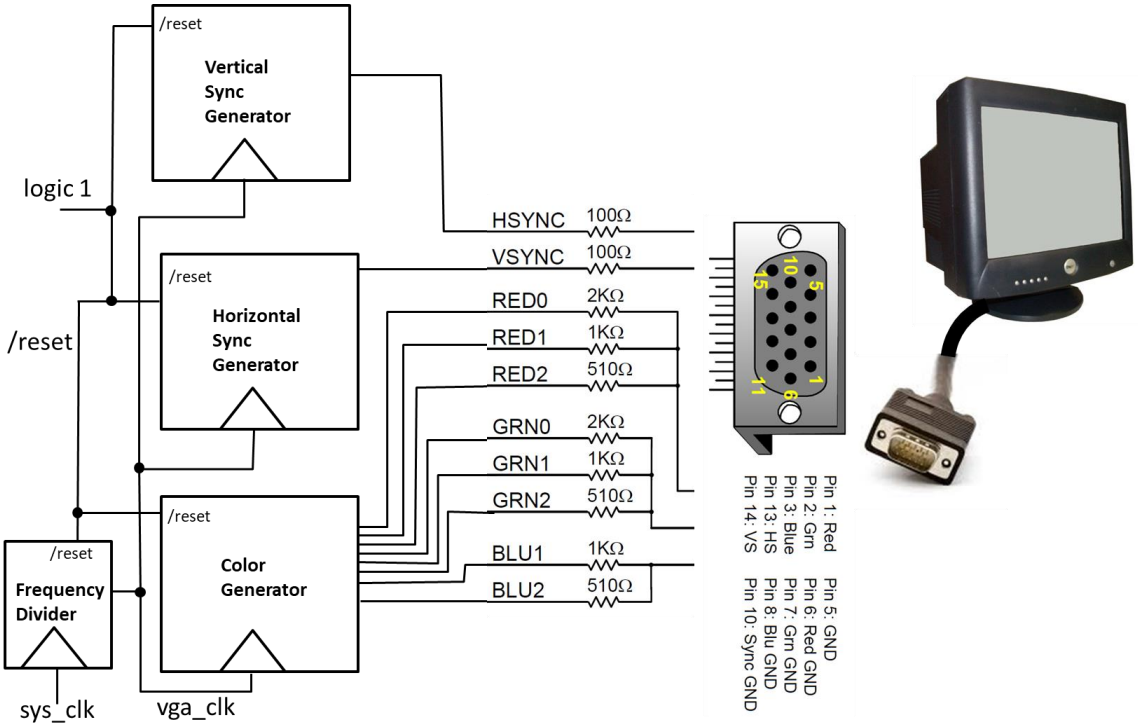
LAB7: SIMPLE VGA DRIVER TO DISPLAY 256 DIFFERENT COLORS

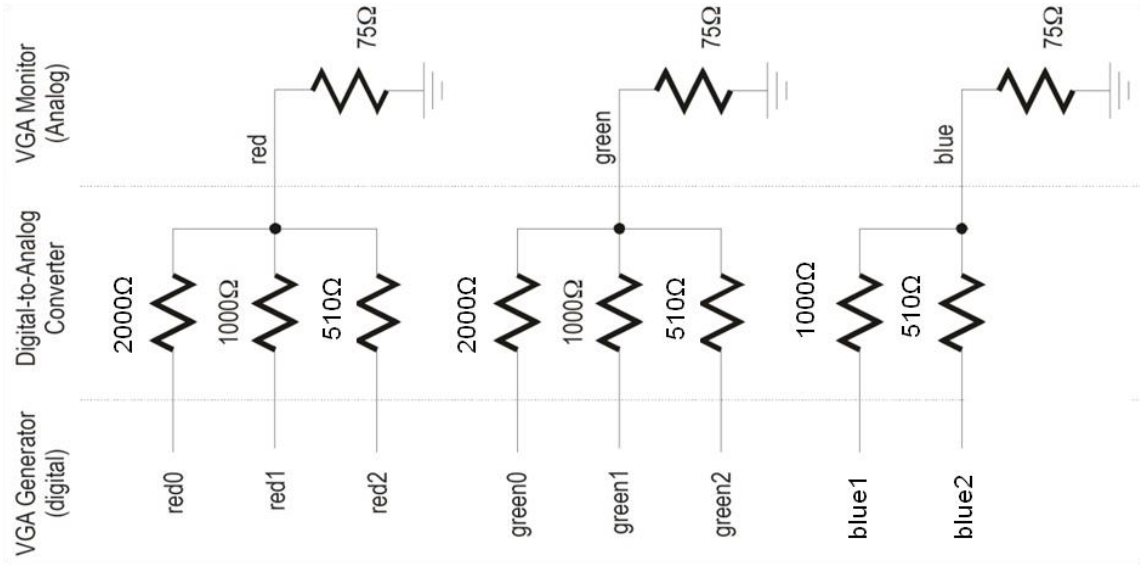
The purpose of this lab is to design a VGA driver to display 256 different colors on a monitor. Two timing signals are generated in this system, vsync and hsync to synchronize the plotting of vertical and horizontal pixels, respectively. The color data of each pixel is generated using an 8 bit counter whose first 3-bits are for red, next 3-bits are for green and last 2-bits are for blue color data while the synchronization signals are generated. This way, as the counter counts, 256 different color combinations are displayed on the screen one after another. This counter is implemented on the FPGA board. Overall schematic for the system is shown below.



