A personal computer digital oscilloscope

by

Wassef M. Masri

A Thesis Submitted to the

Graduate Faculty in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

Department: Electrical Engineering and Computer Engineering
Major: Computer Engineering

Iowa State University
Ames, Iowa
1991

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

The personal computer (PC) is playing a big role in our everyday lives. It is becoming involved in more aspects of our lives than ever, with applications that range from word processing and playing video games to monitoring and controlling other instruments.

An application worth investigating is using the PC to emulate the work of another instrument, namely the oscilloscope. By adding the necessary hardware and/or using the right software, other functions can be added to the set of functions the PC can perform, and thus replacing the oscilloscope.

The design process of the PC Oscilloscope includes designing the hardware circuitry necessary to handle and condition the signal to be displayed on the PC monitor. This hardware is then interfaced to the PC system and the necessary software code is generated to take care of manipulating and displaying the data collected by the PC.

CHAPTER 1. OVERVIEW OF THE IBM PC SYSTEM

For the PC to implement the functions of an oscilloscope, additional circuitry has to be added to it. In order to understand where the PC itself fits in such a scheme. and how it will be integrated with the additional hardware. it is important to understand the PC system.

This chapter discusses the PC system in general. and those parts that are related to our design in particular. Special emphasis is given to interfacing the PC to external circuitry.

1.1 System Board

The system board resides inside the PC cabinet, along with several add on boards, one or more disk drives, a power supply and a fan.

Figure 1.1 shows the block diagram of the system board. On this board. there are five I/O channels (eight on the PC XT system) in which expansion boards can be installed. Up to 256k bytes of RAM and 40k bytes of ROM can be installed.

There are five important chips on the system board:

- The 8088 microprocessor: It has eight 16-bit registers used to hold the addresses, status flags and intermediate results and operands of arithmetic and logical operations.

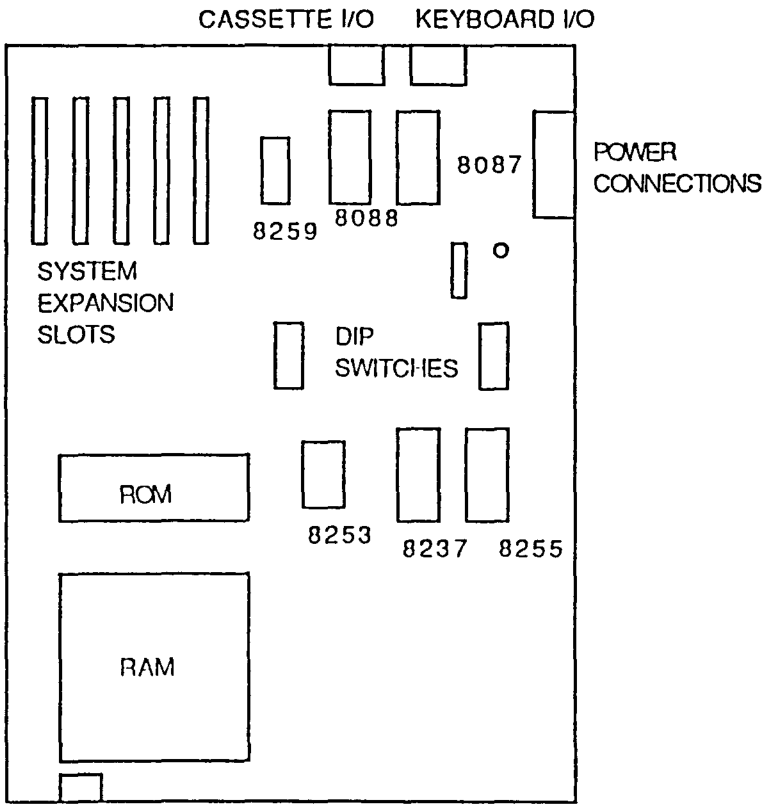


Figure 1.1: The system board block diagram

- The 8254 counter/timer chip: When the power is on, this chip allows the PC to maintain the date and time of day, keep track of refreshing the RAM at specific time intervals, and create sound through the use of a speaker.

- The 8237 DMA chip: It allows slower devices like disk drives controllers to transfer data to and from the main memory without involving the CPU. It also refreshes the RAM together with the 8254 chip.

- The 8255 I/O chip: Also called the PPI (Programmable Peripheral Interface chip), it is used to read the two system board DIP switches, which tells the PC the amount of memory it has, the number of disk drives and the type of screen used. The PPI is also used to read data coming from the keyboard.

- The 8259 eight-channel interrupt controller chip: This chip handles the hardware interrupts. It sends the interrupts to the CPU according to their priority level.

1.2 System Bus

Interfacing a card to the PC is done through the system bus. The system bus, shown in Figure 1.2, has 62 signal lines which consist of 20 address lines. 8 data lines, 27 control lines and 7 power supplies and ground. The function of each of these signals is as follows:

- A0 through A19

  These address signals are the output lines from the CPU or the DMA controller to address the memory or the I/O ports. They are used to address up to 1

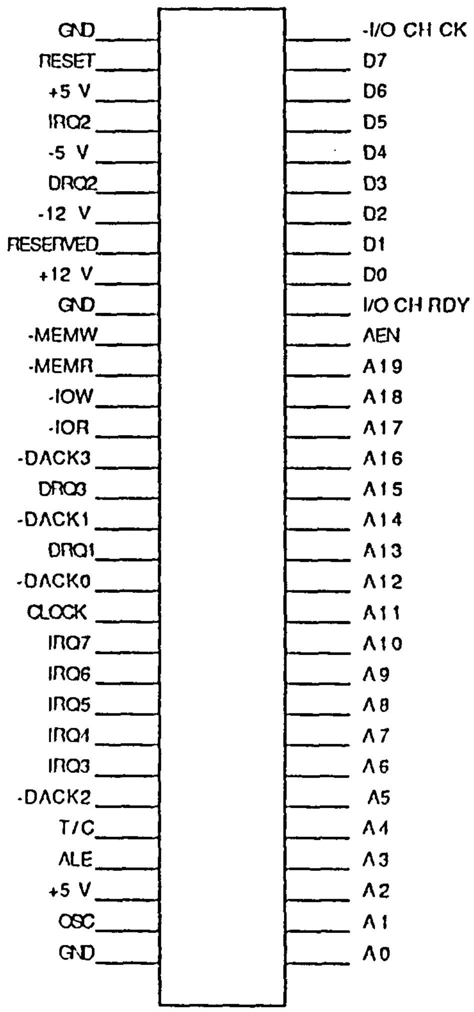


Figure 1.2: The I/O channel bus

megabyte of memory. When addressing I/O ports, only 10 address lines (A0 through A9) are needed.

- D0 through D7

  These are the data lines used to transfer data, one byte at a time, among the CPU, the memory and the I/O ports. Data should be placed on these lines prior to the activation of $\overline{\text{IOW}}$, $\overline{\text{MEMW}}$, $\overline{\text{IOR}}$, $\overline{\text{MEMR}}$.

- $\overline{\text{IOW}}$, $\overline{\text{MEMW}}$, $\overline{\text{IOR}}$, $\overline{\text{MEMR}}$

  These signals are low-level active control signals from the CPU. $\overline{\text{MEMW}}$ and $\overline{\text{IOW}}$ indicate that the CPU wants to write data to the memory or to the I/O ports, respectively. Similarly, $\overline{\text{MEMR}}$ and $\overline{\text{IOR}}$ indicate that the CPU wants to read data from the memory or from the I/O ports, respectively.

- IRQ2 through IRQ7

  These lines are used to request the hardware interrupts (2 through 7) to the CPU. They are input to the 8259 interrupt controller (discussed earlier). If the interrupt is not masked, a signal will be generated to interrupt the CPU.

- ALE

  The address latch enable line. It is output from the bus controller to indicate the validity of the address on the address lines.

- AEN

  AEN is issued by the DMA controller indicating a DMA bus cycle. It is used to disable the CPU address, data, and control buses from the system bus.

- DRQ1 through DRQ3 and $\overline{\mathrm{DACK0}}$ through $\overline{\mathrm{DACK3}}$

  DRQ1 through DRQ3 are direct memory access request lines and are used to request the DMA bus cycle. $\overline{\mathrm{DACK0}}$ through $\overline{\mathrm{DACK3}}$ are low-level active, direct memory access acknowledge lines. They are issued by the 8237 DMA controller to indicate that a DRQ has been granted.

- TC (Terminal Count)

  This signal is issued by the DMA controller to indicate that it has reached the preprogrammed number of transfer cycles.

- I/O Ch Ck (I/O Channel Check)

  This signal is used by interface cards to indicate error conditions.

- I/O Ch Rdy (I/O Channel Ready)

  This signal is used to extend the length of the bus cycles. It allows the memory or the I/O ports that are not fast enough to respond to the CPU bus cycle to still be attached to the system bus.

- OSC (Oscillator) and Clk (Clock)

  OSC is the signal that has the highest frequency on the system bus (14.31818 MHz). Other timing signals, like Clk, are derived from OSC. Clk is the system clock and has a frequency of 4.77 MHz. The period is 210 ns with 70 ns high and 140 ns low time.

- RESET DRV (reset driver)

Table 1.1: The PC 1-megabyte memory addresses

Address	Function
0-3FF	Interrupt vector table
400-47F	BIOS data area
480-5FF	BASIC and special system function RAM
600-9FFFF	Program memory
A0000-AFFFF	EGA graphics mode video RAM
B0000-B7FFF	Monochrome video RAM
B8000-BFFFF	Color/graphics video RAM
C0000-CFFFF	I/O ROM BIOS's, EMS window
D0000-DFFFF	I/O ROM BIOS's, EMS window
E0000-F3FFF	Unused on PC
F4000-F5FFF	PC spare ROM socket
F6000-FDFFF	ROM BASIC
FE000-FFFFF	ROM BIOS

When the system power is turned on, this signal is held high until all levels have reached their operating range. RESET DRV then goes low.

- Power Supplies and Ground

  The power supplies are +5, -5, +12 and -12 Volts DC, and GND (ground).

1.3 Addressing Scheme

The PC addresses its memory and its I/O ports separately. It uses twenty-bit addresses to address up to one megabyte of memory, while it uses only ten-bit addresses to address up to 64k I/O port addresses.

Table 1.1 shows how the PC allocates its one megabyte memory addresses. According to the table, memory addresses E0000h-F3FFFh are unused by the PC. It will be shown later how a 64k memory chip, will be mapped into this range of addresses.

Similarly, Table 1.2 shows how the PC allocates its 64k I/O port addresses. The control circuit used in our design will occupy a portion of the addresses in the range 300h-31Fh, which is allocated for prototype cards.

1.4 Video Display

The IBM PC has addresses A0000h-BFFFFh reserved for memory-mapped displays. Some of the popular display boards for the IBM PC are: the IBM monochrome adaptor, the IBM color/graphics adaptor (CGA), the PCjr video system, and the IBM enhanced graphics adaptor (EGA). All four systems are centered upon the Motorola 6845 CRTC (cathode ray tube controller) chip. The 6845 manages a number of technical tasks that are not ordinarily of concern to programmers. It also sets the screen mode, generates and controls the cursor, and (on the color graphics adaptor) assigns colors.

The monochrome card supports one screen mode, the color card supports seven, the PCjr supports ten, and the EGA supports twelve. These various modes are listed in Table 1.3.

Although the 6845 chip is easy to program directly, it is easier to make use of the large set of video screen subroutines that make up BIOS INT 10H, which can handle most of the chip's operations. Indeed, in our design, these subroutines are used to display the signals on the PC monitor. Table 1.4 shows the available functions in INT 10H. Functions are accessed by executing an INT 10H with register AH containing the function number. DOS INT 21H provides additional higher level screen functions.

