual Interrupt Flag generated, this automatically starts a new Counter Initialization Cycle. The process will continue with frequency comparison being performed on each Gate input cycle until the mode is changed, or a cycle is determined to be above the predetermined limit.

e. Pulse Width Comparison Mode (CRX3 : 1, CRX4 : 1) This mode is similar to the Frequency Comparison Mode except for a positive, rather than negative, transition of the Gate input terminates the count. With CRX5 : 0 an Individual Interrupt Flag will be generated if the zero level pulse applied to the Gate input is less than the time period required for Counter Time Out. With CRX5 : 1, the interrupt is generated when the reverse condition is true.

As can be seen in Table 12 a positive transition of the Gate input disables the counter. With CRX5 : 0, it is therefore possible to directly obtain the width of any pulse causing an interrupt. Similar data for other Time Interval Modes and conditions can be obtained, if two sections of the PTM are dedicated to the purpose.

TABLE I1 – FREQUENCY COMPARISON MODE

CRX3 = 1, CRX4 = 0				
Control Reg Bit 5 (CRX5)	Counter Initialization	Counter Enable Flip-Flop Set (CE)	Counter Enable Flip-Flop Reset (CE)	Interrupt Flag Set (I)
0	$\overline{G}\downarrow\cdot\overline{T}\cdot(\overline{CE}+TO)+R$	$\overline{G}\downarrow\cdot\overline{W}\cdot\overline{R}\cdot\overline{T}$	$W+R+I$	$\overline{G}\downarrow$ Before TO
1	$\overline{G}\downarrow\cdot\overline{T}+R$	$\overline{G}\downarrow\cdot\overline{W}\cdot\overline{R}\cdot\overline{T}$	$W+R+I$	TO Before $\overline{G}\downarrow$

I represents the interrupt for a given timer.

TABLE I2 – PULSE WIDTH COMPARISON MODE

CRX3 = 1, CRX4 = 1				
Control Reg Bit 5 (CRX5)	Counter Initialization	Counter Enable Flip-Flop Set (CE)	Counter Enable Flip-Flop Reset (CE)	Interrupt Flag Set (I)
0	$\overline{G}\downarrow\cdot\overline{T}+R$	$\overline{G}\downarrow\cdot\overline{W}\cdot\overline{R}\cdot\overline{T}$	$W+R+I+\overline{G}$	$\overline{G}\uparrow$ Before TO
1	$\overline{G}\downarrow\cdot\overline{T}+R$	$\overline{G}\downarrow\cdot\overline{W}\cdot\overline{R}\cdot\overline{T}$	$W+R+I+\overline{G}$	TO Before $\overline{G}\uparrow$

2.5. PERIPHERAL INTERFACE ADAPTER (PIA)

The MC6821 Peripheral Interface Adapter provides the universal means of interfacing peripheral equipment to the M6800 family of microprocessors. This device is capable of interfacing the MPU to peripherals through two 8-bit bidirectional peripheral data buses and four control lines. No external logic is required for interfacing to mostperipheral devices.

The functional configuration of the PIA is programmed by the MPU during system initialization. Each of the peripheral data lines can be programmed to act as an input or output and each of the four control/interrupt lines may be programmed for one of several control modes. This allows a high degree of flexibility in the overall operation of the interface.

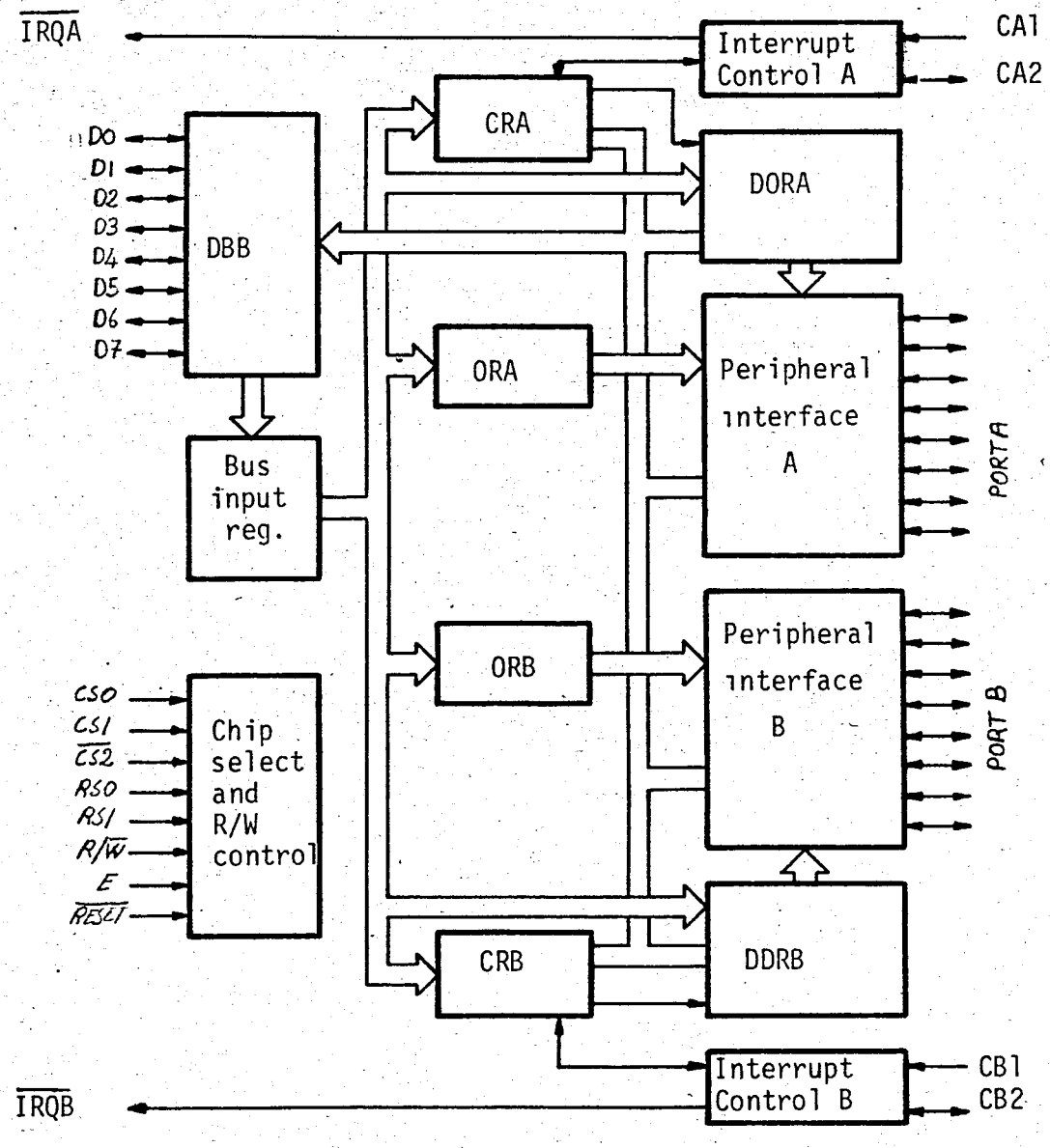


FIGURE 21 : PIA BLOCK DIAGRAM

2.5.1. PIA INTERFACE SIGNALS FOR MPU

The PIA interfaces to the M6800 bus with an 8-bit bidirectional data bus, three chip select lines, two register select lines, two interrupt request lines, a read/write line, an enable line and a reset line. To ensure proper operation with the MC6800, MC6802 or MC6808 microprocessors, VMA should be used as an active part of the address decoding.

(i) Bidirectional Data (DO-D7) The bidirectional data lines (D0-D7) allow the transfer of data between the MPU and the PIA. The data bus output drivers are three-state devices that remain in the high-impedance (off) state except when the MPU performs a PIA read operation. The read/write line is in the read (high) state when the PIA is selected for a read operation.

(ii) Enable (E) - The enable pulse, E is the only timing signal that is supplied to the PIA. Timing of all other signals is referenced to the leading and trailing edges of the E pulse.

(iii) Read/Write ($R/\overline{W}$) This signal is generated by the MPU to control the direction of data transfers on the data bus. A low state on the PIA read/write line enables the input buffers and data is transferred from the MPU to the PIA on the E signal if the device has been selected. A high on the read/write line sets up the PIA for a transfer of data to the bus. The PIA output buffers are enabled when the proper address and the enable E are present.

(iv) $\overline{RESET}$ - The active low $\overline{RESET}$ line is used to reset all register bits in the PIA to a logical zero (low). This line can be used as a power-on reset and as a master reset during system operation.

(v) Chip Selects (CS0, CS1 and $\overline{CS2}$) These three input signals are used to select the PIA, CS0 and CS1 must be high and $\overline{CS2}$ must be low for selection of the device. Data transfers are then performed under the control of the enable and read/write signals. The chip select lines must be stable for the duration of the E pulse. The device is deselected when any of the chip selects are in the inactive state.

(vi) Register Selects (RSO and RS1) The two register select lines are used to select the various registers inside the PIA. These two lines are used in conjunction with internal Control Registers to select a particular register that is to be written or read.

The register and chip select lines should be stable for the duration of the E pulse while in the read or write cycle.

(vii) Interrupt Request ($\overline{IRQA}$ and $\overline{IRQB}$) The active low interrupt Request lines ($\overline{IRQA}$ and $\overline{IRQB}$) act to interrupt the MPU either directly or through interrupt priority circuitry. These lines are "open drain" (no load device on the chip). This permits all interrupt request lines to be tied together in a wire - OR configuration.

Each Interrupt Request line has two internal interrupt flag bits that can cause the Interrupt Request line to go low. Each flag bit is associated with a particular peripheral interrupt line. Also four interrupt enable bits are provided in the PIA which may be used to inhibit a particular interrupt from a peripheral device.

Servicing an interrupt by the MPU may be accomplished by a softwareroutine that, on a prioritized basis, sequentially reads and tests the two control registers in each PIA for interrupt flag bits that are set.

