Course Name: EMBEDDED SYSTEMS I / III

Course Number and Section: **14:332:493:03**

Year: Spring 2021

Lab Report # 4: Now You See It, Now You Don't

Lab Instructor: Philip Southard

Student Name and RUID: Mya Odrick 180004110

Date Submitted: 04/08/2021

GitHub Link:
https://github.com/embedded-systems-1-spring-2020/lb4-now-you-see-it-now-you-don-t-mmo91

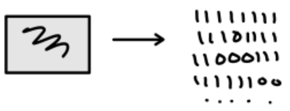
Demo: 04/03/2021

Purpose/Objective:

The purpose of this lab is to utilize the Vivado IP libraries to display an image on a monitor through a VGA and HDMI connection. In this lab, we convert an image into a COE file which stores the pixel data of the image into a format that is accessible through the Block Memory Generator IP. The memory block is used to send each pixel to the "pixel pusher" to decode the corresponding color. This information is sent to either a VGA or HDMI output.

Theory of Operation:

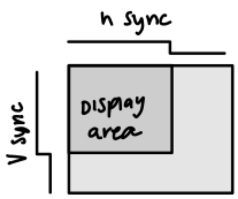
STEP 1: convert image into pixel data



Step 2: Decode pixel data into red, green, blue

$$\underbrace{111}_{red}\underbrace{110}_{green}\underbrace{01}_{blue}$$

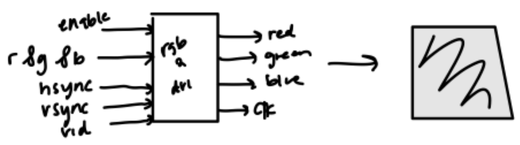
Step 3: send data to VGA port, following horizontal/vertical sync



Step 4: Convert VGA 3:3:2 rgb values to 8:8:8

red3 = "111" → red8 = "11111111"

Step 5: Output to HDMI using rgb2dvi IP Library

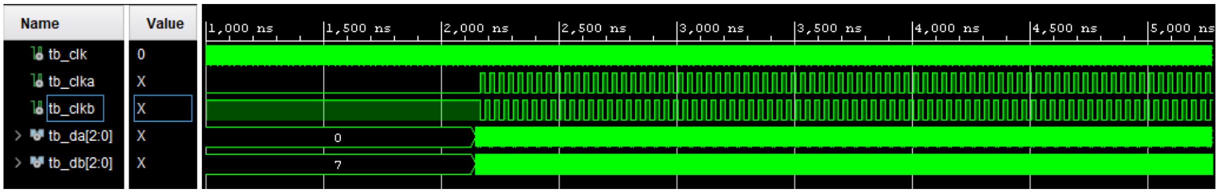


Simulation Waveforms:

1. Part 1

No testbench was created

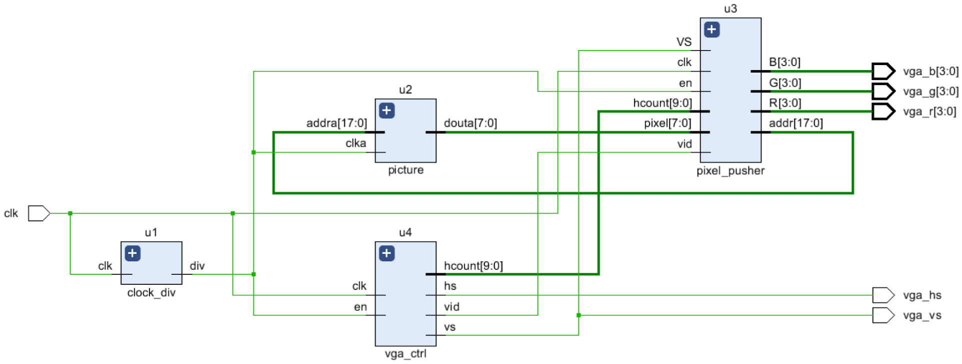
2. Part 2



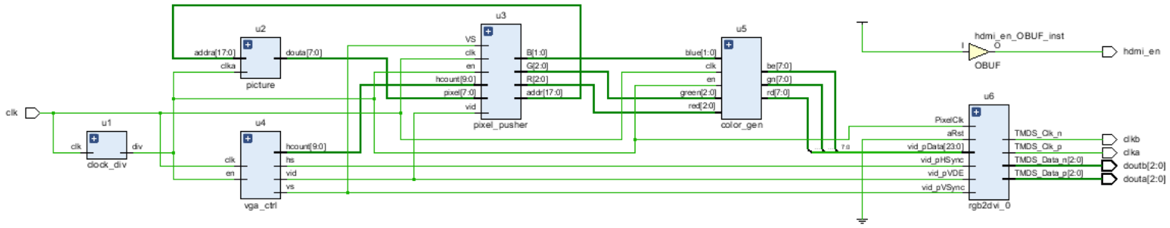
Vivado Schematics:

