VGA DISPLAY MANUAL

Background

Nexys3 FPGA 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 VGA 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.

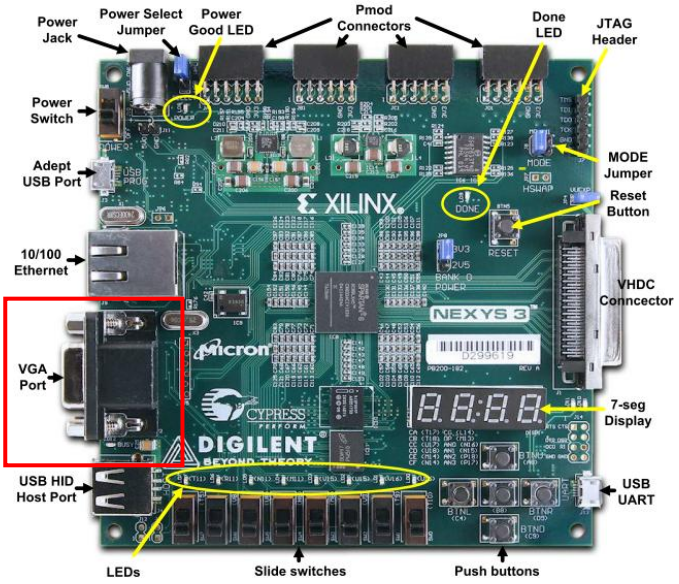


Figure 1 VGA connector on board

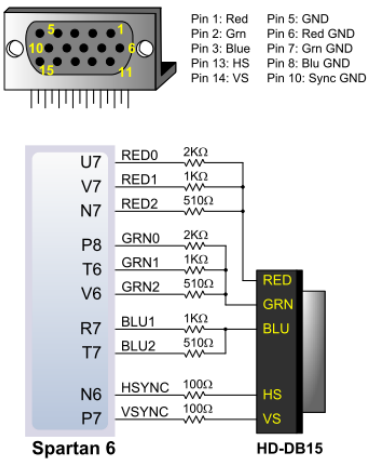


Figure 2 Interface to design

What to do?

CRT-based VGA displays use amplitude-modulated moving electron beams (or cathode rays) to display information on a phosphor-coated screen. We have provided a VGA control module, which generate the HS and VS timings signals and coordinate the delivery of video data based on the pixel clock. This module is defined in "hvsync_generator.v". A testbench is also provided to test this module.

Please make sure you understand how VGA works with the help from your TA before you proceed.

You need to build a core design which generates the corresponding video data so as to display the expected patterns on the monitor/screen. A sample core design is given, named as vga_demo.v.

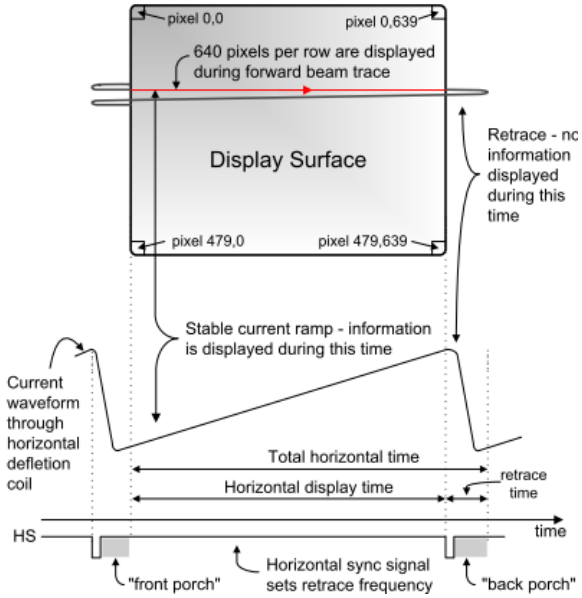
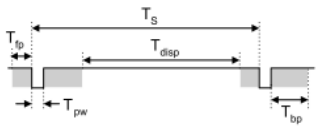


Figure 3 CRT-based VGA



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

Source files

In this folder, you will find following four files for the VGA project:

hvsync_generator.v contains the VGA control module, i.e., *hvsync_generator*, in which (i) *vga_h_sync* and *vga_v_sync* are generated for synchronization. (ii) *inDisplayArea* defines a 480×640 display area, and (iii) *CounterX* and *CounterY* are two counters that keep track of the horizontal and vertical positions. And they are used in vga_demo.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.

vga_demo.v contains the main logic of your design, i.e., *vga_demo*. It receives five signals from *hvsync_generator*, generates color information based on *CounterX* and *CounterY*, and finally outputs video data in the form of 3-bit RED, 3-bit GREEN and 2-bit BLUE.

nexys3.ucf contains interface specifications and timing constraints. The default setting for VGA ports has one pin enabled for each color. Using this setting, you can display up to 8 (i.e., 2^3) colors for each pixel. You may need to enable more pins for RED, GREEN and BLUE to be able to create a fancy image.

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

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