VGA Driver

FPGA can generate a video signal to display it on a VGA monitor. The FPGA board has three bits of red, three bits of green, and two bits of blue color information connected to a simple Digital to Analog Converter (DAC) of resistor-ladder type as shown in the figure below. This provides a palette of 2^3 x 2^3 x 2^2 = 256 different colors. The three analog output wires (for red, green and blue color) of this simple DAC are sent to the RGB inputs of a VGA monitor using three pins of the connectors. The FPGA must also generate horizontal and vertical synchronization pulses [9]. These five signals (including the two synchronization signals) typically drive an electron gun that emits electrons, which paint one primary color at a point on a Cathode Ray Tube (CRT) monitor. Signal levels between 0 (completely dark) and 0.7 V (maximum brightness) control the intensity of each color component. This makes the color of a dot (or pixel) on the monitor screen [1]. Even though, display operation of an LCD monitor is different than the CRT one,

same signals and methods are also used for LCD monitors that are driven through VGA ports.



Digital to Analog Conversion done on the FPGA board and signals going to the VGA port of a monitor.

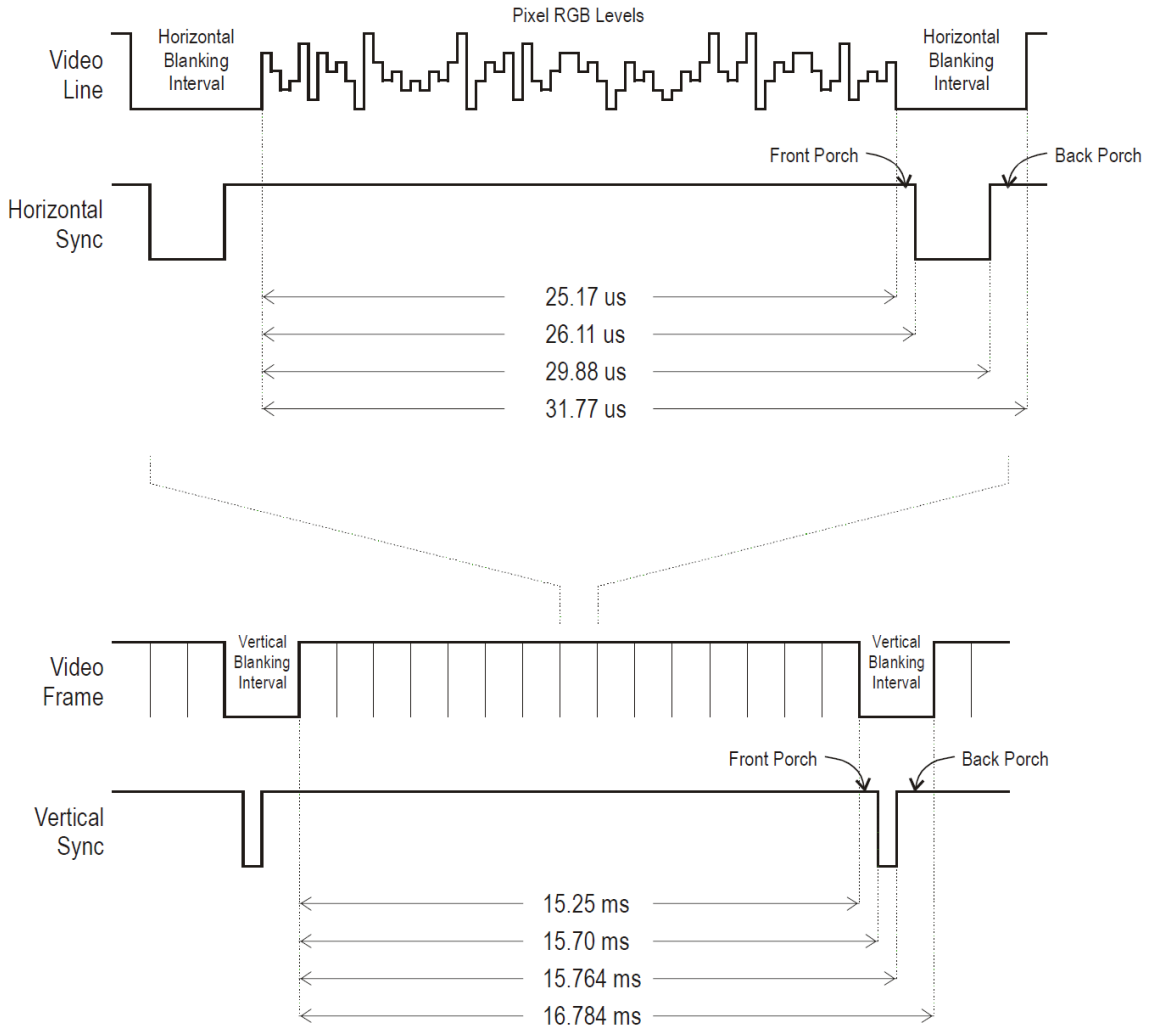
VGA Signal Timing

An image (or frame) on a monitor screen is composed of h lines. Each line has w pixels. VGA frame size is shown as w x h with typical sizes of 640 x 480, 800 x 600, 1024 x 768 and 1280 x 1024. In this lab, we aim for 640x480 size. In order to paint a frame, there are deflection circuits in the CRT monitor that move the beams of electrons from left-to-right and top-to-bottom across the screen (totally three beams for three fundamental colors). These circuits require two synchronization signals in order to start and stop the beams at the right time so that a line of pixels is plotted across the monitor and the lines stack up from the top to the bottom to form an image. An example for the timing of the VGA synchronization signals is shown in the figure below [1].

