UNIVERSITY OF TEHRAN

School of Electrical and Computer Engineering

Digital Logic Laboratory, ECE 045, Fall 1392

Experiment 4 (Sessions 9, 10)

VGA CONTROLLER

Purpose

In this lab, you will learn about:

- The operation of VGA controller
- Using ROM memories in digital systems
- Displaying characters on monitor
- Displaying images on monitor

Background Information

In this experiment, you are to design a VGA controller to display a text and an image on a VGA monitor. Therefore, you need a ROM block to save an image data and a ROM block to save character information in it. You should load image data and characters from these ROMs and display on VGA screen.

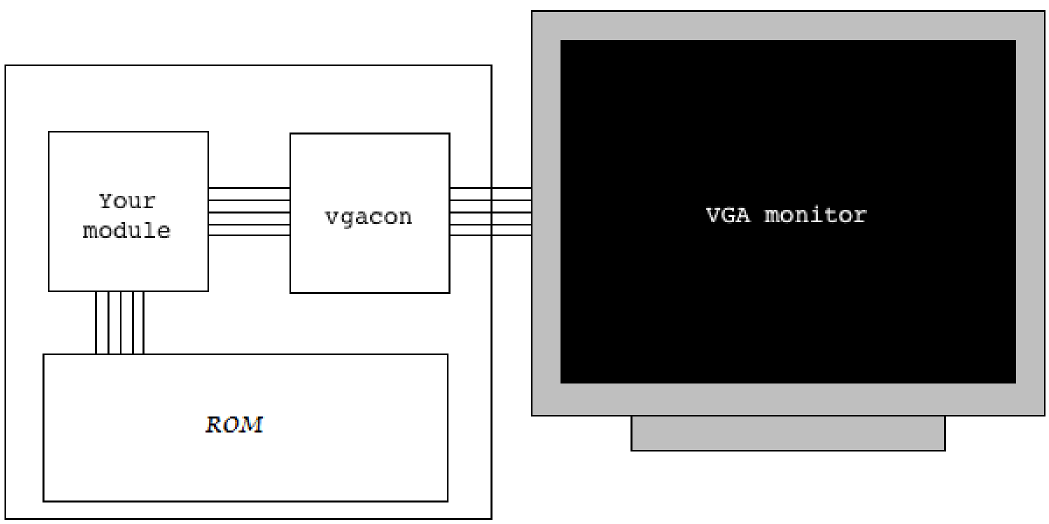

Figure 1-The Design block Diagram

Image ROM

In computer systems, image is a 2-dimentional array of pixels. Each pixel is a small dot in display that has an independent color. The formation of thousands of pixels makes up the image in the computer displays. RGB color model is used in VGA controller. In RGB color model red, green, and blue light are added together in various ways to reproduce a broad array of colors for producing various colors. You need a ROM block initialized with the content of the image you want to display.

At first, you should learn how to use the available blocks in Quartus II. In this design you will use the Available ROM blocks in Cyclone FPGA. For this purpose, you can add available blocks from $Tools \rightarrow MegaWizard\ Plugin\ Manager$. You should choose the Memory Compiler folder and select the 1-port ROM then specify a name for the output file for this module, and set the number of words and their width in this wizard. An 'MIF' file that contains the data for a 160x120 image is provided. In the final stage, you should specify the initialization file for the Wizard. Each word of mentioned ROM contains data for one pixel of the stored image. Each pixel has 3 bits representing Red, Green and Blue.

The Cyclone chip on UP3 board does not have sufficient on-chip memory to save a value for each monitor pixel. A simple solution to this problem is to show the picture on the screen several times (Tile). Each memory location is mapped into several locations on the screen. The screen resolution is 640×480, but the defined ROM memory is for a 160×120 image. Consequently, you should copy each location of memory 16 times on the screen.

The way that you should map memory contentment to picture display pixels are shown in figure2.

