San Jose State University
Department of Electrical Engineering
EE178, Fall 2017, Crabill

Laboratory Assignment #4

Objectives

The goal of this lab is for you to implement a circuit that generates video signal for a specific display format, and output it to a video display, using the VGA connector on the Digilent Basys3 board. When you successfully complete this lab, you will have developed a piece of intellectual property that you might be able to re-use in the future.

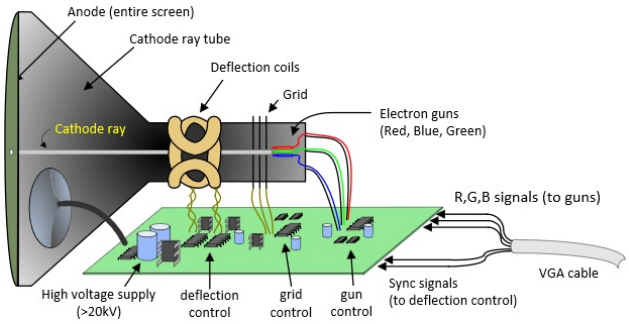

Figure 1: CRT-Based Video Display [Digilent]

Now that you are familiar with the tools from Laboratory Assignment #1, you should be able to concern yourself with digital design. Figure 2 shows a symbol of the design you will create. The inputs are shown on the left and the outputs are shown on the right.

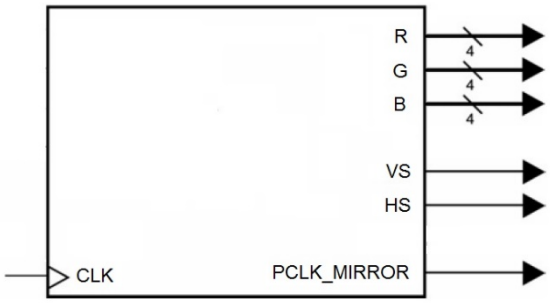

Figure 2: A Symbol of the Design

Bibliography

This lab draws heavily from documents on the Digilent website http://www.digilentinc.com. I would like to thank Digilent for making this material available.

Digilent has an overview of a VGA Display Controller for the 640x480@60Hz display mode, which is relevant to this lab, although we are using a different display mode. Display timing parameters for the 800x600@60Hz display mode are obtained from the Video Electronics Standards Association *VESA Display Monitor Timing Standard* and are reproduced under fair use. Portions of the test bench provided for this lab are based on the TIFF Revision 6.0 Specification from Adobe Systems Incorporated. Figures used in this document to illustrate video timing are obtained from the Digi-Key Corporation Applications

Engineering EEWiki and the Texas Instruments *TMS34010 Graphics System Processor User Guide* and are reproduced under fair use.

Design Description and Requirements

In this design, you are not allowed to use latches. You are allowed to use only one clock. The clock must be a 40 MHz clock signal derived from the 100 MHz clock signal available from the oscillator on the board. You will receive zero points if you do not follow these requirements.

As shown in Figure 2, the design only has one input, a clock. There are six output signals, five of which are used for display signaling, and a sixth which is used to provide an external copy of the 40 MHz clock signal that is derived internal to the design. This external copy of the 40 MHz clock signal exists so that the test bench can operate synchronously with the design logic during simulations.

clk	clock signal, 100 MHz from oscillator
r[3:0]	digital red output, drives a resistive DAC
g[3:0]	digital green output, drives a resistive DAC
b[3:0]	digital blue output, drives a resistive DAC
vs	vertical sync
hs	horizontal sync
pclk_mirror	pixel clock, copy of derived internal clock

The design must generate a video signal as specified in the next section. The design must properly drive a video display and generate a solid image without flicker, snow, or other distortions.

Describing the Design

Create a project using the following files posted on the class website. Instead of creating source files and copying text, use the add source capability to directly add the files to the project. When adding files, you have the option to copy the files into the project directory – which is advisable, so that the files do not get separated from the project directory.

1. testbench.v (top level testbench)
2. tiff_writer.v (simulation display capture, sub-module used by testbench)
3. vga_example.v (top level design, contains timing controller and test pattern generator)
4. vga_example.xdc (top level design constraint file)
5. vga_timing.v (timing controller, sub-module used by vga_example)

After reviewing the files and their contents, you will notice that the timing controller module has a declaration, including port names, but otherwise no contents. The primary goal of this assignment is to design the timing controller and describe it in the timing controller module.

Before you begin writing any code, you must sit down with scratch paper and draw a block diagram of a circuit that will satisfy the design requirements. Once you have a possible solution, write a description of it in Verilog-HDL and proceed to test it in simulation. Use the provided files for your design. You may change the declared port data types if you need to suit your design description.

