

DLD VGA Driver Module Tutorial



June 2005
Revision A
Prepared by: David Fritz
Digital Logic Design
Oklahoma State University

Overview

This document attempts to explain the basic usage of the Digital Logic Design VGA Driver Module written for the Digilent Spartan 3 FPGA by David Fritz and Eric Larson. This document is tutorial in nature and no knowledge of how VGA monitors work is expected. A working knowledge of how to use Xilinx ISE and how to program the Digilent Spartan 3 FPGA is required.

What you need

In order to complete this tutorial you will need the following skills and resources:

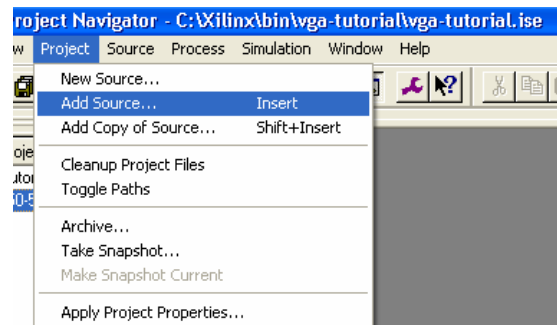
1. Xilinx ISE Software (available in lab and on the web [see Xilinx Tutorial])
2. Computer (available in lab).
3. Digilent Spartan 3 FPGA (available in your parts kit).
4. Monitor (available in lab).

Part I – Creating the VGA Module

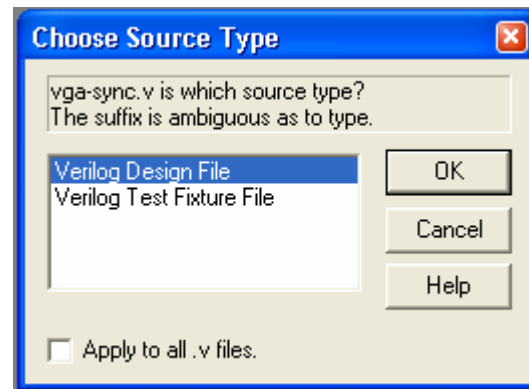
Before designing our test circuit, we need to download and create the VGA module. The module (named vga-sync.v) can be downloaded from the course website in the “Resources” section.

Start a new project in Xilinx ISE. Download the VGA module (vga-sync.v) to your project directory.

Add the module to your project by selecting Project->Add Source. Select the vga-sync.v file and select “Open”.

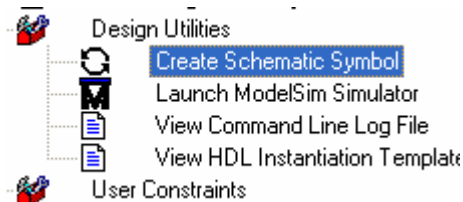


You will be asked what kind of module the file is. Select “Verilog Design File” and press “OK”.



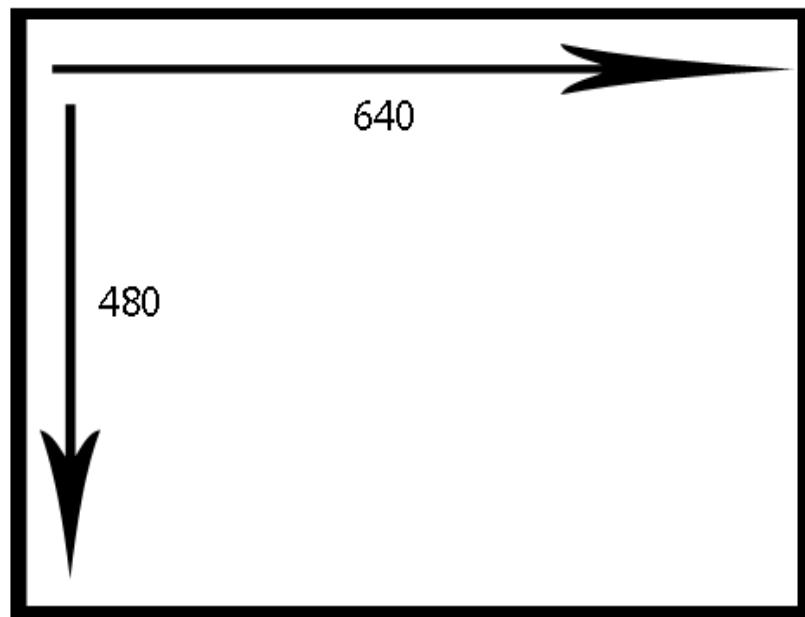
You should now have the vga_sync module added to your project.