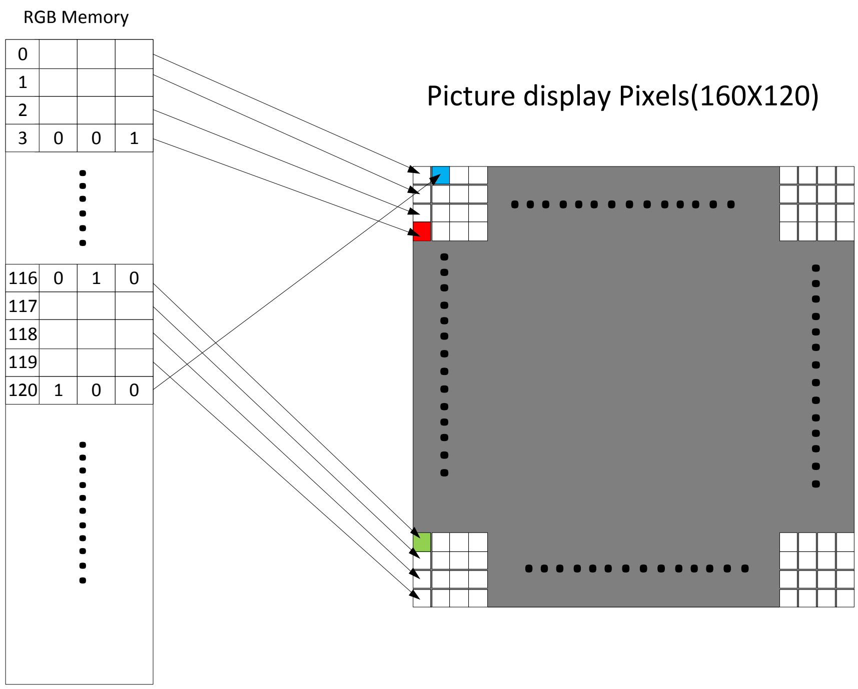

Figure 2-Mapping RGB Memory to pixels

Character Font ROM

Characters are displayed on a VGA monitor using fixed pixel patterns or fonts. The character fonts can be stored in a ROM, which can be accessed by the VGA controller. For the examples in this tutorial, we will use an 8×16 bit array for each character's font. A part of Character Font ROM will hold the fonts of the characters shown in Table 1. A 2048×8 ROM is sufficient to store the 128 character fonts (8×16).

Table 1-Character Address Map for Character Font ROM

Addr	Char	Addr	Char	Addr	Char
30	0	40	@	50	P
31	1	41	A	51	Q
32	2	42	B	52	R
33	3	43	C	53	S
34	4	44	D	54	T
35	5	45	E	55	U
36	6	46	F	56	V
37	7	47	G	57	W
38	8	48	H	58	X
39	9	49	I	59	Y
3A	:	4A	J	5A	Z
3B	;	4B	K	5B	[
3C	<	4C	L	5C	\
3D	=	4D	M	5D	]
3E	>	4E	N	5E	^
3F	?	4F	O	5F	_

Note that the address field in Table 1 only indicates the order of the character fonts in the ROM, not the actual ROM addresses (each character font occupies 16 bytes of ROM, each 'Addr' value must be multiplied by 16 in order to get the ROM starting address for the corresponding character. Another way to think of it is that the Addr field value represents the ROM address bits [10:4] and the four least significant bits of ROM address(addrs[3:0]), specifies a row (0-15) in a character font. Then the bit select logic determines which individual bit from the font row to display to the current pixel. For example, the font map for character '9' is shown below. The character font map can be found in a VHDL file which is located in CECM.

```
"00000000", -- 0
"00000000", -- 1
"01111100", -- 2  *****
"11000110", -- 3 **   **
"11000110", -- 4 **   **
"11000110", -- 5 **   **
"01111110", -- 6  ******
"00000110", -- 7      **
"00000110", -- 8      **
"00000110", -- 9      **
"00001100", -- a     **
"01111000", -- b  ****
"00000000", -- c
"00000000", -- d
"00000000", -- e
"00000000", -- f
```

Figure 3-The font map for character 9

Notice that pixel_row[3:1] is connected to the Character Font ROM address inputs font_row[2:0] and pixel_column[3:1] is connected to the Bit_Select input. This means that the same bit from the Character Font ROM will be displayed at two horizontally adjacent pixels (pixel_row[0] = 0 and pixel_row[0] = 1). Also, the same bit will be selected for two horizontally adjacent pixels (pixel_column[0] = 0 and pixel_column[0] = 1). This has the effect of displaying each character with a font size of 16×16 pixels. Note that the character font size can be easily configured as 8×8 pixels, 16×16 pixels, 32×32 pixels, 64×64 pixels, etc.

The character_address[7:0] input is connected to the sum of the upper 8-bits of the pixel_column[] and pixel_row[] counters. This has the effect of starting each row of characters on the VGA monitor with a different character from the previous row.