Video typically consists of a sequence of images, or frames. Figure 3 shows how a non-interlaced raster display is created by scanning a phosphor-coated screen in a regular pattern with an amplitude modulated electron beam. During the active, or visible, portion of the frame, the electron beam excites the phosphor coating on the screen, generating light. After scanning the active portion of the frame, the electron beam is returned to its starting place and the process is repeated.

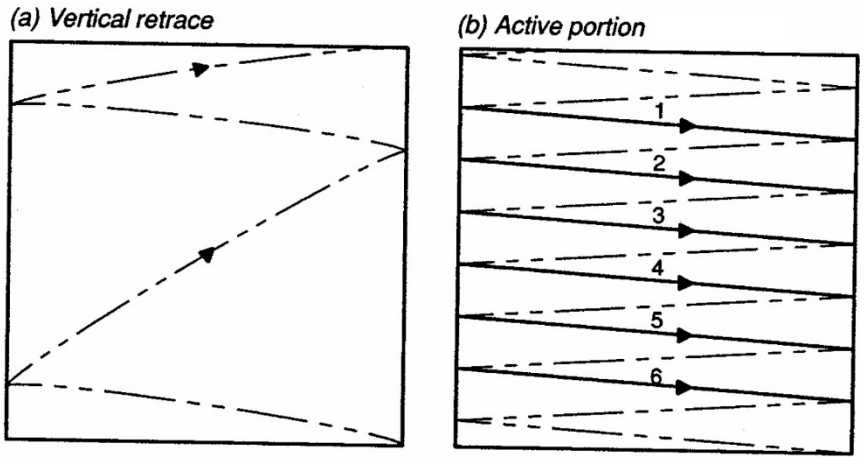

Figure 3: Non-Interlaced Raster Display Scan (Beam) [Texas Instruments]

Common raster displays operate based on a set of inputs from a video timing controller. In addition to color information, the display needs synchronization signals, as shown in Figure 4. A complete interface consists of five signals: R, G, B, horizontal sync HS, and vertical sync VS. Although the traditional CRT display device is growing less and less common – newer display devices are designed to accept the same signals.

The VS, or vertical synchronization signal forces the display to return the beam to the top edge of the display. The HS, or horizontal synchronization signal forces the display to return the beam to the left edge of the display. Figure 4 shows a spatial representation of the synchronization signals. The display time represents time spent drawing the visible portion of the display, with the (0, 0) pixel coordinate marked in the upper left corner of the display time.

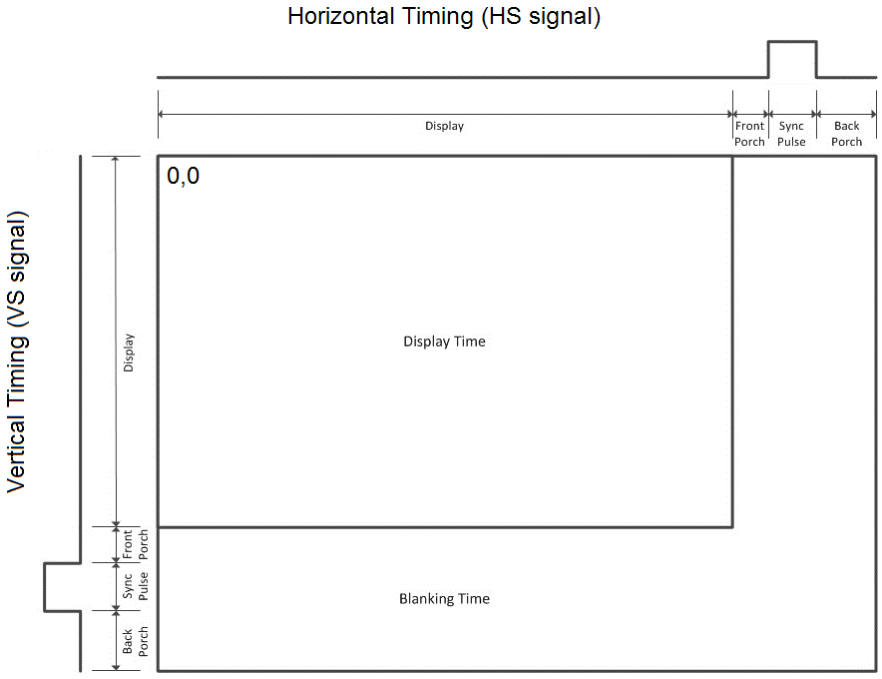

Figure 4: Non-Interlaced Raster Display Synchronization (Spatial) [Digi-Key]

The actual interface is temporal in nature, and the raster display effectively performs a temporal to spatial conversion. Figure 5 shows a temporal representation of the display interface. Two blanking signals, VB and HB, are also shown. These signal when to turn off the electron beam during retrace so that it does not perturb the active display area. The same effect may be achieved by simply turning off the R, G, and B

signals during beam retrace. For this reason, blanking signals are typically not present on an interface to a raster display. Instead, the image generation circuit takes the responsibility to ensure that the R, G, and B signals are turned off when either VB or HB are true.