The interrupt flags are cleared (zeroed) as a result of an MPU Read Peripheral Data Operation of the corresponding data register. After being cleared, the interrupt flag bit cannot be enabled to be set until the PIA is deselected during an E pulse. The E pulse is used to condition the interrupt control lines (CA1, CA2, CB1, CB2). When these lines are used as interrupt inputs at least one E pulse must occur from the inactive edge to the active edge of the interrupt input signal to condition the edge sense network. If the interrupt flag has been enabled and the edge sense circuit has been properly conditioned, the interrupt flag will be set on the next active transition of the interrupt input pin.

2.5.2. PIA PERIPHERAL INTERFACE LINES

The PIA provides two 8-bit bidirectional data buses and four interrupt/ control lines for interfacing to peripheral devices.

(i) Section A Peripheral Data (PAO-PA7) - Each of the peripheral data lines can be programmed to act as an input or output. This is accomplished by setting a "1" in the corresponding Data Direction Register bit for those lines which are to be outputs. A "0" in a bit of the Data Direction Register causes the corresponding peripheral data line to act as an input. During an MPU Read Peripheral Data Operation, the data on peripheral lines programmed to act as input appears directly on the corresponding MPU Data Bus lines. In the input mode, the internal pullup resistor on these lines represents a maximum of 1.5 standard TTL loads.

The data in Output Register A will appear on the data lines that are programmed to be outputs. A logical "1" written into the register will cause a "high" on the corresponding data line while a "0" result in a "low". Data in Output Register A may be read by an MPU "REad Peripheral Data A" operation when the corresponding lines are programmed as outputs. This data will be read properly if the voltage on the peripheral data lines is greater than 2.0 volts for a logic "1" output and less than 0.8 volt for a logic "0" output. Loading the output lines such that the voltage on these lines does not reach full voltage causes the data transferred into the MPU on a Read operation to differ from that contained in the respective bit of Output Register A.

(ii) Section B Peripheral Data (PBO-PB7) The peripheral data lines in the B Section of the PIA can be programmed to act as either inputs or outputs in a similar manner to PAO-PA7. They have three-state capabilty, allowing them to enter a high-impedance state when the peripheral data line is used as an input. In addition, data onthe peripheral data lines PB0-PB7 will be read properly from those lines programmed as outputs even if the voltages are below 2.0 volts for a "high" or above 0.8 V for a "low". As outputs, these lines are compatible with standard TTL and may also be used as a source of up to 1 milliampere at 1.5 volts to directly drive the base of a transistor

switch.

(iii) Interrupt Input (CA1 and CB1) Peripheral input lines CA1 and CB1 are input only lines that set the interrupt flags of the control registers. The active transition for these signals is also programmed by the two control registers.

(iv) Peripheral Control (CA2) The peripheral control line CA2 can be programmed to act as an interrupt input or as a peripheral control output. As an output, this line is compatible with standard TTL: as an input the internal pullup resistor on this line represents 1.5 standard TTL loads. The function of this signal line is programmed with Control Register A.

(v) Peripheral Control (CB2) - Peripheral Control line CB2 may also be programmed to act as an interrupt input or peripheral control output. As an input,this line has high input impedance and is compatible with standard TTL. As an output it is compatible with standard TTL and may also be used as a source of up to 1 milliampere at 1.5 volts to directly drive the base of a transistor switch. This line is programmed by Control Register B.

2.5.3. INTERNAL CONTROLS

INITIALIZATION

A $\overline{\text{RESET}}$ has the effect of zeroing all PIA registers. This will set PA0-PA7, PB0-PB7, CA2 and CB2 as inputs, and all interrupts disabled. The PIA must be configured during the restart program which follows the reset.

There are six locations within the PIA accessible to the MPU data bus two Peripheral Registers, two Data Direction Registers and two Control Registers. Selection of these locations is controlled by the RS0 and RS1 inputs together with bit 2 in the Control Register, as shown in Table 13.

Details of possible configurations of the Data Direction and Control Register are as follows:

TABLE 13 INTERNAL ADDRESSING

		Control Register Bit		
RS1	*RS0*	*CRA-2*	*CRB-2*	*Location Selected*
0	*0*	*1*	*X*	*Peripheral Register A*
0	*0*	*0*	*X*	*Data Direction Register A*
0	*1*	*X*	*X*	*Control Register A*
1	*0*	*X*	*1*	*Peripheral Register B*
1	*0*	*X*	*0*	*Data Direction Register B*
1	*1*	*X*	*X*	*Control Register B*

X : Don't Care

CONTROL REGISTERS (CRA and CRB)

The two Control Registers (CRA and CRB) allow the MPU to control the operation of the four peripheral control lines CA1, CA2, CB1 and CB2. In addition they allow the MPU to enable the interrupt lines and monitor the status of the interrupt flags. Bits 0 through 5 of the two registers may be written or read by the MPU when the proper chip select and register select signals are applied. Bits 6 and 7 of the two registers are read only and are modified by external interrupts occuring on control lines CA1, CA2, CB1 ar CB2.

DATA DIRECTION ACCESS CONTROL BIT (CRA - 2 and CRB - 2)

Bit 2, in each Control Register (CRA and CRB), determines selection of either a Peripheral Output Register or the corresponding Data Direction E Register when the proper register select signals are applied to RSO and RS1. A '1' in bit 2 allows access of the Peripheral Interface Register, while a "0" causes the Data Direction Register to be addressed.

(i) Interrupt Flags (CRA-6, CRA-7, CRB-6 and CRB-7)
The four interrupt flag bits are set by active transitions of signals on the four Interrupt and Peripheral Control lines when those lines are programmed to be inputs . These bits cannot be set directly from the MPU Data Bus and are reset indirectly by a Read Peripheral Data Operation on the appropriate section.

(ii) Control of CA2 and CB2 Peripheral Control Lines (CRA-3, CRA-4, CRA-5, CRB-3, CRB-4 and CRB-5) Bits 3, 4 and 5 of the two control registers are used to control the CA2 and CB2 Peripheral Control lines. These bits determine if the control lines will be an interrupt input or an output control signal. If bit CRA-5 (CRB-5) is low, CA2 (CB2) is an interrupt input line similar to CA1 (CB1). When CRA-5 (CRB-5) is high, CA2 (CB2) becomes an output signal that may be used to control peripheral data transfers. When in the output mode, CA2 and CB2 have slightly different loading characteristics.

(iii) Control of CA1 and CB1 Interrupt Input Lines (CRA-0, CRB-1,CRA-1 and CRB-1) The two lowest-order bits of the control registers are used to control the interrupt input lines CA1 and CB1. Bits CRA-0 and CRB-0 are used to enable the MPU interrupt signals IRQA and IRQB, respectively. Bits CRA-1 and CRB-1 determine the active transition of the interrupt input signals CA1 and CB1.

2.6. ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER - MC 6850

The MC6850 Asynchronous Communications Interface Adapter provides the data formatting and control to interface serial asynchronous data communications information to bus organized systems such as the MC6800 Microprocessing Unit.

The bus interface of the MC6850 includes select, enable, read/write,interrupt and bus interface logic to allow data transfer over an 8 bit bi-directional data bus. The parallel data of the bus system is serially transmitted and received by the asynchronous data interface, with proper formatting and error checking. The functional congiguration of the ACIA is programmed via

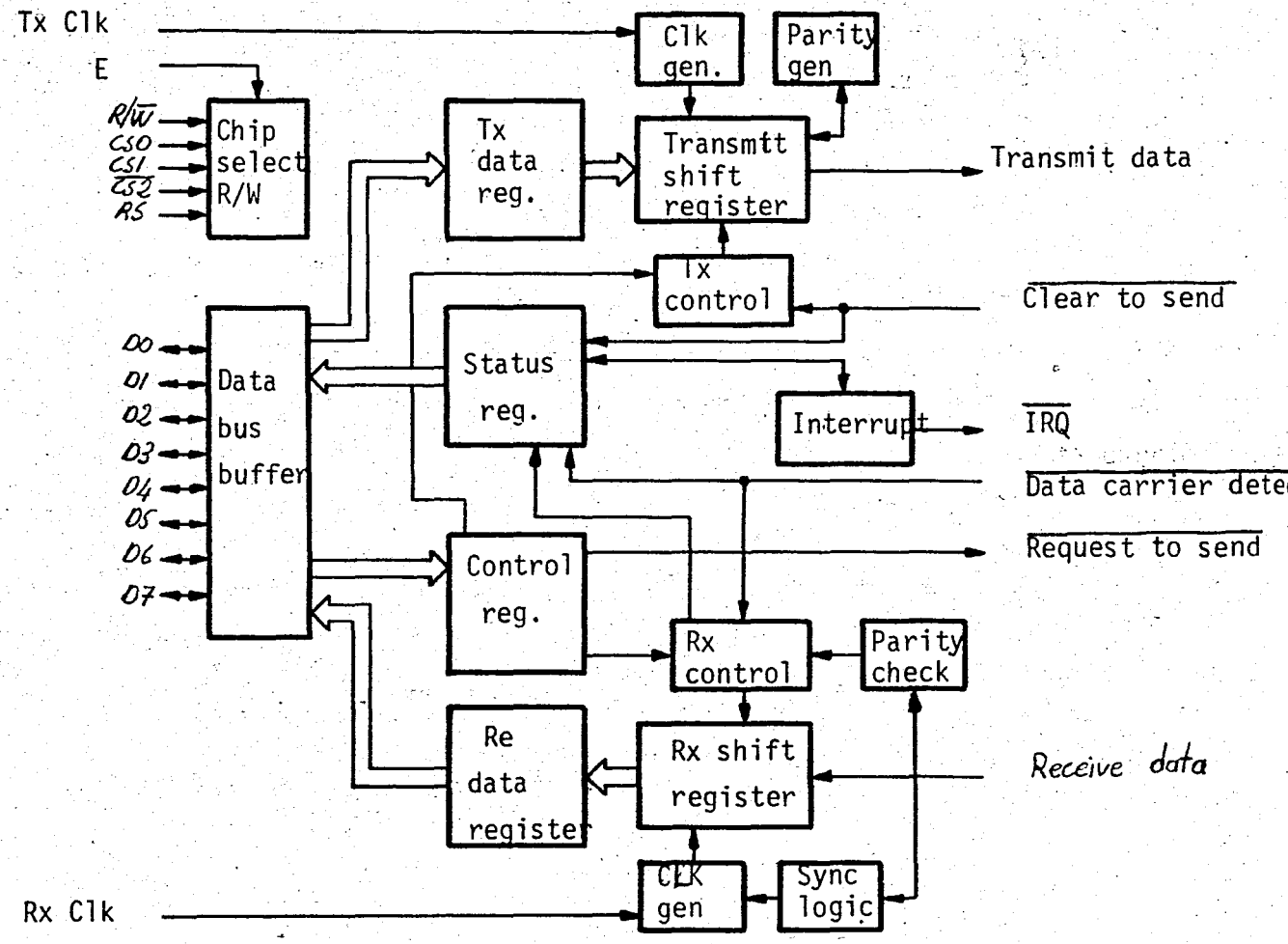


FIGURE 22 ACIA BLOCK DIAGRAM

the data bus during system initialization. A programmable Control Register provides variable word lengths, clock division ratios, transmit control, receive control, and interrupt control. For peripheral or modem operation three control lines are provided. These lines allow the ACIA to interface directly with the MC6860L 0-600 bps digital modem.