Table 1.2: The PC 64k I/O port addresses

Address	Function
0-1F	8237 DMA controller
20-3F	8259 interrupt controller
40-5F	8253 counter/timer
60-7F	8255 PPI
80-9F	DMA 64k page register
A0-BF	NMI reset
C0-DF	Unused on PC
E0-FF	8087 math coprocessor interface
100-1FF	Unused on PC
1F0-1F8	Unused on PC
200-20F	Game I/O adaptor
210-217	Expansion unit
220-24F	Reserved
250-277	Not used
278-27F	Second parallel printer interface (LPT2)
280-2EF	Not used
2F0-2F7	Reserved
2F8-2FF	Second 8250 serial UART interface (COM2)
300-31F	Prototype card
320-32F	XT hard disk
378-37F	First parallel printer interface (LPT1)
380-38C	SDLC
3A0-3AF	Primary Binary Synchronous
3B0-3BF	Monochrome display and LPT1
3D0-3DF	Color/graphics display adapter
3F0-3F7	$5\frac{1}{4}$" floppy disk drive controller
3F8-3FF	First 8250 serial UART interface (COM1)

Table 1.3: Screen display modes in the IBM PC

Number	Mode	Adaptors
0	40X25 (320X200) B&W alphanumeric	color, PCjr, EGA
1	40X25 (320X200) color alphanumeric	color, PCjr, EGA
2	80X25 (640X200) B&W alphanumeric	color, PCjr, EGA
3	80X25 (640X200) color alphanumeric	color, PCjr, EGA
4	320X200 4-color graphics	color, PCjr, EGA
5	320X200 B&W graphics (4 gray shades on PCjr)	color, PCjr, EGA
6	640X200 B&W graphics	color, PCjr, EGA
7	80X25 (720X350) B&W alphanumeric	monochrome, EGA
8	160X200 16-color graphics	PCjr
9	320X200 16-color graphics	PCjr
A	640X200 4-color graphics	PCjr
B	reserved by the EGA	—
C	reserved by the EGA	—
D	320X200 16-color graphics (EGA only)	EGA
E	640X200 16-color graphics (EGA only)	EGA
F	640X350 4-color graphics on monochrome display	EGA
10	640X350 4- or 16-color graphics	EGA

Table 1.4: INT 10H screen handling functions

Function Number	INT 10H Video Routine
0	Set new graphics or text mode
1	Set cursor size
2	Set cursor position
3	Read cursor position
4	Read light pen position
5	Select active display page
6	Scroll display window up
7	Scroll display window down
8	Read character and attribute at cursor position
9	Write character and attribute at cursor position
A	Write character at cursor position
B	Set color palette
C	Write dot
D	Read dot
E	Write teletype (simulate a dumb printer)
F	Read current video state

CHAPTER 2. THE DESIGN STRATEGY

This chapter summarizes the reasons and methodology which led to the decisions made throughout the design process. It briefly discusses the related concepts, assumptions, and limitations imposed by the different trade offs. Technical details and specifications are discussed in later chapters.

The first issue to be considered was converting the analog signal into a digital signal which can be handled and manipulated by the PC. For that reason, an 8-Bit Flash A/D Converter was chosen. The eight-bit resolution gives 256 quantum levels which offers enough accuracy when displaying the signal on the PC monitor. It can also sample the signal at a rate of up to 15 Mhz. If used at its maximum rate, we can obtain a reasonable display for signals of frequencies up to 1 MHz, assuming that 15 samples per cycle will be enough to reproduce a periodic signal.

Storing and manipulating the data samples was the first obstacle in the design process. The dynamic nature of the PC memory makes it unsuitable for handling a real time function, i.e some samples will be lost in the time it takes the PC to refresh its memory. This leads to inaccurate displays.

The solution to this problem was using an external memory. Although the external memory used is also dynamic, it can be controlled such that refreshing its rows will not interfere with storing and manipulating the data.

Considering the A/D Converter to be the core of the design, we can view the rest of the circuitry as being built around the A/D Converter. Thus, the design can be divided into the following segments:

1. The pre-A/D Converter stage (input circuit).

2. The post-A/D Converter stage (external memory).

3. The timing circuit.

4. The control circuit.

2.1 The Pre-A/D Converter Stage (Input Circuit)

From the point of view of the input signal, this is the first stage it goes through and is used to condition the signal prior to its sampling. This is necessary for the following reasons:

- In order to use the full range of input voltage to the A/D Converter and thus achieving maximum accuracy in reading the signal off the screen, small signals need to be amplified.

- The input voltage range to the A/D Converter was chosen to be 0 to 5 volts, because of the availability of these reference voltage values on the PC board. Consequently, input signals are offset by a value of +2.5 volts just before they are input to the A/D Converter, in order to achieve symmetry around the zero value.

For amplification purposes, we need an operational amplifier that can provide a gain of up to 100 at frequencies up to 1 MHz. The chosen amplifier was a video line

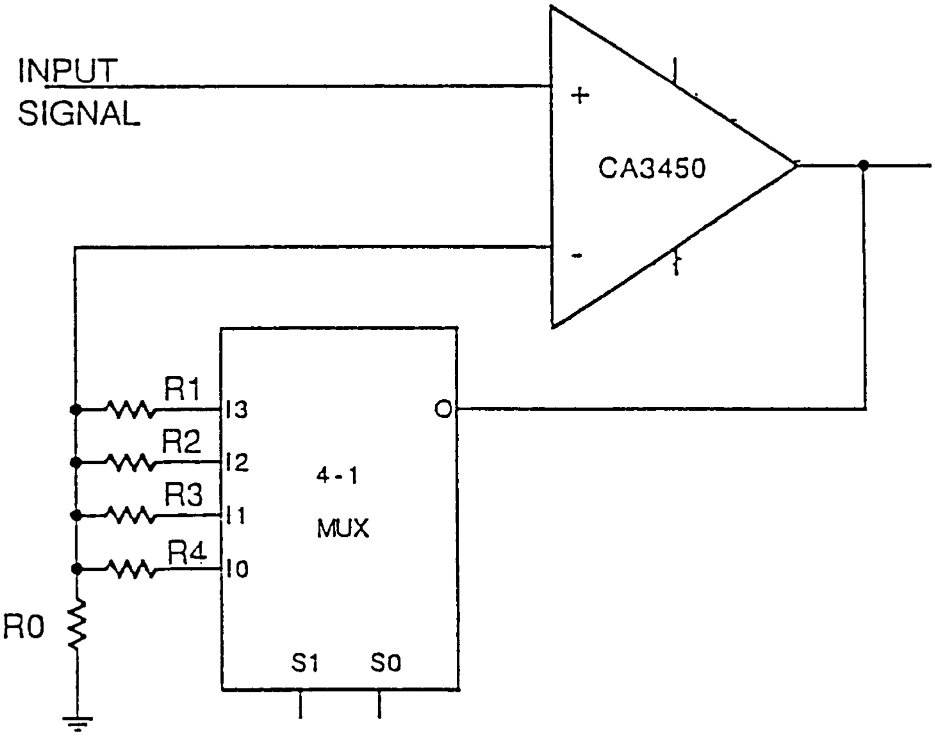


Figure 2.1: A variable gain op amp

driver, high-speed operational amplifier. Its gain-bandwidth product was experimentally determined to be 25 MHz, which necessitates the use of two such op amps in cascade to achieve the gain and bandwidth requirements.

The gain of each op amp can be varied by changing the value of the feedback resistor(s). In order to accomplish this, and in order to be able to control the gain by software means, the circuit in Figure 2.1 is used. This configuration uses a 4-1 multiplexer to switch among the feedback resistors and thus change the gain. The signals that control the multiplexer are generated by software means through the Control Circuit.

2.2 The Post-A/D Converter Stage (The External Memory)

A critical part of the design pertains to collecting the output samples from the A/D Converter, storing them and manipulating them in order to recreate the signal on the PC monitor. A solution is sought where a screen worth of data samples can be conditioned and displayed on the PC monitor, while at the same time, a separate "load" of data samples (another screen worth of data) is placed in another location in memory to be processed later. This scheme is accomplished by using an external memory.

The chosen memory chip is a dual port dynamic random access memory (also known as a video random access memory or a VRAM). It is organized as 65,536 (256 rows X 256 columns) words dynamic RAM port and 256 words serial access memory (SAM) port. Both ports can be accessed asynchronously. The word length is 4 bits which means that two such chips are needed to store the eight-bit data samples.

Figure 2.2 is a block diagram of a VRAM, and it shows the way data is processed. The PC uses the RAM port to access the data in one row and use it to update the display. At the same time, new data samples are shifted continuously into the SAM. These two operations are performed asynchronously until 256 data samples are shifted in, which means the SAM is full. At this point, the data in the SAM is transferred internally into a row in the DRAM. This procedure is kept track of by the PC.

To control the VRAM and generate the necessary timing signals, a Video System Controller (VSC) is used. Its function in this particular design is to access the DRAM port and to perform the internal transfer operation between the SAM and the DRAM, in addition to refreshing the VRAM without threatening the integrity of the data. These operations are only a subset of the functions of the VSC.

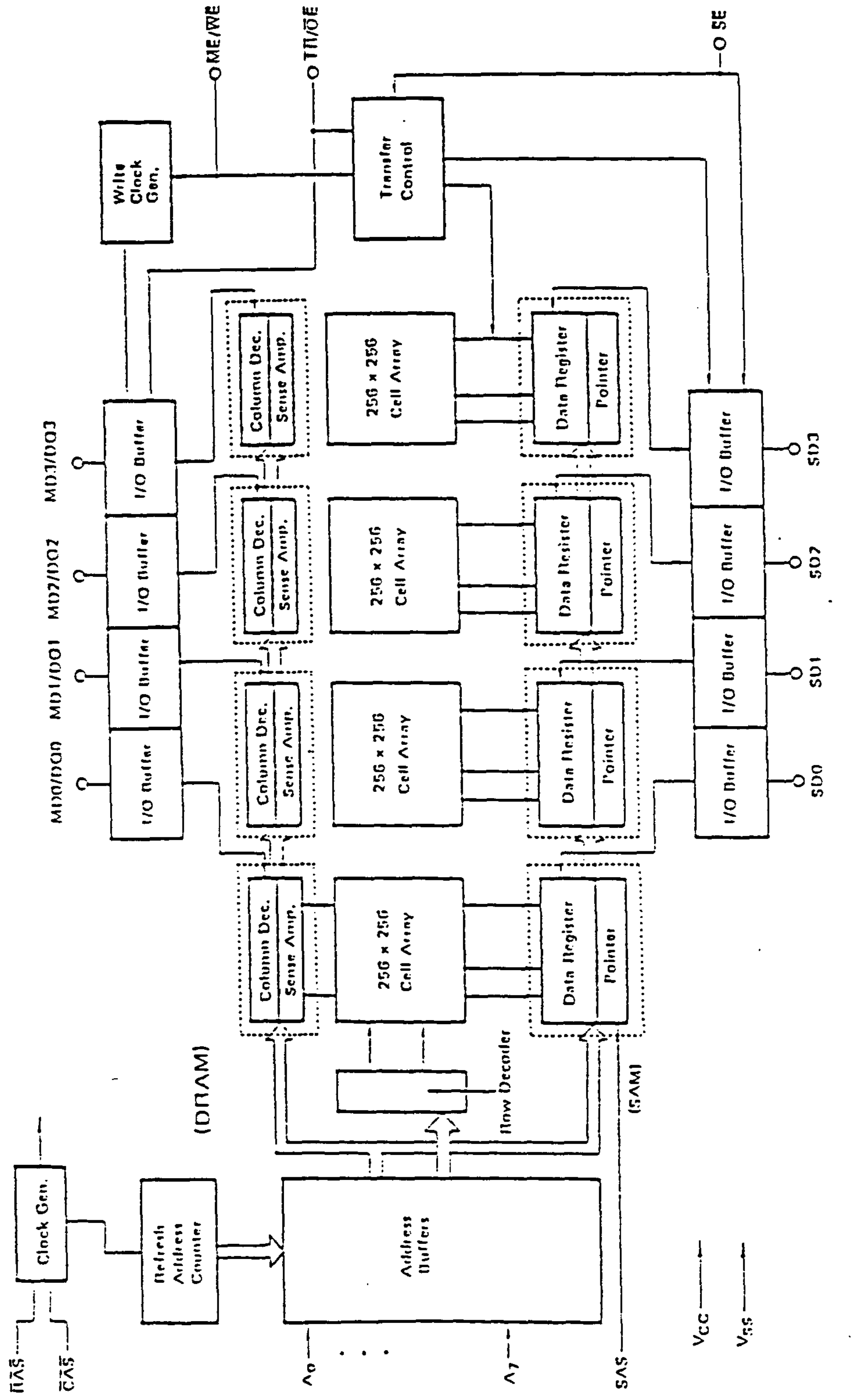


Figure 2.2: VRAM block diagram

2.3 The Timing Circuit

While the clock of the DRAM port is controlled by the PC, other parts of the design require separate timing circuitry, independent of the PC. The proposed timing circuit is needed to perform the following functions:

- Generate the clock signal for the A/D Converter (the sampling rate).
- Generate the clock signal for the SAM (the shifting clock). This signal has the same frequency as the sampling clock, since data is shifted into the SAM at the same rate it is sampled at.
- Trigger the start and end of the process of shifting data into the SAM, since only one screen worth of data (256 samples) is collected at a time.

These functions can be implemented using a crystal clock and a Programmable Interval Timer. With the crystal clock output used as its input, the Programmable Interval Timer can be programmed to generate the sampling clock signal (and hence the shifting clock signal), by simply operating as a frequency divider.

