

Lab 8

VGA Monitor Controller Design

Objectives

- To design and implement sequential control logic circuits using the Schematic Editor and Simulator.
- To learn how a VGA monitor works.
- To implement a VGA monitor controller circuit.
- To download your circuits onto the prototyping board and test it.

Laboratory Instructions

- Create a directory with your name on the C drive of your lab PC. Use this directory to create your project, store your results, bitstreams, etc. during the lab session.
- You can bring complete project files on a floppy disk and then use the **Copy Project** command from the Project Manager menu to copy it into the directory you created above.
- Alternatively, you can create a new project in your directory on the C drive and then copy your files to that new project directory. Remember to **Add** your .SCH file to the project.
- Perform functional simulation of your design and have it checked by your TA.
- Refer to [appendix A](#) for instructions on performing functional simulation.
- Refer to [appendix B](#) for instructions on how to download the circuit to the prototyping board.
- Test and demonstrate your circuit to your TA.
- Before you leave the lab please **remove** the files and directories that you created on your lab PC and leave our workplace clean and tidy.

Design Problems

Using the Xilinx Schematic Editor and Simulation Tools, design, simulate, and demonstrate on the prototyping board a VGA monitor controller that displays the following.

1. Use the VGA monitor controller circuit to display a red boarder around the screen.
2. Use the VGA monitor controller circuit to display a red boarder around the screen, and the two large block letters “CS”.

The operation of the VGA monitor is described below. The description is based on a 25MHz clock. The clock that you will be using is a 12MHz clock. Hence, you will need to make the appropriate changes. When you do the calculations for a 12MHz clock, you will see that your screen resolution will be 305 columns by 480 rows.

The monitor screen for a standard VGA format contains 640 columns by 480 rows of picture elements called pixels as shown in Figure 1. An image is displayed on the screen by turning on or off individual pixels. The monitor continuously scans through the entire screen turning on or off one pixel at a time at a very fast speed. The scan starts from row 0, column 0 at the top left corner, and moves to the right until it reaches the last column in the row. When the scan reaches the end of a row, it continues at the beginning of the next row. When the scan reaches the last pixel at the bottom right corner of the screen, it goes back to the top left corner of the screen, and repeats the scanning process again. In order to reduce flicker on the screen, the entire screen must be scanned 60 times per second or higher. During the horizontal and the vertical retraces, all the pixels are turned off.

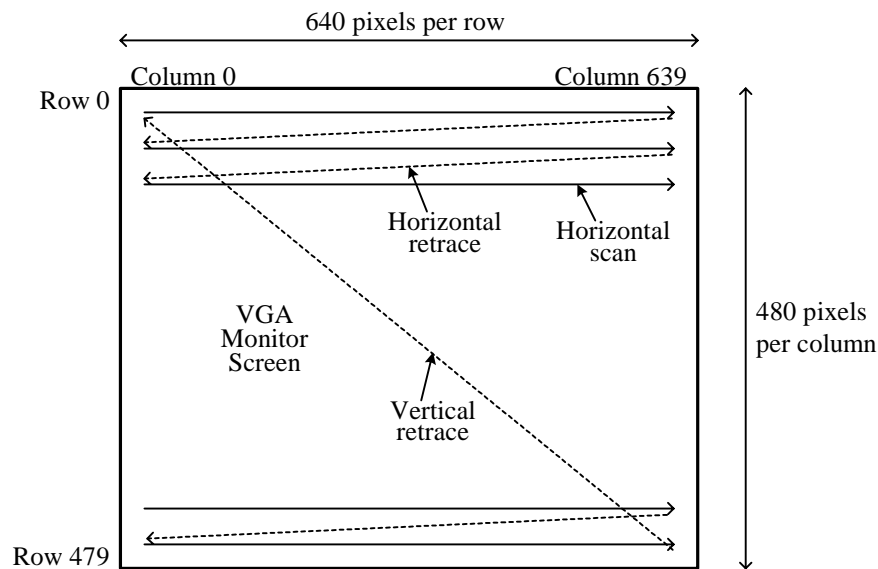


Figure 1. VGA monitor with 640 columns \times 480 rows. Scan starts from row 0, column 0, and moves to the right and down until row 479, column 639.

The VGA monitor is controlled by five signals: red, green, blue, horizontal synchronization, and vertical synchronization. The three color signals, referred to collectively as the RGB signal, are used to control the color of a pixel at a location on the screen. These three color signals are

connected such that they can individually be either turned on or off, hence each pixel can display only one of eight colors. In order to produce more colors, each analog color signal must be supplied with a voltage between 0.7 to 1.0 volts for varying the intensities of the colors. The horizontal and vertical synchronization signals are used to control the timing of the scan rate. The horizontal synchronization signal determines the time to scan a row, while the vertical synchronization signal determines the time to scan the entire screen. By manipulating these five signals, images are formed on the monitor screen.

