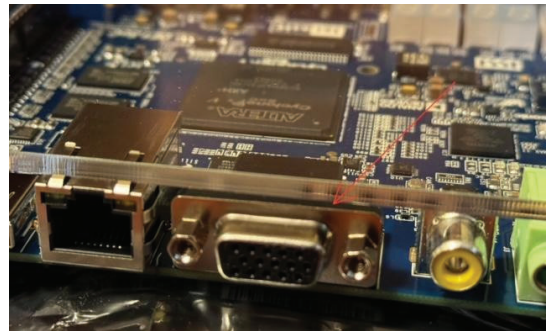
UNIVERSITY OF CALIFORNIA, DAVIS
Department of Electrical and Computer Engineering

EEC 181A **Winter 2024**

Lab 4: Using the VGA Interface on a DE1-SoC Board

I. Introduction

The DE1-SoC board has a 15-pin D-SUB connector designed for VGA (Video Graphics Array) video output. The VGA synchronization signals are generated directly from the Cyclone V SoC FPGA. The Analog Devices ADV7123 triple 10-bit (only the highest 8-bits [9:2] are used) high-speed video DAC transforms the three color signals (red, green, and blue) from digital to analog form. Although the VGA interface can operate at multiple resolutions, this document is written for 640 × 480 pixels, which yields 307,200 pixels per image and 18,432,000 pixels per second @ 60 frames per second.



II. VGA Timing Specifications

For more details of the timing specification for VGA synchronization and RGB (red, green, blue) data, search the web for a good tutorial.

Figure 1 illustrates the basic timing requirements for each row (horizontal) displayed on a VGA monitor. The vga_col_sync (HSYNC) signal is pulsed low to signify the beginning ("back porch") of a row of pixels ("display interval"), and is again pulsed low ("front porch") after the end of the row. During the display interval, pixel RGB data drives each pixel ideally at a rate of 25.175 MHz (39.72 ns per pixel) from left to right across the row being displayed. In our lab to simplify the FPGA's setup, we use the 50.000 MHz default clock which divides to 25.000 MHz.

Horizonal Timing

Horizonal Dots	640	640	640	
Vertical Scan Lines	350	400	480	
Horiz. Sync Polarity	POS	NEG	NEG	
A (us)	31.77	31.77	31.77	Scanline time
B (us)	3.77	3.77	3.77	Sync pulse lenght
C (us)	1.89	1.89	1.89	Back porch
D (us)	25.17	25.17	25.17	Active video time
E (us)	0.94	0.94	0.94	Front porch

```
          ____________________________          ________
________|          VIDEO             |________| VIDEO (next line)
    |-C-|----------D-----------|-E-|
__  ____________________________________  ___________
  |_|                                |_|
  |B|
  |---------------A----------------|
```

Vertical Timing

Horizonal Dots	640	640	640	
Vertical Scan Lines	350	400	480	
Vert. Sync Polarity	NEG	POS	NEG	
Vertical Frequency	70Hz	70Hz	60Hz	
O (ms)	14.27	14.27	16.68	Total frame time
P (ms)	0.06	0.06	0.06	Sync length
Q (ms)	1.88	1.08	1.02	Back porch
R (ms)	11.13	12.72	15.25	Active video time
S (ms)	1.2	0.41	0.35	Front porch

```
          ____________________________          ________
________|          VIDEO             |________|  VIDEO (next frame)
    |-Q-|----------R-----------|-S-|
__  ____________________________________  ___________
  |_|                                |_|
  |P|
  |---------------O----------------|
```

Figure 1. VGA Timing Specifications. [https://www.epanorama.net/documents/pc/vga_timing.html].
(Note: the 25.17 µs "Active video time" seems to be in error; it should probably be 39.72 ns × 640 = 25.42 µs.)

The timing of vertical synchronization using *vga_row_sync* is similar to timing of a single row, except that a pulse signifies the end of one frame and the start of the next, and the data in between comprises all of the rows within a single video frame.

III. The VGA_controller.v Module

The VGA_controller.v module is designed so that the colors that are driven on the signals *ired*, *igreen*, and *iblue* will appear at the pixel location indicated by *col* and *row*.

