SIGNAL PROCESSING ON SATELLITE COMMUNICATION

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of
Dokuz Eylul University
In Partial Fulfilment of the Requirements for
the Master's degree in Electrical and Electronics
Engineering

by
Altay Tan GENÇAY

Supervised by
Prof. Dr. Kemal ÖZMEHMET

July, 1996
İZMİR

M.Sc. THESIS EXAMINATION RESULT FORM

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as thesis for the degree of Master of Science.

Prof. Dr. Kemal ÖZMEHMET
(Advisor)

....
(Committee Member)

....
(Committee Member)

Approved for the
Graduate School of Natural and Applied Sciences

Mtoksoy

Prof.Dr.Macit Toksoy
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

My deepest thanks go to my thesis advisor, Prof. Dr. Kemal ÖZMEHMET, for his guidance and support. I would like to thank Muzaffer ÖZER for providing technical support.

My thanks also to my mother and father for giving me every opportunity to learn, and for being there all this time, and also I would like to thank to my friend İbrahim AVCI.

Altay Tan GENÇAY

ABSTRACT

SIGNAL PROCESSING
ON SATELLITE COMMUNICATION

By

Altay Tan GENÇAY

Master of Science, July 1996

(B. Sc., Dokuz Eylul University, 1991)

Supervised by Prof. Dr. Kemal ÖZMEHMET

In this thesis, it is planned to show the TV signals on a standard PC monitor. At the same time required technical equipment to receive signals from satellite is set. A unit is built to quantise, to store, to process and to restore the signals and drive the PC monitor.

Firstly, a software solution for our purpose was examined. During this study C++, Pascal and Assembler for 80x86 based microprocessors software languages were used, but it's seen that only software solutions do not solve the problem. Then, addition to software hardware solutions were introduced. TV to PC interpreter circuit was designed and realised. The introduced circuit run properly.

ÖZET

Bu tezde, TV sinyalerinin PC ekranında gösterilmesi planlanmıştır. Aynı zamanda sinyaleri elde etmek için gerekli teknik donanım kurulmuştur. Sinyalleri örneklemek, saklamak, işlemek ve tekrar geri elde etmek ve PC monitörünü sürmek için gerekli birim kurulmuştur.

Öncelikle program yazılımları ile çözüm aranmıştır. Bu çalışmada C++, Pascal ve 80x86 mikroişlemci tabanlı Assembler bilgisayar dilleri kullanılmıştır, ancak sadece bilgisayar programı ile çözüm bulunamamıştır. İlave olarak kurulan kartlar ile çözüm geliştirilmiştir. Bunun için TV'den PC'ye sinyal çevrimini sağlayan devre kurulmuştur. Kurulan devre düzgün şekilde çalıştırılmıştır.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACRONIMS

AC	Alternative Current
APA	All Points Addressable
ASCII	American Standard Code for Information Interchange
CPU	Cental Processing Unit
DMA	Direct Memory Access
DRAM	Dynamic RAM
Frame	One motionless screen slice
Hsync	Horizontal Synchronisation signal
I/O Device	Input Output Device
IC	Integrated Circuit
Interrupt	All the assembler programs have different kind of interrupts, these are software interrupts.
mc	megacycle (MHz)
Pixel	One colored dot on graphics screen
RAM	Read Only Memory
RGB	Red, Green, Blue
SHL	Shift Left Operation
sync	Synchronisation signal
Vestigial side-band	Any modulation method
VGA	High color resolution graphics adapter
Vsync	Vertical Synchronisation signal

CHAPTER 1
INTRODUCTION

Digital image processing deals with the enhancement, statistical analysis, gray-scale or colour-image data. This data is generated by sensing of real world objects, which can be either scanned x-ray images, images or broadcasting TV channels of a satellite.

Image processing functions operate on the individual pixel (or data elements) of the image data array. A set of drawing commands and attributes generates the image data. Image processing functions tend to fall into two categories: point and neighbourhood operations. Point operators perform on individual image-pixel data, while neighbourhood operators work on connected groups of pixel.

A digital image processing system is usually composed of two parts, namely an input part and a processing part. The input part receives images from different sources to the image processing system. Input sources include video digitisation, optical scanners, and other digital sensor outputs. The processing part assumes that image data resides within a frame buffer. A frame buffer is typically coupled with a video display processor [5] .

Business employees commute to and from offices, spend countless wasted hours on airplanes and hotel rooms, doing business face-to-face [4] .

In this project, it is aimed to show real-time images on a PC screen. For this purpose, a card is designed, built and tested. No microprocessor is used during the installation of the card. Only buffers, analog multiplexers, RAMs, counters, D-type flip-flops, inverters, high speed ADC and DAC ICs are used to realise the card.

Most of the cards on the market have their own CPU or microcontroller based unit inside. It is necessary if the card will work with the interactive commands of users. Users can change the size of the window on the PC screen at any time by the commands of the Windows Software. The software program for the card must learn the new size of the window and send the related commands to the card. When the card receives the signals, it must change the timing of the image to fit the new window. In this case the image can be stretched to any axis.

The main rule of stretching is to store the digitised data to a RAM, and then to restore the same data from the RAM at a different frequency. The frequency rate between the writing frequency and reading frequency of the RAM gives the stretching ratio. Stretching can be achieved by making the reading frequency higher then the writing frequency. Depending on this rule, a card was designed to achieve the stretching process easily.

The stretching ratio depends on the difference between the timing diagrams of TV signals and PC video card signals. When TV signal is compared with the PC monitor signal it is seen the necessity of a new interface unit, which will be called TV card in this thesis. This card consists of two parts, namely digital and analog part. The complex digital circuit was built to achieve the stretching. This circuit receives the analog signal in, and transmits the stretched analog signals out. The stretching ratio is 0.5. However, the stretching algorithm can be called as "compress twice". This compression process can be satisfied by writing to a RAM and reading from the RAM at a different reading and writing frequency. Therefore, this process is called as "write slow, read fast algorithm".

Software programs given in Chapter two were written to estimate the system performance. This software programs were written in three different software languages; Pascal, C, and Assembler. Additionally, some memory access methods and structures of accessing to the memory are also given in this Chapter.

The technical information about PC and TV timing diagrams, and some signal shapes for different bar images are given and the main elements of VGA video cards are represented in Chapter three.

The instrumentation of the TV card is given in Chapter four. Problems of this circuit are also considered and a new circuit design is proposed in this chapter, as well.

The conclusions are given in Chapter five.

CHAPTER 2
SOFTWARE SOLUTIONS

2.1 PIXEL AND GRAPHICS MODES

Pixel is a dot on a PC graphics screen. There are two main modes for a PC screen: text and graphics modes. There are also other modes for text and graphics screens. All the text modes will not be discussed here as they are somehow similar to the graphics modes. The main text mode is 80x25 text mode for VGA cards. There are more than 30 different graphics resolution modes for some special VGA Video Cards and monitors.

The modes listed above are main modes for EGA and VGA Video Cards. Some other special modes can be listed as 640x480 16 colours, 640x480 256 colours, 800x600 16 colours, 800x600 256 colours, 1024x768 16 colours and 1024x768 256 colours. In other words, the resolution and the range of colours are related to the capacity of video card.

2.2 'PUT-PIXEL' PROGRAM

A colour pixel can be put at a specified position on the screen by any software program. The software program can be written in any computer language, such as C, Pascal, and Assembler. Pascal and C programs run in the same manner. First, some variables are defined, then the program detects the graphics screen to initialise. After initialisation, the graphics screen is checked if any error occurred. Later, a pixel is put on the coordinate of 100,100 and the software waits until the user presses any key. Finally, it closes the graphics screen and returns to the text mode.

Table 2.1 BIOS Video Modes [1]

Mode (Hex)	Type	Colours	Alpha Format	Buffer Start	Box Size	Max Pgs.	Vert. Freq.	PELS
0,1	A/N	16 / 256K	40 x 25	B8000	8 x 8	8	70 Hz	320 x 200
2,3	A/N	16 / 256K	80 x 25	B8000	8 x 8	8	70 Hz	640 x 200
0*,1*	A/N	16 / 256K	40 x 25	B8000	8 x 14	8	70 Hz	320 x 350
2*,3*	A/N	16 / 256K	80 x 25	B8000	8 x 14	8	70 Hz	640 x 350
0+,1+	A/N	16 / 256K	40 x 25	B8000	9 x 16	8	70 Hz	360 x 400
2+,3+	A/N	16 / 256K	80 x 25	B8000	9 x 16	8	70 Hz	720 x 400
4,5	APA	4 / 256K	40 x 25	B8000	8 x 8	1	70 Hz	320 x 200
6	APA	2 / 256K	80 x 25	B8000	8 x 8	1	70 Hz	640 x 200
7	A/N	-	80 x 25	B0000	9 x 14	8	70 Hz	720 x 350
7+	A/N	-	80 x 25	B0000	9 x 16	8	70 Hz	720 x 400
D	A/N	16 / 256K	40 x 25	A0000	8 x 8	8	70 Hz	320 x 200
E	APA	16 / 256K	80 x 25	A0000	8 x 8	4	70 Hz	640 x 200
F	APA	-	80 x 25	A0000	8 x 14	2	70 Hz	640 x 350
10	APA	16 / 256K	80 x 25	A0000	8 x 14	2	70 Hz	640 x 350
11	APA	2 / 256K	80 x 25	A0000	8 x 16	1	60 Hz	640 x 480
12	APA	16 / 256K	80 x 25	A0000	8 x 16	1	60 Hz	640 x 480
13	**APA**	**256 / 256K**	**40 x 25**	**A0000**	**8 x 8**	**1**	**70 Hz**	**320 x 200**

* Enhanced modes from the IBM Enhanced Graphics Adapter.

+ Enhanced Modes.

Mode 13 hex is 320x200 256 colour mode. All VGA cards support this mode.

