SOC implementation of Universal Text Drivers for VGA using Nexys 3 Spartan 6 FPGA board

RAMANDEEP SINGH1, ANKIT KULSHRESTHA2, CHAITANYA MEHTA3
1 Assistant Professor, Dept. of EECE, ITM University, Gurgaon, INDIA
2-3 U.G. EECE Student, ITM University, Gurgaon, INDIA
1ramandeepsingh@itmindia.edu, 2ankitkulshrestha0912@gmail.com, 3chaitanyamehta@outlook.com

ABSTRACT
The project aims to demonstrate the usefulness of FPGA’s by using it as a device to drive an analog VGA output. The project by the use of novel methods to produce a string of text when connected to a monitor. The string is produced by using the character generator used in the program to tell the VGA to light specific pixels.

Keywords: Verilog, VGA, Spartan 6, Nexys 3, Text, Drivers, Monitor

1. INTRODUCTION
The utility of FPGA boards1,1 in general are that they offer a relatively cheap and easy means for engineers to test their ideas. Besides providing a very handy hardware solution they usually come bundled up with their own design suites through which many ideas can be tested out before they are actually implemented in some product. We chose Nexys 3 Spartan 6 FPGA board primarily because it offered a VGA port with 8-bit colour and the two standard sync signals, 100 MHz frequency oscillator and 16MByte of RAM. These specs were more than sufficient for our project and apart from the high processing speed we also got a Xilinx design suite to write the program in.

Interfacing on surface sounds a very trivial task, but when it is actually implemented, it becomes difficult since two different pieces of electronics must know and interact with each other to produce a coherent output. There were many challenges that we had to overcome in this project some of them being clock mismatch or code sections running faster than expected, or sometimes producing a blank on the screen, countering these challenges was a very rewarding and productive experience and that was precisely the reason why the project was chosen by us.

2. BASICS OF VGA
To drive a VGA monitor only 5 signal HSYNC, VSYNC Red, Green and Blue are needed1. HSYNC and VSYNC are synchronisation signals and synchronise start of new line and frame respectively. These are digital signals and have a value of 0v or 5v. Red, Green and Blue signal the monitor the colour of the pixel1. These are analog signals with range 0v to 0.7v (0V for black level, 0.7V for full colour intensity)2. The resolution at which the monitor displays is decided by the timings used to generate HSYNC and VSYNC. Below are given the timings used for 640x480 display resolution3.

Table 1. Timings for 640x480 resolution.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Horizontal Sync</th>
<th>Vertical Sync</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Time</td>
<td>Clocks</td>
</tr>
<tr>
<td>T_{s}</td>
<td>Sync Pulse</td>
<td>32µs</td>
<td>800</td>
</tr>
<tr>
<td>T_{disp}</td>
<td>Display Time</td>
<td>25.6µs</td>
<td>640</td>
</tr>
<tr>
<td>T_{pw}</td>
<td>Pulse Width</td>
<td>3.84µs</td>
<td>96</td>
</tr>
<tr>
<td>T_{fp}</td>
<td>Front Porch</td>
<td>640ns</td>
<td>16</td>
</tr>
<tr>
<td>T_{bp}</td>
<td>Back Porch</td>
<td>1.92µs</td>
<td>48</td>
</tr>
</tbody>
</table>

3. BLOCK DIAGRAM
The block diagram4 depicts the working of our project. The source clock is the on board clock (site V104) and has a frequency of 100 MHz. The pixel clock (of frequency 25 MHz) is derived using the divide by 4 counter which is fed by the 100 MHz clock. The counter takes four clocks to roll over and whenever it rolls the output of 25 MHz line is inverted, thus the pixel clock is obtained for a resolution of 640x480 with a refresh rate
of 60Hz. It must be noted that for different resolution a different pixel clock is required. The HSYNC and VSYNC pulse are generated by using Horizontal counter (HC) and Vertical Counter (VC). The counters also tell which pixel is currently being dealt with. A basic framework of character generator is being shown below.