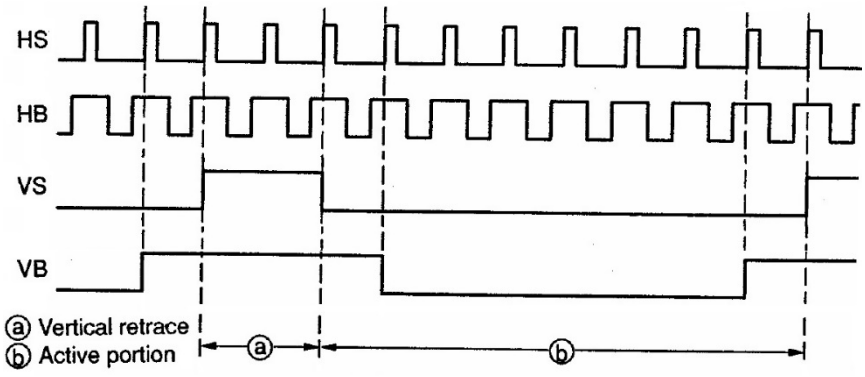

Figure 5: Non-Interlaced Raster Display Synchronization (Temporal) [Texas Instruments]

The video timing controller module you are to design relies on two counters and some associated logic.

The first counter counts pixels on a given line at the pixel clock rate. The output of this counter is called the horizontal count, hcount[10:0], which is an indicator of the current beam position on the horizontal axis. Comparators in the timing controller watch the horizontal count and determine the value for hblnk, hsync, and when to reset the horizontal counter.

The second counter counts lines on a given frame. The output of this counter is called the vertical count, vcount[10:0], which is an indicator of the current beam position on the vertical axis. This counter is enabled to increment each time the horizontal counter reaches its terminal count. Comparators in the timing controller watch the vertical count and determine the value for vblnk, vsync, and when to reset the vertical counter.

The video timing controller makes hcount[10:0], hblnk, hsync, vcount[10:0], vblnk, and vsync available as module outputs. The horizontal and vertical counts, taken together at any instant in time, may be interpreted as an (h, v) coordinate – or (x, y) coordinate, if you prefer – of the current pixel being drawn on the display. The upper left pixel is (0, 0).

VGA stands for Video Graphics Array. Initially, it referred specifically to the display hardware first introduced with IBM® PS/2 computer in 1987. With the widespread adoption, it now usually refers to the analog computer display standards (defined by VESA), the DE-15 connector (commonly known as VGA connector), or the 640x480 resolution itself (commonly known as VGA resolution).

In this exercise, you will design a video timing controller for an 800x600 display resolution (commonly known as Super VGA, or SVGA resolution) with a 60 Hz frame rate. This is a common display mode supported by most monitors, panels, and projectors. The display timing parameters are shown in Figure 6.

You will notice the required pixel clock frequency is 40.000 MHz, but the clock available on the Basys3 board is 100 MHz. Most current programmable logic devices contain clock management circuits based on PLLs or DLLs which enable users to perform a variety of operations such as phase shifting and frequency synthesis (not just division, they also support multiplication).

In this lab, we are using the Xilinx Multi-Mode Clock Manager, or MMCM, to perform frequency synthesis of a 40 MHz clock from the 100 MHz clock that is available. Inspect the code showing the instantiation of the MMCM primitive, how it is configured by parameters, and how it is connected. Then look at the 7 Series FPGA Libraries Guide, UG953 and scan through it to find the MMCM information. Browse the document to get an idea of what else is available.

Timing Name	=	**800 x 600 @ 60Hz;**	
Hor Pixels	=	**800;**	// Pixels
Ver Pixels	=	**600;**	// Lines
Pixel Clock	=	**40.000;**	// MHz
Scan Type	=	**NONINTERLACED;**	
Hor Sync Polarity	=	**POSITIVE;**	
Ver Sync Polarity	=	**POSITIVE;**	
Hor Total Time	=	1056	Pixels
Hor Addr Time	=	800	Pixels
Hor Blank Start	=	800	Pixels
Hor Blank Time	=	256	Pixels
Hor Sync Start	=	840	Pixels
// H Right Border	=	0	Pixels
// H Front Porch	=	40	Pixels
Hor Sync Time	=	128	Pixels
// H Back Porch	=	88	Pixels
// H Left Border	=	0	Pixels
Ver Total Time	=	628	lines
Ver Addr Time	=	600	lines
Ver Blank Start	=	600	lines
Ver Blank Time	=	28	lines
Ver Sync Start	=	601	lines
// V Bottom Border	=	0	lines
// V Front Porch	=	1	lines
Ver Sync Time	=	4	lines
// V Back Porch	=	23	lines
// V Top Border	=	0	lines