2.2.1 Pascal and C Programs

```
{ Example for PutPixel }
Uses CRT, Graph;
Var
    Gd, Gm, Err    : Integer;
```

```
   Colour              : Word;
Begin
   Gd := Detect;                    {auto-detect the graphics driver and mode }
   InitGraph(Gd ,Gm , '  ');
   Err := GraphResult;              { check for any errors }
   If Err <> grOk Then Begin
      WriteLn('Graphics Error ', grapherrormsg( Err ) );
      Halt( 1 );
   End;

   PutPixel( 100 , 100 , Red );     { put a pixel on the screen}
   Repeat Until KeyPressed;         { pause screen }
   CloseGraph;
End.
```

```
/* C Program */
#include <graphics.h>
#include <conio.h>
#include <stdio.h>
#include <process.h>
int main(void)
{
   /* auto-detect the graphics driver and mode */
   int gdriver=DETECT, gmode, errorcode;
   initgraph(&gdriver, &gmode, " ");
   errorcode = graphresult( );              /* check for any errors */
   if (errorcode != grOk)
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      exit(1);
```

```
    }
    putpixel( 100, 100, RED );             /* put a pixel on the
    screen */
    getch( );                              /* pause screen */
    closegraph( );
    return 0;
}
```

2.2.2 Assembler

The same program can be written in Assembler which is expected to be the fastest since it is the machine language and does not require compiling and interpreting. Assembler programs are easily understood even if you do not know much about them, because all the structure is similar to high level computer language programs. Only difference is that Assembler has a few simple move and interrupt commands. The whole process can be accomplished by only writing to some registers and invoking interrupts.

The following assembler program puts a pixel on the PC screen in 320x200 resolution and 256 colours mode. Like the other two, Assembler code puts only one pixel on the coordinate of 100,100.

```
; Assembler Program that puts a red dot on graphics screen that resolution
; is 320x200 by using DOS and BIOS interrupts

mov ah,0            ; function number
mov al,13h          ; mode selector, 320x200x256
int 10h             ; BIOS Interruptϕv
                    ; graphics mode is initialised
mov ah,0ch          ; function number
mov al,4            ; colour of dot
mov dx,100          ; row
mov cx,100          ; column
int 10h             ; BIOS Interrupt
                    ; A colored dot is put

mov ah,07h          ; function number
int 21h             ; DOS Interrupt
                    ; getch( ) No Echo

mov ah,0            ; function number
mov al,3h           ; 80x25 16 colour text mode
int 10h             ; BIOS Interrupt
mov ah,4ch          ; function number
int 21h             ; DOS Interrupt
                    ; Terminate program
```

2.3 MEMORY ACCESS ALGORITHM

Memory Access Method is the fastest way to put a pixel on the screen. The main purpose is to write the data directly into the specified address of the video screen. Both text and graphics screens have a starting address in the memory. The address table is given in Table 2.1. Only ASCII *(American Standard Code for Information Interchange)* characters can be written on the text screen. ASCII data is normally used to represent alphanumeric characters in the memory of a computer system [2]. If the video card and monitor are VGA type, the starting address of text mode is B8000 for 80x25 colored text mode.

2.3.1 20 Bit Addressing Method

The address bus width of a video card is 20 bits, but all IBM compatible PCs have 16-bit address bus structure. For this reason, the 20-bit address bus is divided into segment and offset parts. Segment addresses are shifted 4 bits left and added with the offset address to find the absolute address value. In this case, B8000 address can be written as B800h segment, and 0000 offset. At the same time, B7FFh segment, and 0010h offset point out the same absolute address. (B7FFh SHL 4) + 0010h = B8000h. As it is seen, the same memory cell can be reached from different segment and offset addresses. The offset address is set to zero to make accessing easy. B8000h is the starting address of the 80x25 16 colour text mode display. The memory block has a sequential form, which means that each additional memory address points the sequential boxes on the text screen mode. B800:0000 address is the character address of the 0, 0 coordinate of the screen. The 0,0 coordinate is the top left corner of the screen. B800:0001 address is the colour of the character and background. This form is called the attribute of the character. B800:0002 is the second character box on the screen, B800:0003 is the attribute of the same second box, and so on. There is 80x25 (2000) bytes of characters on the text screen, and 2000 bytes of attributes, as well. There are totally 4000 bytes for the text screen. While all even numbered addresses show the characters, odd numbered addresses show the attributes. All information written above is valid for the graphics modes. The only differences between the text and graphics modes are the starting addresses and attributes. For example, the starting address is A0000 for 320x200 256 colours mode and there is no attribute values for graphics modes. If a byte is written to the address A0000, the written byte is accepted as the colour code. A0000 is the top left corner of the screen. A0001 is the second pixel on the graphics screen, and so on. A0000 address can be separated as A000 segment and 0000 offset easily.

2.3.2 Pascal and C Programs and Memory Organization

When a byte, such as 65, is written to B800:0000 memory address, character "A" will be shown up on the top left corner of the screen. If another byte is written to B800:0001 memory address, the colour and background of the character "A" will be changed. If the byte, 65, is written to A000:0000 memory address a colour pixel will be shown up on the top left corner of the graphics screen. But in order to realise this operation, the screen mode must be initialized to the graphics mode. If a different screen mode has to be used, the user manual of the VGA card should be read in order to determine the code numbers of the high resolution modes, because every VGA card has its own codes for the same resolution. Additionally, there is a memory bank problem for high resolution modes. The segment and offset addresses are 16 bit numbers. Thus, the maximum range of the address is 65535 (64kbyte) in high resolution mode. This is the bus structure problem of IBM Compatible Personal Computers. They have segment and offset addresses to point a cell in a memory. If the resolution is 320x200, there are 64000 pixels on the graphics screen. The colour range of each pixel is 256. In this case, only one memory bank can hold that memory area. The number of pixels will increase for the high resolution modes. Therefore, one memory bank will not be sufficient for this amount of memory. The memory bank has to be changed by software for direct addressing. The bank registers of the card have to be known to change the memory bank. For the problem proposed in this report, it would be better to use 320x200 resolution.

```
{Pascal Program writes a byte to memory}
Begin
        Mem[ $B800 : 0000 ] := 65;
        Mem[ $B800 : 0001 ] := 20;
End.
{ 65 is "A" character on ASCII table }
{20 sets the colour of that character }
{ to red on blue background }
```

```
/* Same program in C */
#include <dos.h>
#include <conio.h>
int main( void )
{
    poke( 0xB800, 0000, 65 );
    poke( 0xB800, 0001, 20);
    getch( );
    return 0;
}
```

2.3.3 Assembler

The assembler code program below puts a pixel at any location on the screen in 320x200 256 colours mode.

```
;Assembler Program that puts a red dot on graphics screen
;where the  resolution is 320x200
;by writing data to video memory

mov ah,0          ; function number
mov al,13h        ; mode selector, 320x200x256
int 10h           ; graphics mode is initialized BIOS Interrupt
mov dx,0A000h  ; Graphics Mode Segment A000
mov es,dx         ; 0,0 is upper left corner
mov al,4          ; Colour of dot
; ( Row - 1 ) * 320 + Column - 1 = ( 100 - 1 ) * 320 + 100 - 1 = 31799
mov di,31799      ; 31799 decimal is equal to 7C37 hex.
es:
mov [di],al       ; write a byte to memory which is located A000:7C37
mov ah,07h     ; function number
int 21h           ; getch( ) No Echo, DOS Interrupt
mov ah,0          ; function number
mov al,3h         ; 80x25 16 colour text mode
int 10h           ; BIOS Interrupt
                  ; Terminate program
mov ah,4ch        ; function number
int 21h           ; DOS Interrupt
```

2.3.4 Estimating the Speed of the Memory Access

Assembler routine given below puts 64000 bytes in to the memory in mode 320x200 by only changing the location of the pixel. Because the colour range is 256, all the pictures can be seen perfectly. The assembler routine paints a frame readily and additional frames can be put without any delay. In continuous form all the pictures can be shown consequently.

The program also estimates the speed of an assembler program for an IBM Compatible PC, Intel 80486 CPU, 4 Mbytes RAM, 256 Byte Cache Memory.

```
; Assembler Program that fills the graphics screen writing data to video memory.
; the resolution is 320x200.
mov ah,0            ; function number
mov al,13h          ; mode selector, 320x200x256
int 10h             ; BIOS Interrupt
                    ; graphics mode is initialized

mov ah,9            ; function number
mov dx,offset data  ; offset value of data
int 21h             ; writes a string to screen

mov ah,07h          ; function number
int 21h             ; DOS Interrupt
                    ; getch( ) No Echo
mov cx,1000         ; Number of frames

mov dx,0A000h       ; Graphics Mode Segment A000
mov es,dx           ; 0,0 is upper left corner
mov dx,64000        ; 320x200

repeat
mov al,cl           ; Colour of dot
```

```
xor di,di          ; set di to zero
go     es:         ;
mov [di],al        ; write a byte to memory which is located in A000:DI
inc di
cmp di,dx          ; compare with di and 64000
jne go             ; repeats 64000 times to fill a frame
loop repeat        ; decrement CX and if not equal to zero jump to the label repeat
mov ah,07h         ; function number
int 21h            ; DOS Interrupt
                   ; getch( ) No Echo

mov ah,0           ; function number
mov al,3h          ; 80x25 16 colour text mode
int 10h            ; BIOS Interrupt

mov ah,4ch         ; function number
int 21h            ; DOS Interrupt
                   ; Terminate program
data  db     'Press any key to fill different colored frames',
             'press any key again to terminate program$'
                   ; the last character of the string must be "$" for the function 9
                   ; of int 21h (  print string )
```

The program paints the screen 1000 different colours. The number of cycle is written into the register of CX. When the program runs, the speed of frames per second is found to be 15.2. In fact, this speed is fast enough for human eye, because human eye can not recognise the transitions between the frames. Number of frames is 25 on cinema and TV screens. But there is an important problem in that it is assumed that all the frames are stored in the memory. But if they are read from the external devices, there will be some delay while reading from the port. In this case, the speed of the software will reduce, but there is no other way to get external TV signals without using an internal card. So, a card has to be installed inside the PC, which receives the TV signals and changes them in to the signal format of a PC monitor.