Each synchronization signal is composed of two periodic negative pulses. Negative pulses on the horizontal synchronization signal mark the start and end of a line and ensure that the monitor displays the pixels between the left and right edges of the visible screen area. Only certain scanning angle range of the electron beam is visible to viewers. The start and end information of this visible scanning angle of the beam movement are given by the synchronization signals. The pixels are sent on the red, green and blue signal lines within a 25.17 μs window. After this, an interval called front porch of 0.94 μs is inserted before the horizontal synchronization signal pulses stays low for 3.77 μs. After an interval called back porch of 1.89 μs, the next line of pixels begins. Therefore, a single line of pixels occupies 25.17 μs of a 31.77 μs interval. The red, green and blue signals are blanked during the 6.6 μs interval comprised of the front porch, synchronization pulse and back porch [1].

In a similar fashion, negative pulses on the vertical sync signal mark the start and end of a frame made up of video lines and ensure that the monitor displays the lines between the top and bottom edges of the visible monitor screen. The lines are displayed within a

15.25 ms window. After this, a front porch interval of 0.45 ms is inserted before the vertical sync signal pulses stay low for 64 µs. After a back porch interval of 1.02 ms, the next frame begins. Therefore, a single frame of pixels occupies 15.25 ms of a 16.784 ms interval (almost 60 Hz refresh rate). The red, green and blue signals are blanked during the 1.534 ms interval comprised of the front porch, synchronization pulse and back porch [1].