a) Vivado Elaboration Schematic

i) Part 1

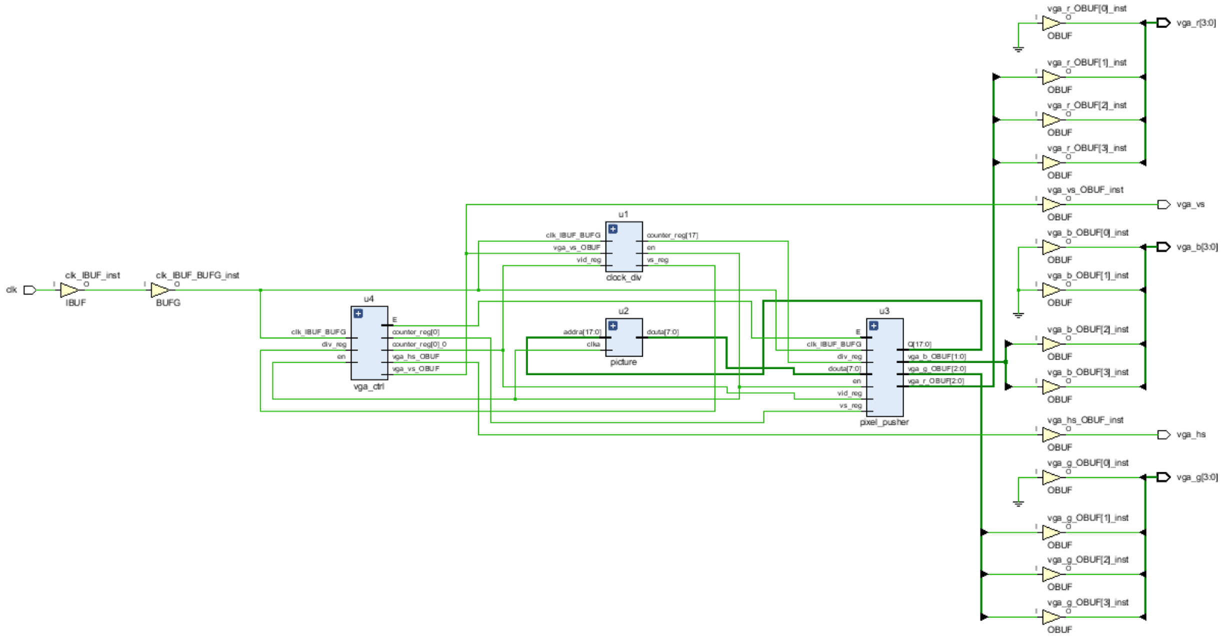


ii) Part 2

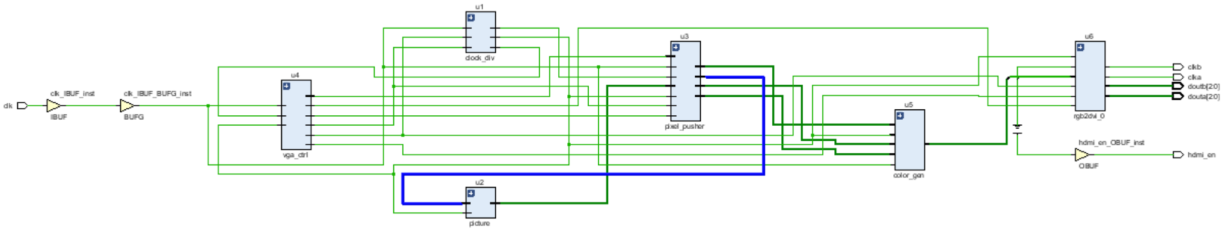


b) Vivado Synthesis Schematic

i) Part 1

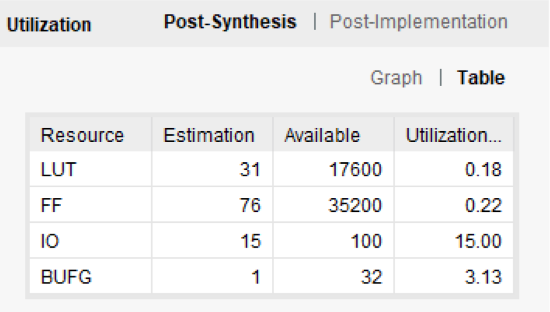


ii) Part 2



c) Post- Synthesis Utilization Table

i) Part 1

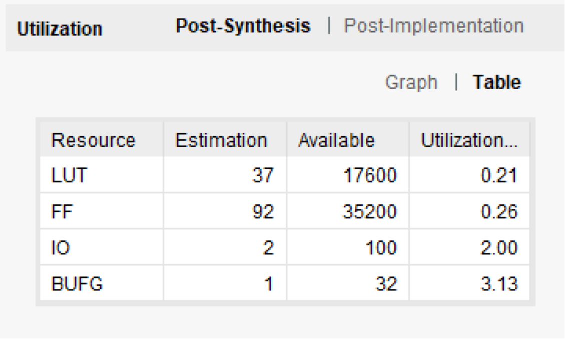


Utilization   Post-Synthesis | Post-Implementation

Graph | Table

Resource	Estimation	Available	Utilization...
LUT	31	17600	0.18
FF	76	35200	0.22
IO	15	100	15.00
BUFG	1	32	3.13

ii) Part 2



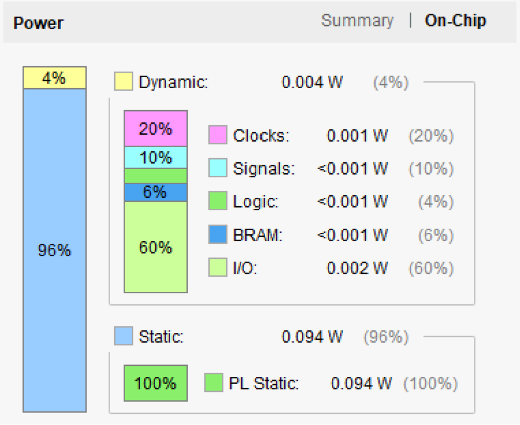
Utilization   Post-Synthesis | Post-Implementation

Graph | Table

Resource	Estimation	Available	Utilization...
LUT	37	17600	0.21
FF	92	35200	0.26
IO	2	100	2.00
BUFG	1	32	3.13

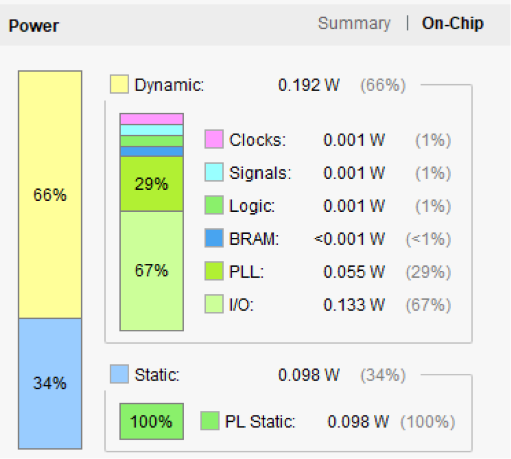
d.) On-Chip Power Graphs

i) Part 1



Dynamic:	0.004 W	(4%)
Clocks:	0.001 W	(20%)
Signals:	<0.001 W	(10%)
Logic:	<0.001 W	(4%)
BRAM:	<0.001 W	(6%)
I/O:	0.002 W	(60%)
Static:	0.094 W	(96%)
PL Static:	0.094 W	(100%)

ii) Part 2



Dynamic:	0.192 W	(66%)
Clocks:	0.001 W	(1%)
Signals:	0.001 W	(1%)
Logic:	0.001 W	(1%)
BRAM:	<0.001 W	(<1%)
PLL:	0.055 W	(29%)
I/O:	0.133 W	(67%)
Static:	0.098 W	(34%)
PL Static:	0.098 W	(100%)

Constraint File:

1. I used the VGA Pmod in ports JD and JE. So, I assigned T14, T15, P14, R14, U14, U15, V17, V18, V12, W16, J15, and H15 to represent the rgb colors and V13 and U17 to represent the horizontal and vertical sync.
2. I used the HDMI signals designated on the zybo board.

Conclusion:

During this lab, I became familiar with Vivado's IP Library. The library allows us to use premade modules without having to develop the logic. At first, I tried to write my own logic for the conversion between the vga rgb to the hdmi rgb. This became very complicated, but I was able to figure out how to import the rgb2dvi module and import the top file as a component into my "image_top". During the demo, I was able to showcase my hdmi results on my local machine without any issues, however it didn't show up the same way on the vlab. We discovered that the way I converted from 3:3:2 rgb to 8:8:8 rgb was decoding incorrectly. This was fixed by converting the bit values through a bit conversion equation.

Follow Up:

This lab gave me more confidence as I'm beginning to work on the final project. Now that I am more familiar with Vivado's IP, I began looking into how I can use the IP Libraries for my pmods. While working on this lab, I've also noticed that I'm getting better at locating an issue in my logic and figuring out how to solve them.