The Intel 8254 consists of three timers that can be programmed separately. One of these timers can be used to generate a retriggerable one-shot pulse to control the process of loading data into the memory, as will be explained later.

The Programmable Interval Timer can be interfaced directly to the PC, but we chose to handle it through the control circuit in order to avoid using the extra hardware associated with addressing it directly. The Timer is capable of handling clock inputs up to 10 MHz, which limits the sampling rate to that frequency, and lowers the bandwidth of our oscilloscope.

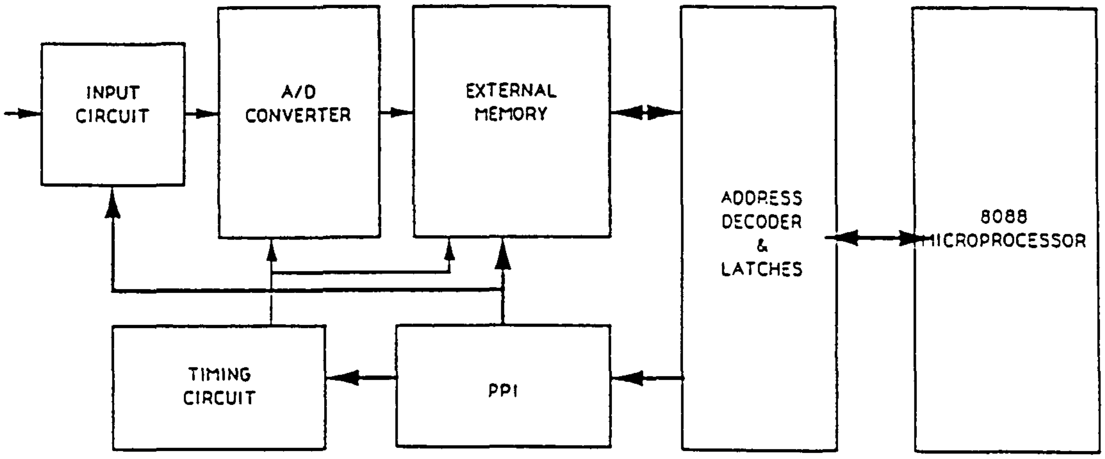


Figure 2.3: Block diagram of the PC board

2.4 The Control Circuit

As concluded from the previous sections, the control circuit is necessary to generate the select signals for the multiplexers in the pre-A/D Converter stage. In addition to that, it is through the control circuit that the processor actuates the timing circuit.

To implement these functions, it was found suitable to use a Programmable Peripheral Interface chip. This I/O device, addressable by the PC, has three eight-bit registers that can be used to communicate with the PC board. In this case, these registers will be used to communicate with the input circuit gain control channels as well as the Programmable Interval Timer.

The availability of unused pins on the Programmable Peripheral Interface (PPI) chip suggests making use of these pins to select the function (mode of operation) of the VSC. This arrangement saves the hardware circuitry otherwise necessary to decode the address and generate the select signals for the VSC. These concepts will be elaborated upon further in the following chapter discussing the hardware design.

Figure 2.3 shows a block diagram and thus the guidelines for this design. It is based on the design strategy adopted throughout this chapter.

CHAPTER 3. THE HARDWARE DESIGN

The circuitry integrated on the PC board is discussed in this chapter. This encompasses the different parts used, their specifications and their functionality as dictated by the design goals. The block diagram in Figure 2.3 shows that the design consists mainly of the following parts:

1. The Flash A/D Converter.
2. The input signal circuit.
3. The external memory.
4. The timing circuit.
5. The Programmable Peripheral Interface chip.
6. The address decoders and latches.

These circuits are implemented on a prototype card.

The analog input signal is conditioned before it is sampled and converted to an eight-bit digital signal. This signal is stored in the external memory, a dual port RAM, through the serial port. The samples are then read by the PC through the random access port and displayed on the screen.

3.1 The Flash A/D Converter

The A/D Converter used in the design is the CMOS Video Speed 8-Bit Flash Analog-to-Digital Converter (CA3318CE). It uses a parallel conversion technique to achieve a sampling rate of up to 15 MHz. The maximum input bandwidth allowed is 2.5 MHz.

The CA3318 uses a parallel technique to obtain its high-speed operation. The sequence consists of the "Auto-Balance" phase, $\phi 1$, and the "Sample Unknown" phase, $\phi 2$, as shown in Figure 3.1. Each conversion takes one clock cycle, where $\phi 1$ and $\phi 2$ are the high and the low periods of the same clock.

During the "Auto-Balance" phase, the first 256 commutating capacitors are connected to their associated ladder reference tap. The other side of these capacitors are connected to single-stage amplifiers whose outputs are shorted to their inputs by switches. This charges the first set of capacitors to their associated tap voltages.

In the "Sample Unknown" phase, all ladder tap switches and comparator shorting switches are opened. At the same time, V_{IN} is switched to the first set of commutating capacitors. Since the other end of the capacitors are now looking into an effectively open circuit, any input voltage that differs from the previous tap voltage will appear as a voltage shift at the comparator amplifiers. All comparators that had tap voltages greater than V_{IN} will go to a "high" state at their outputs. All comparators that had tap voltages lower than V_{IN} will go to a "low state".

The second-stage comparator amplifiers ac couple the status of all these comparators and store it at the end of $\phi 2$ by a latching amplifier stage. The latch feeds a second latching stage, triggered at the end of $\phi 1$. This delay allows comparators extra settling time. The status of the comparators is decoded by a 256 to 9-bit decode

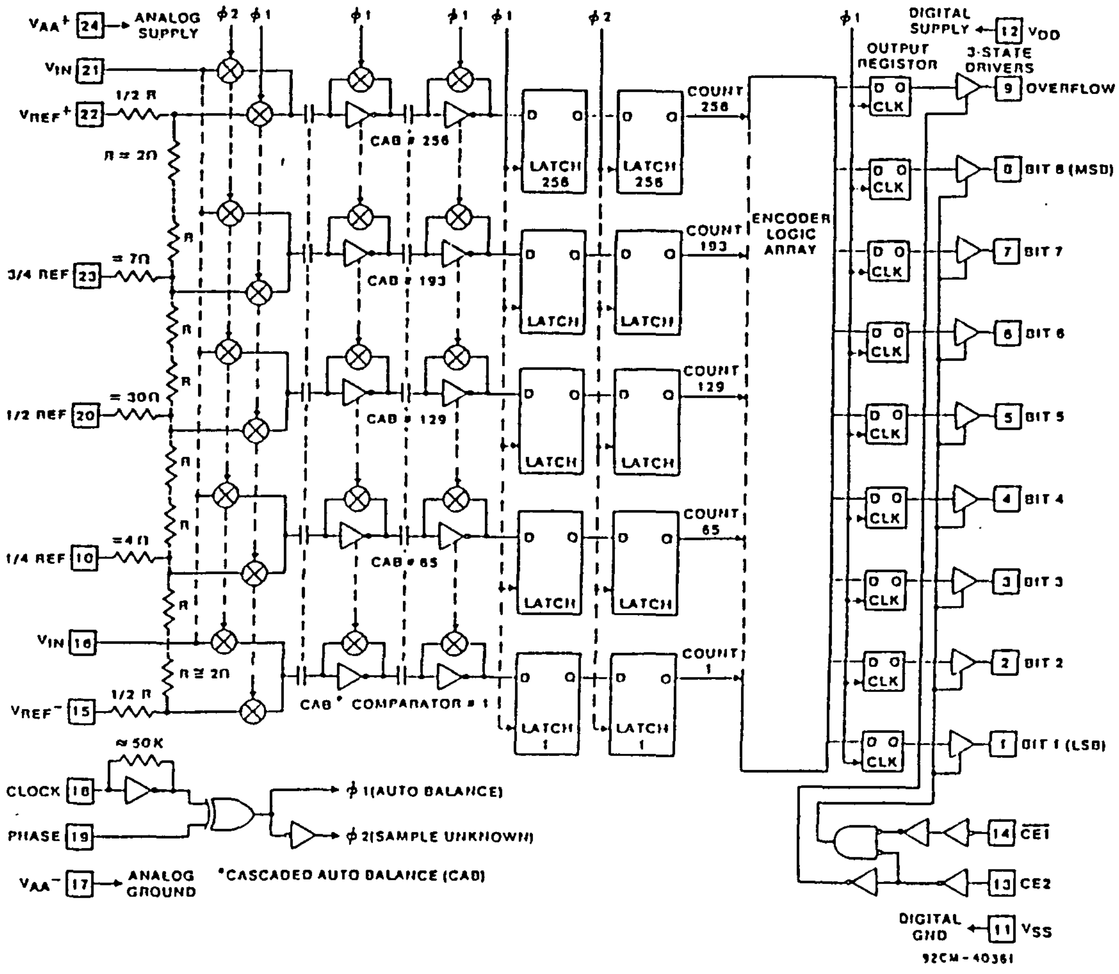


Figure 3.1: Block diagram for the CA3318 Flash A/D Converter

array, and the results are clocked into a storage register at the end of the next $\phi 2$.

The output of the 9 storage registers goes through a last stage of 3-state buffers which are controlled by two chip-enable signals: CE1 and CE2. In this design, these inputs are always enabled.

The circuit configuration for the A/D Converter is shown in Figure 3.2. From the figure, it is apparent that the input signal should range between 5 and 0 Volts. These are the levels of $V_{REF}+$ (higher reference voltage) and $V_{REF}-$ (lower reference voltage), respectively. These values are chosen because they are already supplied by the PC system. and at the same time. they satisfy the constraint that V_{IN} and ($V_{REF}-$)-($V_{REF}-$) full scale range should be between 4 and 7 Volts. Therefore. reasonable accuracy is achieved while maintaining a minimum amount of glue logic.

3.2 The Input Signal circuit

The Input Circuit is basically the stage at which the input signal is conditioned prior to the sampling stage. This entails amplifying the signal and dc offsetting it to meet the requirements set by the A/D converter.

Figure 3.3 shows the circuit diagram of the Input Circuit. The control signals needed to control the multiplexers' select signals and thus the gain are provided by software means through the Programmable Peripheral Interface chip, which is to be discussed later in this chapter.

The offset source, seen in Figure 3.3, is needed to offset the output of the Input Circuit by 2.5 volts. This means that signals that range between -2.5 volts and 2.5 volts will fall in the range 0 to 5 volts to meet the requirements of the A/D Converter set earlier.