![Block Diagram of VGA operation](image)

**Fig.2. Block Diagram of VGA operation.**

```plaintext
8'b00000000, // 0
8'b00000001, // 1
8'b00010000, // 2
8'b00111000, // 3
8'b01011000, // 4
8'b10000110, // 5
8'b11000010, // 6
8'b11111110, // 7
8'b11100010, // 8
8'b11001010, // 9
8'b10000110, // a
8'b10000000, // b
8'b00000000, // c
8'b00000001, // d
8'b00000010, // e
8'b00000011); // f
```

Using HC, VC and Font Map the Digital to Analog Converter assigns different RGB values to individual pixels. The pixels corresponding to 0 are set Black while RGB value of those corresponding to 1 are set as per desired colour.

4. **FLOW CHART**

![Flow chart for character generation](image)

**Fig.1. Flow chart for character generation.**

The given flowchart provides a method to generate a character on the screen using VGA. The given method may be expanded to generate other characters as well.
5. ACTUAL PICTURES
The intended output was to display a multi case string with different colours. The given picture depicts the obtained output.

Fig.2. Capture of Output
Fig.3. Kit and Connections
The shown kit is a Nexys 3 Spartan 6 board. The board is powered by a USB connected to a terminal and utilizes the female DB-15 port. The 100 MHz clock is on board.

6. RTL VIEW

Fig.4. RTL View of VGA
The implemented design has one input pin clock_100 and eight output pins red2, red1, red0, green2, green1, green0, blue1, blue0, HSYNC and VSYNC.

7. SIMULATION RESULT
7.1 HSYNC Timings

Fig.5. HSYNC Timing Diagram.

7.2 VSYNC Timings

Fig.6. VSYNC Timing Diagram
Table 2. Obtained Timings for HSYNC and VSYNC

<table>
<thead>
<tr>
<th>Parameter</th>
<th>HSYNC</th>
<th>VSYNC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sync pulse duration</td>
<td>3.7 µs</td>
<td>0.06392 ms</td>
</tr>
<tr>
<td>Complete duration</td>
<td>31.82 µs</td>
<td>16.6192 ms</td>
</tr>
</tbody>
</table>

These timings are found to be in agreement with the specification mentioned earlier and can be used to display at a resolution of 640x480 @60Hz.

8. ADVANTAGES OF VGA
A VGA display unlike previously used Alphanumeric LCD is made of one continuous block of many pixels rather than many discrete blocks of fewer segments, this gives VGA to display a wide variety of content which was not possible earlier, including characters of different sizes, font, language and moving objects. VGA also adds the ability to set RGB colour of individual pixels which gives the ability to display not only monochromatic but also coloured content.

9. APPLICATIONS
The VGA Text Driver can be used in several places like music players, information display systems, kiosks, etc.

10. LIMITATIONS
The given method produces the desired output, but it does have some limitations. For example, to generate a long string we have to generate a longer character map of ASCII values. This would result in a larger source code. In addition, the method is not dynamic i.e. only pre-coded strings can be displayed. Improvement upon the design is left for future iterations of the project.

11. CONCLUSION AND FUTURE SCOPE
Text Driver for VGA were successfully implemented using Xilinx ISE design suite 14.7 and the Nexys3 Spartan-6 FPGA Board. The given method works well for displaying a pre-coded string on a VGA monitor. The basic design may be implemented using a keyboard to display a user entered data. Also, the multicolour text according to the user choice may be added.

ACKNOWLEDGEMENTS
I would like to thank ITM, University Gurgaon for allowing me to use their DSD Lab and Nexys 3 kits.

REFERENCES
[1] Build a VGA monitor controller by Enoch Hwang, La Sierra University.
[12] Van-Huan Tran, Xuan-Tu Tran – “An Efficient Architecture Design for VGA Monitor Controller“