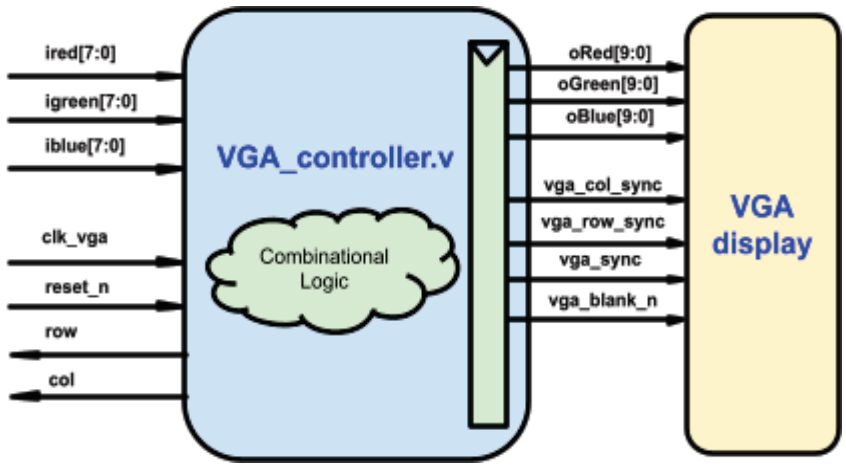


Figure 2. VGA_controller module block diagram

Port names (Verilog)	Direction	Width	Description
clk_vga	Input	1	VGA clock @ 25.000 MHz
reset_n	Input	1	VGA reset, active low, asynchronous
ired	Input	8	The input value of red color
igreen	Input	8	The input value of green color
iblue	Input	8	The input value of blue color
vga_col_sync	Output	1	VGA horizontal synchronization signal
vga_row_sync	Output	1	VGA vertical synchronization signal
vga_sync	Output	1	VGA synchronization signal
vga_ blank_n	Output	1	VGA blank signal, active low
ored	Output	8	The value of red color to VGA
ogreen	Output	8	The value of green color to VGA
oblue	Output	8	The value of blue color to VGA
col	Output	13	Indicates the horizontal pixel position
row	Output	13	Indicates the vertical pixel position

Table 1. Ports of Interface

The timing of when a pixel in the image is displayed is indicated by two signals, *col* and *row*. *col* indicates the current horizontal pixel position, and *row* indicates the current vertical pixel position. *col* ranges from 0–639 and *row* ranges from 0–479.

IV. Sample Project Description

To prepare for the future possibility of using the 50.000 MHz SDRAM, the example code is written with a main 50.000 MHz clock and a 25.000 MHz divided clock (done in a risky way that you should *not* do as described in Rule #7 in the *Single-bit Memories* handout). So the example code requires the mildly-awkward situation where your control circuits operate at 50.000 MHz and the pixel rate controller operates at 25.000 MHz. Fortunately the situation is entirely handled by merely calculating and outputting an RGB pixel color appropriately coordinated by the *row* and *col* signals, even though *row* and *col* will change value only every other cycle.

A module or always block is needed to control all of the signals in Table 2. The signals *row* and *col* indicate the row and column position for RGB information on the VGA monitor.

For example, to display the color white at the pixel position (10, 10), when the *row* and *col* signals are (`13'd0010`, `13'd0010`), the circuit must drive the *ired*, *igreen*, and *iblue* signals all to `8'hFF`.

The overall system reset signal is called *reset_n*, is asserted low, and is driven by the push button KEY[0].

V. Design Descriptions [100 points]

Download and modify the sample project contained in the posted file `vga_top.zip` Details not specified must be chosen by you and listed in your pdf submission.

1) [20 points] Modify the Verilog code on the sample project to draw 2 triangles that overlap each other on the VGA monitor. The color of the triangles is selectable from one of two values chosen by the SW switches. Choose colors, sizes, and locations on the screen.

2) [30 points] Draw an 16-pixel wide × 6-pixel tall green square in the center of the display, then the square grows larger by one pixel in each direction per video frame until the whole display is green; when that happens, the process repeats.

3) [40 points] Draw a square on the screen that moves a specified number of pixels each video frame. The number of pixels of movement is determined by four SW switches: two switches determine the horizontal speed (0, 1, 2, 4 pixels per frame) and two determine the vertical speed (0, 1, 2, 4 pixels per frame). When the square hits a frame boundary, it must bounce back in the direction from which it came.

Submitted work [10 points] Submit a pdf document to canvas of the following for each of the three designs:

- choices you made for your designs (e.g., colors, object sizes, key signals, etc.)
- circuit diagram of the circuits you designed
- all Verilog code in pdf file

Extra credit [5 points max] Add two more squares to part (3) that bounce independently. Choose some method to set the velocity of each square to a different value.