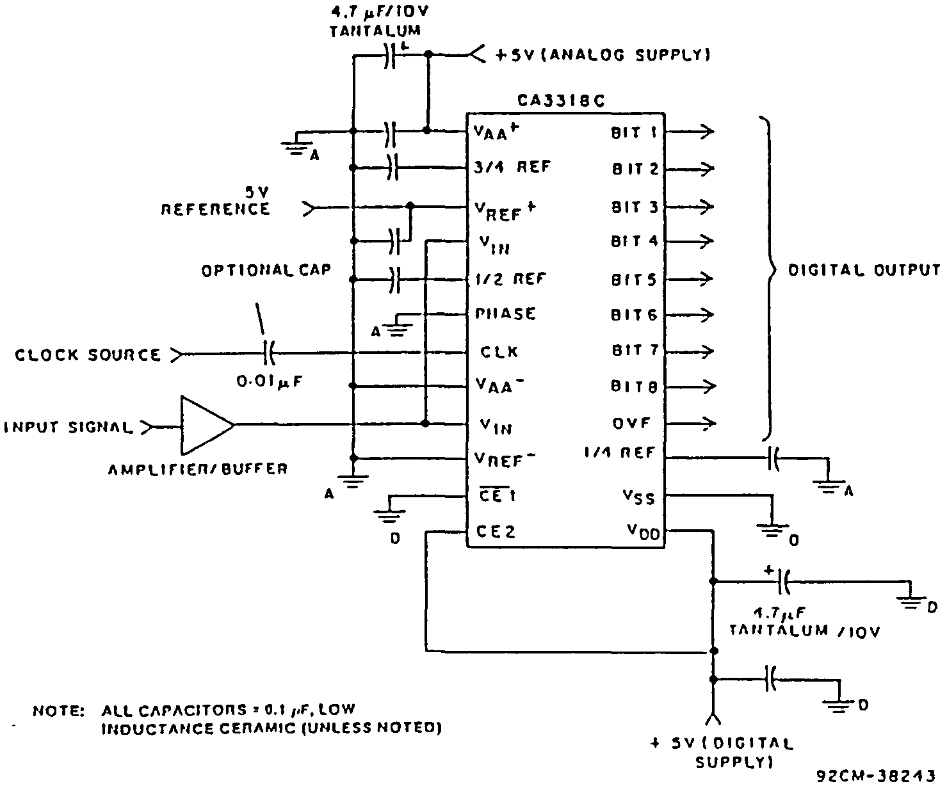


Figure 3.2: Circuit configuration for the CA3318 Flash A/D Converter

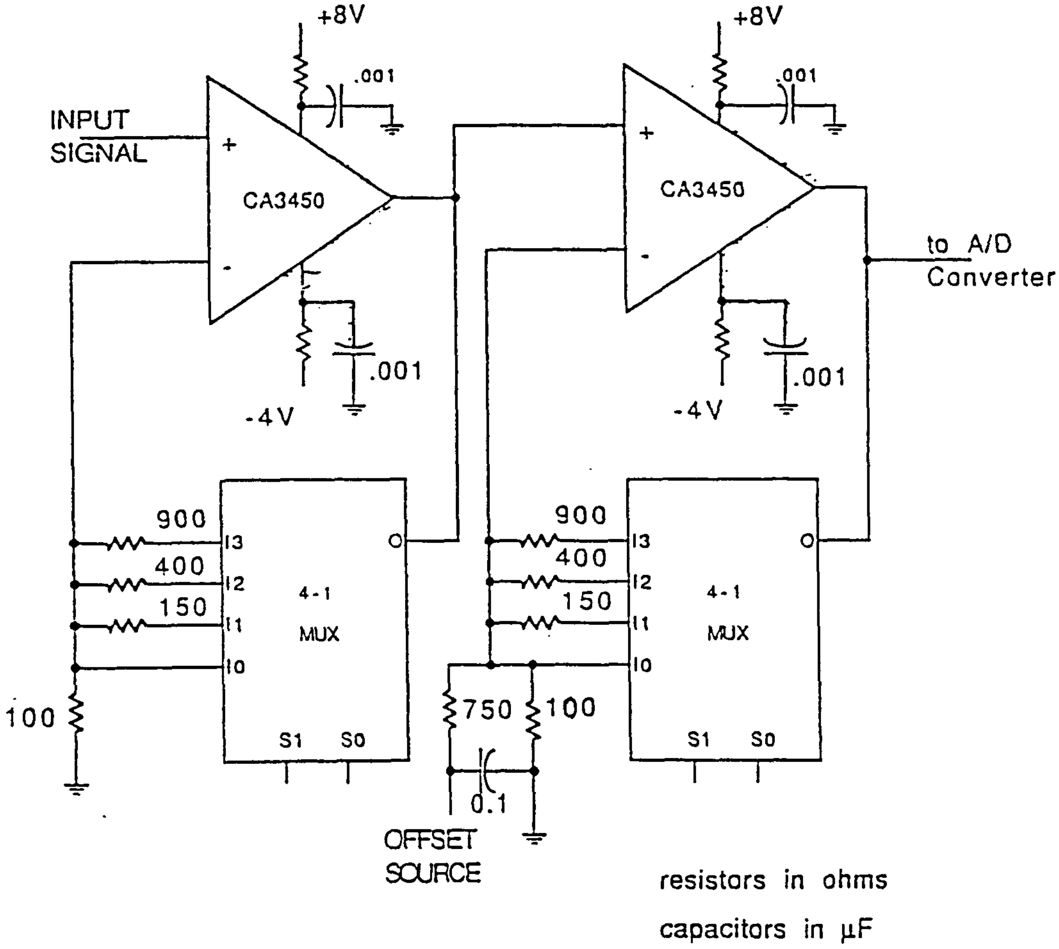


Figure 3.3: The input circuit

3.3 The External Memory

The external memory consists of a memory chip and a memory controller chip. The TMS34061 Video System Controller interface to the 256k dual port video memory requires a minimum of glue logic and is briefly discussed here. The microprocessor/VSC interface, on the other hand, is elaborated upon in more detail taking into consideration the decoding and timing requirements.

3.3.1 The dual port RAM (VRAM)

The Fujitsu MB81461B-12, used as an external memory in this design, is a fully decoded dual port NMOS dynamic random access memory organized as 65,536 words by 4 bits dynamic RAM port and 256 words by 4 bits serial access memory (SAM) port. Two MB81461-12's are used to store the 8-bit data sampled by the A/D Converter.

The DRAM port has four bits parallel random access I/O, while the SAM port is designed as four 256 bit registers each operating as a serial I/O. The four serial registers operate in parallel with each other during SAM port operation. Internal interconnects give the device the capability to transfer data bi-directionally between the DRAM memory array and the SAM data registers.

An important characteristic of the MB81461B is that it offers complementely asynchronous access of both the DRAM and SAM ports except when data is transferred between them internally. This signifies that we can perform two operations at the same time:

- Update the display by reading data through the DRAM port.

- Shift new samples into the SAM prior to transferring it internally to the DRAM.

Once the registers in the SAM port are filled with new data, the 256 8-bit samples are transferred simultaneously to the DRAM. Figure 3.4 shows the block diagram of the MB81461B. It is noticed from the figure that the 16 address bits are multiplexed over inputs A0-A7 to select 4 bits of 262,144 memory cell locations. The eight row address inputs are strobed by $\overline{\text{RAS}}$, then eight column address inputs are strobed by $\overline{\text{CAS}}$. The multiplexing process and the generation of other control signals is handled by the TMS34061 Video System Controller, which will be discussed next.

3.3.2 The memory controller

The Video System Controller (VSC) is a monolithic NMOS device which controls multiport video RAMs (VRAM or dual port RAMs), as well as the conventional 64K and 256K dynamic RAMs.

The VSC has four major functions:

1. Allows the host virtually unimpeded access to VRAMs, directly (host direct) or indirectly (X-Y).

2. Automatically generates the DRAM refresh cycles needed to maintain data stored within the DRAMs.

3. Performs display-update cycles needed to periodically load new video data into the VRAM shift registers.

4. Generates sync and blanking signals necessary for monitor control.

For the purposes of this design, only the first two functions are exploited. Specifically:

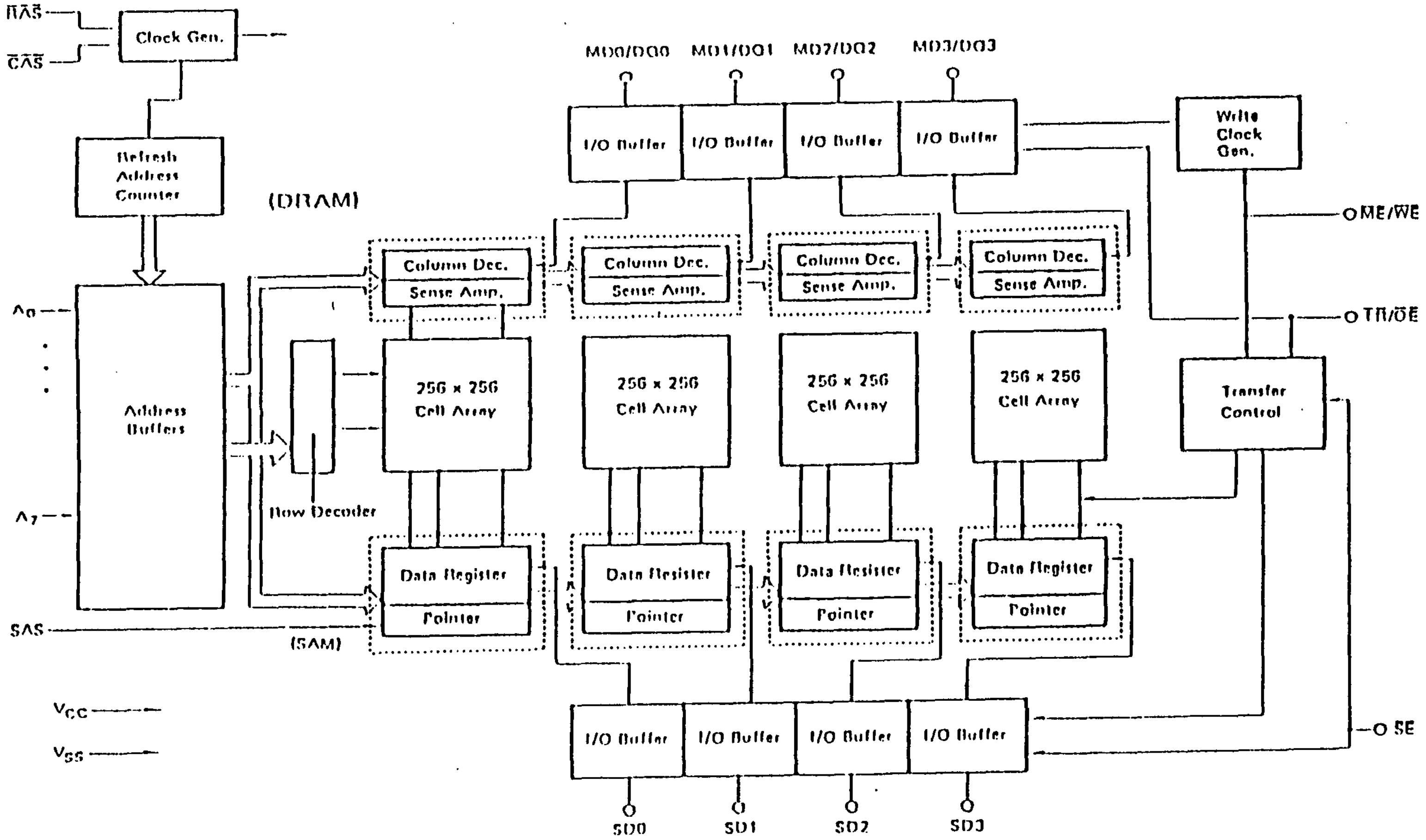


Figure 3.4: Block diagram of the MB81461B

- Allowing access to the VRAM, including the performance of data transfers from the SAM to the DRAM.
- Refreshing the DRAM.

3.3.3 The VRAM/VSC interface

The pin connections of the VRAM are shown in Figure 3.5. As can be seen in the figure, the address signals and most of the control signals are supplied directly by the VSC. $\overline{\mathrm{WM}}/\overline{\mathrm{W}}$, however, is generated differently. In Figure 3.5, gate A prevents the Write Mask Select/Write Enable ($\overline{\mathrm{WM}}/\overline{\mathrm{W}}$) signal on the VRAM from being active in the interval from $\overline{\mathrm{RAS}}$ time to 40 ns after $\overline{\mathrm{RAS}}$ time, thus disabling the write-per-bit feature on the VRAM. Gate B is used to allow $\overline{\mathrm{WM}}/\overline{\mathrm{W}}$ to be active low at $\overline{\mathrm{RAS}}$ time, in turn allowing the memories to do shift-register-to-memory transfers and memory-to-shift-register transfers. The outputs of gates A and B are ANDed together to provide the VRAM with the $\overline{\mathrm{WM}}/\overline{\mathrm{W}}$ signal.

3.3.4 The VSC/PC interface

In interfacing the external memory to the microprocessor, the following criteria were taken into consideration:

The minimum memory access time for an 8-MHz 8088 is four clock cycles, or 500 ns. These four clock cycles, T1 through T4, do not provide enough time to support DRAM read cycles because data is required by the processor before the end of cycle T3, while a DRAM read cycle extends into cycle T4. The solution to this problem is simply adding one wait state and thus increasing the memory cycle to 625 ns. This does not require any additional hardware since the VSC can be

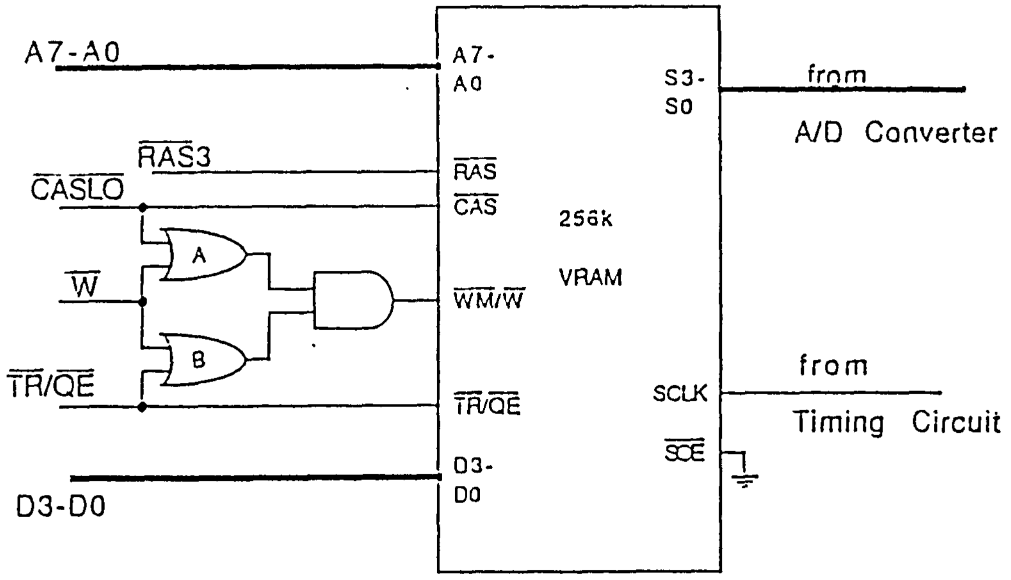


Figure 3.5: VRAM/VSC Interface

programmed to insert a delay of up to 7 wait states. A cycle is delayed by postponing the ready signal to the host processor.