Figure 6: Display Timing for 800x600@60Hz [Video Electronics Standards Association]

The test pattern generator is a circuit that evaluates the current horizontal and vertical coordinates of the beam to decide what color should be displayed. The Basys3 board has triple 4-bit digital to analog converters (built with resistors) to support analog RGB signaling for the VGA connector.

The obvious (or rather, easy) choice for representation of pixel color in a digital design for the Basys3 board is RGB 4:4:4 which means the color space is RGB and each color channel is represented by a 4-bit value. This provides a palette of 4,096 colors.

Testing the Design

You must perform some minimal functional simulation of the design. This is important for two reasons. First, it will give you confidence your design is working properly before you implement it. Second, if the design does not behave as expected when you download it, you will have a mechanism to quickly create additional test cases to help debug the problem. The instructor will not help you debug logic problems (incorrect design behavior) unless you have a block diagram and are able to run a simulation.

A simple test bench is provided. The test bench can be used for functional simulation as well as timing simulation after the implementation step. Feel free to enhance the basic test bench as you see fit.

For initial evaluation and debug, you should be using the waveform display to check what your circuit is doing, to gain confidence it is behaving as you expect. You can use the cursors to measure the timing, or probe into the design to compare the counter values to the synchronization and blanking signal behavior. The test bench is a great way to verify your design in simulation, although it can take a while to simulate a full frame.

The provided test bench automatically generates TIFF output files that show your simulation results as they would appear on the display, frame by frame. The files will be located in one of the project simulation directories, for example lab4\lab4.sim\sim_1\behav and are sequentially named frame000.tif, frame001.tif, etc… Figure 7 shows the expected output of the unmodified test bench that is provided to you, assuming you have implemented a correct display timing controller. If your timing controller is not correct, what shows up in the TIFF output files is undefined and may not even be viewable.

Figure 7: Expected Result with Correct Display Timing Controller

The non-black region is the visible area – it is mostly gray, but bounded by multi-color perimeter that is one pixel wide: the top is a yellow line, the right is a blue line, the bottom is a red line, and the left is a green line. Inspect your results *carefully*. The black regions around the visible area represent the display blanking time and are only partially visible on an actual display.

A common flaw is an "off by one" error, where a design is nearly correct but some parameter or sequence is off by one. In this lab, symptoms of such flaws are simulation results showing shearing of the image (e.g. display of a rhombus instead of a rectangle) or misplaced lines of the multi-color perimeter. Check that you have correctly applied the display timing parameters from Figure 6, that each line is the same number of pixels, and that each frame is an integer number of lines. You can take measurements in the simulation waveform window for confirmation.

Synthesizing and Implementing the Design

Synthesize and implement your design exactly as you did in the tutorial. Do not forget to check the reports. As a general practice, you will want to review all errors and warnings. These point to areas of concern that you should either address or justify. If the design fails one or more timing specifications the reports will indicate this is the case.

Test your design in hardware. Does the circuit behave as you expect? If it does not, seek assistance.

Customizing the Design for Demonstration

You must modify the vga_example test pattern code to display your first and last initial, somewhere in the center of the visible area. It can be any size you like, any distinguishable color you like, and does not need to be fancy. For example, in my own implementation, the finished result would display EC. You will do this by decoding combinations of the horizontal count and vertical count, and using that information to change the color output. Once you are confident the design works properly and your initials are legible, demonstrate your final result to the instructor.

Laboratory Hand-In Requirements

Once you have completed a working design, prepare for the submission process. You are required to demonstrate a working design. Within four hours of your demonstration, you are required to submit your

entire project directory in the form of a compressed ZIP archive. You must include a simulation-generated TIFF file showing the output from your customized design. Use WinZIP to archive the entire project directory, and name the archive lab4_yourlastname_yourfirstname.zip. For example, if I were to make a submission, it would be lab4_crabill_eric.zip. Then email the archive to the instructor. Only WinZIP archives will be accepted.

Given the amount of simulation in this lab, the ZIP archive may be extremely large. Please reduce the ZIP archive size by reviewing the ZIP archive contents and deleting any WDB, waveform data base, files in the simulation directories. Further, delete all TIFF files except for the one you intend to submit.

Demonstrations must be made on or before the due date. If your circuit is not completely functional by the due date, you should turn in what you have to receive partial credit.