2.6.1. ACIA OPERATION

At the bus interface, the ACIA appears as two addressable memory locations. INternally, there are four registers: two read only and two write-only registers. The read-only registers are Status and Receive Data: the write only registers are Control and Transmit Data. The serial interface consists of serial input and output lines with independent clocks, and three peripheral modem control lines.

POWER ON/MASTER $\overline{RESET}$

The master reset (CR0,CR1) should be set during system initialization to insure the reset condition, and prepare for programming the ACIA functional configuration when the communications channel is required. Control bits CR5 and CR6 should also be programmed to define the state of RTS whenever master reset is utilized. The ACIA also contains internal power-on reset logic to detect the power line turn-on transition and hold the chip in a reset state to prevent erroneous output transitions prior to initialization. This circuitry depends on clean power turn-on transitions. The power-on reset is released by means of the bus-programmed master reset which must be applied prior to operating the ACIA. After master resetting the ACIA, the programmable Control Register can be set for a number of options such as variable clock divider ratios, variable word length, one or two stop bits,parity (even, odd, or none) etc.

TRANSMIT

A typical transmitting sequence consists of reading the ACIA Status Register either as a result of an interrupt or in the ACIA's turn in a polling sequence. A character may be written into the Transmit Data Register if the status read operation has indicated that the Transmit Data Register is empty. This character is transferred to a Shift Register where it is serialized and transmitted from the Transmit Data output preceded by a start bit and followed by one or two stop bits. Internal parity (odd or even) can be optionally added to the character and will occur between the last data bit and the first stop bit. After the first character is written in the Data Register, the Status Register can be read again to check for a Transmit Data Register Empty condition and current peripheral status. If the register is empty, another character can be loaded for transmission even though the first character is in the process of being transmitted (because of double buffering). The second character will be automatically transferred into the Shift Register when the first character transmission is completed. This sequence continues until all the characters have been transmitted.

RECEIVE

Data is received from a peripheral by means of the Receive Data input. A divide-by-one clock ratio is provided for an externally synchronized clock (to its data) while the divide-by-16 and 64 ratios are provided for internal synchronization. Bit synchronization in the divide - by - 16 and 64 modeis is initiated by the detection of the leading mark-to-space transition of the start bit. False start bit deletion capability insures that a full half bit of a start bit has been received before the internal clock is synchronized to the bit time. As a character is being received, parity (odd or even) will be checked and the error indication will be available in the Status Register along with framing error overrun error and Receive Data Register full. In a typical receiving sequence, the Status Register is read to determine if a character has been received from a peripheral. If the Receiver Data Register is full the character is placed on the 8-bit ACIA bus when a Read Data command is received from the MPU. When parity has been selected for an 8-bit word (7 bits plus parity), the receiver strips the parity bit (D7: 0) so that data alone is transferred to the MPU. This feature reduces MPU programming. The Status Register can continue to be read again to determine when another character is available in the Receive Data Register. The

receiver is also double buffered so that a character can be read from the data register as another character is being received in the shift register. The above sequence continues until all characters have been received.

2.6.2. ACIA INTERFACE SIGNALS FOR MPU

The ACIA interfaces to the MPU with an 8-bit bi-directional data bus, three chip select lines, a register select line, an interrupt request line, read/write line, and enable line.

(i) ACIA BI-Directional Data (DO-D7) The bi-directional data lines (DO-D7) allow for data transfer between the ACIA and the MPU. The data bus output drivers are three-state devices that remain in the high-impedance (off) state except when the MPU performs an ACIA read operation.

(ii) ACIA Enable (E) - The Enable signal E, is a high impedance TTL compatible input that enables the bus input/output data buffers and clocks data to and from the ACIA. This signal will normally be a derivative of the Ø2 Clock.

(iii) Read/Write (R/$\overline{W}$) The Read/Write line is a high impedance input that is TTL compatible and is used to control the direction of data flow through the ACIA's input/output data bus interface. When Read/Write is high (MPU Read cycle), ACIA output drivers are turned on and a selected register is read. When it is low, the ACIA output drivers are turned off and the MPU writes into a selected register. Therefore, the Read/Write signal is used to select read-only or write-only registers within the ACIA.

(iv) Chip Select (CSO,CS1,$\overline{CS2}$) These three high impedance TTL compatible input lines are used to address the ACIA. The ACIA is selected when CS0 and CS1 are high and $\overline{CS2}$ is low. Transfers of data to and from the ACIA are then performed under the control of the Enable signal, Read/Write and Register Select.

(v) Register Select (RS) - The Register Select line is a high impedance input that is TTL compatible. A high level is used to select the Transmit/Receive Data Registers and a low level the Control/Status Registers. The Read/Write signal line is used in conjunction with Register Select to select the read-only or write-only register in each register pair.

(vi) Interrupt Request ($\overline{IRQ}$) Interrupt - Request is a TTL compatible, open-drain (no internal pullup), active low output that is used to interrupt the MPU. The $\overline{IRQ}$ output remains low as long as the cause of the interrupt is present and the appropriate interrupt enable within the ACIA is set. The $\overline{IRQ}$ status bit, when high, indicates the $\overline{IRQ}$ output is in the active state.

Interrupts result from conditions in both the transmitter and receiver sections of the ACIA. The transmitter section causes an interrupt when the Transmitter Interrupt Enabled condition is selected (CR5. CR6) and the Transmit Data Register Empty (TDRE) status bit is high. The TDRE status bit indicates the current status of the Transmitter Data Register except when inhibited by $\overline{Clear\text{-}to\text{-}Send}$ ($\overline{CTS}$) being high or the ACIA being maintained in the Reset condition. The interrupt is cleared by writting data into the Transmit Data Register. The inturrupt is masked by disabling the Transmitter Interrupt via CR5 or CR6 or by the loss of CTS which inhibits the TDRE status bit. The Receiver section causes an interrupt when the Receiver Interrupt Enable is set and the Receive Data Register Full (RDRF) status bit is high, an Overrun has occured, or Data Carrier Detect (DCD) has gone high An interrupt resulting from the RDRF status bit can be cleared by reading data or resetting the ACIA. Interrupts caused by Overrun or loss of DCD are cleared by reading the status register after the error condition has occurred and then reading the Receive Data Register or resetting the ACIA. The receiver interrupt is masked by resetting the Receiver Interrupt Enable.

2.6.3. CLOCK INPUTS

Separate high impedance TTL compatible inputs are provided for clocking of transmitted and received data. Clock frequencies of 1, 16 or 64 times the data rate may be selected.

2.6.4. SERIAL INPUT/OUTPUT LINES

(i) Receive Data (Rx Data) - The Receive Data line is a high impedance TTL compatible input through which data is received in a serial format. Synchronization with a clock for detection of data is accomplished internally when clock rate of 16 or 64 times the bit rate are used. Data rates are in the range of 0 to 500 kbps when external synchronization is utilized.

(ii) Transit Data (Tx Data) - The Transit Data output line transfers serial data to a modem or other peripheral. Data rates are in the range of 0 to 500 kbps when external synchronization is utilized.

2.6.5. PERIPHERAL/MODEM CONTROL

The ACIA includes several functions that permit limited control of a peripheral or modem. The functions included are Clear-to-Send, Request-to Send and Data Carrier Detect.

(i) $\overline{\text{Clear-to-Send}}$ ($\overline{\text{CTS}}$) - This high impedance TTL compatible input provides automatic control of the transmitting end of a communications link via the modem $\overline{\text{Clear-to-Send}}$ active low output by inhibiting the Transmit Data Register Empty (TDRE) status bit.

(ii) $\overline{\text{Request-to-Send}}$ ($\overline{\text{RTS}}$) - The Request - to - Send output enables the MPU to control a peripheral or modem via the data bus. The $\overline{\text{RTS}}$ output corresponds to the state of the Control Register bits CR5 and CR6. When

CR6 : 0 or both CR5 and CR6 : 1, the RTS output is low (the active state) This output can also be used for Data Terminal Ready (DRT)

(iii) $\overline{\text{Data Carrier Detect}}$ ($\overline{\text{DCD}}$) - This high impedance TTL compatible input provides automatic control, such as in the receiving end of a communications link by means of a modem Data Carrier Detect output. The $\overline{\text{DCD}}$ input inhibits and initializes the receiver section of the ACIA when high. A low to high transition of the Data Carrier Detect initiates an interrupt to the MPU to indicate the occurrence of a loss of carrier when the Receive Interrupt Enable bit is set.

2.6.6. ACIA REGISTERS

The expanded block diagram for the ACIA indicates the internal registers on the chip that are used for the status, control, receiving, and transmitting of data.

TRANSMIT DATA REGISTER (TDR)

Data is written in the Transmit Data Register during the negative transition of the enable (E) when the ACIA has been addressed and RS. R/W is selected. Writing data into the register causes the Transmit Data Register Empty bit in the Status Register to go low. Data can then be transmitted. If the transmitter is idling and no character is being transmitted, then the transfer will take place within one bit time of the trailing edge of the Write command. If a character is being transmitted, the new data character will commence as soon as the previous character is complete. The transfer of data causes the Transmit Data Register Empty (TDRE) bit to indicate empty.