2.3.5 Estimating the Speed of the I/O Device.

Another assembler program is used to estimate the speed of the I/O device. The speed is found to be 8.33 frames per second by using the following Assembler code. The program reads the I/O device continuously and counts the number of bytes coming in. Then, the program takes the average of them and finds the number of frames per second.

```
; Assembler Program that reads data from I/O device
mov ah,9                ; function number
mov dx,offset data      ; offset value of data
int 21h                 ; writes a string to screen

mov ah,07h              ; function number
int 21h                 ; DOS Interrupt
                        ; getch( ) No Echo
mov cx,1000             ; number of frame will be read
mov bx,64000            ; number of byte for one frame
repeat
xor di,di               ; reset counter
again
mov dx,300h             ; 300h I/O device number
in al,dx                ; result in AL register
inc di                  ; add counter by 1
cmp di,bx               ; compare weather one frame is completed or not
jne loo                 ; if not go to label again

loop repeat             ; repeat CX times
                        ; CX * BX = 1000 * 64000 = 64.000.000 data will be read
                        ; Terminate program
mov ah,4ch              ; function number
int 21h                 ; DOS Interrupt
data  db 'Press any key to read data from I/O port$'
```

2.3.6 Estimating the Speed of the Combined Program

I/O and memory access speed estimation programs can be brought together to read the data from I/O device and write the data directly into the video memory. An assembler program is written to combine both programs. This program is run and the speed of it is estimated to be 7.2 frames per second which means that only 7.2 frames can be shown in one second while reading the data from the I/O device and writing that data to the video memory. But there is also another problem. The device connected to the I/O device must sample one frame very fast and store the data in its own memory. Then, it has to transfer the data to the I/O device.

```
; Assembler Program that fills the graphics screen that resolution
; is 320x200 by writing data to video memory while getting data from I/O port

mov ah,0              ; function number
mov al,13h            ; mode selector, 320x200x256
int 10h               ; BIOS Interrupt
                      ; graphics mode is initialized

mov ah,9              ; function number
mov dx,offset data    ; offset value of data
int 21h               ; writes a string to screen

mov ah,07h            ; function number
int 21h               ; DOS Interrupt
```

```
                 ; getch() No Echo
mov cx,500

mov dx,0A000h    ; Graphics Mode Segment A000
mov es,dx        ; 0,0 is upper left corner
mov bx,64000     ; 320x200
repeat
xor di,di        ; set di to zero
go
mov dx,300h      ; 300h I/O device number
in al,dx         ; result in AL register
es:              ; Colour of dot
mov [di],cl      ; write a byte to memory which is located A000:7C37
inc di
cmp di,bx        ; compare with di and 64000
jne go           ; repeats 64000 times to fill a frame
loop repeat

mov ah,0         ; function number
mov al,3h        ; 80x25 16 colour text mode
int 10h          ; BIOS Interrupt

                 ; Terminate program
mov ah,4ch       ; function number
int 21h          ; DOS Interrupt

data  db     ‘Press any key to fill different colored frames,’,
             ‘press any key again to terminate program$’
```

2.4 DIRECT MEMORY ACCESS (DMA)

Direct memory access normally occurs between an I/O device and memory without using the microprocessor. A DMA reads data from the memory to the I/O device and writes the data from I/O device into memory. In both operations, the memory and I/O device are controlled simultaneously and that is why the system contains separate memory and I/O control signals. This special control bus structure of the microprocessor allows DMA transfers [2].

The DMA I/O technique provides direct access to the memory while the microprocessor is temporarily disabled. This allows the data to be transferred between memory and the I/O device at a rate that is limited only by the speed of the memory components in the system or the DMA controller. The DMA transfer speed can approach 10-12 Mbytes transfer rates with today's high-speed RAM memory components. DMA transfers are used for many purposes, but more common are DRAM refresh, video displays for refreshing the screen, and disk memory system reads and writes. The DMA transfer is also used to make high-speed memory-to-memory transfers [2].

The 8237 DMA controller supplies the memory and I/O with control signals and memory address information during the DMA transfer. The 8237 is a special purpose microprocessor whose function is high-speed data transfer between memory and the I/O.

DMA programs do not work for the purpose of this project because of the RGB (Red, Green, Blue) table problem. A picture can have maximum 256 different colours on a PC screen with a VGA card. That range of colours is enough for a good image quality. Commercial software programs shows the pictures on the PC screens by reading and setting the RGB palette of the picture. All picture file formats have their own compression methods and they have to store the RGB palette table inside the picture. Each picture has its own palette table. But TV channels have 25 frames per second with different colour pallets and none of the TV standards keep the colour palette for the VGA cards. Thus, if DMA runs properly, RGB palette problem occurs. To solve this problem an RGB True Colour PC card and a monitor have to be used. This kind of PC cards and monitors do not need any colour table. The range of colours on these screens is 256x256x256. But they are not very common. Therefore, true colour pictures have to be simulated on the VGA cards and a new external card has to be designed to decode the TV signals and to prepare them for the PC VGA monitors.

2.5 Real-time Image Mixers

There are so many different video cards on the market. These kind of cards receive TV signals in and process them with a programmable high speed video processor, and drives the PC monitor. After receiving the TV signals by antenna cable, the card amplifies, decodes and separates the components of RGB. Then they take the output of the video card of the PC and overwrites the RGB values on the original PC signals. The card mixes both signals and generates a new image for the PC monitor. The mixing method is described as follows.

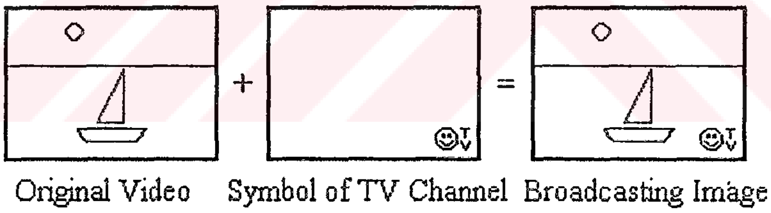


Figure 2.1 Mixing with the logo and original TV frame.

Each TV channel has its own sign or figure as a logo and puts it on each frame of TV. The original video signal and the logo is overwritten on each image of TV as seen in Figure 2.1. The insertion process is almost the same for the video insertion process of a PC. Original digitised signals are sent to the video card of the PC. The card converts the signals to analog signals, synchronises the pictures, and drives the monitor. This process can be done with a software program, but in this case the speed of the displayed frames will not be high enough. In order to increase the number of frames, an external card has to be build.

This card should grab the original video signals and also receive the original TV signals. It has to mix these two signals properly, as well (Figure 2.2).

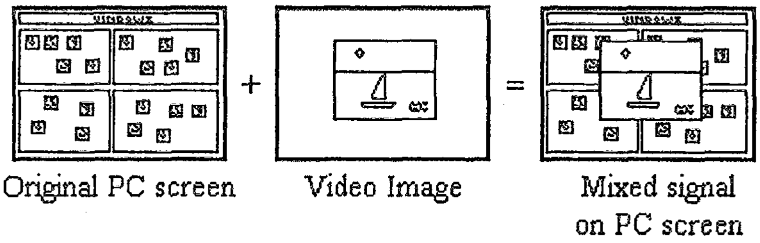


Figure 2.2 Mixing the TV signals and video signals.

CHAPTER 3
TECHNICAL INFORMATION ABOUT TV and PC SCANNING ALGORITHM

3.1 SCANNING TV AND PC SCREENS

The electron tube starts to gun at the top left corner of the screen to the right side of the same line and goes to the second line, when the second line is completed, it goes to the third line, etc. This process continues until the last line of the screen. After sending the last electron to the bottom right corner of the screen, the electron gun goes again to the top left corner of the screen. This process is valid for both TV and PC monitors. When the electrons reach the right side of the screen for each line, Hsync signal is set to the logic level low until the electron gun starts to scan a new line. While the screen is scanned, Hsync signals are always in logic level high. After scanning all the lines, Hsync drops to the logic level low and Vsync signal is set to the logic level low by the video card. The Vsync signal shows the end of the frame. The Vsync signal stays at low until scanning starts again from the top left corner of the screen. During the scanning process, RGB signals may have any analog voltage level. The voltage level is related to the power of the

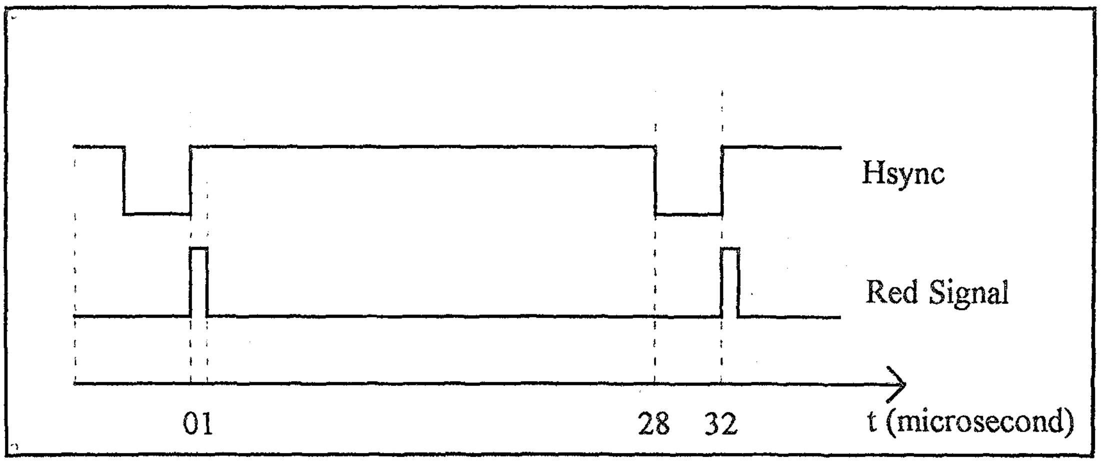


Figure 3.1 Timing diagram of a PC video card and RGB signals

pixel colour. As far as the red signal is concerned, if the voltage level of the red signal is zero, it means that there is no red component of the pixel put on the screen. On the other hand, if the voltage level of the red signal is maximum, the red component of this pixel is in the highest range. If the red signal has the maximum voltage level and the other RGB signals have the minimum voltage level at the same time, only the red signal will be displayed on the screen. The blue and red signals can be mixed by increasing the voltage level of the blue signal. The same rule is valid for the green and blue signals. Black colour may be obtained by combining all red, green and blue signals.

One picture line of a PC monitor is scanned by electron gun in 32 microseconds. Scanning time includes the fly back time. Visual part of the screen is scanned in 28 microseconds. In this case, if the video card sets all the RGB components to the analog level zero, the screen will be black. If the PC video card sets the red signal to the analog voltage level of 2 volts during 1 microsecond of the scanning time of each line, about 3% of the PC screen will become red as seen in Figure 3.1. Approximately, 10 pixels become red in 320x200 resolution mode. To create a colour bar on the PC screen, an on-off switch can be used as follows. Original signal coming from the video card is fed to the monitor during the on period of the switch. On the off period of the switch, either no signal or another signal is applied to the monitor. This process is repeated continuously. If no signal is applied while the switch is off, a black bar is shown on the screen. Otherwise, the applied signal is shown on the screen as a bar. This bar can be either a visible image or a scrambled image depending on the form of the signal.