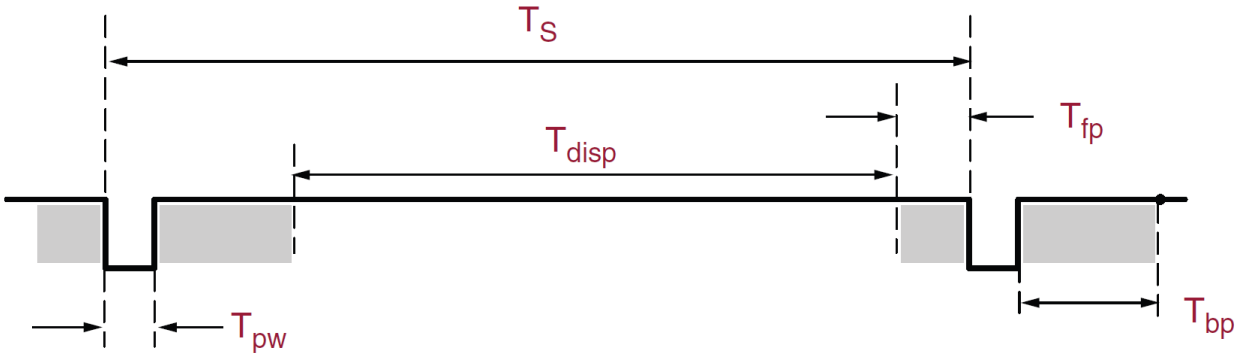
VGA Signal Timing [1].

The timing values given above are just part of one possible solution to drive a VGA monitor. Another possible timing summary is given in the table below. Timing parameters used in this table is explained in the figure below this table. The given timing summary is for a 25 MHz pixel clock and 60 Hz ± 1 refresh rate. These timing parameters are found by observing various VGA monitors. You are advised to use these parameters in your design. You may divide the 100 MHz system clock to generate the 25 MHz pixel clock. Generally, a counter clocked by this pixel clock controls the horizontal timing. Content of this counter is used by logic gates to generate the horizontal syncronization signal. It tracks the current pixel display location on a given row. A separate counter with the same structure as the horizontal syncronization counter tracks the vertical timing. The vertical synchronization counter increments with each

horizontal synronization pulse. Logic gates are used to generate the vertical synhronization signal from the content of this counter. This counter tracks the current display row [2]. All sequential components operate synchronously.

Timing summary for a 640x480 VGA, 25 MHz pixel clock [2]

25 MHz pixel clock and 60 Hz ± 1 refresh						
Symbol	**Parameter**	**Vertical Sync**			**Horizontal Sync**	
		Time	**Clocks**	**Lines**	**Time**	**Clocks**
T_S	Sync pulse time	16.7ms	416,800	521	32 µs	800
T_{DISP}	Display time	15.36ms	384,000	480	25.6 µs	640
T_{PW}	Pulse width	64 µs	1,600	2	3.84 µs	96
T_{FP}	Front Porch	320 µs	8,000	10	640 ns	16
T_{BP}	Back Porch	928 µs	23,200	29	1.92 µs	48



Explanations of timing parameters in the table given above [2]

Overall Design

Horizontal and vertical synchronization signals are simply signals with just two-negative-pulses. Vertical and horizontal synchronization generator circuits generate these two pulse signals. These circuits are identical to each other except for the values that determine the pulse timing. In your design you may want your horizontal synchronization generator to communicate with your vertical synchronization generator. For example, the horizontal sync generator may output a single-cycle signal coincident with the leading edge of the horizontal sync pulse to the vertical sync generator. This signal may be

connected to the clock-enable of the vertical sync generator so it only updates its timing counter once per line of pixels. Furthermore, a single-cycle signal may be generated in the vertical sync generator to be used as an end-of-frame indicator to the color generator to reset its contents so the VGA generator starts from a completely cleared state on every frame [1].

Color generator is simply an 8-bit binary up-counter. The first three bits of the counter form the RED0, RED1 and RED2 signals. The next three bits form the GREEN0, GREEN1 and GREEN2 signals. Last two bits form BLUE0 and BLUE1 signals. The counter rolls back when it reaches the maximum count. This way, as the counter counts up, all possible color combinations are displayed on the screen. It is not important from which count the counter starts with respect to the synchronization signals. The counter simply creates data to plot on the monitor. Otherwise, to display a meaningful image on the monitor, 640x480x3x3x2=5,529,600 bits have to be presented for each frame (every 0,0167 second). This can be done by writing the image data to an SRAM and reading from it. However, since this would complicate the laboratory assignment, we form display information by using a continuously running counter.