RECEIVE DATA REGISTER (RDR)

Data is automatically transferred to the empty Receive Data Register (RDR) from the receiver deserializer (a shift register) upon receiving a complete character. This event causes the Receive Data Register Full bit (RDRF) in the status buffer to go high (full). Data may then be read through the bus by addressing the ACIA and selecting the Receive Data Register with RS and R/W high when the ACIA is enabled. The non-destructive read cycle causes the RDRF bit to be cleared to empty although the data is retained in the RDR. The status is maintained by RDRF as to whether or not the data

is current. When the Receive Data Register is full, the automatic transfer of data from the Receiver Shift Register to the Data Register is inhibited and the RDR contents remain valid with its current status stored in the Status Register .

CONTROL REGISTER

The ACIA Control Register consists of eight bits of write-only buffer that are selected when RS and R/W are low. This register controls the function of the receiver, transmitter, interrupt enables, and the Request-to-Send peripheral/modem control output.

(i) Counter Divide Select Bits (CR0 and CR1) The Counter Divide Select Bits (CR0 and CR1) determine the divide ratios utilized in both the transmitter and receivre sections of the ACIA. Additionally, these bits are used to provide a master reset for the ACIA which clears the Status Register (except for external conditions on $\overline{CTS}$ and $\overline{DCD}$) and initializes both the receiver and transmitter. Master reset does not affect other Control Register bits. Note that after power-on or a power fail/restart, these bits must be set high to reset the ACIA. After resetting, the clock divide ratio may be selected. These counter select bits provide for the following clock divide ratios:

CR1	CR0	Function
0	0	1
0	1	16
1	0	64
1	1	Master Reset

Word Select Bits (CR2, and CR4) The Word Select bits are used to select word length, parity and the number of stop bits. The encoding format is as follows:

CR4	CR3	CR2	Function
0	0	0	7 Bits :Even Parity 2 Stop Bits
0	0	1	7 Bits : Odd Parity 2 Stop Bits
0	1	0	7 Bits : Even Parity 1 Stop Bit
0	1	1	7 Bits Odd Parity 1 Stop Bit
1	0	0	8 Bits 2 Stop Bits
1	0	1	8 Bits 1 Stop Bit
1	1	0	8 Bits Even Parity 1 Stop Bit
1	1	1	8 Bits Odd Parity 1 Stop Bit

Word lenght, Parity Select, and Stop Bit changes are not buffered and therefore become effective immediately.

(ii) Transmitter Control Bits (CR5 and CR6) – Two Transmitter Control bits provide for the control of the interrupt from the Transmit Data Register Empty condition the Request-to-Send ($\overline{RTS}$) output, and the transmission of a Break level (space). The following encoding format is used:

CR6	CR5	Function
0	0	$\overline{RTS}$: low,Transmitting Interrupt Disabled
0	1	$\overline{RTS}$: low,Transmitting Interrupt Enabled
1	0	$\overline{RTS}$: high,Transmitting Interrupt Disabled
1	1	$\overline{RTS}$: low,Transmits a Break level on the Transmit Data Output, Transmitting Interrupt Disabled.

(iii) Receive Interrupt Enable Bit (CR7)– The following interrupts will be enabled by a high level in bit position 7 of the Control Register (CR7): Receive Data Register Full, Overrun or a low to high transition on the Data Carrier Detect ($\overline{DCD}$) signal line.

STATUS REGISTER

Information on the status of the ACIA is available to the MPU by reading the ACIA Status Register. This read-only register is selected when RS is low and R/W is high. Information stored in this register indicates the status of the Transmit Data Register, the Receive Data Register and error logic, and the peripherol/modem status inputs of the ACIA.

(i) Receive Data Register Full (RDRF) Bit 0 - Receive Data Register Full indicates that received data has been transferred to the Receive Data Register, RDRF is cleared after an MPU read of the Receive Data Register or by a master reset. The cleared or empty state indicates that the contents of the Receive Data Register are not current. Data Carrier Detect being high also causes RDRF to indicate empty.

(ii) Transmit Data Register Empty (TDRE) Bit 1 - The Transmit Data Register Empty bit being set high indicates that the Transmit Data Register contents have been transferred and that new data may be entered. The low state indicates that the register is full and that transmission of a new character has not begun since the last write data command.

(iii) $\overline{\text{Data Carrier Detect}}$ ($\overline{DCD}$) Bit 2 - The Data Carrier Detect bit will be high when the $\overline{DCD}$ input from a modem has gone high to indicate that a carrier is not present. This bit going high causes an Interrupt Request to be generated when the Receive Interrupt Enable is set. It remains high after the $\overline{DCD}$ input is returned low until cleared by first reading the Status Register and then the Data Register or until a master reset occurs. If the $\overline{DCD}$ input remains high after read status and read data or master reset has occurred, the interrupt is cleared, the $\overline{DCD}$ status bit remains high and will follow the $\overline{DCD}$ input.

(iv) $\overline{\text{Clear-to-Send}}$ ($\overline{CTS}$) Bit 3 - The Clear to send bit indicates the state of the Clear-to-Send input from a modem. A low $\overline{CTS}$ indicates that there is a Clear-to-Send from the modem. In the high state, the Transmit Data Register Empty bit is inhibited and the Clear-to-Send status bit will be

high. Master reset does not affect the Clear-to-Send Status bit.

(v) Framing Error (FE) Bit 4 - Framing error indicates that the received character is improperly framed by a start and a stop bit and is detected by the absence of the lst stop bit. This error indicates a synchronization error, faulty transmission, or a break condition. The framing error flag is set or reset during the receive data transfer time. Therefore, this error indicator is present through-out the time that the associated character is available.

(vi) Receiver Overrun (OVRN) Bit-5 Overrun is an error flag that indicates that one or more characters in the data stream were lost. That is, a character or a number of characters were received but not read from the Receive Data Register (RDR) prior to subsequent characters being received. The overrun condition begins at the midpoint of the last bit of the second character received in succession without, a read of the RDR having occured. The Overrun does not occur in the Status Register until the valid character prior to Overrun has been read. The RDRF bit remains set until the Overrun is reset. Character synchronization is maintained during the Overrun condition. The Overrun indication is reset after the reading of data from the Receive Data Register or by a Master Reset.

(vii) Parity Error (PE) Bit 6 - The parity error flag indicates that the number of highs (ones) in the character does not agree with the preselected odd or even parity. Odd parity is defined to be when the total number of ones is odd. The parity error indication will be present as long as the data character is in the RDR. If no parity is selected, then both the transmitter parity generator output and the receiver parity check results are inhibited.

(viii) $\overline{\text{Interrupt Request}}$ ($\overline{IRQ}$)-Bit 7 The $\overline{IRQ}$ bit indicates the state of the IRQ output. Any interrupt condition with its applicable enable will be indicated in this status bit. Anytime the $\overline{IRQ}$ output is low the $\overline{IRQ}$ bit will be high to indicate the interrupt or service request status. $\overline{IRQ}$ is cleared by a read operation to the Receive Data Register or a write operation to the Transmit Data Register.

CHAPTER 3

SYSTEM DESCRIPTION

In the following paragraphs the descriptions of the main units of the intelligent terminal are given. At the end of the chapter a complete schematic diagram is presented.

3.1. DECODING UNIT

For the simplicity and speed, of the hardware to be used in decoding circuit a two step decoding is realised using the 74S 471 programable read only memory - PROM - and the 74S 138 1- of - 8 decoder.

As will be explained in the memory unit section the intelligeant terminal has three different memory options. One can chose either the 32 K RAM or 48 K RAM or 58K RAM options by making appropriate jumper connections and using the appropriate decoding ROM. Therefore for the decoding there are 3 different PROM contents to be programmed. This is represented in Table 14. When the address ECOD-ECFF is decoded by the programmable ROM the decoded pin enable the 74 S 138 1- of - 8 decoder which selects the input/output units, PTM, CRTC, internal PIA and user outputs, using A7, A6 and A5. (Table 15)

The ACIA - MC 6850 - is decoded using the address pin A4 so that its intern registers are located at the addresses ECDO-ECD1. The second PIA-MC 6821 - is decoded by A3 so that it is located at the addresses between ECC8-ECCB or ECD8-ECDB. The first PIA-MC 6821 - is decoded by A2 and it is located at the addresses ECC4-ECC7 or ECD4-ECD7.