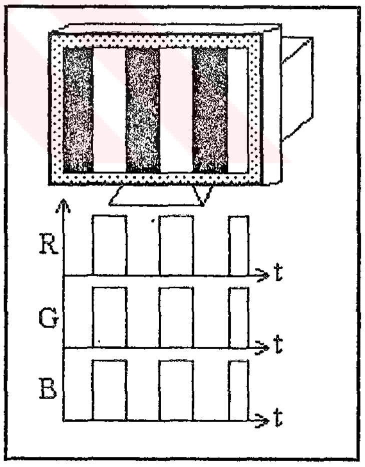


Figure 3.2 Timing Diagrams of PC monitor

The location of the colour pixel on the graphics screen depends on the voltage level of RGB signals in the time domain. A test can be made to see the different tones of the red colour. First, the connection between the PC monitor and video card is removed and the Hsync and Vsync signals are short-circuited. Then, the Hsync signal is applied to both the trigger input of a signal generator and the monitor. Later, a triangle wave in the voltage level of between 0 and 3 volts is fed to the red pin of the monitor. After all these connections, the different tones of the red colour is seen on the PC screen.

A program is used to paint the screen black & white as shown in Figure 3.2. Then it is seen on the oscilloscope screen that each RGB signal has the same constant voltage level, because the image contains only two colours. If the image is gray scaled, a different voltage signal shape will be seen for RGB signals and each of them will be the same, because of the gray scale. If the image is a colour one, each RGB signal will have its own separate voltage signal shape. Given timing diagrams of each RGB value are valid for each line on the screen. This means that RGB values repeat themselves at every 32 microseconds. Figure 3.3 shows another example of timing and signal diagram of the PC monitor.

Most of the TV cards can calculate the window size on the screen, and change the timing signals. In order to change the window size of the TV picture on the PC screen, a powerful and fast external processor is needed. They also send the control signals to the adjustable switches. All video cards should be able to calculate the timing diagrams and mix both pictures coming from the PC and TV. The signals coming from the TV have to be tuned, decoded, and converted to the digital form. Then, they can be mixed with the PC video signals. The speed of the TV cards should be noted. They have to run at high frequencies, and have to produce the necessary control signals on time.

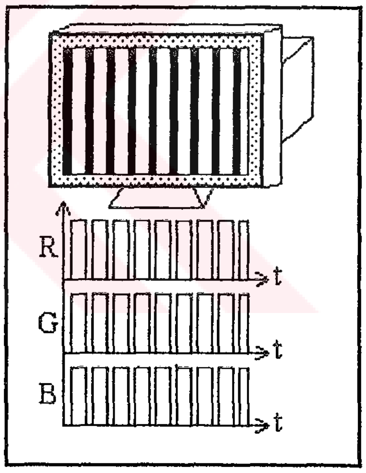


Figure 3.3 Timing Diagrams of PC monitor

A TV card is build to show the TV channels in real-time on the PC screen. It puts 25 frames per second on the screen. All the timing diagrams of the TV and video card are evaluated, and a different digital circuit is built.

3.2 TV SIGNAL CHARACTERISTICS

TV channels are received without any distortion with a good TV set and an antenna system. It is better to know the structure of the antenna signals to have a good understanding of how the TV signals are composed. The antenna signal is called as RF signal. All TV channels have their own channel numbers and frequencies during broadcasting. A TV tuner is an adjustable device and filters the channels. Only the filtered channel signal can be sensed at the output of the tuner.

3.2.1 Structure of a TV Channel

The picture signal is transmitted by amplitude modulation. In conventional amplitude modulation, two sideband frequencies are generated for each modulating frequency. These sidebands appear symmetrically above and below the carrier frequency. Since the ineffective usage of the frequency band, vestigial sideband system is used in TV broadcasting. TV channel frequency spectrum is given in Figure 3.4.

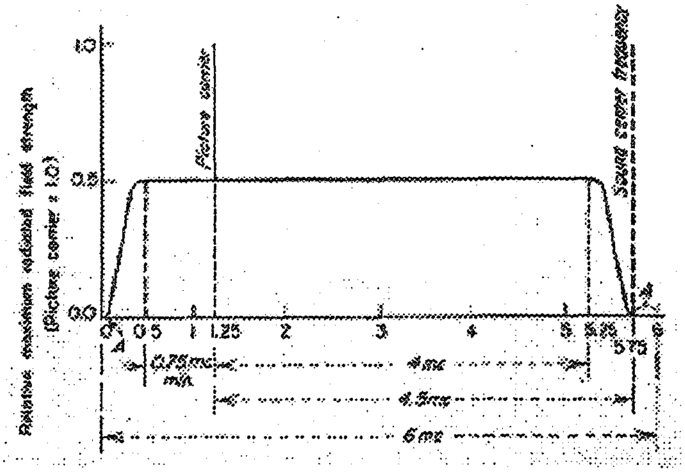


Figure 3.4 The standard TV channel in USA.
The regions marked "A" must exhibit 20 dB attenuation of the picture sidebands [7].

3.2.2 Scanning

The method used in practice in general is known as "uniform linear scanning". The word "scanning" arises from its similarity to the action of a reader scanning a page of printed type. The eye begins at the upper left hand corner of the page and travels along the first line of the type until it reaches the right hand edge of the page. There the eye quickly reverses its motion and returns to the beginning of the next line, where it resumes its slower left-to-right motion, travelling to the end of the line, then back, and so on.

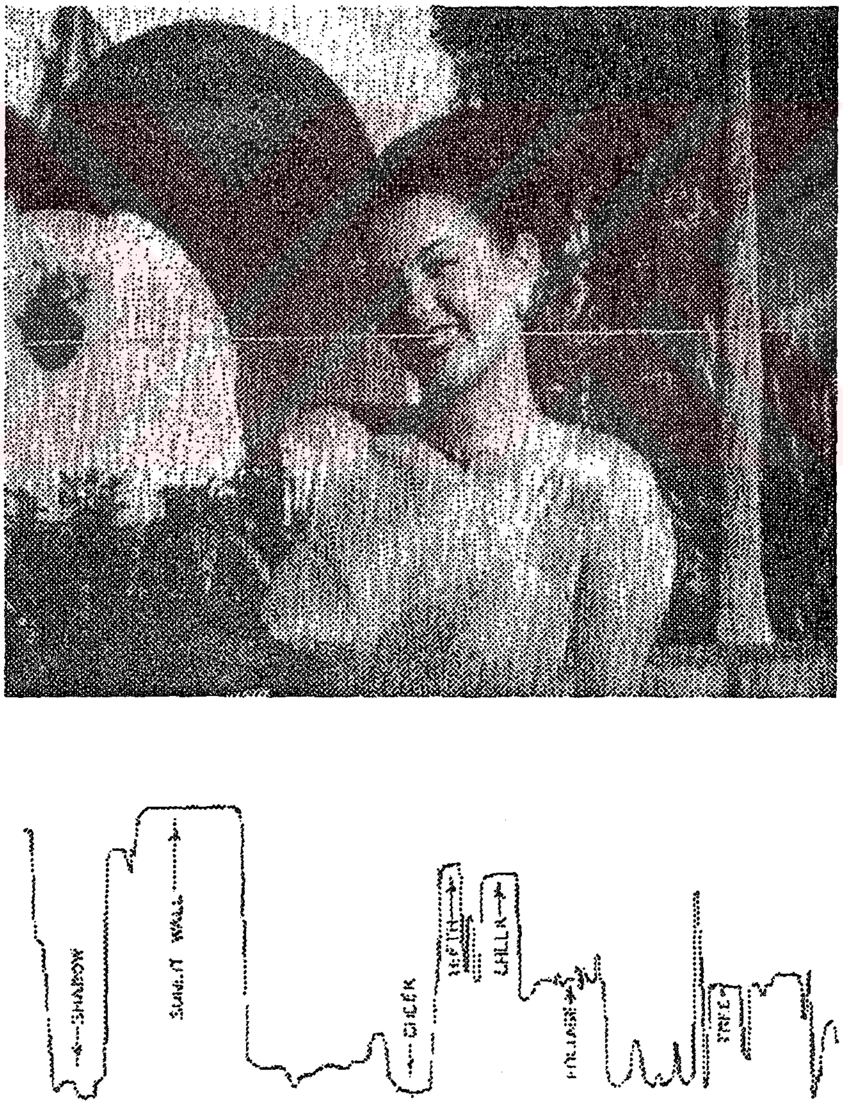


Figure 3.5 The scanning technique [7].

A very similar method is employed in TV scanning. The picture elements, actually distributed uniformly over the picture area, are considered to lie in horizontal rows, much as the letters of print are arranged on a page. When the transmission of the picture begins, all the picture elements present in the top most row are selected, one after the other from left to right, and converted into corresponding electric signals which are sent successively over the communication channel. When the first row has been scanned, the picture elements in the third row are selected in the same fashion, followed by the elements in the fifth row, and so on, until the bottom row of the picture is reached. The voltage level of an image is given in Figure 3.5. The horizontal white line in the middle of the image represents the voltage diagram.

The standard scanning pattern consist of a total of 625 lines, divided into two interlaced groups of 312 1/2 lines each. The scanning 625 lines takes place in 1/25 second, which means 25 complete pictures are transmitted each second.

3.3 VGA VIDEO CARD

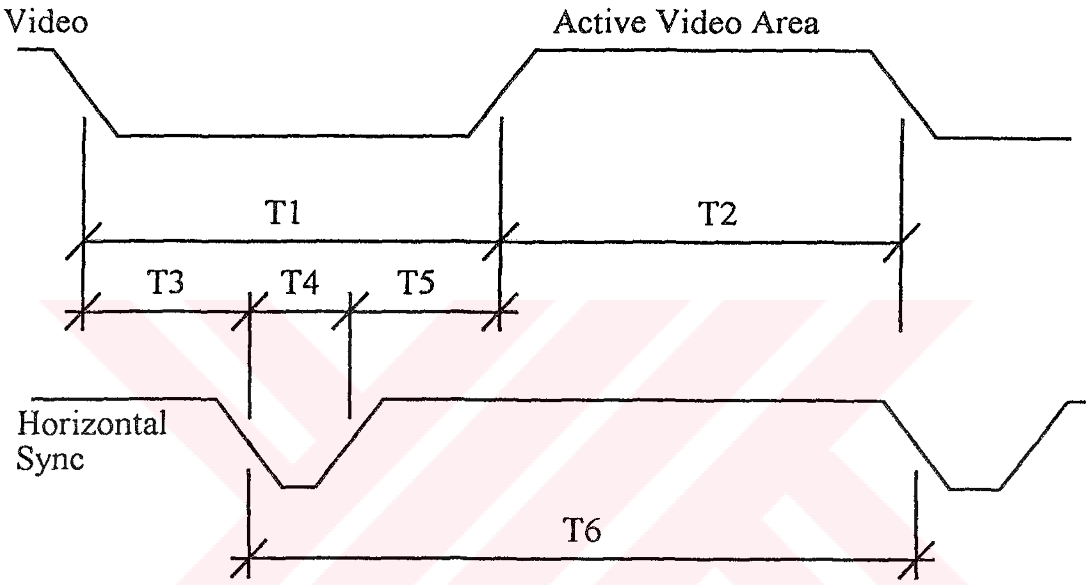
Most of the logic system for the VGA card is contained in one module. This module contains all the circuits necessary to generate the timing for the video memory, and generates the video information which goes to the video digital-to-analog converter (DAC).