Select “Create Schematic Symbol” from the “Design Utilities” submenu.



The vga-sync module provides a layer of abstraction between the user circuit and the relatively complex task of drawing to the monitor. With the vga-sync module running, your circuit will have to present color data to it on demand. This is accomplished by a sync signal that the module creates. When the sync signal is generated, your circuit will have to use the available row and column address (provided by the vga-sync module) and return a RGB 3-bit color value. This may sound difficult, but is surprisingly easy to do using the StateCAD editor.

The area of the monitor is divided into several thousand pixels. More specifically, it is divided into a 640 column, 480 row matrix of 3-bit color pixels as depicted below. Note that point (0,0) is in the upper left hand corner, and that the y-axis is pointed down in the positive direction.

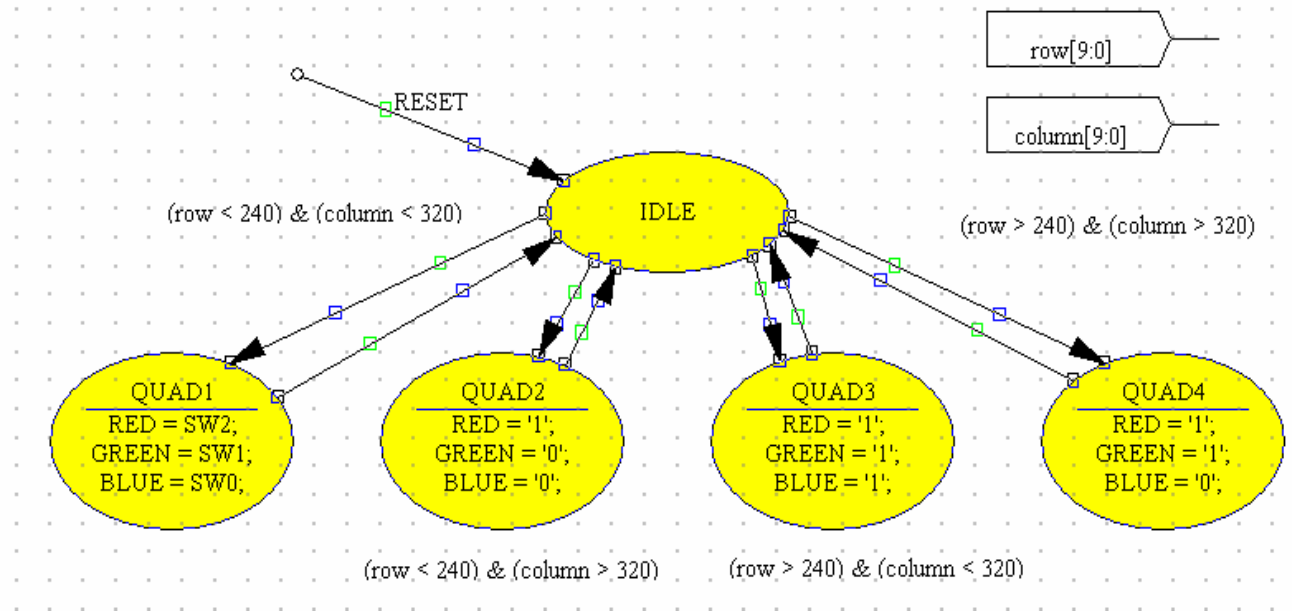


■ 000	■ 001	■ 010	■ 011
■ 100	■ 101	■ 110	■ 111

RGB Colors

Part II – Creating a Test Circuit

For our test circuit, we will create a simple test pattern that divides the screen into 4 quadrants. The top left quadrant's color will be determined by 3 switches. The other three will be hard-coded into the design.