TABLE 14

DECODING TABLE

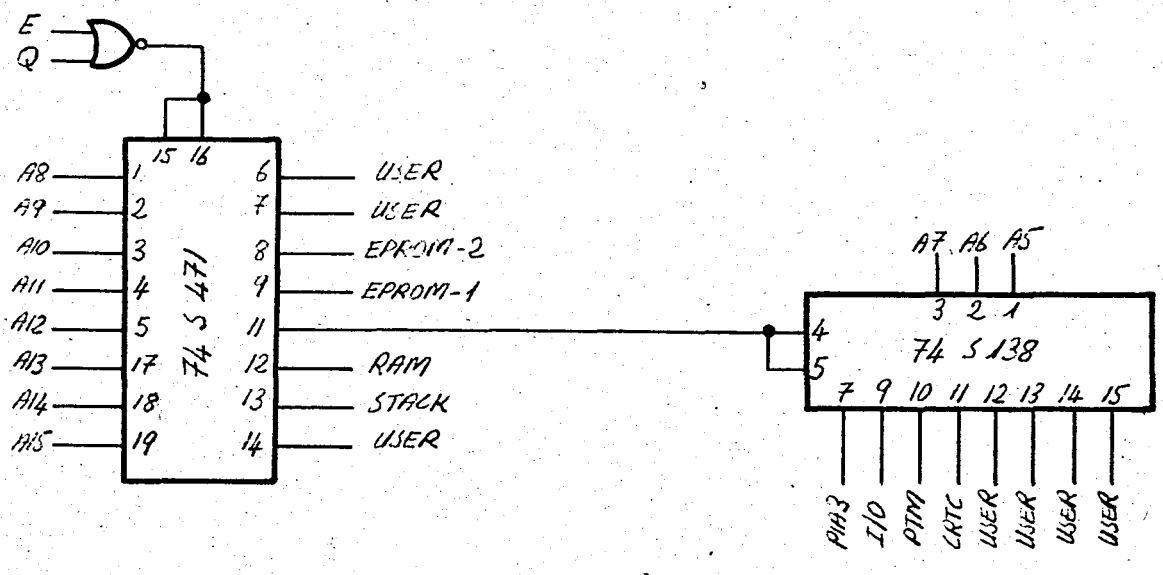
FROM ADDRESS INPUT								DATA OUTPUT								DECODING
A15	A14	A13	A12	A11	A10	A9	A8	00	01	02	03	04	05	06	07	
0	0	0	0	0	0	0	0									
								1	1	1	1	1	0	1	1	0000-7FFF 32 KRAM
0	1	1	1	1	1	1	1			(FB)						
0	0	0	0	0	0	0	0									
								1	1	1	1	1	0	1	1	0000-BFFF 48 KRAM
1	0	1	1	1	1	1	1			(FB)						
0	0	0	0	0	0	0	0									
								1	1	1	1	1	0	1	1	0000-E7FF 58 K RAM
1	1	1	0	0	1	1	1			(FB)						
1	1	1	0	1	0	0	0									
								1	1	1	1	1	1	1	0	E800-EBFF USER
1	1	1	0	1	0	1	1			(FE)						
1	1	1	0	1	1	0	0									
								1	1	1	1	1	1	1	1	ECOD-EOFF I/O USE
1	1	1	0	1	1	0	1			(E7)						
1	1	1	0	1	1	1	0									
								1	1	1	1	1	1	0	1	EEOO-EFFF STACK
1	1	1	0	1	1	1	1			(FD)						
1	1	1	1	0	0	0	0									
								1	1	1	0	1	1	1	1	FOOO-FFFF EPROM-
1	1	1	1	1	1	1	1			(EF)						
1	1	0	0	0	0	0	0									
								1	1	0	1	1	1	1	1	COOO-CFFF EPROM-
1	1	0	0	1	1	1	1			(DF)						(Not decoded in 5 option)
1	1	0	1	0	0	0	0									
								1	0	1	1	1	1	1	1	DOOO-DFFF
1	1	0	1	1	1	1	1			(BF)						(Not decoded in option)

TABLE 15

A7	A6	A5	DECODING	
0	0	0	ECOD-ECIF	USER 1
0	0	1	EC20-EC3F	USER 2
0	1	0	EC40-EC5F	USER 3
0	1	1	EC60-EC7F	USER 4
1	0	0	EC80-EC9F	CRTC
1	0	1	ECAD-ECBF	PTM
1	1	0	ECCO-ECDF	I/O
1	1	1	ECEO-ECFF	INternal/PIA

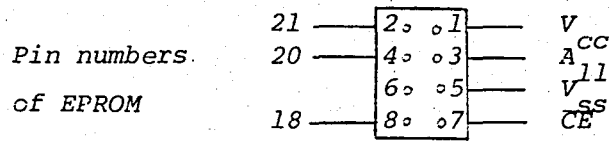
FIGURE 23

DECODING CIRCUITRY



3.2. THE MEMORY UNIT

The RAM and EPROM units handle the resident monitor and terminal software and user memory together with the display memory. There are 2 EPROM sockets available for 2K and 4K EPROM's so that from 2K byte upto 8K byte resident software can be used. Either one of the sockets can be used for 2 K byte (2716) or 4 K byte (2732, 2532) EPROM's by making the appropriate jumper connections.



PIN	2716	2732	2532
21	V_{pp}	A11	V_{pp}
20	$\overline{G}$	$\overline{G}/V_{pp}$	E/P_{rgr}
19	A10	A10	A10
18	E	E	A11

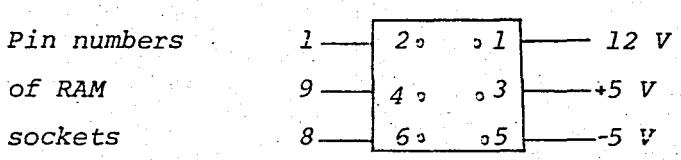
The main EPROM (EPROM-1) is located at the address FOOO-FFFF and the user EPROM is located at COOO-CFFF. For the 58 Kbyte RAM option the user EPROM (EPROM-2) is not decoded.

The RAM section has been choosedto be dynamic because of the large capacity required. Dynamic RAM's occupy less surface on the board compared to static ones and they are more cost effective, although an additional refresh circuit is required.

In this system there are 2 RAM areas available for sixteen 4116's so that 32 K bytes of memory can be obtained. The same RAM area can also be used for eight 4164's for 48 Kybte or 58 K byte options. At the right of each row of eight RAM's there are jumpers available so that one can use 4116's or 4164. The 4116 has three different supplies V_{CC}, V_{BB} and V_{DD}, but 4164 has a single V_{CC} (5 V) supply the difference in the pin description of 4116 and 4164 is gi below.

Pin Numbers	4116 Description	4164 Description
1	V_{BB} -5V	NC
8	V_{DD} 12 V	V_{CC} 5V
9	V_{CC} 5 V	A7
16	V_{SS} GROUND	V_{SS} GROUND

Jumper connections are described below:



REFRESH THEORY

The static MOS memory cell uses about six transistors. It consists of a flip flop plus one input and one output transistors. In the dynamic memory cell, the MOS transistor number is reduced to four, the reason beign the use of stray capacitances to store the state of the cell. But if a write operation has not been performed for an extended time the leakage of the capacitor charge may cause the information in the cell to be lost. It is therefore necessary to perform a refresh operation periodically to not lose the information.

The 4116 is a 16384 x 1 bit dynamic RAM using an advanced N-channel silicon ga technology, organized in form of 128 x 128 memory array so that each address decoder accepts 7 address lines, the total beeing 14. The multiplexed addre permits the 4116 to be packed in a standart 16-pin DIP. Refresh of the dynami cell matrix is accomplished by performing a memory cycle at each of the 128 ro addresses within each 2 msec time interval.

The 4164 is a 65536 x 1 bit dynamic RAM organized in a 256 x 256 matrix memory array, so that each address decoder accepts 8 address lines, the total beeing 16. The multiplexed addresses and single 5V supply permit the 4164 to be packe in a standart 16-pin DIP. Refresh of the dynamic cell matrix is the same as

the 4116, but now 256 row addresses are refreshed.

There are three refreshing techniques: the first one is to stop the system every 2 msec and perform the 128 or 256 refresh cycles or tho distribute them over a 2 msec time, the second method is to steal the $\emptyset_1$ time by stretching the clock. By stretching program execution is delayed. In these first two methods the reduction in program execution time may not be tolerated because of real time applications and software requirements. To avoid the loss of time a hidden or transparent refresh technique is used. The hidden refresh cycle is accomplished in the most easiest way with the technique mentioned as the $\overline{RAS}$ - only refresh. First $\overline{RAS}$ is held high and the row address to be refreshed is prepared. Then $\overline{RAS}$ is lowered to let the address in and it is held low until the refresh occurs, then it is pulled up at the and of the cycle. In a read or write operation the row address is latched in with a $\overline{RAS}$ signal, then thc colum address and $R/\overline{W}$ is prepared and all the information is latched with the $\overline{CAS}$ line.

The refresh and memory control circuitry is shown in figure 24. When the powe ison the Q output assumes either a low or high state. If it assumes a low state the first inverter output is high, the parallel capacitor to the second inverter will be charged from left. If a transition occurs, the capacitor will discharge into the first inverter causing a delay until the second inverter input goes to a low level. When the second inverter goes high it charges the capacitor from right. If a transition occurs it will discharge into the second inverter causing the same delay. Now if the output of the third invertor is low the capacitor at the reset input discharges through R1 and the R input of the flip flop goes low to reset the Q output. When Q becomes low the third invcrter charges C1 to a high state pulling up the reset input. So the flip- flop acts as a monastable. The delay introduced can be arranged experimentall It should be noted that the total pulse length is equal to the delay introduce by the three inverter circuit plus the discharge time of C1. The third inverte output is used to generate the $\overline{CAS}$ signal together with the second flip flop's inverter output, so the $\overline{CAS}$ signal is delayed from $\overline{RAS}$. The second flip flop

logic is the same as the first one only the difference beeing that the D input is the RAM select output of the decoder PROM. The $\overline{RAS}$ signal is generated whe the Q outputs of both flip-flops are high. And the $\overline{CAS}$ signal is gencrated with the delay introduced as explained above with the two flip flops. The $\overline{CAS}$ signals are gated separately by A14 of the processor and MA14 generated by the full adder circuitry, So two $\overline{CAS}$ signals are produced for each bank of RAM.

Since the 4164 requires an A7 address input two banks of multiplexer sockets are used so that when the 74LS 257 multiplexers are put on the left sockets a 4164 RAM system will be used.

It should be also noted that the flip-flops used are schotky TTL devices since no big delays are permitted for the refresh timing.