Software support is provided by video BIOS. Video BIOS is a part of the system BIOS. BIOS is contained in a read-only-memory (ROM) on the system board. This ROM BIOS contains the character generators and the control code to run the video subsystem [1].

The digital video output is sent to the digital-to-analog converter (DAC), which contains a colour look-up table. Three analog signals (red, green, blue) are output from the DAC and then sent to the display. The sync signals to the monitor are in TTL levels. The analog video signals are between 0 to 0.7 volts.

3.3.1 CRT Controller

The Cathode Ray Tube Controller (CRTC) generates horizontal and vertical synchronous timings, addressing for the regenerative buffer, cursor and underline timings, and refresh addressing for the dynamic RAMs [1].



T1	6.365	microseconds
T2	25.422	microseconds
T3	0.636	microseconds
T4	3.813	microseconds
T5	1.907	microseconds
T6	31.778	microseconds

Figure 3.6 Horizontal timing(s)-40/80 column, no border [1]

3.3.2 Sequencer

The sequencer generates basic memory timings for the dynamic RAMs and the character clock for controlling regenerative memory fetches [1].

3.3.3 Graphics Controller

The graphics controller is the interface between video memory and the attribute controller during video memory reading or writing. During display times, memory data is latched and sent to the attribute controller. In All Points Addressable (APA) modes, the parallel memory data is converted to serial bit-plane data before being sent; in alphanumeric (A/N) modes, the parallel attribute data is sent directly. During a system microprocessor write or read to the video memory, the graphics controller can perform logical operations on the memory data before it reaches video memory or the system microprocessor data bus, respectively [1].

3.3.4 Display Support

The video subsystem supports attachment of 31.5 kHz horizontal sweep frequency direct drive analog displays. These displays have a vertical sweep frequency capability of 50 to 70 cycles per second, providing extended colour and sharpness and reduced flicker in most modes [1].

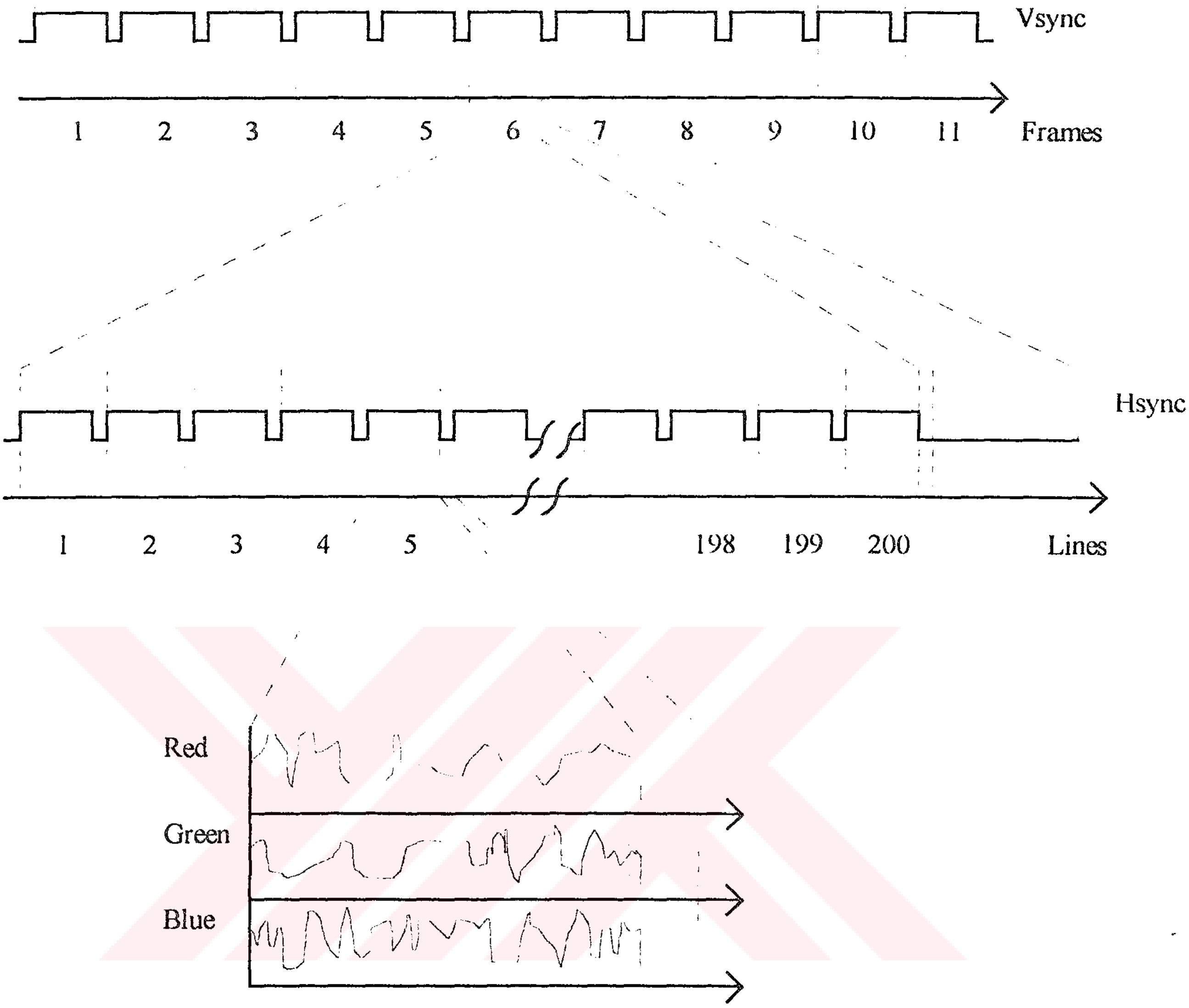


Figure 3.7 Horizontal and vertical signal diagram of PC video card.

3.3.5 Video Digital-to-Analog Converter (DAC)

The video digital-to-analog converter (DAC) integrates the function of a colour look-up table with three internal DACs for driving an analog display. The size of the colour look-up table is 256 by 18 bits to allow the display of 256 colours from palette of 256K possible colours. Each RGB analog output is driven by a 6-bit DAC. Each register in the colour look-up table contains the data for the red, green, and blue DACs [1] .

3.3.6 15 Pin Display Connector Timing (SYNC Signals)

BIOS sets the VGA registers to generate the video modes. The video modes are shown in Table 2.1. All of the modes are 79 Hz vertical retrace except for the modes 11 and 12. These two modes are 60 Hz vertical retrace. The VGA timing generators are within the specifications for the supported displays using these modes. The analog displays operate from 50 to 70 Hz vertical retrace frequency. The active video and Hsync signal timing diagram is given in Figure 3.6 for a screen with no border.

Hsync signals repeats itself 200 times for each cycle of the Vsync signal. The number of lines is given for the 320x200 resolution mode in Figure 3.7. There are 200 cycles of Hsync at every cycle of Vsync signal for this resolution mode.

CHAPTER 4
INSTRUMENTATION OF TV CARD

4.1 BLOCK DIAGRAM OF THE TV CARD

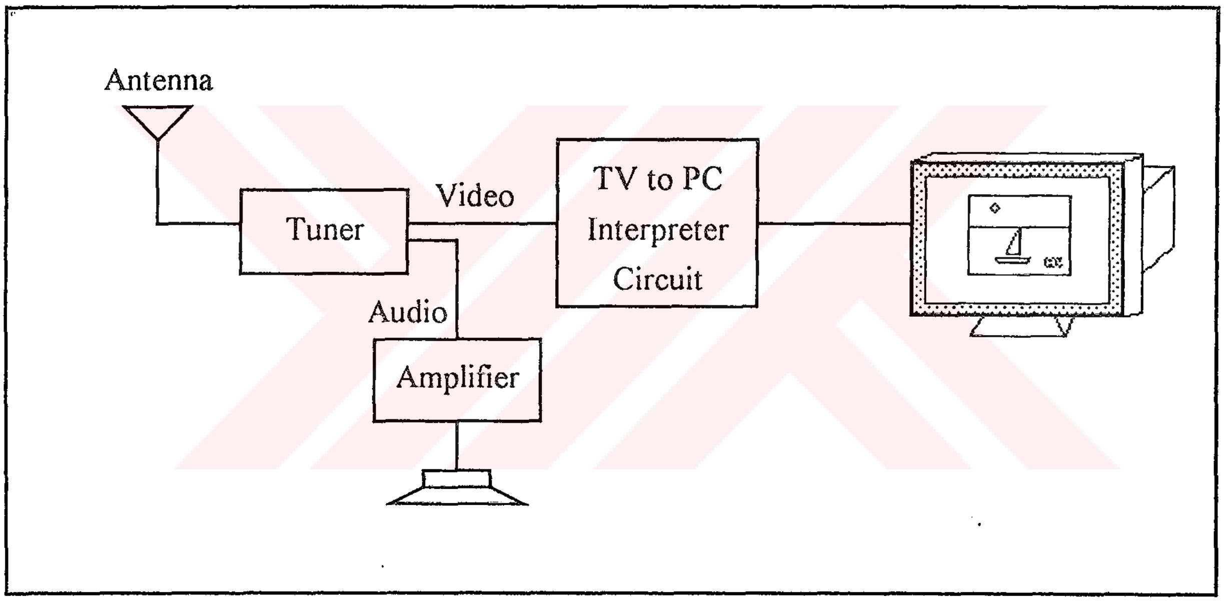


Figure 4.1 Antenna, tuner, interpreter block and PC monitor block diagrams

An antenna and a tuner have to be used in order to receive TV channels. Composite video signal is obtained at the output of tuner with a simple process. The composite video signal has two main components; the video and audio. Information about TV signal components and diagrams is given in section 3.2. The global block diagram is shown in Figure 4.1. As seen clearly, timing diagrams of a TV and PC monitor are not similar. The PC monitor has to be driven by RGB signals. TVs have composite video signals. They have to be separated into red, green, and blue (RGB) components. That process can be accomplished by using a Philips TDA3561A IC. The timing diagram of RGB signals is almost the same as composite video signals. The RGB signals do not have any burst or any other component while TV signals have. They only carry the colour components of each picture. Each cycle of RGB component repeats itself for every 64 microseconds. During this time, one line of the TV screen is scanned by the electron gun. On the contrary, a line on the PC monitor is scanned in 32 microseconds. Thus, RGB signals of TV have be compressed twice in the time domain. This process can not be achieved on the analog signals. Therefore, first, the RGB signals have to be digitised and processed and then they have to be converted again into the analog signals to drive the PC monitor.

