Lab 4 - Now You See It, Now You Don't

Embedded Systems 1
Lab Report 4
Tim Petersen - trp87
4/8/2020

Part 1: vga_ctrl

Purpose

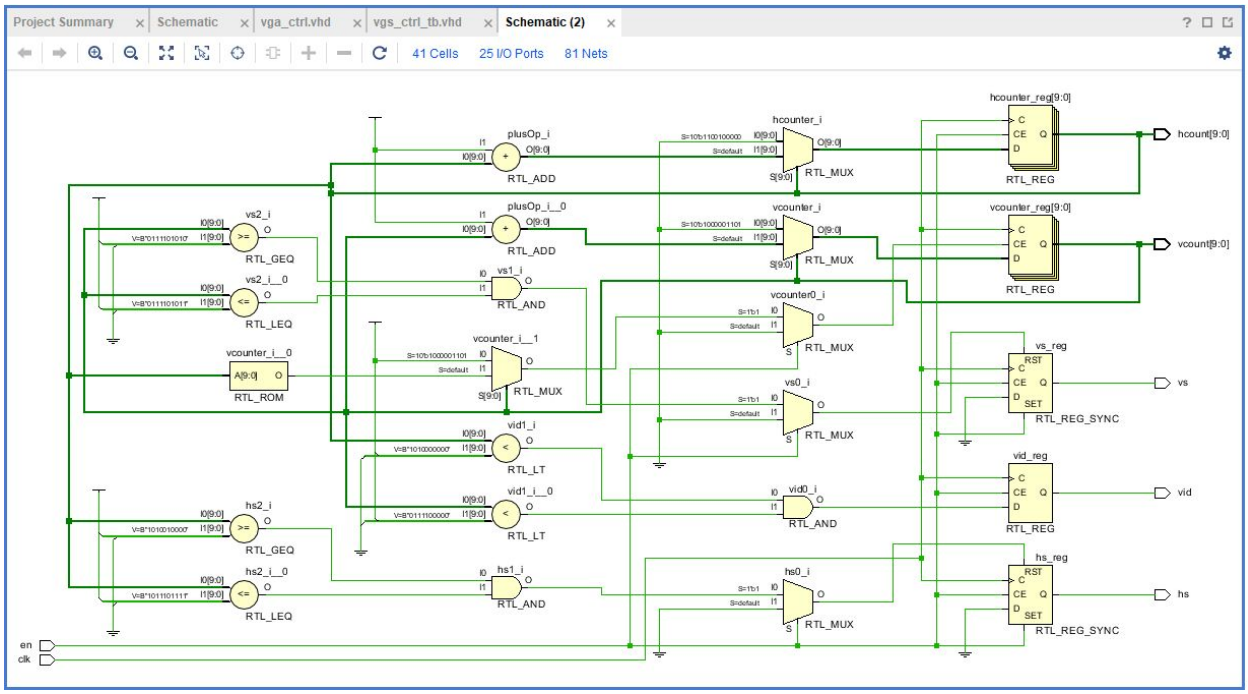
Create an entity called vga ctrl that takes as input a clock and a clock enable and produces the following outputs.

- hcount: a 10-bit std logic vector that is the value of the horizontal counter
- vcount: a 10-bit std logic vector that is the value of the vertical counter
- vid: a 1-bit signal that is 1 when the display should be on, otherwise it is 0
- hs: a 1-bit signal that is the 'HS' pulse
- vs: a 1-bit signal that is the 'VS' pulse