The VSC requires its ALE (Address Latch Enable) and $\overline{\text{CEL}}$ (Column Address Enable) inputs to be synchronous to the system clock SYSCLK. This means that these inputs must be stable 20 ns before and 10 ns after the rising edge of SYSCLK. The 8088 processor specification states that for an 8-MHz processor, $\overline{\text{MEMR}}$ and $\overline{\text{MEMW}}$ fall between 10 and 70 ns after the falling edge of its output clock. Because the $\overline{\text{MEMR}}$ and $\overline{\text{MEMW}}$ signals can fall either before or after the rising edge of CLK, CLK cannot be used as SYSCLK directly. However, inverting CLK and using the inverted signal as SYSCLK input allows the $\overline{\text{MEMR}}$ and $\overline{\text{MEMW}}$ signals to meet the setup and hold times with respect to the VSC SYSCLK input.

The timing diagram in Figure 3.6 shows the processor's outputs relative to the processor's clock output. The figure shows the wait states inserted by the VSC and denoted by T_W. After activating ALE the TMS34061 responds by pulling RDY/$\overline{\text{HOLD}}$ low ready. This informs the processor that additional clock cycles (wait states) will be inserted in the current memory access.

It is not recommended that the processor ALE be used to drive the ALE signal required by the TMS34061, since there is not enough time to perform an address decode while using the processor's ALE signal. Another signal is thus to be used to indicate when a TMS34061 cycle is to take place. The $\overline{\text{MEMR}}$ and $\overline{\text{MEMW}}$ outputs work well; as they go active, they indicate that a cycle is taking place, and they leave enough time for address decode.

Figure 3.7 shows the processor/VSC interface. This diagram includes the address decoding circuit which is used to select the VSC chip. It is noted also that the

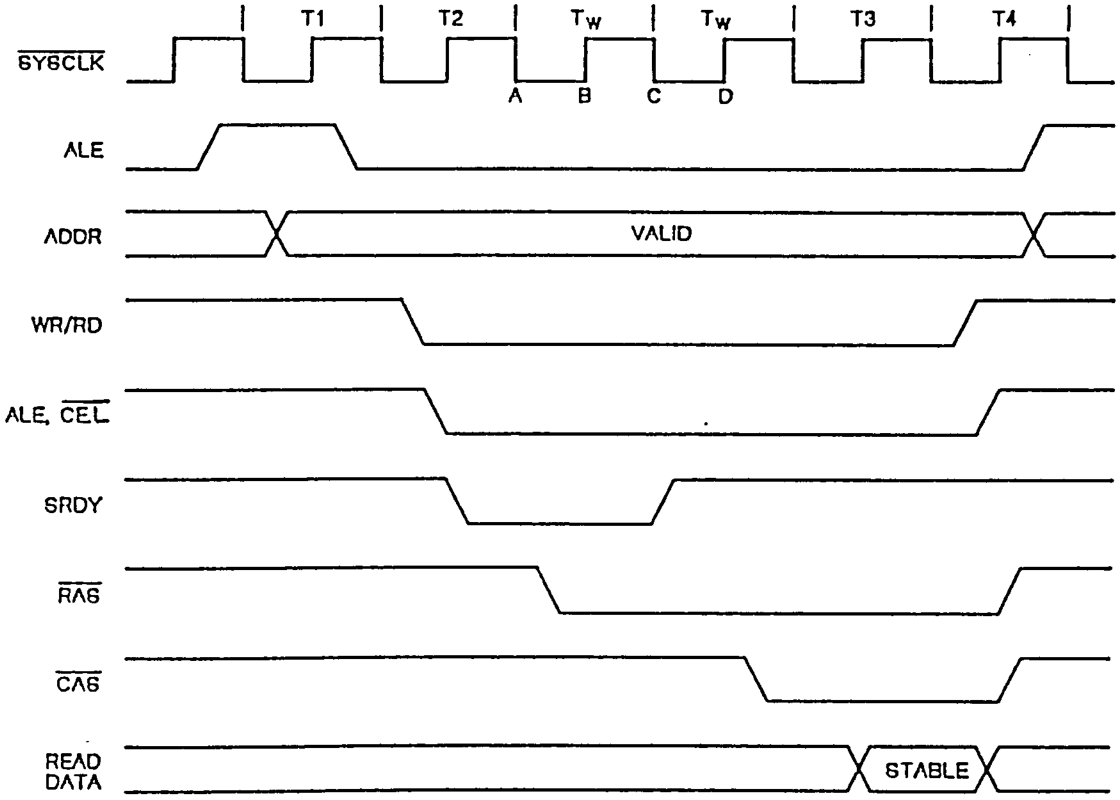


Figure 3.6: 8088 output timing

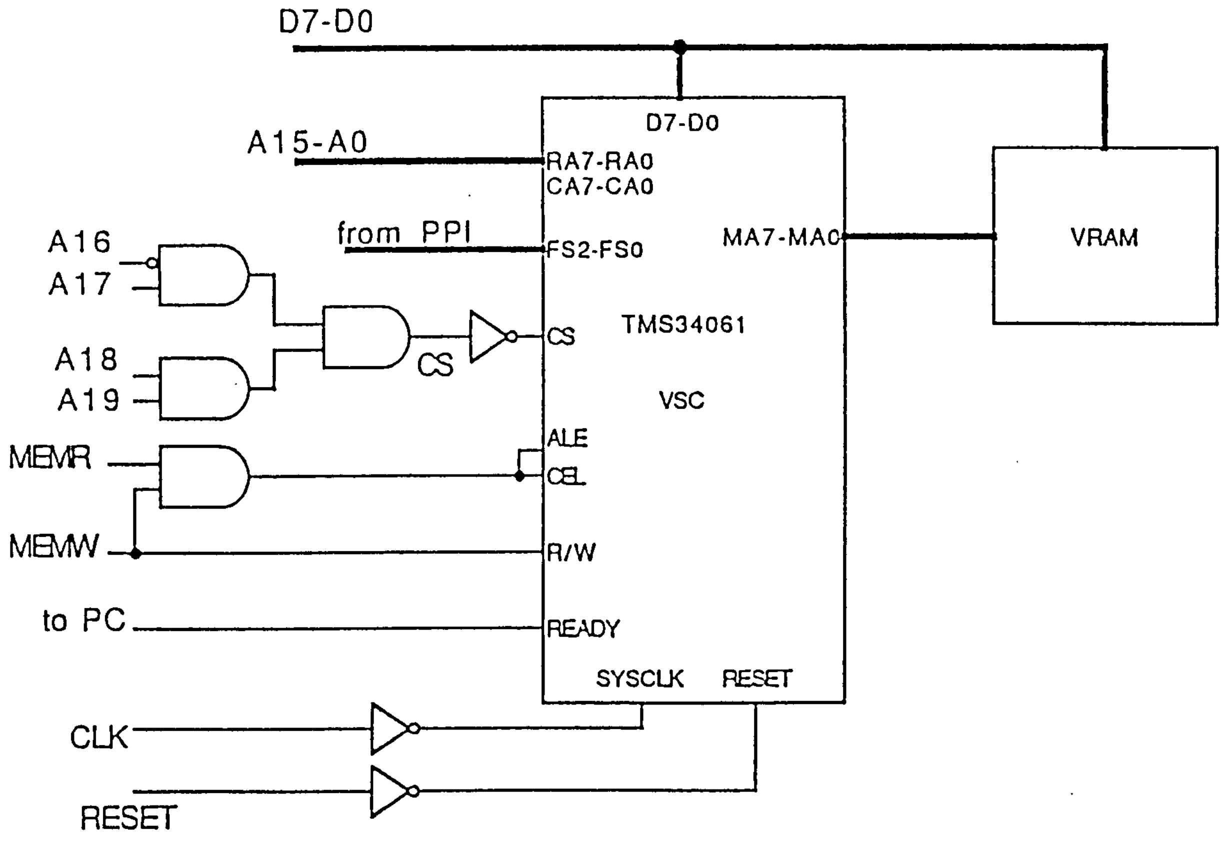


Figure 3.7: 8088/TMS34061 Interface

Function Select inputs to the TMS34061 (FS2-FS0) are controlled by software means through the PPI.

3.4 The Timing Circuit

For timing purposes, a 10 MHz crystal clock is used. This provides the maximum sampling frequency for the A/D converter. However, slower sampling frequencies are required for slower signals. To achieve this a counter/timer device is used.

The Intel 8254 counter/timer device, shown in Figure 3.8. is a software programmable chip that generates accurate time signals under software control. It is used in this design as a frequency divider of the basic 10 MHz sampling clock signal. as well as a trigger to control the number of data samples shifted into the SAM.

Operating the 8254 in mode 2 (one of eight modes of operation available) provides us with the frequency divider needed. When programmed in this mode. a counter functions as a divide-by-N counter. where N ranges between 1 and 10^7 for the purposes of this design, resulting in sampling frequencies that range between 10MHz and 1Hz. respectively.

Since 10^7 is too large a number for one counter to divide by (N_{max}=FFFFh), it takes two counters in cascade to generate the sampling signal, hence leading to the configuration shown in Figure 3.9 where the output of counter 1 is used as input to counter 0. The input to counter 1 is the 10 MHz clock signal and the sampling clock is available at the output of counter 0.

Counter 2 operates in the hardware retriggerable one-shot mode. Its output is initially low. When its GATE input becomes high, it begins a one-shot pulse, and remains low until its counter reaches zero. If the counter is initially set to 256,