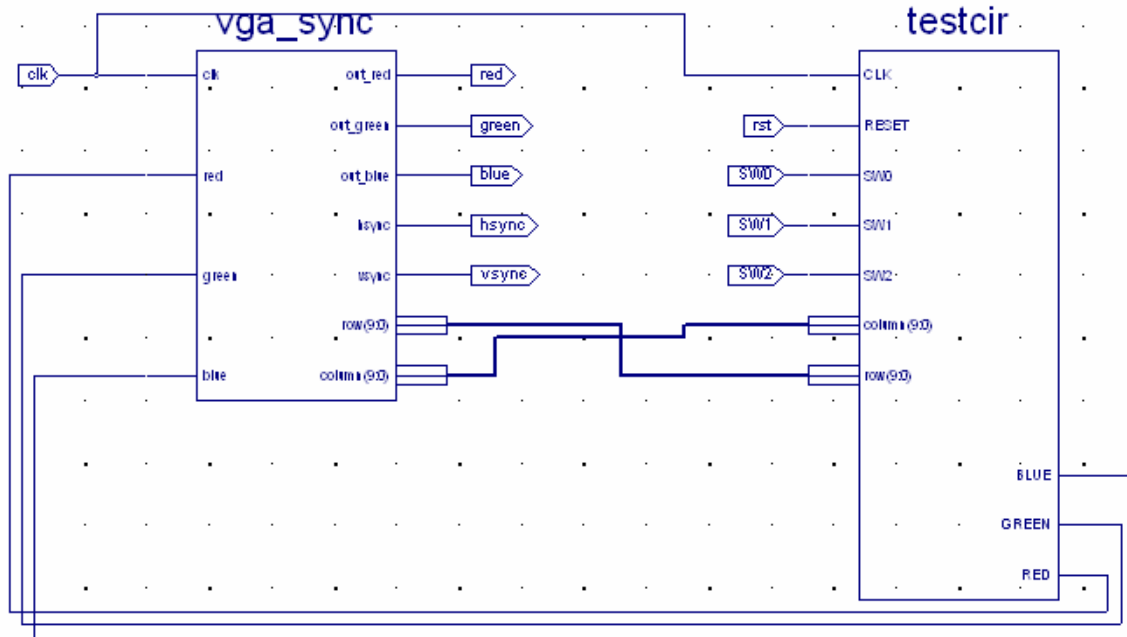
Open StateCAD and create the following state machine (see StateCAD tutorial).

Note that the transitions leaving IDLE for state QUAD2 and QUAD3 are listed below the states. The transitions going back to IDLE from all other states have no condition. Also note that row and column are 10-bit vectors. Remember from the StateCAD tutorial to turn on the 'Retain Output Values' option.

This state machine should make the color in quadrant 1 equal to the color defined by the three switches we will assign. Quadrant 2 will be red, quadrant 3 will be white, and quadrant 4 will be yellow.

Compile, import, and create a schematic symbol for this state machine.

Now create a schematic and join the two modules as shown.



Save and close the schematic. Open the UCF constraints editor and assign the pins as follows. Please note that the pins listed here are very specific. These pins **must** be assigned properly for the device to work. Please refer to the FPGA tutorial for an explanation of the constraints file.

```

1 NET "clk" LOC = "T9";
2 NET "red" LOC = "R12";
3 NET "green" LOC = "T12";
4 NET "blue" LOC = "R11";
5 NET "hsync" LOC = "R9";
6 NET "vsync" LOC = "T10";
7 NET "SW2" LOC = "H14";
8 NET "SW1" LOC = "G12";
9 NET "SW0" LOC = "F12";
10 NET "rst" LOC = "M13";

```

Build the .bit file and program the FPGA. Hook up the monitor to the FPGA's HD-15 (VGA) port. You should see an image like the one to the right. The first quadrant is controlled with Switch 2, Switch 1, Switch 0.



Part III – Conclusion

At this point, you should be able to construct and display a simple VGA test pattern. With this knowledge you should be able to construct more complex designs and patterns. Feel free to ask the TA's for help or advice on more advanced screen drawing techniques.