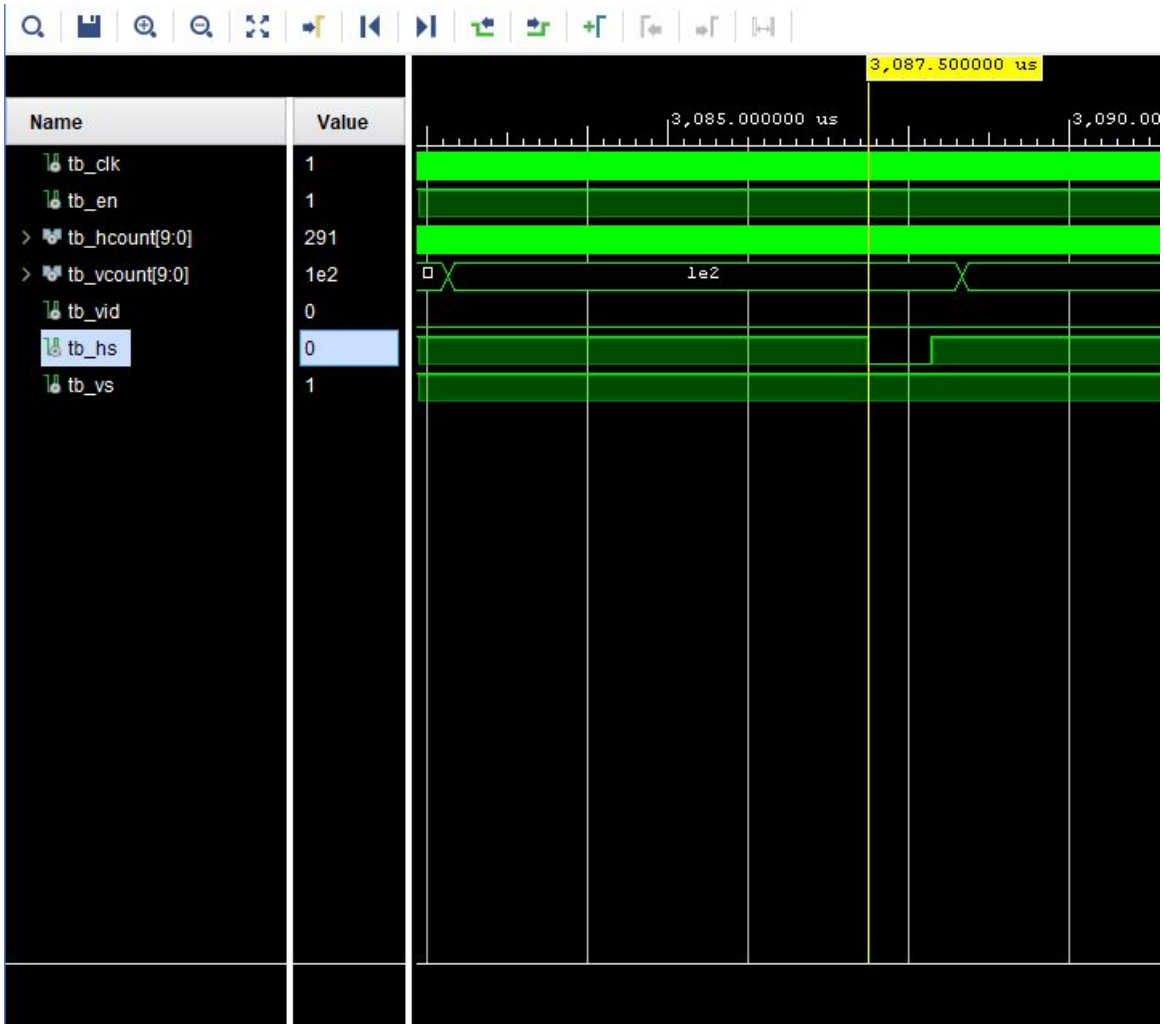
vga_ctrl.vhd

Code is on github: trp87-rutgers

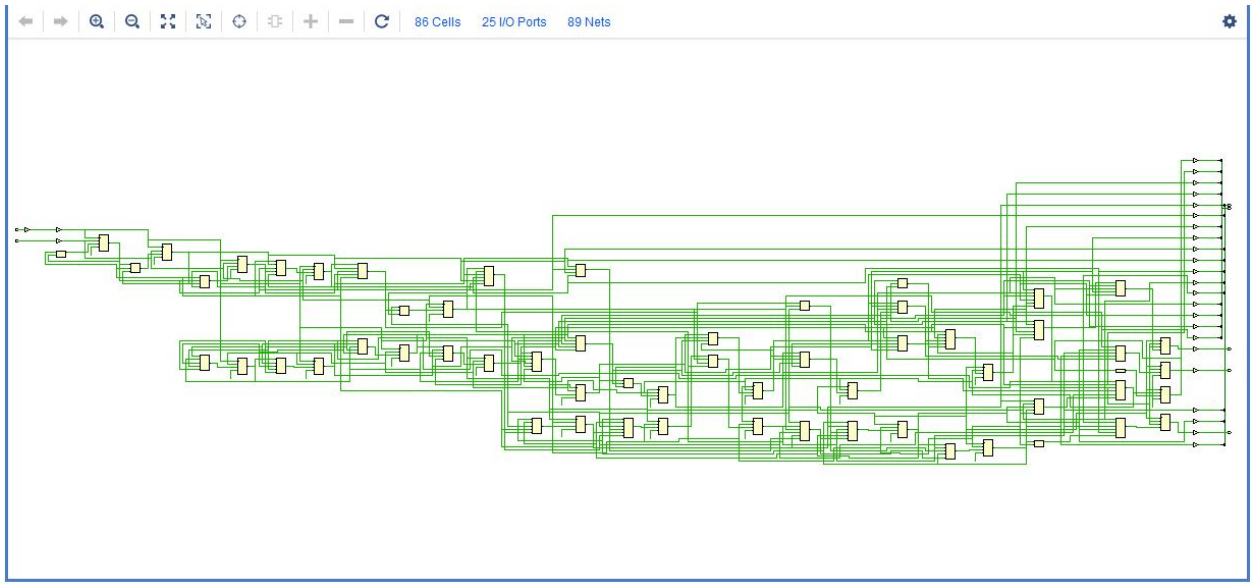
RTL Schematic



Simulation Waveform



Synthesis Schematic:



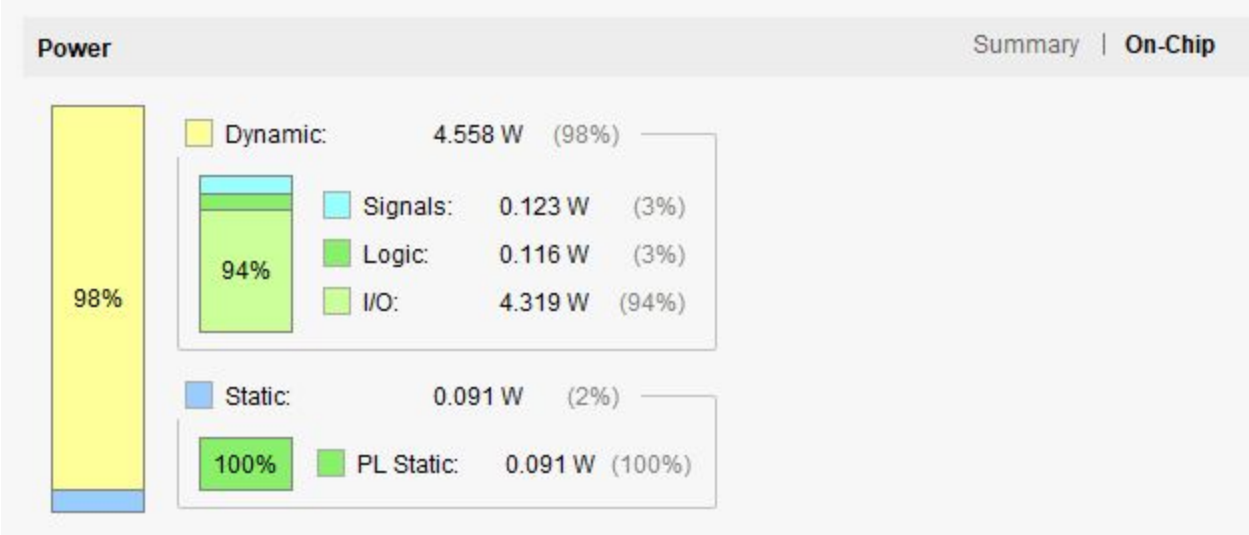
Post Synthesis Utilization Table

Utilization — **Post-Synthesis** | Post-Implementation

Graph | **Table**

Resource	Estimation	Available	Utilization %
LUT	27	41000	0.07
FF	23	82000	0.03
IO	25	300	8.33
BUFG	1	32	3.13