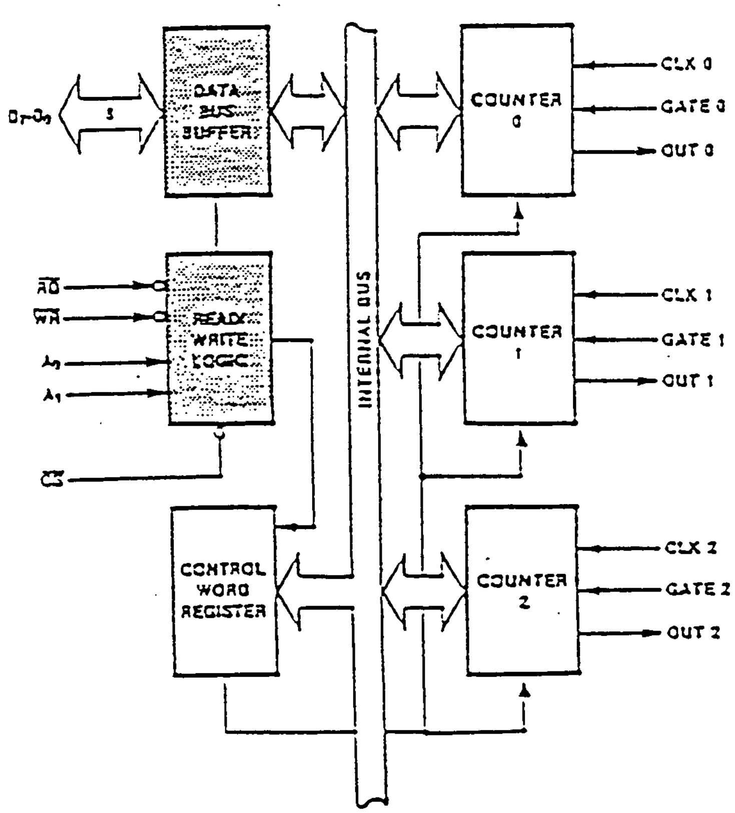


Figure 3.8: 8254 block diagram

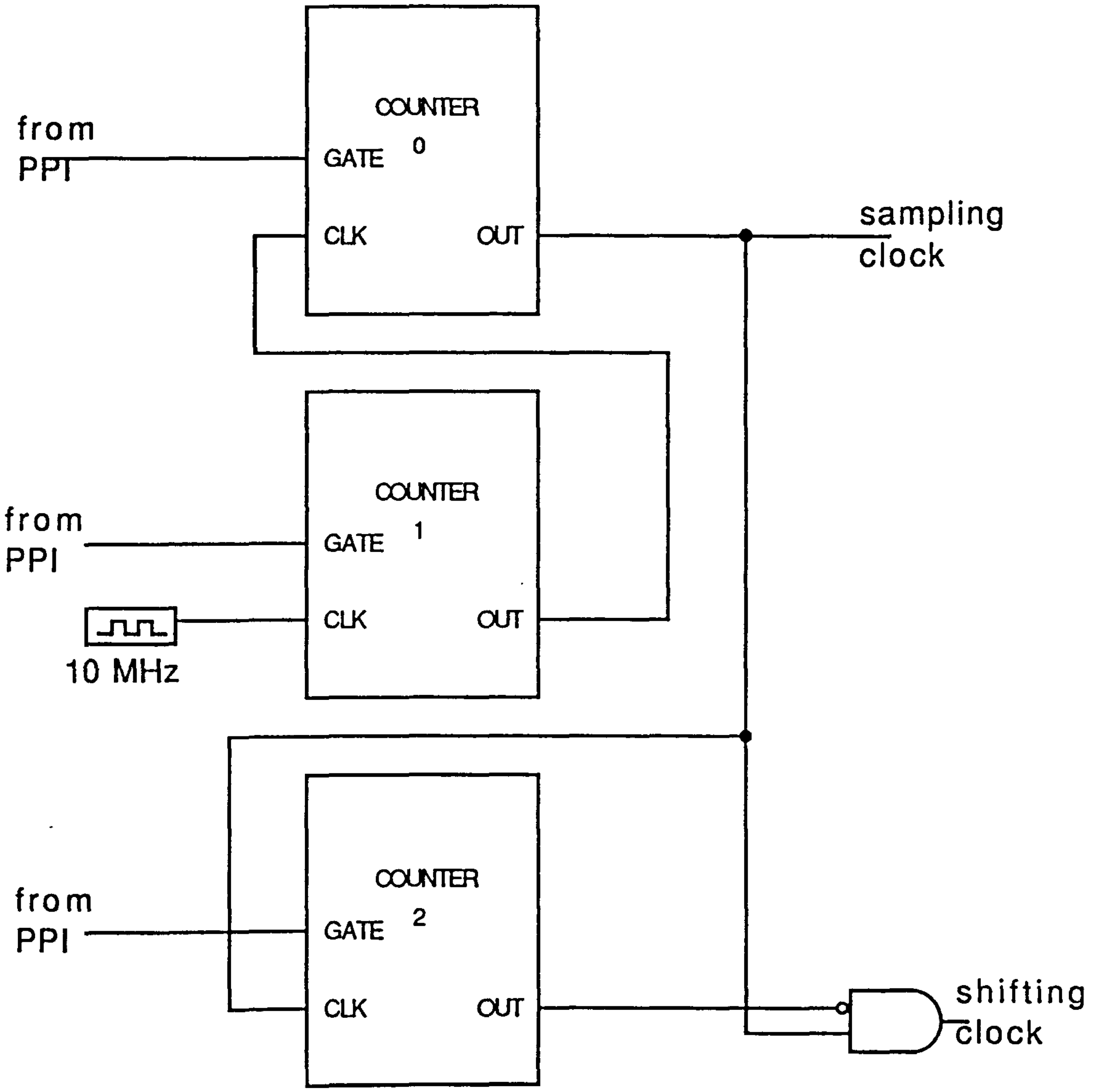


Figure 3.9: 8254 configuration

the one-shot pulse will have a duration of 256 sampling clock pulses (the output of counter 0).

To generate the shifting clock for the SAM, the output of counter 2 is complemented and ANDed with the sampling clock, resulting in 256 clock pulses that will allow the SAM to load exactly 256 data samples, i.e. a screen worth of data.

The 8254 interfaces to the 8088 microprocessor in the same manner as all other peripherals of the Intel Microcomputer Systems family. However, it was found convenient to use the extra pins on the PPI to access the 8254 chip, and thus minimizing the complexity of the control circuitry.

3.5 The Programmable Peripheral Interface chip

The Intel 8255 PPI is a general purpose programmable I/O device designed for use with Intel microprocessors. It has 24 I/O pins which may be individually programmed in 2 groups of 12 and used in three major modes of operation. Figure 3.10 shows the block diagram of the PPI.

In this design, the PPI is used to send three groups of control signals out to different parts of the circuit. These groups are:

- Function Select signals on the VSC (FS2-FS0). These three signals are controlled by software through pins B3-B5 on the PPI.

- Data Signals and Control Signals on the 8254 timer chip. These signals shown in Figure 3.8 are controlled through pins A0-A7 for data signals and pins C0-C7 for control signals.

- Input Circuit control signals. These signals are the multiplexers' select signals

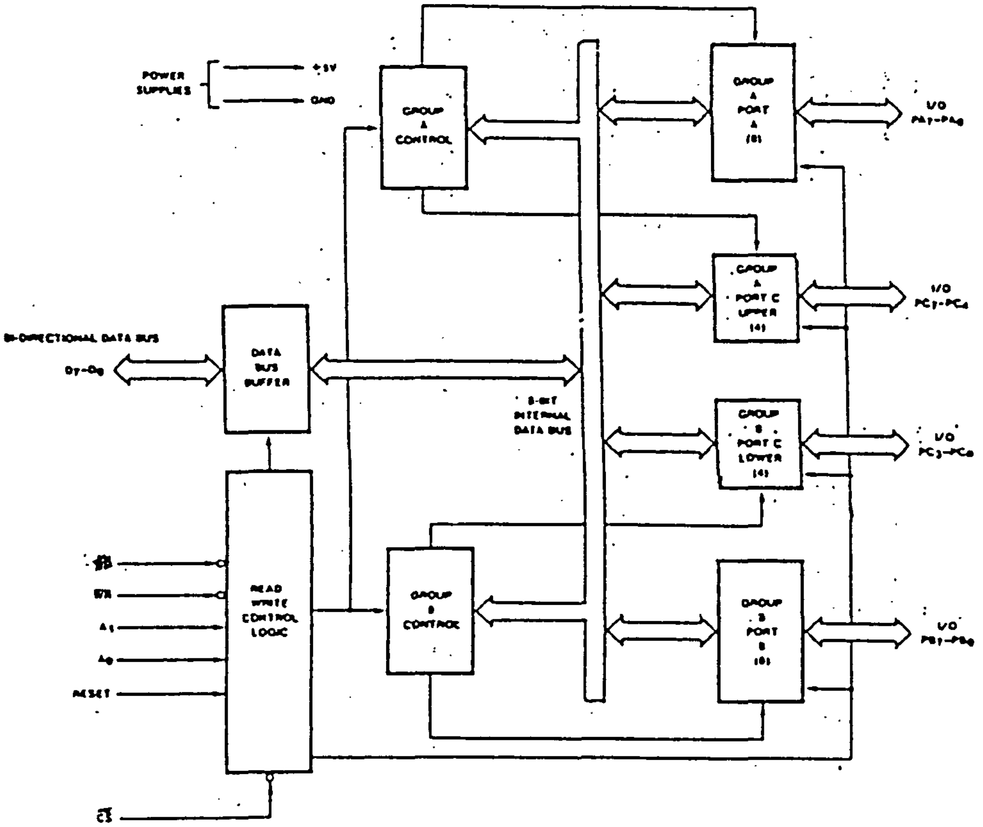


Figure 3.10: Block diagram of the Programmable Peripheral Interface chip

and are controlled through pins B0-B1 and B6-B7 on the PPI.

Needed for simple output operations, the PPI is operated in the first mode (MODE 0) where data is simply written to or read from a specified port. Once the mode select word is written to the control word register (Figure 3.10), the other three port registers can be accessible by the microprocessor.

Interfacing the PP.I to the microprocessor is simply done by mapping it into the Prototype card address range shown in Table 1.2. The address range chosen for the PPI is 30Ch-30Fh. Therefore, the PPI is addressed as four I/O port registers. This

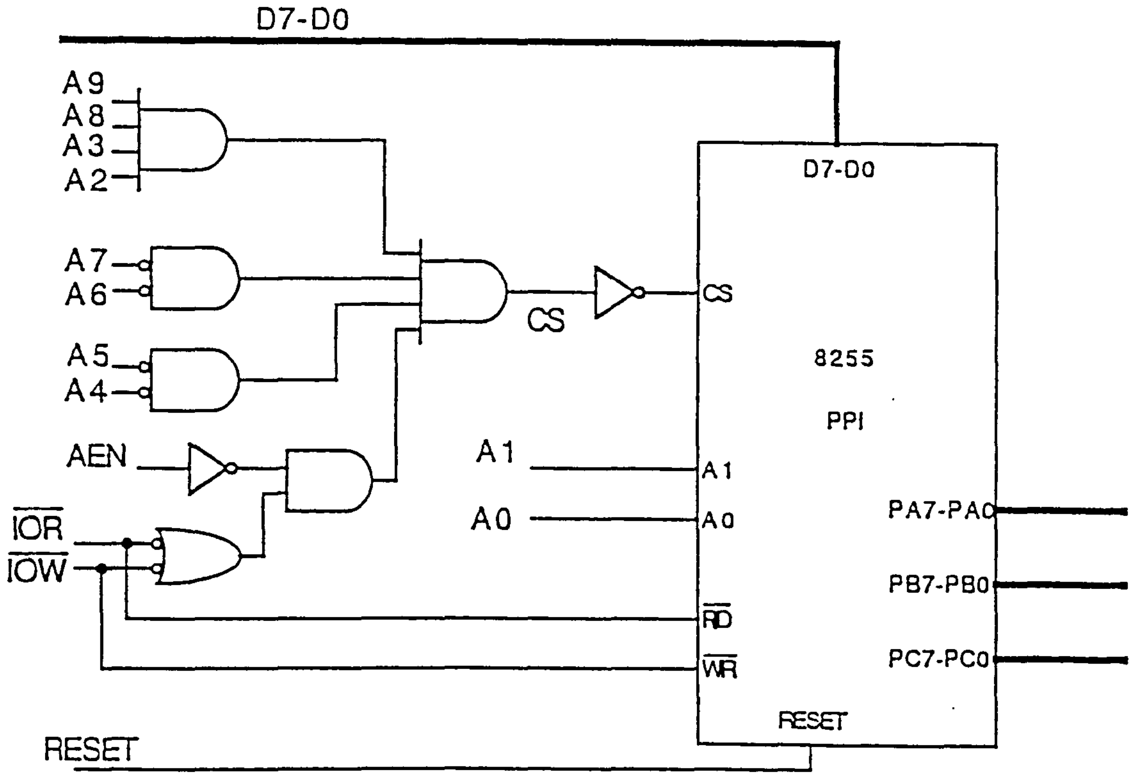


Figure 3.11: The PPI interface to the processor

is illustrated in Figure 3.11 which shows the signals needed to access the PPI as well as the address decoder circuit that enables the chip.

3.6 The Address Decoders and Buffers

To interface the system bus to the rest of the design, address decoders and buffers are essential. These circuits drive the interface circuits and separate them from the system bus, thus avoiding loading the system bus with too many signals.

Figure 3.12 shows the latches used to buffer signals between the processor and the PC board. Some of these signals is used in the memory address decoders and the I/O address decoders.

3.6.1 The buffers

Two types of chips are used for that purpose: The 74LS245 and the 74LS244. The former is an octal bidirectional tristate buffer with Schmitt-trigger inputs. It is used to buffer and drive the eight bidirectional data lines (D0-D7). The direction of data is controlled by pin 1 on the chip. This pin has to be driven low for data to flow from the board circuits to the system bus, hence it is driven by logically ANDing $\overline{\text{MEMR}}$ and $\overline{\text{IOR}}$.

In order to avoid overloading the PC data bus, the 74LS245 chip should only be enabled when the board circuits are accessed. The chip enable pin is thus driven by logically ANDing $\overline{\text{CS}}$ (chip select signal) of the memory controller and $\overline{\text{CS}}$ of the PPI.

The 74LS244 is an octal tristate buffer with Schmitt-trigger inputs. Four such chips are needed to buffer the address signals A0-A19, and control signals $\overline{\text{IOW}}$, $\overline{\text{MEMW}}$, $\overline{\text{IOR}}$, $\overline{\text{MEMR}}$, Reset and AEN. These are unidirectional signals.

3.6.2 The memory address decoders

As mentioned earlier, the 64k VRAM is mapped into address range E0000h-EFFFFh. Therefore the circuit shown in Figure 3.7 is used to decode address signals A16-A19 and generate the VSC chip select signal $\overline{\text{CS}}$. The rest of the address signals, A0-A15, are used to drive the memory inputs on the VSC.