The necessity of the compression is shown in figure 4.2.

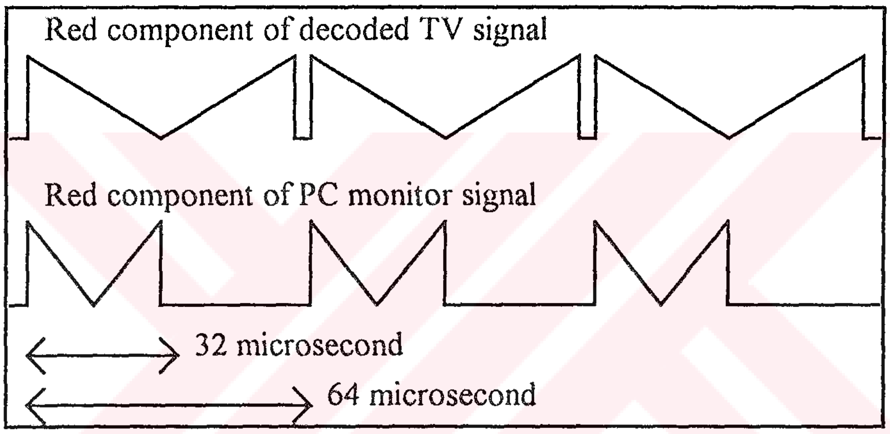

Red component of decoded TV signal
Red component of PC monitor signal
32 microsecond
64 microsecond

Figure 4.2 Comparison of the same signals for TV and PC

In order to show TV pictures on the PC monitor, all TV signals have to be compressed two times in the time domain. The compression process can be achieved by writing to a RAM (Random Access Memory) at an f_s frequency and by reading the RAM at $2\ f_s$ frequencies. In this case, written data can be read twice fast than the writing time. The visual quality of the pictures on the screen depends on the sampling frequency, f_s. The higher the sampling frequency is , the better the image quality is. When the sampling frequency becomes higher, the synchronisation problem gets more difficult. The sampling frequency, starting time to the sampling and other synchronisation timings are the main problems of the system.

4.1.1 The Block Diagram of Digital Circuit

The main block diagram of the system is shown below in figure 4.3. Red, green, and blue signals come from the TDA3561A or TDA3562 ICs of Philips [3]. This three different signals have to be written to the RAM. Therefore, an analog switch must be used to select one channel out of three channels for a few times. An ADC (Analog-to-Digital Converter) samples the signals at the end of the multiplexer. An analog buffer is located at the output of the DAC (Digital-to-Analog Converter). Analog buffers usually have two modes, one of these on-state, and the other one is off-state. While writing to the RAM, the first buffer must be on-state and the second buffer must be off-state. While reading mode all the buffers is set to the inverse mode. After completing sampling process, the multiplexer sends another signal while switching the input signals. Then the data is written to the increased address of the RAM. Incremental RAM address is supported by the second counter which is connected to the address bus of the RAM. The counter increases the logic output pins at the rising edge of the clock signal. This process continues until one picture frame is completed.

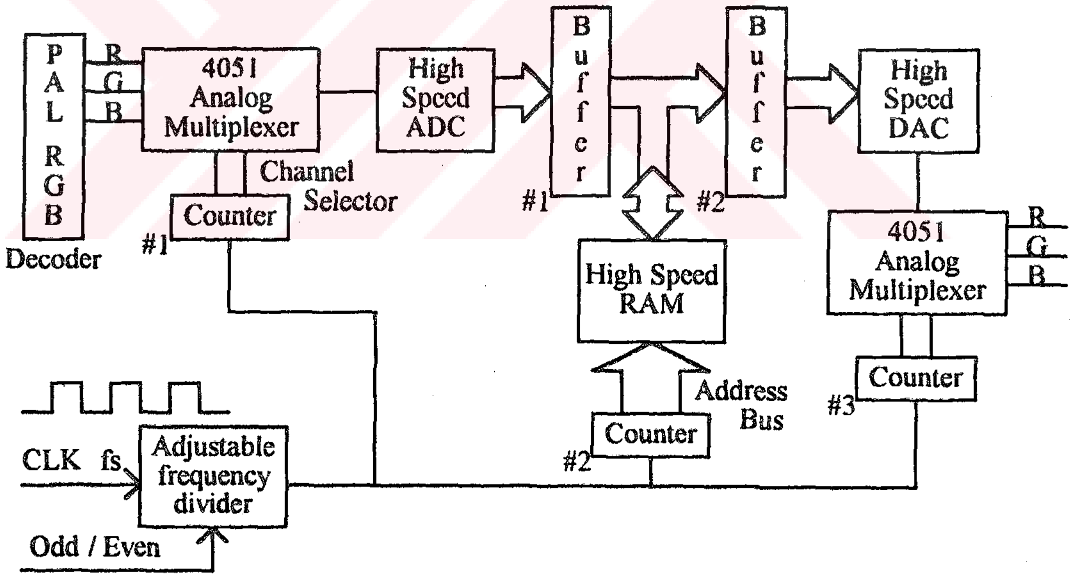


Figure 4.3 Block diagram of one of the solution

Table 4.1 The running modes of each component of digital circuit.

ODD / EVEN	RAM	Divider Factor	CS of Buffer #1	CS of Buffer #2
L	Write Mode	CLK / 2	L	H
H	Read Mode	CLK / 1	H	L

TV signals are transmitted in two stages. First, all the odd lines, then all the even lines are transmitted. When the odd lines are completed, the first buffer is set to the High-Z (off-state mode), and the second buffer is set to the on-state. Meanwhile, the RAM is set to read mode. In this case, counter restarts counting to support the RAM address. The RAM gives out the data written previously. Since the first buffer is off-state, the data coming from RAM must go through the second buffer which is on-stateed. These data coming from buffer is converted to the analog form. The shape of output signal of the DAC is similar to the input signal of the TV card except their frequencies. The input signals come from three different sources, because PAL RGB decoder IC [3] converts the composite video signals into the RGB form. Signals, which are converted into the analog form, are switched and separated to each RGB values.

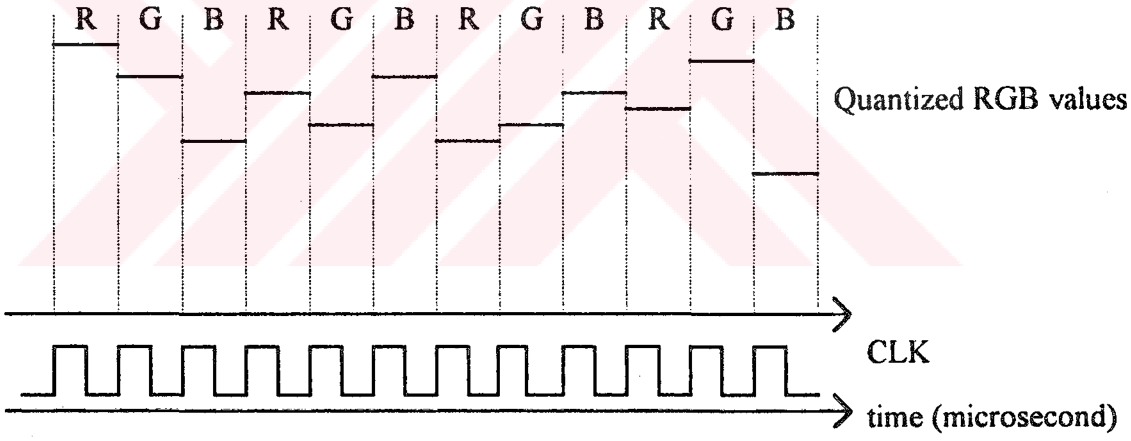


Figure 4.4 Quantised and sequenced RGB values

If the adjustable frequency divider sends the same clock frequency for each write and read modes, input of the ADC and output of the DAC will be same, but our aim is to prepare the TV signals for PC monitor. While writing to the RAM, the ratio of the adjustable frequency divider must be set to two. Given clock frequency is divided by two and writing frequency is determined as $f_{write} = f_{clk}/2$. While reading the RAM, first buffer will be off-state, second buffer will be on-state. At the same time, the scale factor of the adjustable frequency divider has to be set to one which means $f_{read} = f_{clk}$. The main solution is summarised as $f_{write} = f_{read}/2$. Optimum f_{write} frequency can be adjusted depending on the frequency of the input signal. Input signal frequency of TVs is 5.5 MHz. If we set sampling frequency according to the Nyquist rule, it must be at least 11 MHz. But the Nyquist frequency is not sufficient enough to realise such an application circuit. The frequency must be at least five times greater than the maximum frequency component of the input signal.

If the Nyquist frequency is enough, the sampling frequency should be at least 11 MHz. The sampling frequency is equal to the f_{write} frequency and the f_{read} frequency is two times higher than the f_{write} frequency depending on the rule mentioned above.

$$f_{write} = f_{sample} = f_{clk} / 2 = f_{read} / 2$$

In this case, f_{clk} has to be at least 22 MHz.

While the RAM is being read at f_{clk} frequency, the multiplexer also processes at the same frequency. During this time period, the output signals of ADC are not mixed with the output signals of the RAM, because the first buffer is off-state.

As seen in Table 4.1, the block diagram switches the analog signals. Quantised signals is written to the RAM at the sampling frequency. During the writing process, second buffer is off-state; therefore, there will be no signal at the output of the device. When all of the lines of a TV frame is completed, the device switches itself to the output mode. At this time, first buffer is set to the off-state, and the second buffer is set to the on-state. The RAM is also set to the reading mode. System clock frequency is set to the f_{read} frequency, as well. Therefore, if the writing mode takes t_1 milliseconds, the reading mode will take only

$^{t_1}/_2$ milliseconds. At the rest of the time, the system must wait to receive another odd line from the TV.

4.2 PROBLEMS OF THE DESIGNED SYSTEM

The circuit stores each red, green and blue values of data in the RAM. This storing process is carried out until the whole frame is completed. In order to achieve this process, the RAM capacity should be high enough. In the lowest resolution mode, 320x200, there are 320 pixels per line. Each pixel is generated by three components of RGB. Therefore, 320x3=960 bytes have to be stored in the RAM for each line. There are 312.5 odd and 312.5 even lines on the screen for each frame. If only odd lines are received, $312x960 = 299520$ bytes per frame has to be stored in the RAM. Therefore, the capacity of the RAM must be high enough to store this much data. A RAM with a high speed and capacity should be used, and the access time of it should be at least 30 nanoseconds.