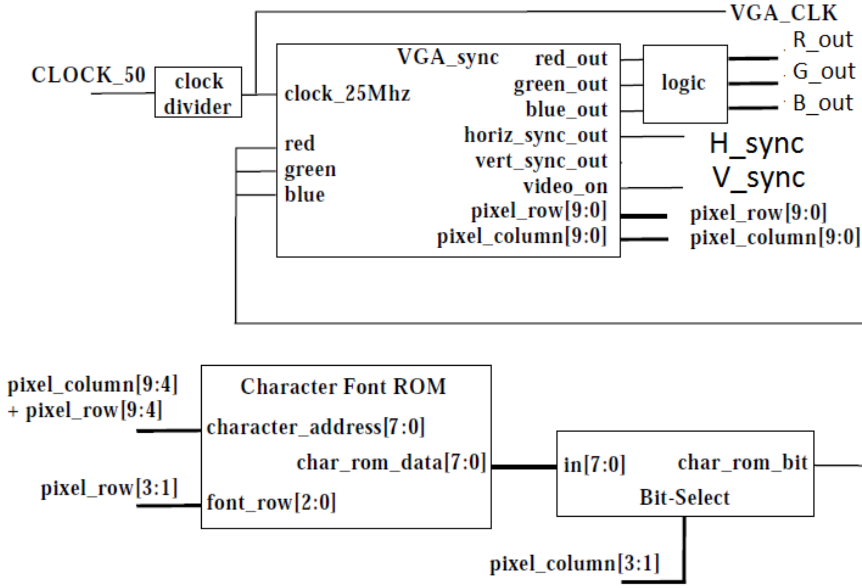

Figure 4-Character display block diagram

VGA Signal Generator

A VGA video signal contains 5 active signals:

- Horizontal sync: digital signal, used for synchronization of the video
- Vertical sync: digital signal, used for synchronization of the video
- Red (R): analog signal (0-0.7 v), used to control the color
- Green (G): analog signal (0-0.7 v), used to control the color
- Blue (B): analog signal (0-0.7 v), used to control the color

By changing the analog levels of the three RGB signals all other colors are produced. The electron beam must be scanned over the viewing screen in a sequence of horizontal lines to generate an image. The RGB color information in the video signal is used to control the strength of the electron beam.

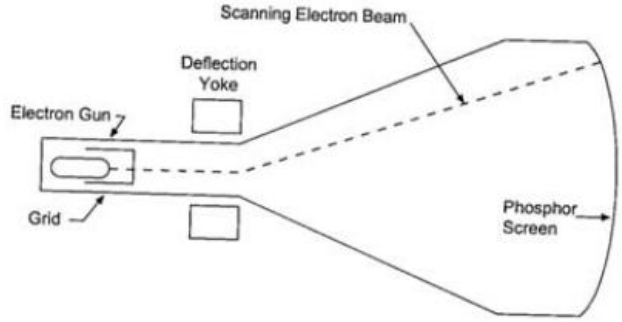

Figure 5-A CRT monitor

The screen refresh process begins in the top left corner and paints 1 pixel at a time from left to right. At the end of the first row, the row increments and the column address is reset to the first column. Once the entire screen has been painted, the refresh process begins again.

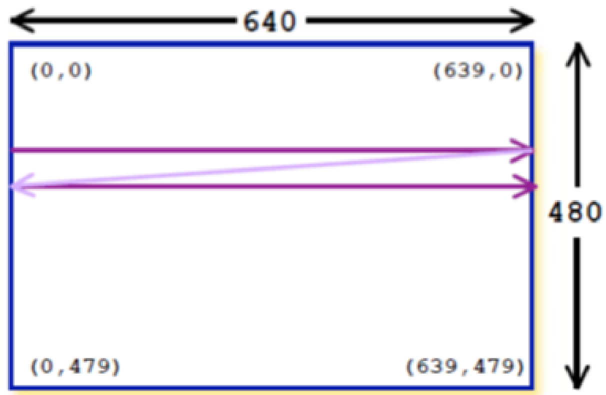

Figure 6-VGA Active region painting

The video signal redraws the entire screen 60 times per second to provide for motion in the image and to reduce flicker; this period is called the refresh rate. Refresh rates higher than 60Hz are used in PC monitors. In 640 by 480-pixel mode, with a 60Hz refresh rate, this is approximately 40 ns per pixel. A 25 MHz clock has a period of 40 ns.

The vertical sync signal tells the monitor to start displaying a new image or frame, and the monitor starts in the upper left corner with pixel (0, 0).

The horizontal sync signal tells the monitor to refresh another row of 640 pixels. After 480 rows of pixels are refreshed with 480 horizontal sync signals, a vertical sync signal resets the monitor to the upper left corner and the process continues.