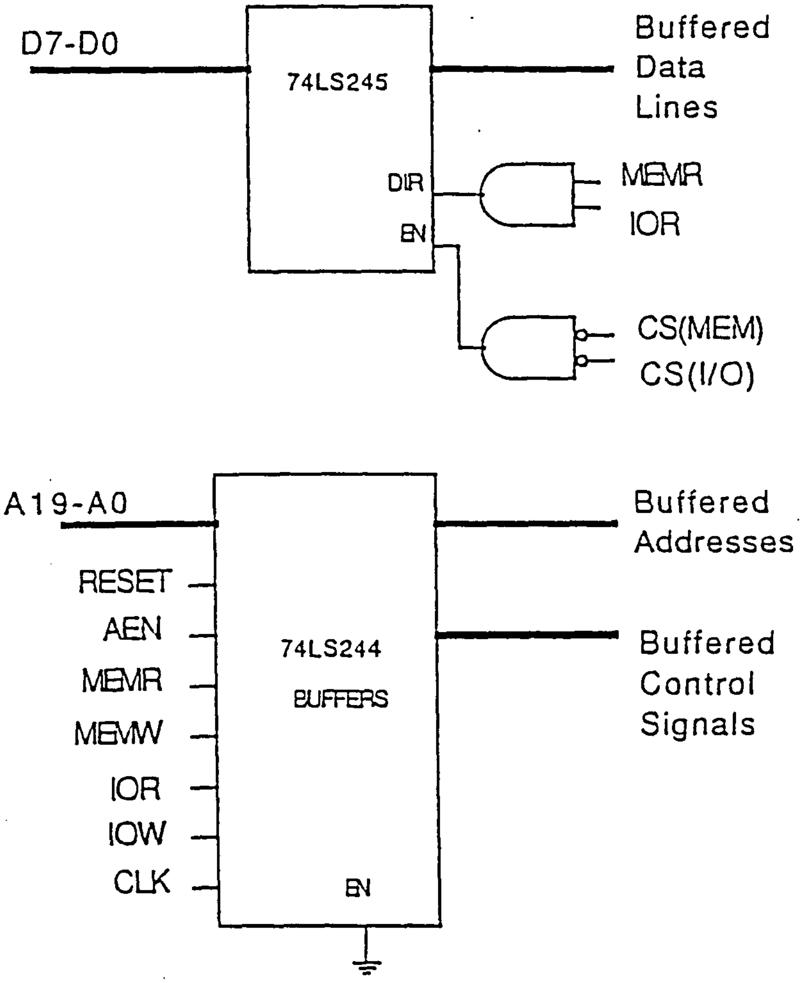


Figure 3.12: Buffers for data, address and control signals

3.6.3 The I/O address decoders

The PPI is the only I/O device in this design that is addressed directly by the microprocessor. It is mapped into I/O addresses 30Ch-30Fh. To implement this, address signals A9-A2 are decoded to generate the PPI chip select signal $\overline{\text{CS}}$. Address signals A1 and A0, on the other hand, are used to select one of four registers on the chip.

The decoding circuit in Figure 3.11 shows also that for $\overline{\text{CS}}$ to be low and the PPI enabled, either $\overline{\text{IOR}}$ or $\overline{\text{IOW}}$ should be low indicating an I/O read or I/O write cycle, respectively. AEN should also be low indicating that no DMA bus cycle is taking place.

CHAPTER 4. THE IMPLEMENTATION

This chapter pertains to the software aspects of the design. In order to generate the code necessary to display signals on the monitor, it is imperative to understand the steps involved in programming and/or sending control signals to the following segments of the design:

1. The input circuit.

2. The timing circuit.

3. The Video System Controller.

4. The Programmable Peripheral Interface chip.

Sending data and/or control signals to the first two segments is performed entirely through the PPI, which is accessed directly by the processor. The VSC, on the other hand, is programmed directly by the processor, except for the Function Select signals (FS2-FS0) which are controlled through the PPI to minimize the circuit complexity, as mentioned earlier.

4.1 The Input Circuit

Controlling the gain of the Input Circuit, and thus the vertical scale of the displayed signal, is carried out by multiplexing the input signal over different channels, as illustrated in the previous chapters.

Pins PB0-PB1 and PB6-PB7 on the PPI are used to control the select signals of the multiplexers. Each time the gain of the Input Circuit is to be changed, a new word is written to port B on the PPI.

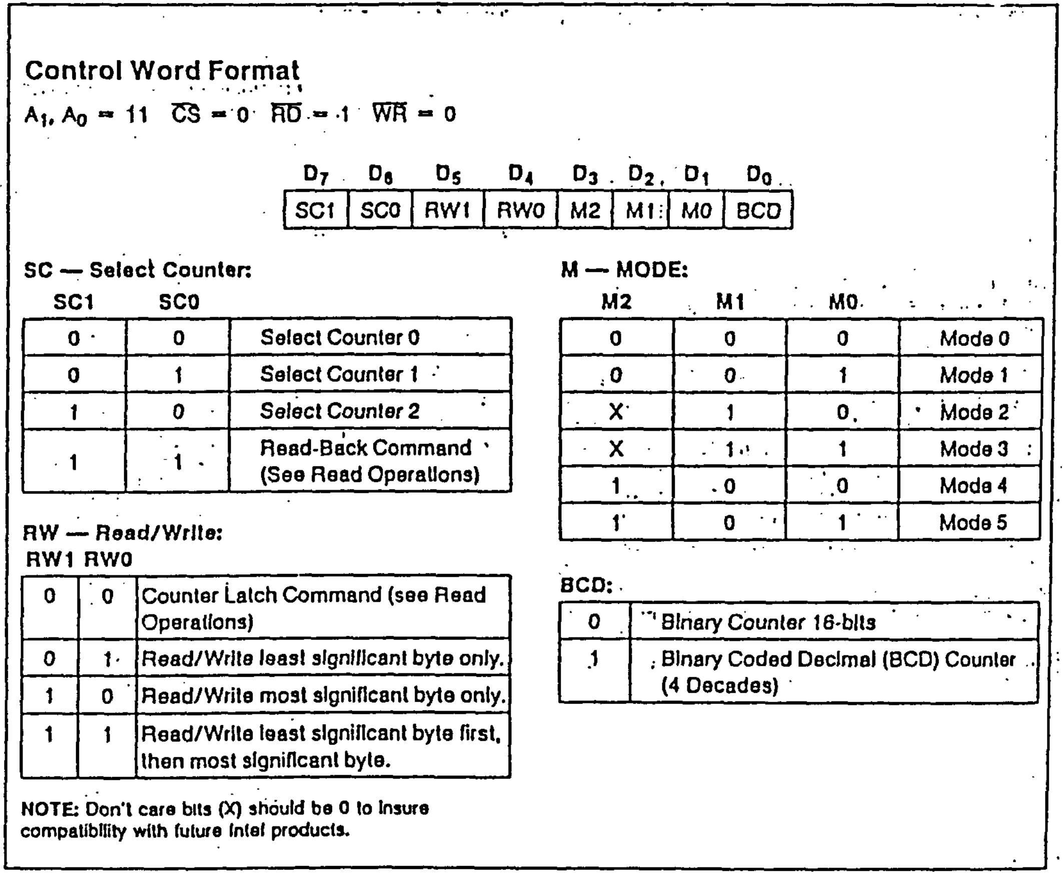
4.2 The Timing Circuit

Counters in the 8254 timer chip are programmed separately. For each of the three counters, a control word is written into the Control Word Register, which is selected when the chip address inputs A1,A0 = 11. The control word format is shown in Figure 4.1 and it specifies, among other parameters, which counter is being programmed.

Following each control word, the initial count is written into the corresponding counter. The A1,A0 address inputs are used to select the counter to be written into. The format of the initial count is determined by the control word used.

For the purposes of this design, two counters are used in mode 2, counter 0 and counter 1. To program the timer, a control word for counter 0 is written into the Control Word Register, followed by writing the value N that specifies the frequency dividing factor for counter 0. The same procedure is followed to program counter 1.

Counter 2 is operated in mode 1. After writing its corresponding control word into the Control Word Register, a value of 256 is written into it to specify the duration of the one-shot pulse.



Control Word Format

$A_1, A_0 = 11$ $\overline{CS} = 0$ $\overline{RD} = 1$ $\overline{WR} = 0$

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

SC — Select Counter:

SC1	SC0	
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Read-Back Command (See Read Operations)

M — MODE:

M2	M1	M0	
0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

RW — Read/Write:

RW1	RW0	
0	0	Counter Latch Command (see Read Operations)
0	1	Read/Write least significant byte only.
1	0	Read/Write most significant byte only.
1	1	Read/Write least significant byte first, then most significant byte.

BCD:

0	Binary Counter 16-bits
1	Binary Coded Decimal (BCD) Counter (4 Decades)

NOTE: Don't care bits (X) should be 0 to insure compatibility with future Intel products.

Figure 4.1: Control Word Format for the 8254 Timer

4.3 The Video System Controller

The TMS34061 Video System Controller contains 18 programmable registers. Each register is nominally 16 bits wide and can be written to and read from by the host processor through the 8-bit data path one byte at a time. Since this design makes use of only a subset of the functions the VSC can potentially perform, only two of these registers need to be accessed. These registers are:

1. Control Register 1. The functions assigned to the individual bits within this register are shown in Figure 4.2. Most of these functions are irrelevant to the modes of operation we are interested in. The bits which are of interest to us are bits B14-B12. The three-bit binary number in this field specifies the frequency of DRAM-refresh cycles.

2. Control Register 2. Figure 4.3 shows the functions assigned to the individual bits within this register. The bits that are of relevance to this implementation are:

   - B5-B2: RAS overrides.

     Each of the four RAS-override bits, B5-B2, is dedicated to one of the four RAS outputs, $\overline{\text{RAS}}3$ to $\overline{\text{RAS}}0$. When an override bit is set to one, the corresponding RAS output is forced to its active-low level during a host-direct write, a shift-register write or a shift-register read cycle.

     For this design, these bits are set to zero, which means that the corresponding RAS outputs are controlled by other means. Namely, during host-direct write cycles, RS1-RS0 input pins are used to select one of four row address strobes to be activated $\overline{\text{RAS}}3$ in this case.

B15	B14-B12	B11	B10	B9	B8	B7	B6	B5	B4	B3-B0
RESERVED 0	REFRESH BURST LENGTH	ERROR INTERRUPT ENABLE	VERTICAL INTERRUPT ENABLE	INTERLACE ENABLE	EXTERNAL SYNC ENABLE	DISPLAY UPDATE RAS MODE	DISPLAY UPDATE DIRECTION	DISPLAY UPDATE INHIBIT	RESERVED 0	LINE COUNT LIMIT

Figure 4.2: Control register 1

B15	B14	B13	B12-B11	B10-B8	B7	B6	B5-B2	B1-B0
RESERVED 0	RAS/CAS READY	BLANK ENTIRE DISPLAY	RDY/$\overline{\text{HOLD}}$ MODE SELECT	WAIT STATE LIMIT	X-Y ADDR. POINTER RAS MODE	EXTENDED RAS MODE	RAS OVERRIDES	EXTENDED RAS SELECT

Figure 4.3: Control register 2

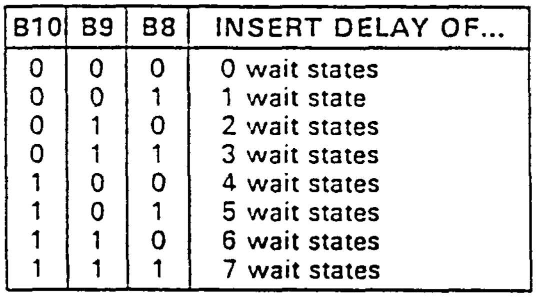
- B6: Extended RAS Mode.

  When B6 is zero, the values input on the RS1-RS0 pins select one of four RAS outputs during a host-direct or shift-register transfer cycle. When B6 is one, the two-bit code in B1-B0 selects one of the strobes.

  In this design, only one strobe is needed. Therefore B6 is set to zero and input pins RS1-RS0 are always set to 11 choosing $\overline{\text{RAS}}$3.

- B10-B8: Wait State Limit.

  The three-bit code in B10-B8 determines the number of wait states inserted into cycles initiated by the host processor. A wait state is invoked by postponing the ready signal to the host processor. B10-B8 are encoded as follows:

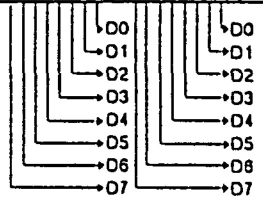


B10	B9	B8	INSERT DELAY OF...
0	0	0	0 wait states
0	0	1	1 wait state
0	1	0	2 wait states
0	1	1	3 wait states
1	0	0	4 wait states
1	0	1	5 wait states
1	1	0	6 wait states
1	1	1	7 wait states

- B12-B11: RDY Mode Select.