The horizontal and vertical synchronization signals timing diagram is shown in Figure 2. When inactive, both synchronization signals are at a 1. The start of a row scan begins with the horizontal sync signal going low for 3.77 μsec as shown by region B in Figure 2. This is followed by a 1.79 μsec high on the signal as shown by region C. Next, the data for the three color signals are sent, one pixel at a time, for the 640 columns as shown in region D for 25.42 μsec . Finally, after the last column pixel, there is another 0.79 μsec of inactivity on the RGB signal lines as shown in region E before the horizontal sync signal goes low again for the next row scan. The total time to complete one row scan is 31.77 μsec .

The timing for the vertical synchronization signal is analogous to the horizontal sync signal. The 64 μsec active low vertical sync signal resets the scan to the top left corner of the screen as shown in region P, followed by a 1020 μsec high on the signal as shown by region Q. Next, there are the 480 row scans of 31.77 μsec each, giving a total of 15250 μsec as shown in region R. Finally, after the last row scan, there is another 450 μsec as shown in region S before the vertical sync signal goes low again to start another complete screen scan starting at the top left corner. The total time to complete one complete scan of the screen is 16784 μsec .

In order to get the monitor to operate properly, we simply have to get the horizontal and vertical synchronization signals timing correct, and then send out the RGB data for each pixel at the right column and row position. It turns out that it is fairly simple to get the correct timing for the two synchronization signals. Assuming that the clock frequency is 25.175MHz. The clock period is then $1 / 25.175 \times 10^6$, or about 0.0397 μsec per clock cycle. For region B in the horizontal synchronization signal, we need 3.77 μsec , which is approximately $3.77 / 0.0397 = 95$ clock cycles. For region C, we need 1.79 μsec , which is approximately 45 clock cycles. Similarly, we need 640 clock cycles for region D for the 640 columns of pixels, and 20 clock cycles for region E. The total number of clock cycles needed for each row scan is, therefore, 800 clock cycles. Notice that with a 25.175MHz clock, region D requires exactly 640 cycles, giving us the 640 columns per row. Hence, a different clock speed will produce a different screen resolution.

The vertical timings are multiples of the horizontal cycles. For example, region P is 64 μsec , which is approximately two horizontal cycles (2×31.77). The calculation for region R is $15250\mu\text{sec} / 31.77\mu\text{sec} = 480$. Of course it has to be exactly 480 times, since we need to have 480 rows per screen.

In order to get the VGA monitor controller to work properly, we simply have to generate the horizontal and vertical sync signals timing correctly. In addition, to be able to turn on a specific pixel, the circuit needs to keep track of the current column within the D region, and the current row within the R region of the scan. The controller will also output the current column and row counts. To make a particular pixel green for example, you need to test the values of the column

and row counts. If they are equal to the location of the pixel that you want to turn on, then you assert the green signal, and that pixel will be green.

The circuit for the VGA monitor controller is shown in Figure 3.