During the time when pixel data is not being displayed and the beam is returning to the left column to start another horizontal scan, the RGB signals should all be set to black color (all zero). In a PC graphics card, a dedicated memory location is used to store the color value of every pixel in the display. This memory is read out as the beam scans across the screen to produce the RGB signals. The detailed signaling of Sync signals are demonstrated in Figure 6. The colored areas in Figure 6 are in Active region that is visible in monitor, but the gray area is the region that you must set R, G, and B signals to zero.

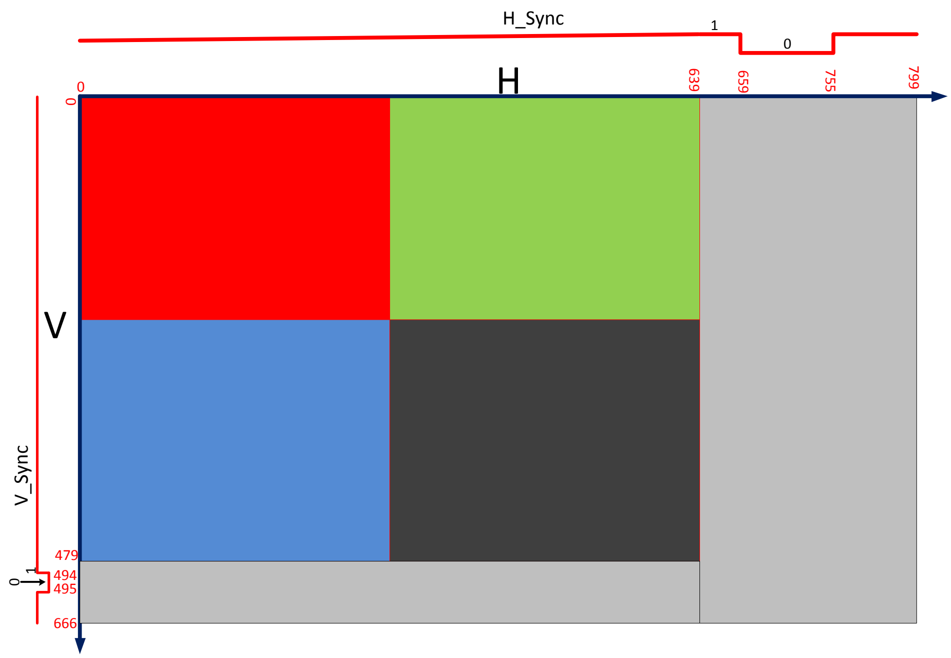

Figure 7-H_Sync and V_Sync signaling

The block diagram of VGA controller is shown below.

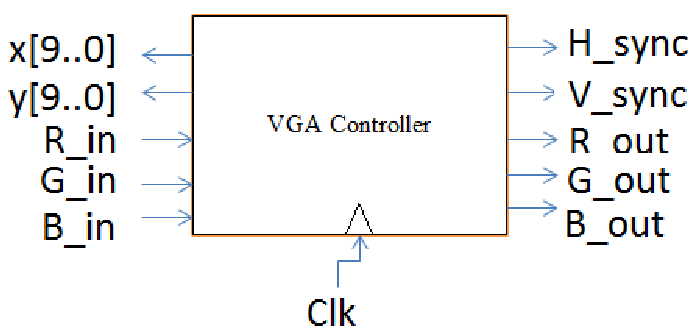

Figure 8-VGA Controller

Pre-Lab assignment

1. Read about the *.MIF files structure on http://quartushelp.altera.com/9.1/mergedProjects/reference/glossary/def_mif.htm
2. Write down a MIF in your prelab paper for initialization of font ROM that contains a 8X16 array like for first character of your last name in Persian.
3. Write a formula that maps each monitor pixel to memory address and shows picture in tile mode (Tile mode is the mode that shows a picture multiple times). This function gets x and y of a pixel and returns the address of image ROM.
4. Sketch a block diagram in RTL level (using multiplexers, decoders, adders, and etc.) for VGA controller.

Lab Work

1. Implement a VGA controller to produce RGB and control signals.

2. Using your design of part 1 implement a system that paints monitor like figure 6 with four colors.
3. Implement a system that shows a picture in tile mode using your function of prelab. Use the ‘picture.mif’ file (provided in CECM) for initialization of Image ROM .
4. Show picture in the middle of the monitor and write your group code in the caption of the picture. Use the ‘font_rom.vhd’ file (provided in CECM) as the character ROM.