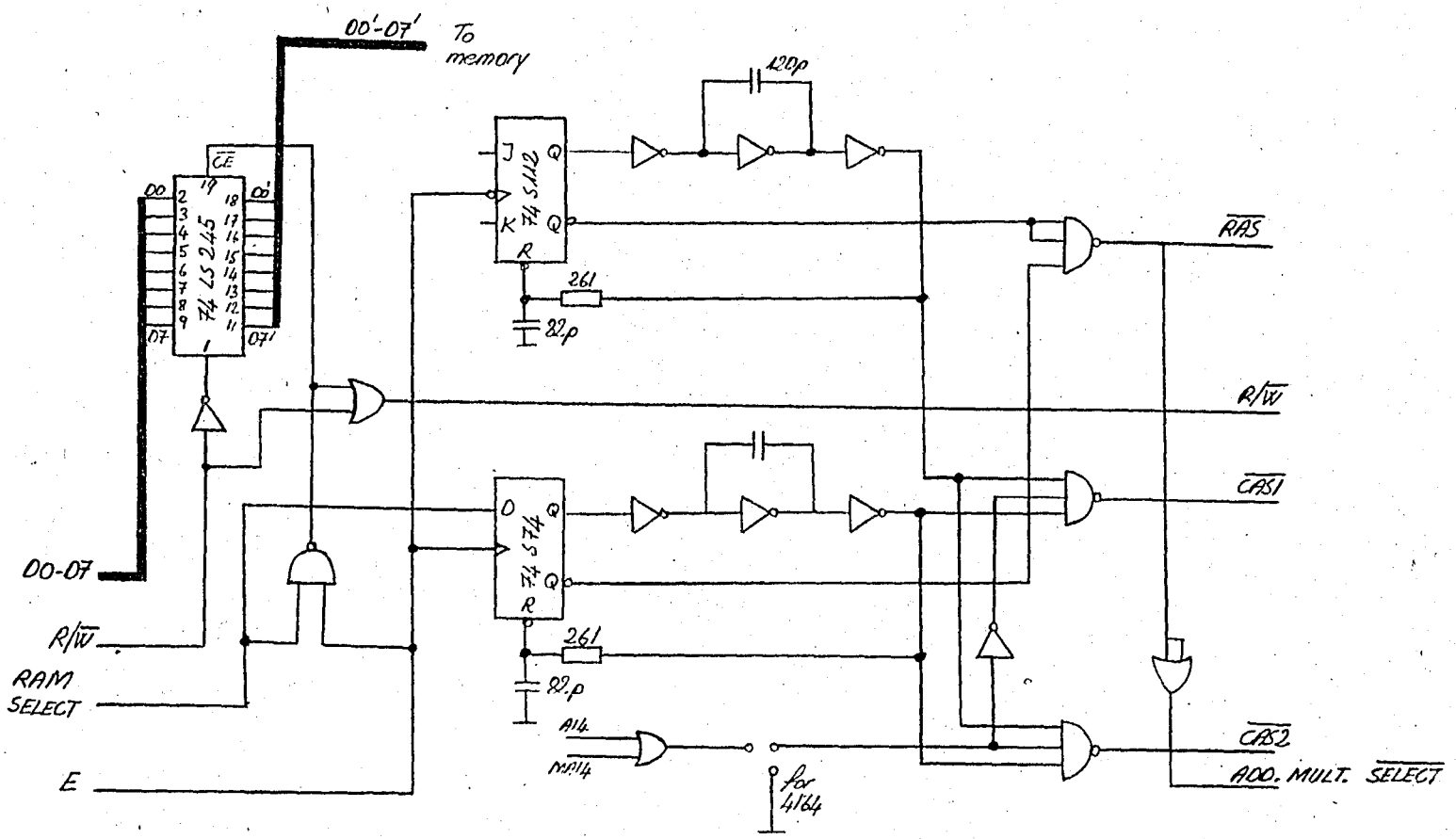
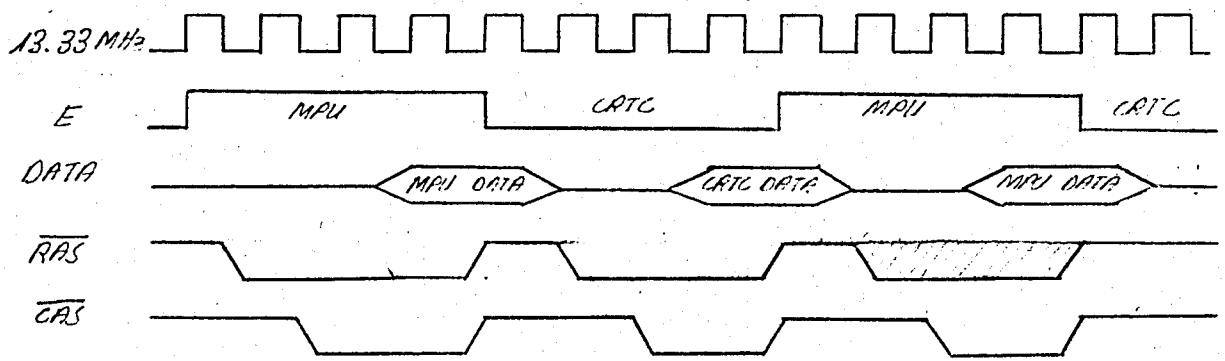


FIGURE 24

RAM CONTROL AND REFRESH -



Shaded are is when there is no MPU access to memory

FIGURE 25

MEMORY TIMING

3.3. CHARACTER MODE/GRAPHICS MODE SELECTION UNIT

One of the important features of the intelligent terminal is that the character set can be changed at any instant during a program or it can be a part of the system operating software so that when the system is on first the character set is generated. For this purpose is a 2K by 8 bit (6116) RAM instead of a read only memory which is generally used.

The peripheral interface adapter (PIA-3) is used for writing the character set into the character generator and for selecting the character or graphics mode of operation.

The character set is written into the character RAM in a two step process. First the memory address is latched using port A and PBO-PB2 of the PIA. PAO-PA7 is latched by the 74 LS 374 octal latch 1. Then to the address whose 8 bits are now latched and the other three bits are present at ports PBO-PB1 and PB2 using again the 8 bits of the port A data is written into the character generator. PB7 enables the outputs of the latch and CB2 controls the clock signal.

In the character mode of operation latch - 1 is disabled and the outputs of latch 2 are enabled using BB7 of the PIA. Now the character generator is addressed by latch-2 and the raster address RAO-RA3 of the CRCT. The data from the 6116 is now present at the shift register inputs which forms the serial video output signal.

In the graphics mode the character generator is disabled and now data from the latch-2 by-passes the character generator.

The operatin g modes are summarized in Table 17.

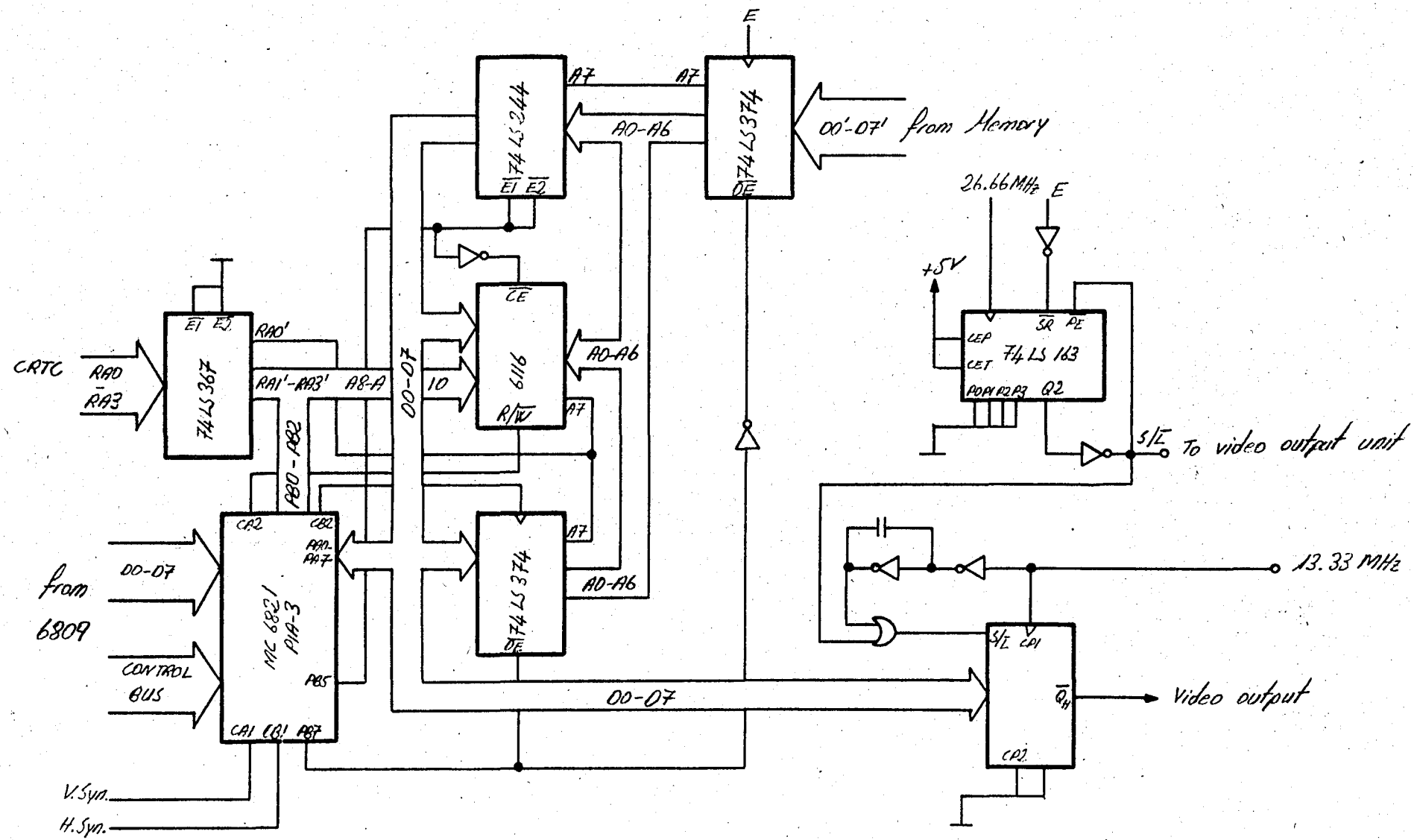


FIGURE 26

CHARACTER/GRAPHICS MODE COTROLLER

TABLE 17

CHARACTER/GRAPHICS CONTROLLER OPERATION

	DIA				OPERATING MODE
PB5	PB7	CA2	CB2	PORT A	
1	1	0		A0-A7	6116 is addressed
1	0	0		D0-07	Data is written onto 6116
1	1	1			Character mode
0	1	X			Graphics mode

3.4. VIDEO OUTPUT UNIT

This unit consists of the video attribute circuit and video output signal driving circuit (Figure 27). Available video attributes are:

(i) Invisible video

(ii) Blinking video

(iii) Half intensity

(iv) Video inverse

(v) Underline

The user can have either the composite video output signal or the video signal, vertical synchronisation and horizontal synchronisation signals separately. There is also an output for the light Pen Strobe (LPSTR) input of the CRT controller.