Another problem is called colour shifting or shading. Figure 4.4 shows the multiplexed RGB by 4051 Analog Multiplexer IC [6]. There is a time delay between each RGB component during the multiplexing process. This time delay is related to the clock frequency. First, the red signal is converted to the digital signal and stored, then the green and blue signals are processed in the same way. This process continues until all the odd lines are completed. Sampling of the RGB signals does not occur at the same time; therefore, a colour shifting and a deflection on the image occur.

The same problem occurs on the second multiplexer connected to the output of the DAC.

Because of the colour deflection and high capacity fast RAM requirement, the block is not implemented. In order to overcome these problems, a new circuit is designed.

4.3 MAIN BLOCK DIAGRAM

The way in which the first circuit works is valid for the current block diagram. The output of the PAL RGB decoder is connected to the high speed ADC. Digitised data is connected to the input port of two separated buffers. The first buffer behaves like a on-state for a few microseconds. Although the input signals of these two buffers are the same, the output signals are different as seen in Figure 4.5. Figure 4.7 shows the running modes of the buffers and RAMs. Additionally, the path of the data transferred from the ADC to the DAC can be seen.

If there were two data busses for the RAM, there would be no need to this circuit because the read and write operations would be proceeded at the same time. This system is the main part of the circuit. The input part of the ADC and the output part of the DAC are the additional parts of the circuit. The system processes the data in digital form.

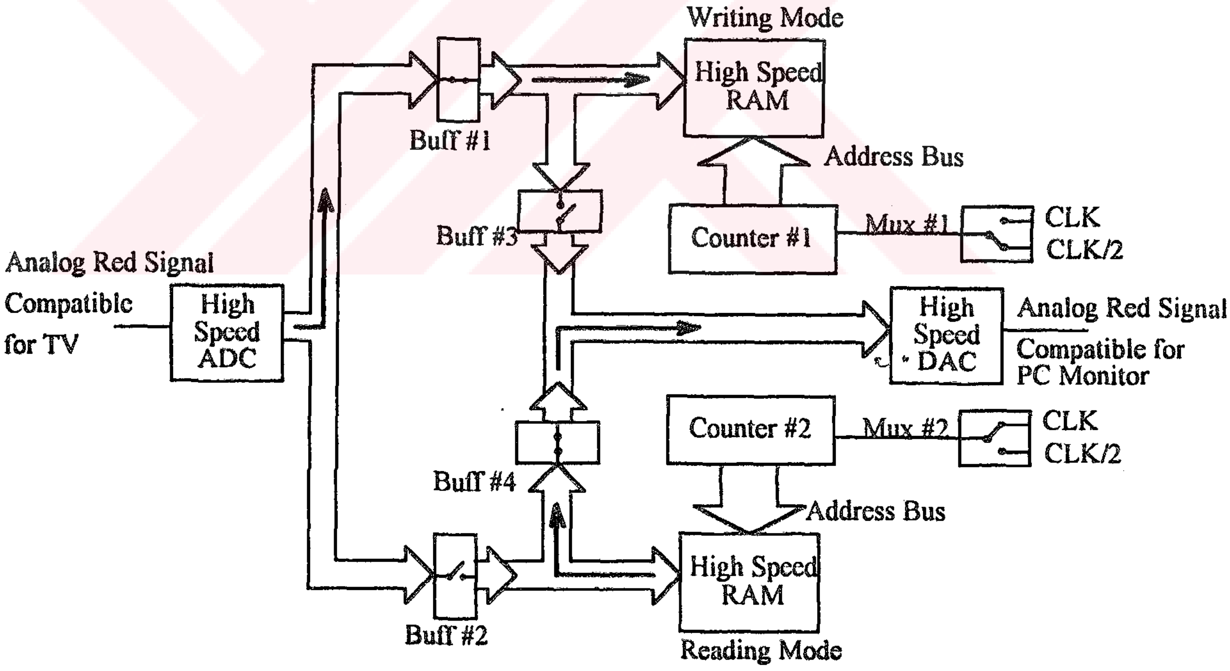


Figure 4.5 Flow Diagram of the Digital Part

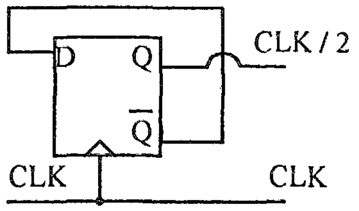


Figure 4.6 Frequency Divider

The flow chart of the digital block is summarised in Table 4.2. As seen in the table, there are two main modes in the digital part of the circuit. The status of each element is given in this table.

The system is started in the first mode and stays in this mode until the horizontal fly back signal comes. The next line comes a few microseconds later. At this time, digital circuit switches all the system sync signals to the second mode. When the next line data comes to the data path, it is changed and the new data is written to the second RAM. During this time, the first RAM is read and the data of the first RAM is fed to the DAC in order to be converted to the analog signal. While the data is being written to the RAM, the address bus incremented at the CLK / 2 frequency. On the other hand, while reading the data from the RAM the address bus is incremented at the CLK frequency. This is called "*write slow, read fast*" process and it is the main trick of this circuit. However, the analog signal has to be compressed twice in the time domain without any loss.

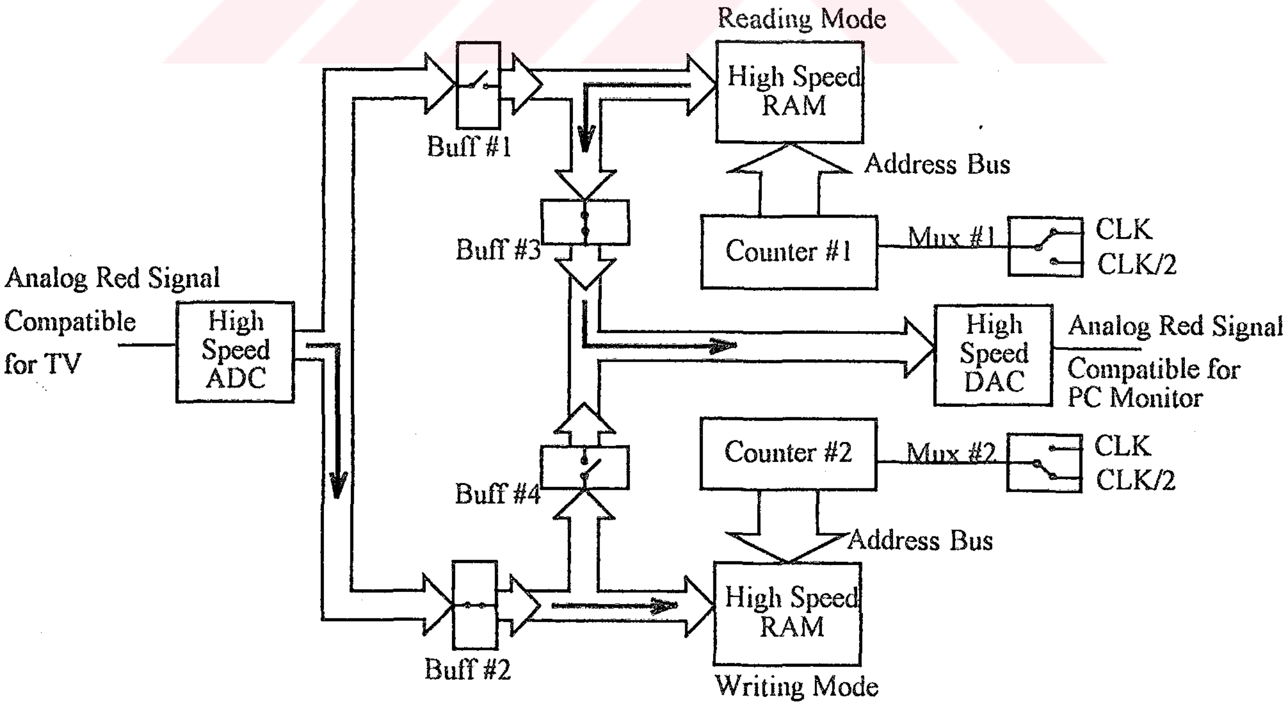


Figure 4.7 Flow Diagram of the Digital Part

The block diagram shown in Figure 4.5 is realised and worked properly. The output voltage level of the DAC has 3.5 Volts DC offset. The output interface circuit reduces the DC offset and amplifies the AC voltage level to drive the PC monitor.

Table 4.2 Running modes and status of each element

Buffer #1,#3	**Buffer #2,#4**	**RAM #1**	**RAM #2**	**Mux #1**	**Mux #2**
Enable	Disable	Write	Read	CLK / 2	CLK
Disable	Enable	Read	Write	CLK	CLK / 2

Although the main digital part of the system was realised, the PC monitor was driven directly by the signal generator because the tuner and the PAL RGB parts were not available.

The PAL RGB decoder application circuit is given in the Philips Video Catalogue and in the appendices.

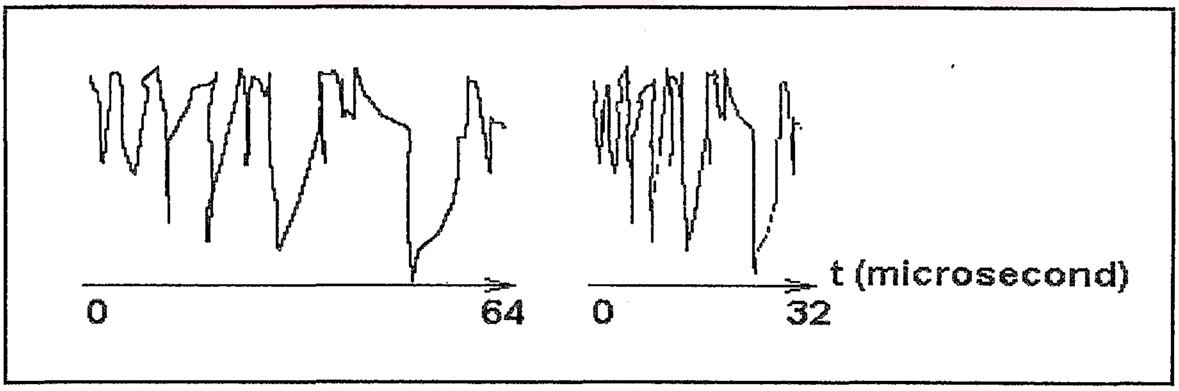


Figure 4.8 A random signal and a two times compressed signal in the time domain.