The clock signal for the timing signals and counters can be generated from system clock (100MHz, pin V10) of the Nexys3 board.

Unfortunately, we cannot use push button switches on the Nexys3 board due to remote lab access. Hence, we cannot reset the system and put the circuit into a known state. Therefore, make sure that you also account for “don’t care” states. If your design wakes in an unknown state (D flip-flop values which do not form desired next states), they will generate valid next state values in couple clock cycles.

Your VHDL code for this design should have the following structure. You should complete the architecture. Your VHDL code can use structural and/or dataflow and/or behavioral style VHDL.

```
-- Synthesizable Simple VGA Driver to Display All Possible Colors
-- EE240 class Bogazici University
-- Implemented on Xilinx Spartan VI FPGA chip

LIBRARY ieee ;
USE ieee.std_logic_1164.all ;
USE ieee.std_logic_signed.all ;
USE ieee.std_logic_arith.all ;
USE ieee.std_logic_unsigned.all;

entity ee240_vgadriver is
        port (
                board_clk: in std_logic;
                vsync: out std_logic;
                hsync: out std_logic;
                red: out std_logic_vector(2 downto 0);
                green: out std_logic_vector(2 downto 0);
                blue: out std_logic_vector(1 downto 0));
end;

architecture arch_vga_driver of ee240_vgadriver is
```

```
--- board_clk comes from the 100 MHz board clock

--vsync is connected to VGA-VSYNC#
--hsync is connected to VGA-HSYNC#
--red(0) is connected to VGA-RED0
--red(1) is connected to VGA-RED1
--red(2) is connected to VGA-RED2
--green(0) is connected to VGA-GRN0
--green(1) is connected to VGA-GRN1
--green(2) is connected to VGA-GRN2
--blue(0) is connected to VGA-BLU1
--blue(1) is connected to VGA-BLU2

end arch_vga_driver;
```

Pin Assignment

After successfully completing the design, compilation, ISim simulations and synthesis, you should show that your design work on the FPGA board. By looking at the manual of the Nexys3 Board, we can find which FPGA pins are connected to which components. The following is the ucf file for this lab including the explanations for the pin assignments.

```
#
# Pin assignments for the Nexys3 Board.
#
net board_clk             loc=V10;  # 100 MHz board clock

net vsync              loc=P7;  # VGA-VSYNC
net hsync              loc=N6;  # VGA-HSYNC
net red<0>             loc=U7; # VGA-RED0
net red<1>              loc=V7; # VGA-RED1
net red<2>              loc=N7; # VGA-RED2
net green<0>            loc=P8; # VGA-GRN0
net green<1>            loc=T6; # VGA-GRN1
net green<2>            loc=V6; # VGA-GRN2
net blue<0>            loc=R7; # VGA-BLU1
net blue<1>             loc=T7; # VGA-BLU2
```

Preparation (Prelab)

For Frequency Divider

- Write a VHDL code for these components.
- Run ISim and verify that they function as expected by running test sequences.

For Horizontal and Vertical Synchronization Signal Generators

- Write a VHDL code for this component.
- Run ISim and verify that it functions as expected by running a few test sequences.

For the Color Generator

- Write a VHDL code for a resettable 9-bit up-counter.
- Run ISim and verify that it functions as expected by running a few test sequences.

For the Overall Design

- Combine the design components under arch_vga_driver architecture and run ISiM on it and verify that it functions as expected by running a few test sequences.

In Lab

- Use Xilinx ISE to synthesize the design based on Xilinx Spartan VI family. Use either the given ucf file or enter pins manually to make pin assignment.
- Generate the bit file and upload it to your FPGA.
- On the hardware show that your design works

References

[1] http://www.xess.com/appnotes/an-101204-vgagen.pdf "XESS VGAgen." (accessed 2021).

[2] http://www.xilinx.com/support/documentation/boards_and_kits/ug230.pdf (pages 56-58) "Xilinx Boards and Kits." (accessed 2021).