  The RDY output is configured by B12-B11 to operate as a "ready","wait" or "hold" signal to accommodate the interfacing requirements of a variety of host processors. For this design, B12-B11 are set to 00 to configure the RDY signal in the "ready" mode to meet the requirements of the 8088 processor.

INPUT PIN VALUES							REGISTER NAME
CA6	CA5	CA4	CA3	CA2	UPPER BYTE (CA1=1)	LOWER BYTE (CA1=0)	
0	0	0	0	0	XXXXAAAA	AAAAAAAA	HORIZONTAL END SYNC
0	0	0	0	1	XXXXAAAA	AAAAAAAA	HORIZONTAL END BLANK
0	0	0	1	0	XXXXAAAA	AAAAAAAA	HORIZONTAL START BLANK
0	0	0	1	1	XXXXAAAA	AAAAAAAA	HORIZONTAL TOTAL
0	0	1	0	0	XXXXAAAA	AAAAAAAA	VERTICAL END SYNC
0	0	1	0	1	XXXXAAAA	AAAAAAAA	VERTICAL END BLANK
0	0	1	1	0	XXXXAAAA	AAAAAAAA	VERTICAL START BLANK
0	0	1	1	1	XXXXAAAA	AAAAAAAA	VERTICAL TOTAL
0	1	0	0	0	XXXXXXXX	XXXXAAAA	DISPLAY UPDATE
0	1	0	0	1	XXXXAAAA	AAAAAAAA	DISPLAY START
0	1	0	1	0	XXXXAAAA	AAAAAAAA	VERTICAL INTERRUPT
0	1	0	1	1	XAAAAAAA	AAAXAAAA	CONTROL REGISTER 1
0	1	1	0	0	XXAAAAAA	AAAAAAAA	CONTROL REGISTER 2
0	1	1	0	1	XXXXXXXX	XXXXXAAA	STATUS REGISTER
0	1	1	1	0	XXXXAAAA	AAAAAAAA	X-Y OFFSET REGISTER
0	1	1	1	1	AAAAAAAA	AAAAAAAA	X-Y ADDRESS REGISTER
1	0	0	0	0	XXXXAAAA	AAAAAAAA	DISPLAY ADDRESS REGISTER
1	0	0	0	1	XXXXAAAA	AAAAAAAA	VERTICAL COUNT REGISTER



NOTE: "A" = ACTIVE REGISTER BIT, "X" = BIT NOT IMPLEMENTED.

Figure 4.4: Programmable register address map

The processor accesses the programmable registers within the VSC by means of special read and write cycles. A register-access cycle is selected by setting the Function-Select input pins FS2-FS0 (through the PPI) to one of two 3-bit codes, either 000 or 010. One of 18 registers is selected by the 5-bit register address input on column-address input pins CA6-CA2, as indicated in Figure 4.4. Binary codes 00000 through 10001 are valid register addresses. Codes 10010 through 11111 are reserved. The high or low byte of the register is selected by the value input on CA1. If CA1 is zero, the register low byte is selected; otherwise, the register high byte is selected.

The Function-Select input pins are addressed through the PPI, as mentioned earlier. These signals select the type of host-initiated access cycle to be performed by the TMS34061, according to the following table:

FS2	FS1	FS0	CYCLE
0	0	0	Register-access
0	0	1	X-Y-indirect
0	1	0	Register-access
0	1	1	Host-direct access
1	0	0	Shift-register, SR to memory
1	0	1	Shift-register, memory to SR
1	1	0	Reserved
1	1	1	Reserved

The DRAM refresh cycle is performed automatically by the VSC.

4.4 The Programmable Peripheral Interface chip

Programming the PPI is simply done by writing a control word into the Control Word Register. This control word specifies the mode of operation and thus the configuration of the three 8-bit ports on the 8255 chip. Figure 4.5 shows the Mode Definition Format for the Control Word in the PPI.

This PC board design requires that the PPI operates in MODE 0, with all ports configured as output ports. The control word that is necessary to realize this configuration is 80h. The way these ports are used is as follows:

- PA0-PA7: Data lines to the 8254 timer chip.
- PB0-PB1, PB6-PB7: Control signals to the Input Circuit.
- PB3-PB5: Function-Select inputs to the TMS34061 VSC.

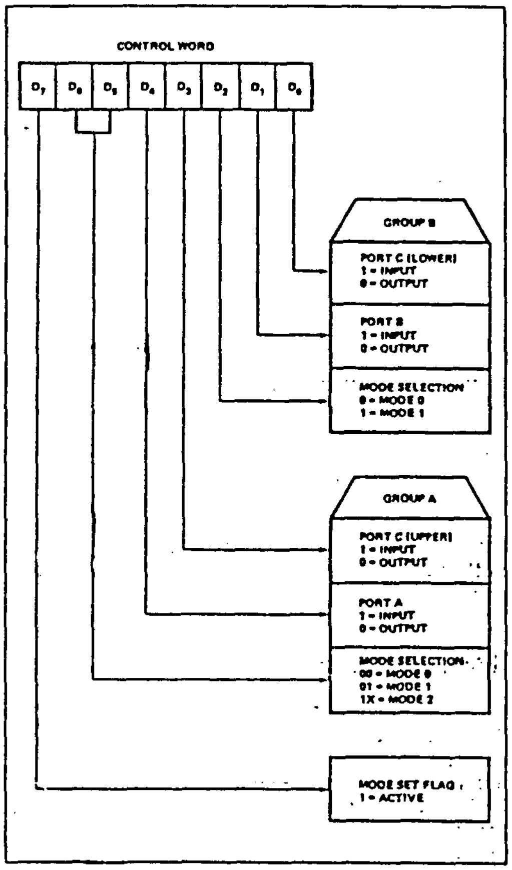


Figure 4.5: Control word register mode definition format

- PB3-PB5: Function-Select inputs to the TMS34061 VSC.

- PC0-PC7: Control signals to the 8254 timer chip.

The code necessary to display a signal on the PC monitor is listed in the appendix. The software initializes the PPI, the VSC and the 8254 timer. It then transfers data from the SAM to the DRAM and displays it on the screen.

CHAPTER 5. RESULTS AND CONCLUSION

Using the design discussed here, signals were digitized and displayed on the PC monitor thus simulating the functions of an oscilloscope. The displayed signal was to a great extent a replica of the input signal (using an oscilloscope as a reference), with no visible distortion.

Special emphasis is placed on interfacing the VRAM to the PC and using it to collect data samples. This stems from the potential to use this scheme in other data acquisition systems where it is required to collect data and load it into memory without interrupting the data collecting process.

The design also involved interfacing other circuitry to the PC: the PPI and the timer. The PPI was interfaced directly to the processor while the timer was unconventionally accessed through the PPI. Although this meant less hardware complexity, it lengthened the software code and thus the execution time.

The code needed to operate the PC oscilloscope is included in the appendix. Since the design components are all highly programmable, the program code can be easily modified to fit various applications.

BIBLIOGRAPHY

[1] *IBM PC Technical Reference.* Boca Raton, FL: IBM Corporation. 1983.

[2] Jourdain, R. *Programmer's Problem Solver for the IBM PC, XT & AT.* New York, NY: A Brady Book Published by Prentice Hall Press, 1986.

[3] Novogrodsky, S., Davis, Frederic E. and the Editors of PC World. *The Complete IBM Personal Computer: The Authoritative Guide to Hardware for Expanding the IBM PC, XT, AT, and Compatibles.* New York, NY: PC World Communications, Inc., 1985.

[4] Rollins, Dan. *IBM-PC 8088 Macro Assembler Programming.* New York, NY: Macmillan Publishing Company, 1985.

[5] Somerson, Paul. *PC Magazine Power Tools: Techniques, Tricks and Utilities.* A Bantam Book by Paul Somerson and Ziff Communications Company, 1988.

[6] *TMS34061 User's Guide: Video System Controller.* Texas Instruments Incorporated, 1986.

[7] Uffenbeck, John. *Microcomputers and Microprocessors: The 8080, 8085, and Z-80 Programming, Interfacing, and Troubleshooting.* Englewood Cliffs, New Jersey: Prentice Hall, Inc., 1985.

[8] Woen, Ting-Woen. *Interfacing a speed control processor to the IBM PC.* Ames, Iowa: Iowa State University, 1988.

ACKNOWLEDGEMENT

I wish to thank my major professor, Dr. A. Pohm, for giving me the opportunity to do my thesis. I also thank Dr. R. Horton and Dr. G. Prabhu for serving on my committee and Mr. G. Bridges for his help in setting up the equipment for my project. Special thanks to Mr. Kyle Holland for his valuable technical counsel.

Most of all, I wish to express my gratitude to my parents. Salwa and Munther Masri, for their patience and continuous support.

APPENDIX The Program Listing

```
;-- This program initializes the PPI and the timer.

CODEX  segment ;Program Code Segment
            assume cs:CODEX

START:          ;Program Entry Point

        MOV AL,80H
        MOV DX,30FH
          OUT DX,AL      ; initialize the PPI

        MOV AL,0DFH
        MOV DX,30EH
          OUT DX,AL      ; address for control register

        MOV AL,15H
        MOV DX,30CH
          OUT DX,AL      ; control word for counter 0

        MOV AL,5FH
        MOV DX,30EH
          OUT DX,AL      ; address for control register

        MOV AL,0DFH
        MOV DX,30EH
          OUT DX,AL      ; address for control register

        MOV AL,10H
        MOV DX,30CH
        OUT DX,AL     ; LSB
```

```
MOV AL,5CH
MOV DX,30EH
 OUT DX,AL      ; address for counter 0

MOV AL,0DCH
MOV DX,30EH
 OUT DX,AL      ; address for counter 0


 MOV  AL,75H
MOV DX,30CH
  OUT DX,AL      ; control word for counter 1

 MOV  AL,5FH
MOV DX,30EH
  OUT DX,AL      ; address for control register

 MOV  AL,0DFH
MOV DX,30EH
  OUT DX,AL      ; address for control register

 MOV AL,00H
MOV DX,30CH
 OUT DX,AL     ; LSB

 MOV AL,5DH
MOV DX,30EH
 OUT DX,AL      ; address for counter 1

MOV AL,0DDH
MOV DX,30EH
 OUT DX,AL      ; address for counter 1

 MOV AL,01H
MOV DX,30CH
OUT DX,AL     ; MSB
```

```
        MOV AL,5DH
        MOV DX,30EH
         OUT DX,AL       ; address for counter 1

        MOV AL,0DDH
        MOV DX,30EH
         OUT DX,AL       ; address for counter 1

         MOV  AH,4CH          ; return to DOS
         INT  21H


CODEX   ends
          end    START
```

```
; written by Wassef Masri
;  3-15-91
; this program initializes the VSC

CODEX  segment ;Program Code Segment
         assume cs:CODEX

START:          ;Program Entry Point

     MOV AL,0H
     MOV DX,30CH
      OUT DX,AL          ;FS2,FS1,FS0 = 000 (access control registers)

     MOV  AX,0E000H
     MOV  ES,AX           ;base address

      MOV AL,010H           ;Upper byte of Ctrl Register 1
     MOV ES:[02E20H],AL

      MOV AL,0A0H          ;Lower byte
     MOV ES:[02C00H],AL

      MOV AL,02H            ;Upper byte of Ctrl Register 2
      MOV ES:[03200H],AL

      MOV AL,03H           ;Lower byte
      MOV ES:[03000H],AL

      MOV  AL,ES:[03000H]

      MOV  AH,14            ;display the contents of AL
      INT  10H

      MOV  AH,4CH          ;back to DOS
      INT  21H

CODEX   ends
          end    START
```

```
;-- This program transfers data from SR to the VRAM and display it on the
screen

CODEX  segment ;Program Code Segment
         assume cs:CODEX

START:          ;Program Entry Point

      MOV AL,18H
      MOV DX,30DH
      OUT DX,AL           ; FS2,FS1,FS0 = 011

      MOV  AX,0E000H
      MOV  ES,AX              ; base address

      MOV  AL,20H             ; FS2,FS1,FS0 = 100
      MOV  DX,30DH
      OUT  DX,AL              ; SR to ROW transfer

       MOV  AL,ES:[0000H]  ; transfer

      MOV  AH,0
      MOV  AL,0FH
       INT 10H                ; set screen in graphics mode

      MOV  AX,0E000H
      MOV  ES,AX              ; base address

      MOV BX,256
      MOV CX,50
      MOV DI,0

 LUP: MOV  AH,0CH
      MOV AL,3
      MOV DL,ES:[DI]
      MOV DH,0
      ADD DX,70
       INT 10H
```

```
        INC DI
        INC  CX
       DEC  BX
        JNZ  LUP

        MOV  AH,4CH            ; return to DOS
        INT  21H

CODEX    ends
            end     START
```