As described previously PC monitors work with analog RGB signals and digital sync signals. The main purpose of this circuit is to show the TV channel on the PC screen. Because of the unexpected problems, the tuner and the PAL RGB decoder could not be built. But the main part of the system runs properly. As there was no tuner, a real TV channel could not be used to test the system. Thus, original PC signal was fed to the video card.

The digital part of the card stores the converted and decoded signals at a frequency. The card reads the same data from the RAM at a frequency which is two times faster then the writing frequency. Later on, read data is converted again to the analog form. In this case, the data fed to the ADC and received from the DAC are almost the same. But the signal shape of the output signal is compressed twice. The necessity of the compression depends on the timing diagrams of both TV and PC video signals. The relationship between these two timing signals is considered. If the original PC signal is fed to the video card instead of the TV signal, the original signal is compressed twice and the picture on the PC screen is compressed twice horizontally, as well. If there is no compression, the picture will be seen normally. If the compression factor is set to two, image is compressed twice and the RAM is also read two times faster than the writing time. However, the second read operation must be accomplished after the writing operation on the second RAM is completed. Therefore, the reading unit of the circuit has to wait until the target RAM is not busy. When the read operation is started, the write operation is also started on the other RAM. But the read operation is completed earlier than the write operation because of the difference between the writing and reading frequencies. Therefore, the reading unit must wait to read the other target RAM, until the RAM is not occupied. This synchronisation operation repeats itself until the vertical fly back signal appears. When the Vsync signal appears, the picture frame is completed. However, the digital part must wait until a new frame arrives.

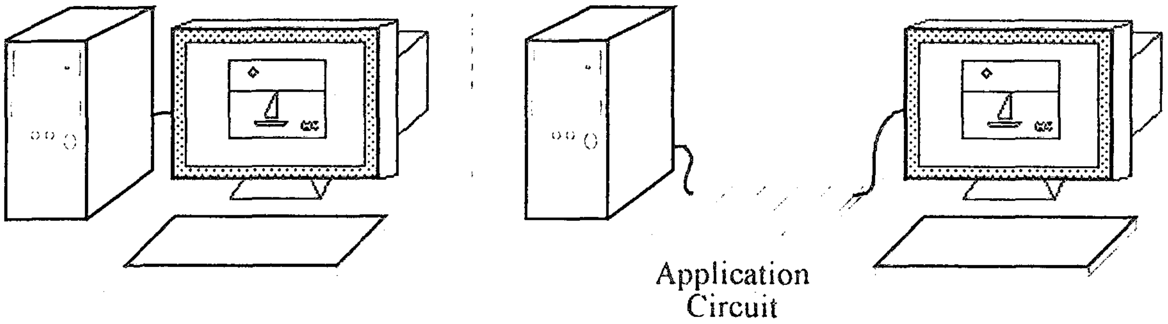


Figure 4.9 Installation of the video card.

4.4 PROBLEMS OF THIS CIRCUIT

In order to have an easy application, the output signal of the video card is used as a source signal. If the tuner and the PAL RGB decoder unit were build, an antenna could be connected to the system. The tuner [8] of the system is controlled by an IIC bus. Therefore, a microprocessor unit or a PC controlled unit has to be built to drive the tuner. The application circuit of the PAL RGB decoder is given in the Philips Video And Associated Systems Catalogue [3]. The application circuit has many external components and it takes much more time than other blocks to be built. Therefore, that unit was not realized. The main part of the system was considered and built. There is no any microprocessor controlled unit inside the system. The speed of the system is too high for the microcontroller based systems. In order not to overload the CPU and reduce the efficiency of the I/O devices, the PC is not used.

The PAL RGB decoder IC receives the composite video signal in, and gives the RGB signals out. Additionally, external RGB signals can be inserted in the IC which mixes them. But the mixing operation occurs on TV standards. Therefore, this mixing operation is not useful for our purpose.

The designed card runs properly and the sampling frequency can be set up to 2.5 MHz. Although the sampling rate is low, the card achieves desired goals. Different tests are made for the card and the results of the tests considered. If the sampling rate gets higher, the quality of the images becomes higher.

4.4.1 Interface Problems

When all the parts of the system are built, each individual part runs properly. But, after bringing all the parts together, the interface problems occur. The cascade connected systems have such kind of problems. The output voltage level of the first block does not match the input voltage level of the second block. To overcome this problem, a new interface block has to be built between these two blocks. This interface block buffers the signals to protect the circuits from overloading and adjusts the voltage levels.

CHAPTER 5
CONCLUSIONS

In order to realise *"write slow read fast algorithm"* a digital block was built. This block runs properly. The designed circuit assumes that the given input signal is in the form of a demodulated RGB TV composite video signal. Therefore the output of the tuner must be connected to a PAL RGB decoder IC. The output of the tuner [8] is in the form of video and audio standards. The audio output of the tuner is not processed anywhere.

The RGB processing unit was built. It runs at the highest frequency which is allowed by the characteristics of the board. Analog-to-digital and digital-to-analog parts of the circuit was built on a printed circuit board, but the other components of the system was built on a board. Thus, when the running frequency is increased, the system is effected more by the noise. If all the system were installed on the printed circuit board and the clock parts of the system were isolated to reduce the spikes, the running frequency would be increased.

As mentioned in the previous chapters PA RGB decoder and TUNER are not used, if the tuner and the PAL RGB decoder parts of the system were built, the system would run completely. The PAL RGB decoder was not built, as well. This part of the system was given in some catalogues. On the contrary, the main part of the system was designed and tested for a long time. It was noted that printed circuit board should be used to increase the running frequency.

The main purpose of this project was to build a cheaper card in our laboratories. Algorithm can be introduced to have more efficient stretching ratio. In visual telephone system the proposed card can be used.

REFERENCES

[1] IBM Technical Reference, (April 1987). Personal System / 2, Model 80.

[2] BREY, Barry B. , Third Edition, The Intel Microprocessors 8086/8088, 80186, 80286, 80386 and Interfacing, DeVRY Institute of Technology

[3] Philips Semiconductors (1991). Integrated Circuits Data Handbook. Video and associated system. Types TDA2658 to A733C.

[4] Yager, Tom (March 1993). Better than being there. BYTE, p129.

[5] Penney, Walt (December 1989). Images. BYTE, p248.

[6] CMOS Logic and Databook (1988)

[7] Fink, D.G. (1952). Television Engineering. (2nd Edition) New York, The Maple Press Company.

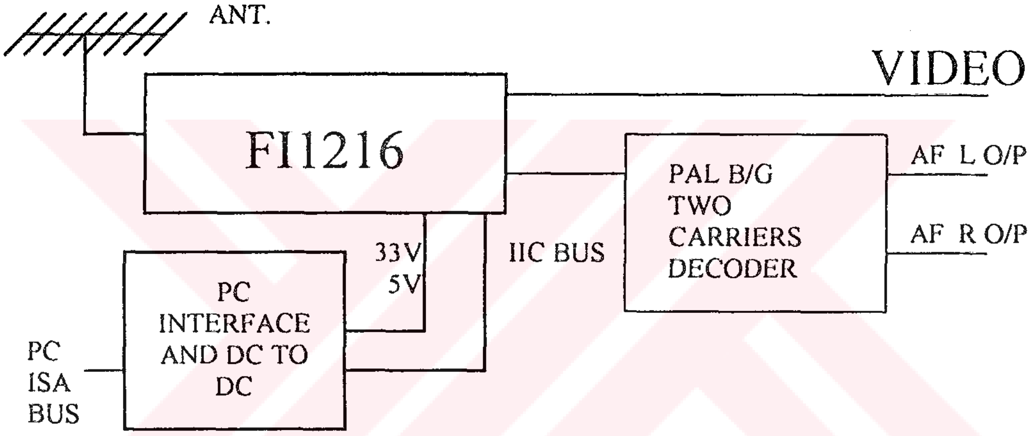
[8] Liang, Jerry & Yu, Sherman. Philips Semiconductors (Sep, 1993). PAL B/G Tuner Add-on Card Using FI1216 for Desktop Video Applications. Industrial Regional Application Laboratory Asia/Pacific-Taipei

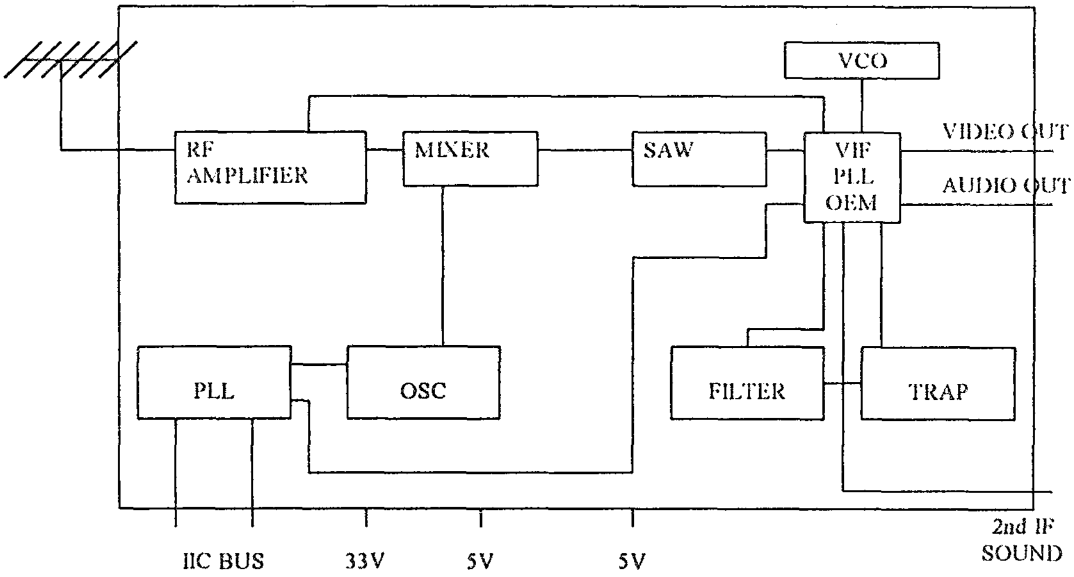
APPENDIX

TV Tuner Board

The PAL B/G tuner board consist of 3 sections.

- FI1216 front end module
- DC-DC converter
- P-AT bus interface


ANT.
VIDEO
FI1216
PAL B/G TWO CARRIERS DECODER
AF L O/P
AF R O/P
33V
5V
IIC BUS
PC INTERFACE AND DC TO DC
PC ISA BUS


VCO
RF AMPLIFIER
MIXER
SAW
VIF PLL OEM
VIDEO OUT
AUDIO OUT
PLL
OSC
FILTER
TRAP
IIC BUS
33V
5V
5V
2nd IF SOUND