VGA DISPLAY PORT INSTRUCTIONS

The Xilinx3 board includes a VGA display port as shown in Figure 1. You can connect this port directly to most PC monitors or flat-panel LCDs using a standard monitor cable.

The Nexys3 board uses 10 FPGA signals to create a VGA port with 8-bit color and the two standard sync signals (HS – Horizontal Sync, and VS – Vertical Sync). The color signals use resistor-divider circuits that work in conjunction with the 75-ohm termination resistance of the VGA display to create eight signal levels on the red and green VGA signals, and four on blue (the human eye is less sensitive to blue levels). Using this circuit, 256 different colors can be displayed, one for each unique 8-bit pattern. The 8 color signals (3 for red, 3 for green and 2 for blue) and 2 sync signals are displayed in Figure 2.

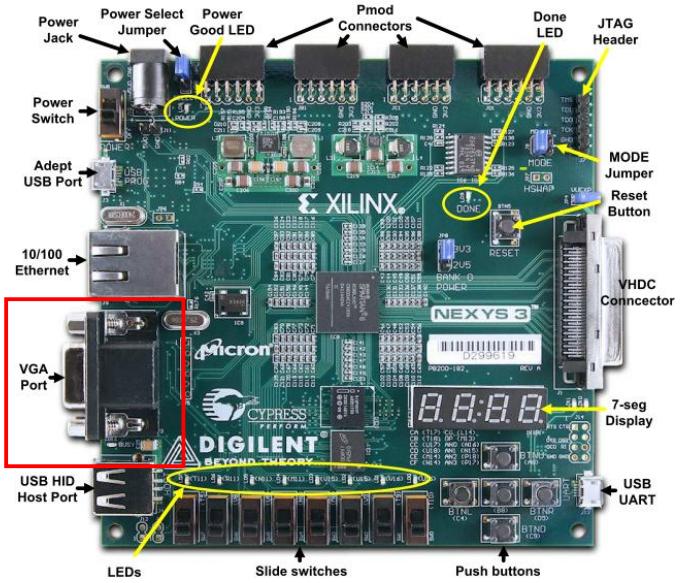


Figure 1 VGA connector on board

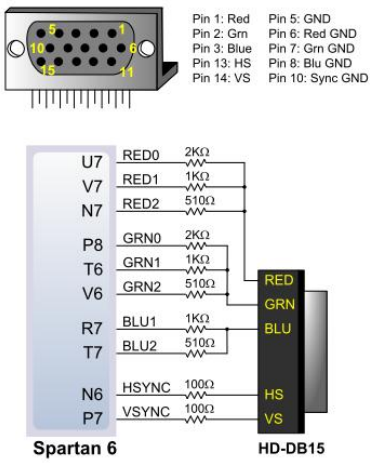


Figure 2 Interface to design

CRT-based VGA displays use amplitude-modulated moving electron beams (or cathode rays) to display information on a phosphor-coated screen. A VGA controller circuit must generate the HS and VS timings signals and coordinate the delivery of video data based on the pixel clock.

In the folder provided by your TA, you will find four files: game.v, hvsync_generator.v, hvsync_generator_tb.v and nexys3.ucf.

In **hvsync_generator.v**, (i) vga_h_sync and vga_v_sync are generated for synchronization. If you are interested, you may look at Figure 3 and Figure 4 to see how they are generated, (ii) inDisplayArea defines the display area 480×640, (iii) CounterX and CounterY are two counters that keep track of the horizontal and

vertical positions. And they are used in game.v to coordinate video data. **hvsync_generator_tb.v** is a testbench file so that you can simulate hvsync_generator.v to view waveforms of output signals.

game.v contains the main logic of your design. It receives five signals from hvsync_generator, generates color information based on counter and counterY, and finally output video data in the form of 3-bit RED, 3-bit GREEN and 2-bit BLUE.

nexys3.ucf contains interfaces specification and timing constraints. Default setting has one pin enabled for each color, that is, you can display up to 8 (2^3) colors. You may need to enable more pins for RED, GREEN and BLUE to be able to display more colors.

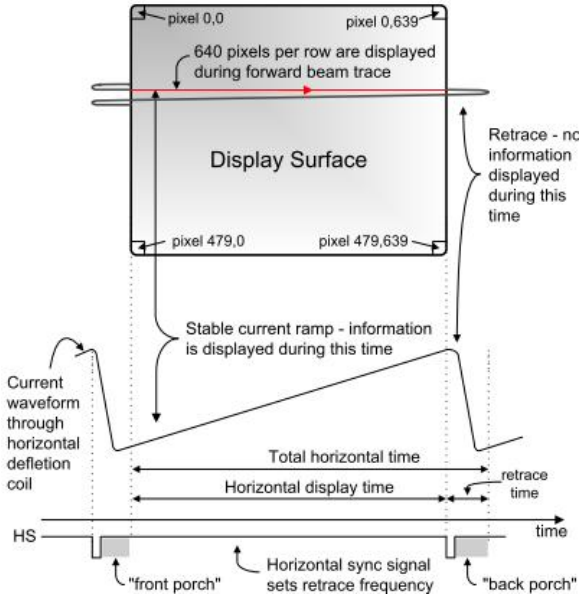
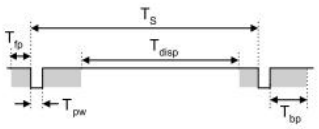


Figure 3 CRT-based VGA



Symbol	Parameter	Vertical Sync			Horiz. Sync	
		Time	Clocks	Lines	Time	Clks
T_S	Sync pulse	16.7ms	416,800	521	32 us	800
T_{disp}	Display time	15.36ms	384,000	480	25.6 us	640
T_{pw}	Pulse width	64 us	1,600	2	3.84 us	96
T_{fp}	Front porch	320 us	8,000	10	640 ns	16
T_{bp}	Back porch	928 us	23,200	29	1.92 us	48

Figure 4 Timing specification

Hint 1: To display a moving object using VGA, you can use two variables (X and Y) to represent its position of the screen, and define the entire object by setting relative distance of each pixel of the object.

Hint 2: To display repetitive patterns, you can use a simple statement, such as "R=CounterX[8:5]==?" in game.v. This will display repetitive red vertical strips each with width equal to 2^4 pixels, and repeating every 2^8 pixels. The "?" represents the displacement of the object on the screen.