On Chip Power Graph



Part 2: pixel_pusher

Purpose

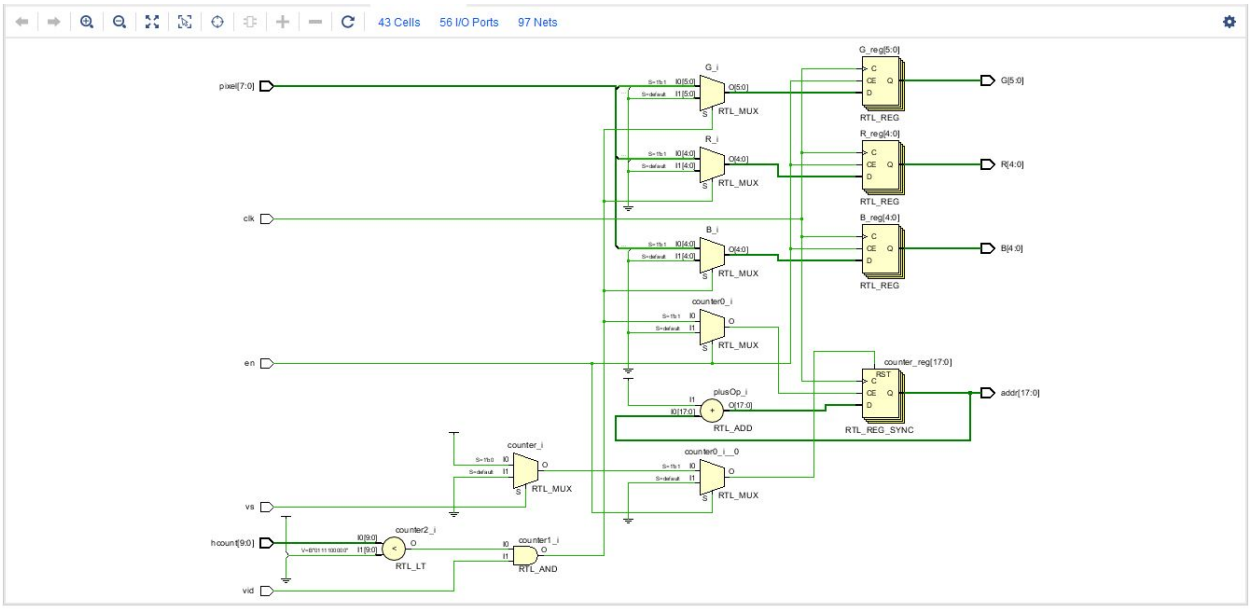
Create an entity called pixel pusher that takes as input a clock, a clock enable, a 1-bit VS, an 8-bit pixel signal, a 10-bit hcount signal, and a vid signal. It should output two 5-bit R and B signals and a 6-bit G signal, as well as an 18-bit addr. It should have the following behavior:

- It contains an internal 18 bit counter called addr with the following behavior:
    - Every clock tick when enable is 1, vid is 1, and hcount is less than 480, it increments. It resets synchronously when VS is 0
- Every clock tick when enable is 1, vid is 1, and hcount is less than 480…
    - R <= pixel(7 downto 5) & "00"
    - G <= pixel(4 downto 2) & "000"
    - B <= pixel(1 downto 0) & "000"
    - Otherwise, R, G, and B are 0.