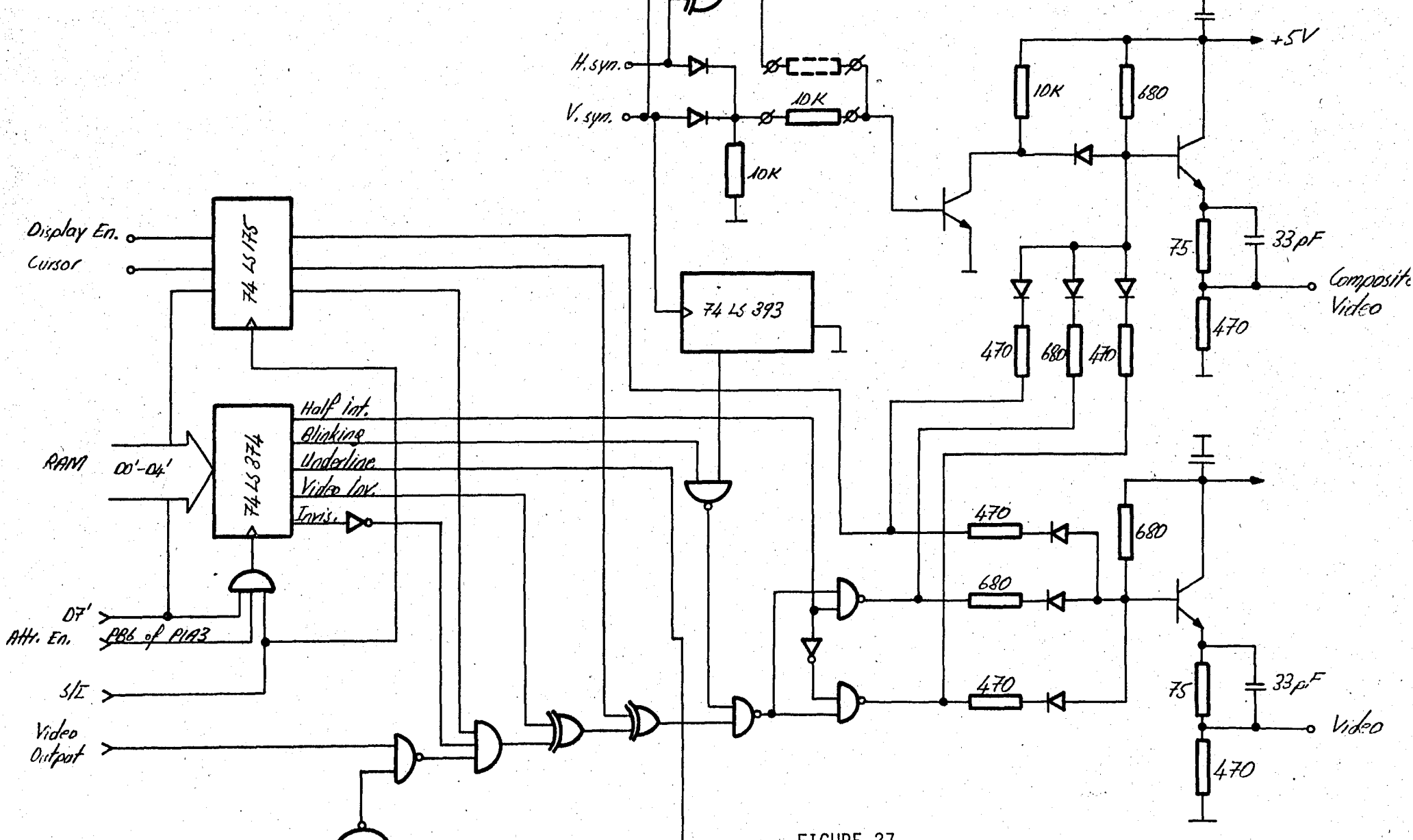


FIGURE 27
VIDEO OUTPUT UNIT

CHAPTER 4

VIDEO MONITOR

The video monitor of the intelligent terminal consists of three main units, the horizontal and vertical drivers, video amplifier and CRT. The horizontal and vertical driving circuits are obtained from BEKO-HITACHI TİP 92, 31cm. television set. Since a terminal requires a large bandwith video amplifier for the sharpness of the characters and graphic displays an amplifier circuit is designed and realized. As shown in figure 28 the amplifier consists of a FET input stage, a cascode amplifier and and a push-pull output amplifier.

The cascode stage has the advantage of having a high gain with a low D-factor. The cascode arrangement presents a low impedance at the collector of transistor T2. The load at this point consists of the impedance to ground looking into the emitter of T3. Since this impedance is small, the voltage gain and the voltage swing at the collector of T2 are relatively small. Althought the voltage swing at the collector of T2 is small, transistor T3 acts as a common base stage with high voltage gain.

There are several types of power output stages: class-A, class-B and class-C power amplifiers.

The class-A stage is one that conducts output current continuously during the complete cycle of a periodic input signal. This stage operates as a linear stage at all times. This stage is less complex than the class-B or class-C stage but is less efficient and delivers less power to the load than do the class-B and class-C stages.

Class-B operation requires two output stages. One stage conducts output current only when the input signal moves in the positive direction from its zero level. The other stage conducts current only when the input signal moves in the negativ direction from the zero level. The currents from both stages are summed and used to create a voltage that is an amplified version of the input signal.

A push-pull amplifier stage is a commonly known class-B power amplifier. The advantages of the push-pull stage are

(i) Very small, no signal power dissipation.

(ii) Class-B stage delivers more power to the load than the class-A stage.

(iii) Push-pull stage has less distortion than the single ended stage.

(iv) The class-B stage has higher theoretical maximum circuit efficiency.

This video amplifier circuit is realized and tested. The bandwith obtained is 25 MHz which is sufficient for the intelligent terminal.

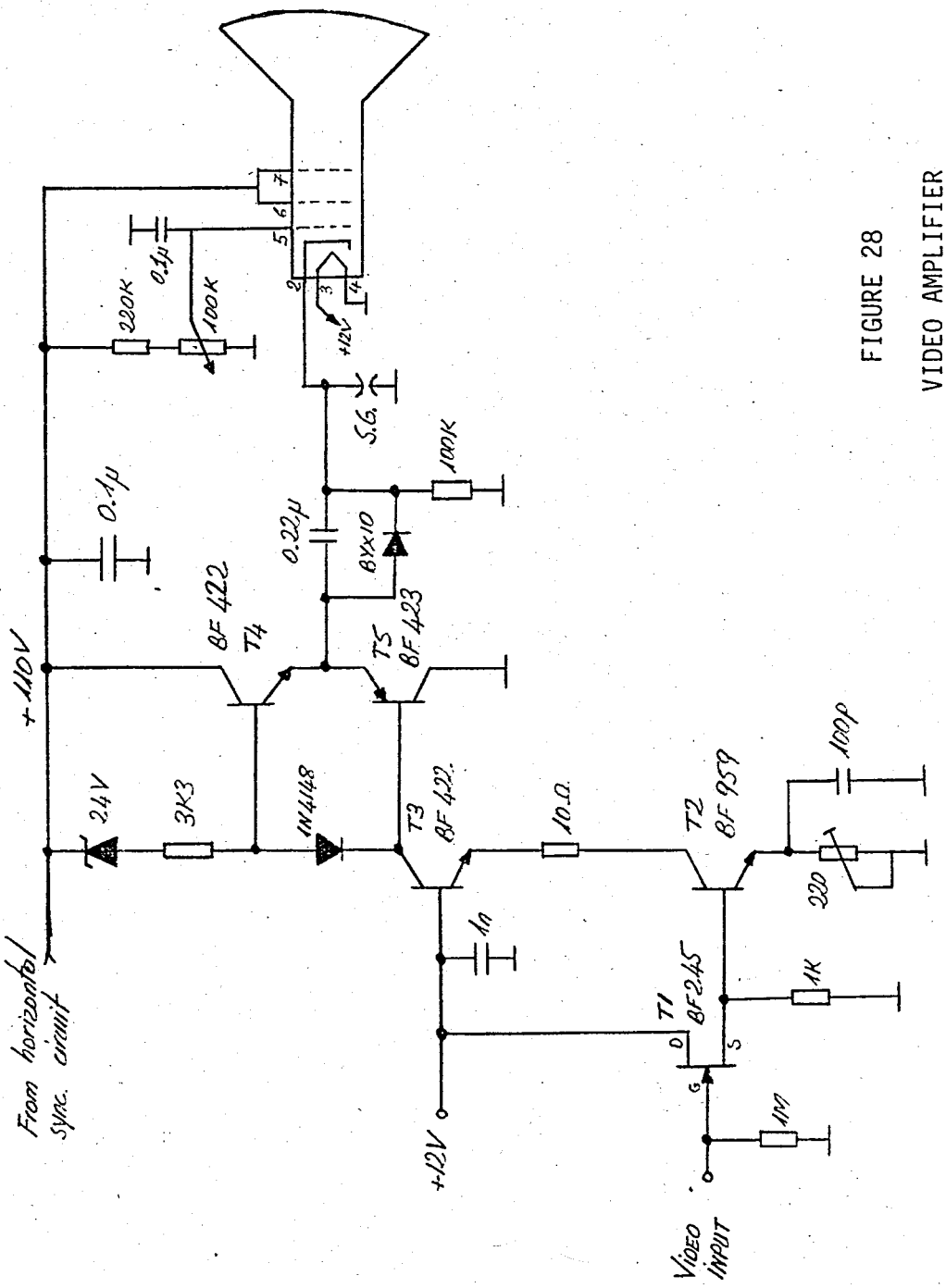


FIGURE 28

VIDEO AMPLIFIER

CHAPTER 5

DISCUSSION

The intelligent CRT terminal with the following features is realized:

1.	Character matrix:	7 x 9 dots.
2.	Character block : shifted lower case	8 x12 dots.
3.	Frame rate: interlaced raster scan	25 Hz
4.	Rows of characters	24
5.	Characters per row	80
6.	Character rate	1.666 MHz
7.	Dot clock rate	13.333 MHz
8.	Character time	0.6 μsec.
9.	Dot time	0.075 μsec.
10.	MPU clock	1.666 MHz
11.	Active graphic display area	256 x 512 dot matrix.

To lower the cost of the system a single printed circuit board is preferred but this caused a problem on the realization of the system because of the production difficulties of PCB's in Turkey. Since the system requires a large PCB area the production caused problems so the design of two separate boards would be more efficient.

CHAPTER 6

CONCLUSION

The complete hardware of the intelligent terminal is designed and realized during the thesis work.