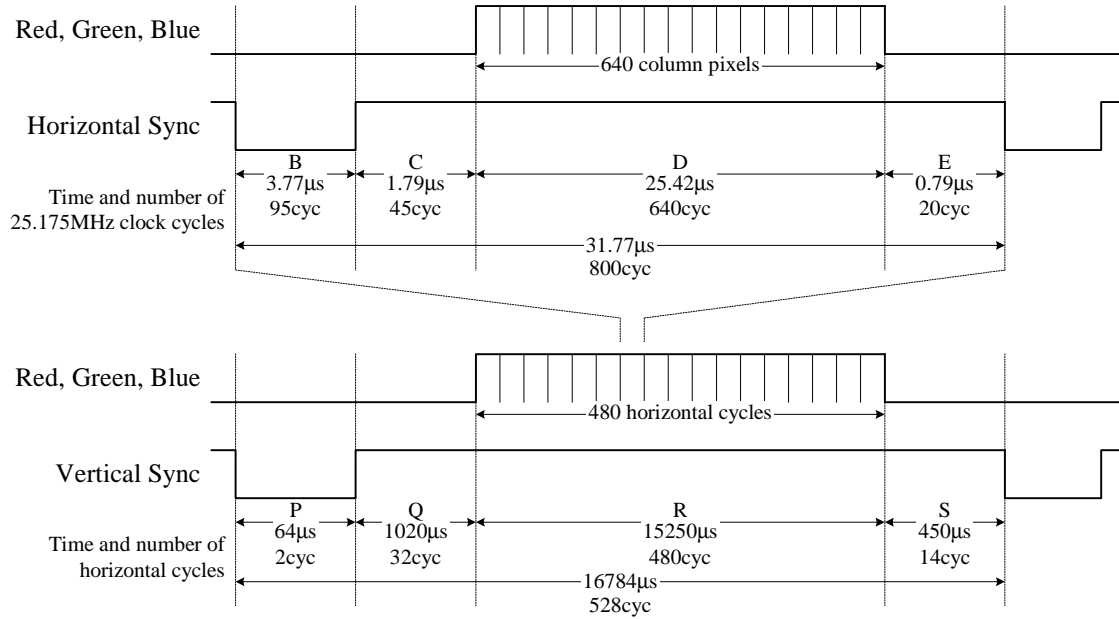


Figure 2. Horizontal and vertical synchronization signals timing diagram for a 25.175MHz clock.

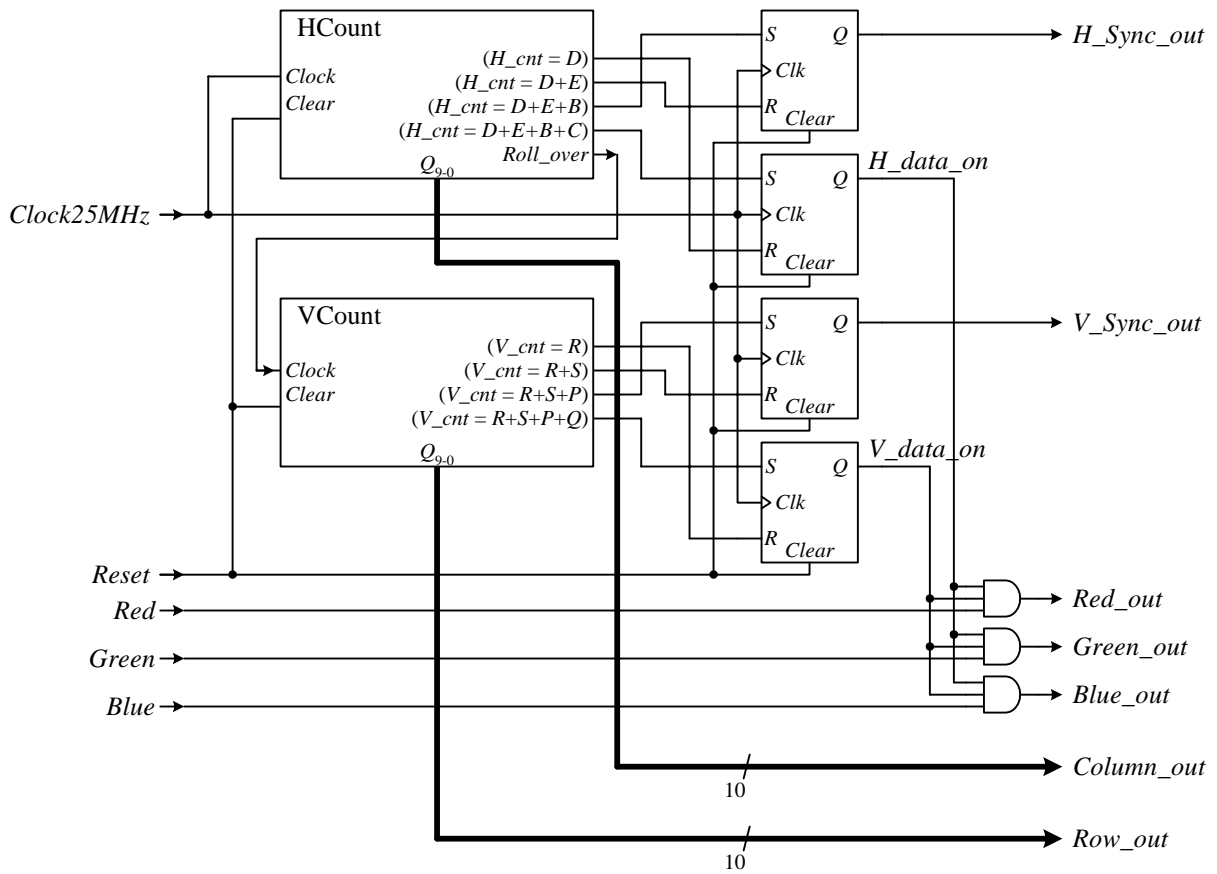


Figure 3. VGA monitor controller circuit.