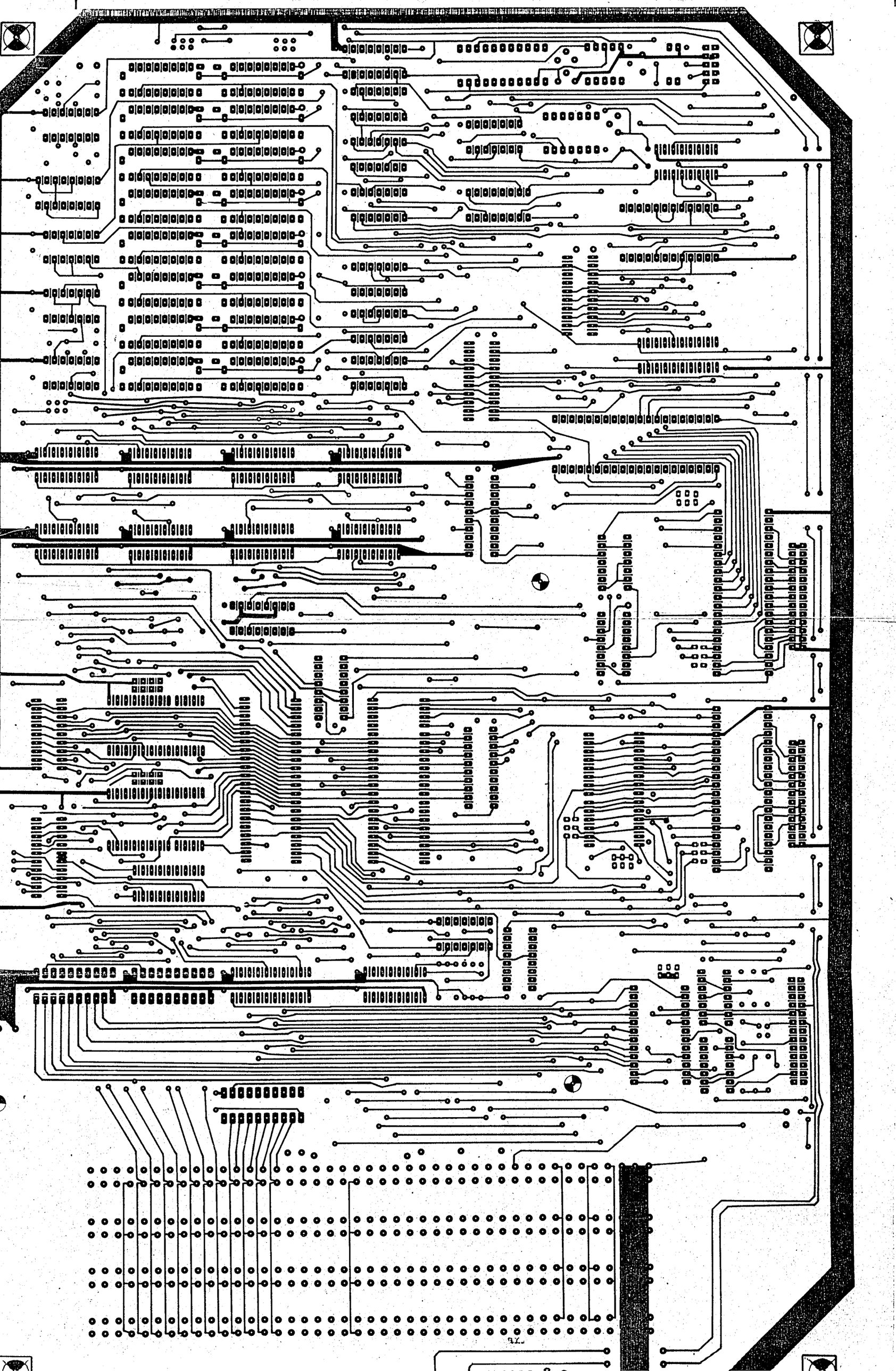
The terminal has expandable memory capacity up to 58 K bytes, software controlled character set generation, one RS-232 interface with software selectable baud rate generation, four parallel output ports for keyboard and printer interfaces and four output slots for expansion cards such as a floppy disk controller.

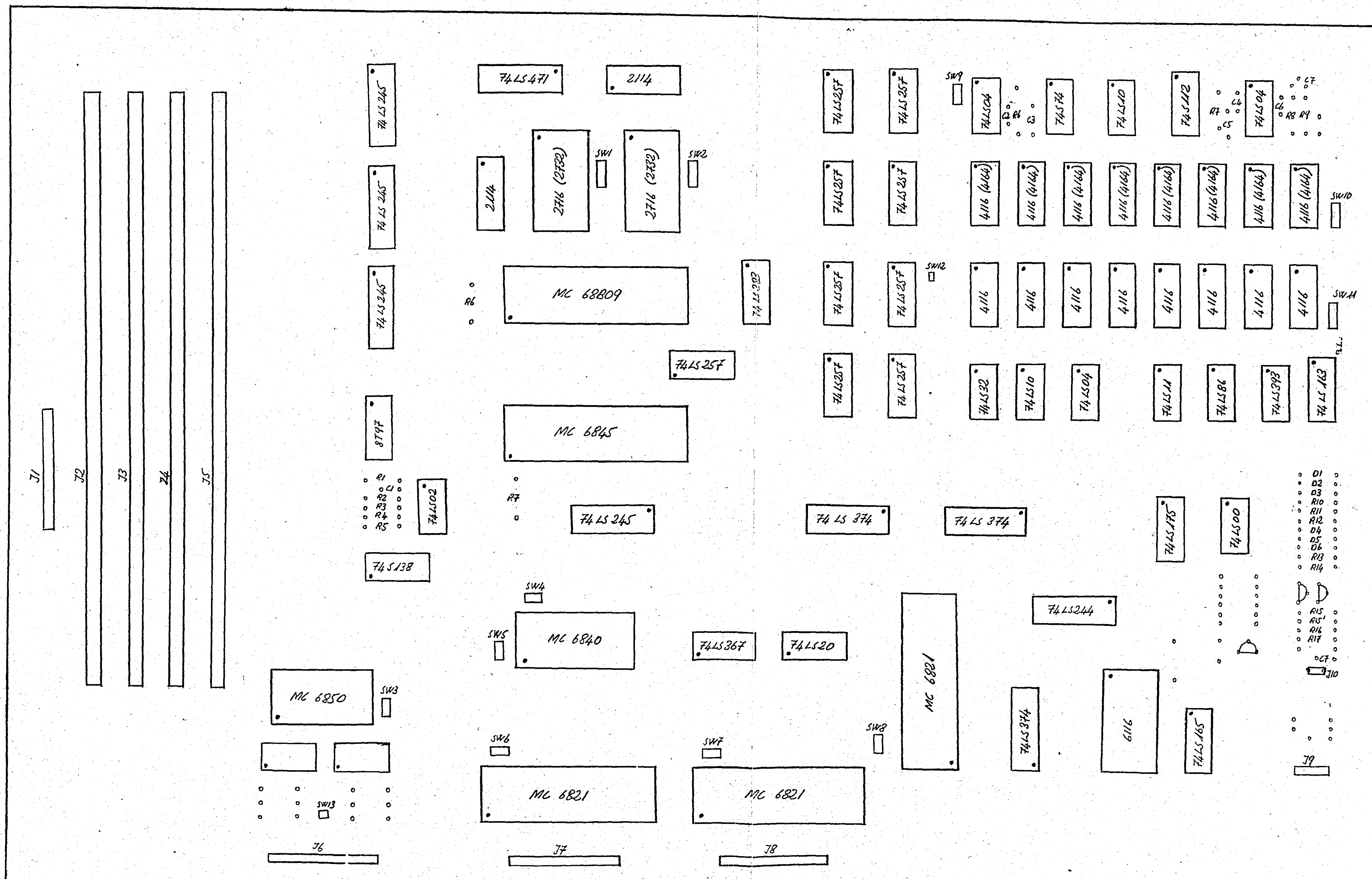
The design is not only an intelligent CRT terminal but with its hardware features and a powerfull operating system software, it can be used as a self contained microcomputer system.

BIBLIOGRAPHY

1. C.Melear "An Intelligent Terminal With Data Link Capability" MOTOROLA Sem.Pro.Inc. AN-830, 1981.
2. D.L.Ruhberg "Using the MC 68000 and the Mc 6845 For a Color Graphics System" MOTOROLA Sem.Pro.Inc. Application Note AN-834, 1981.
3. B.Bratt "A CRT Display System Using NMOS Memories" MOTOROLA Sem.Pro Inc. Application Note AN-706
4. J.Roy and D.Morris "A CRT Terminal Using The M6800 Family" MOTOROLA Sem.Pro.Inc. Application Note AN-773 January, 1977.
5. A.Watson III. " A Simplified Theory of Video Graphics " BYTE November 1980
6. E.G. Booch "Micrograph" BYTE November 1980
7. "Chip Controls CRT Attributes" Electronics June , 1978.
8. C.K.Adams " A Beginner's Guide To Computers and Microprocessors" TAB Books Inc. July, 1980 U.S.A.
9. D.J. Comer "Modern Electronic Circuit Design" Addison Wesley, 1976 U.S.A.

APPENDICES





J1
J2
J3
J4
J5
74LS245
74LS245
74LS245
74LS471
2114
2114
2716 (2732)
2716 (2732)
SW1
SW2
MC 68809
R6
74LS257
MC 6845
74LS02
74S138
R1
C1
R2
R3
R4
R5
R7
74LS245
74LS374
74LS374
SW4
MC 6840
SW5
74LS367
74LS20
MC 6850
SW3
SW13
SW6
SW7
SW8
MC 6821
MC 6821
MC 6821
J6
J7
J8
SW9
SW10
SW11
SW12
74LS257
74LS257
74LS257
74LS257
74LS257
74LS257
74LS257
74LS257
74LS04
74S74
74LS10
74LS112
74LS04
4116 (4164)
4116
74LS32
74LS10
74LS04
74LS00
74LS244
74LS374
6116
74LS165
D1
D2
D3
R10
R11
R12
D4
D5
D6
R13
R14
R15
R16
R17
J9
J10