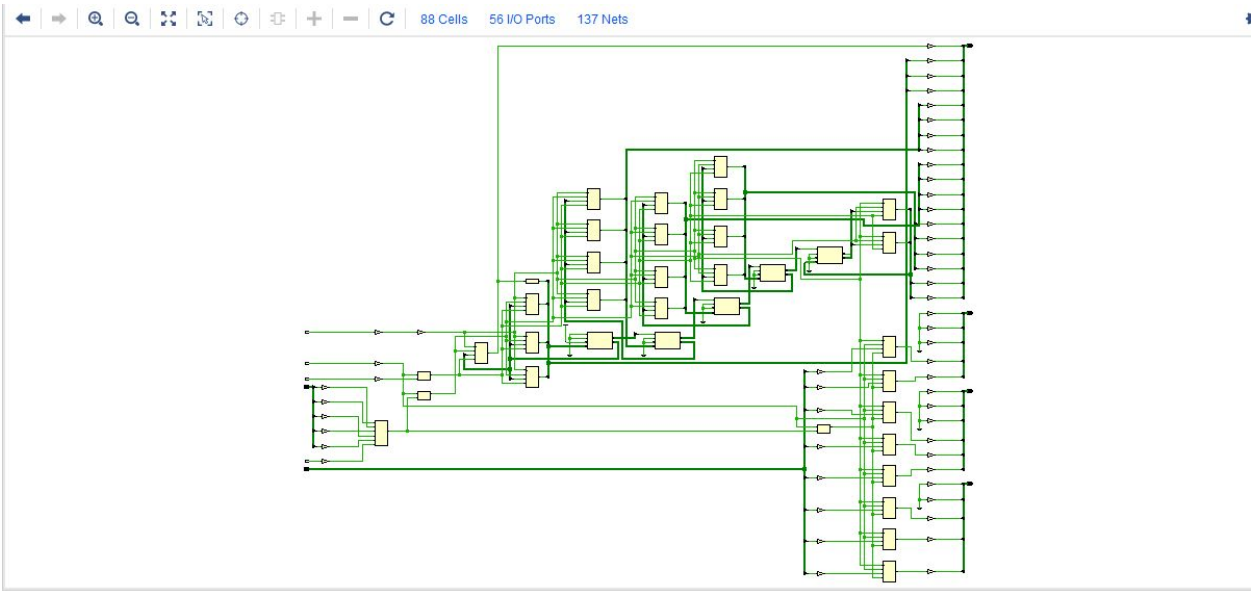
pixel_pusher.vhd

Code is on github for all files: trp87-rutgers

RTL Schematic



Synthesis Schematic:



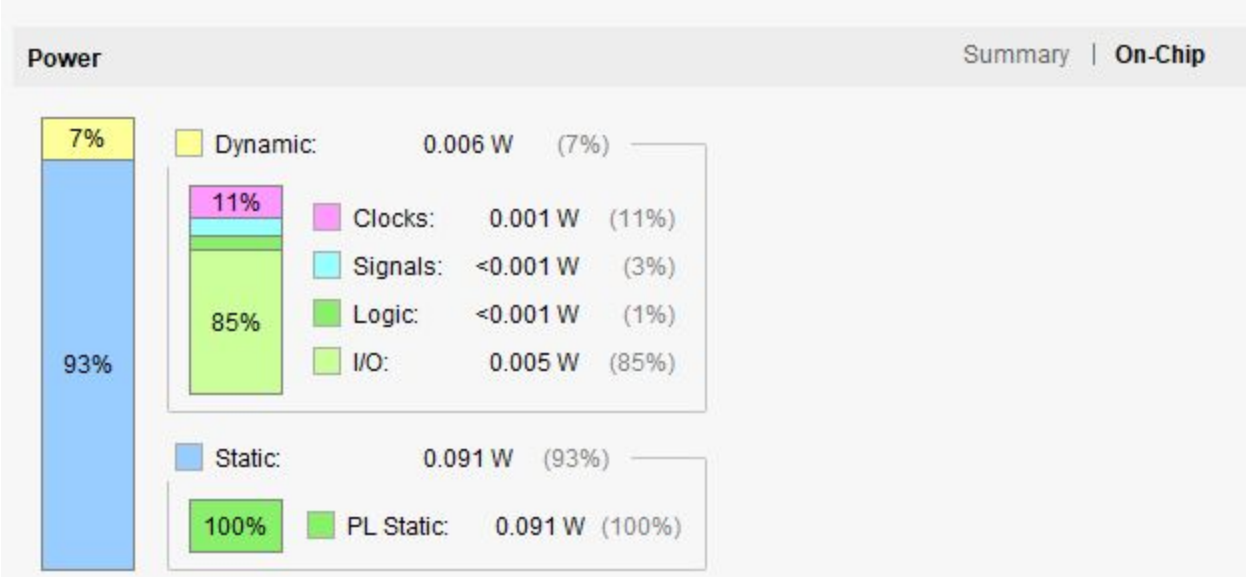
Post Synthesis Utilization Table

Utilization — **Post-Synthesis** | Post-Implementation

Graph | **Table**

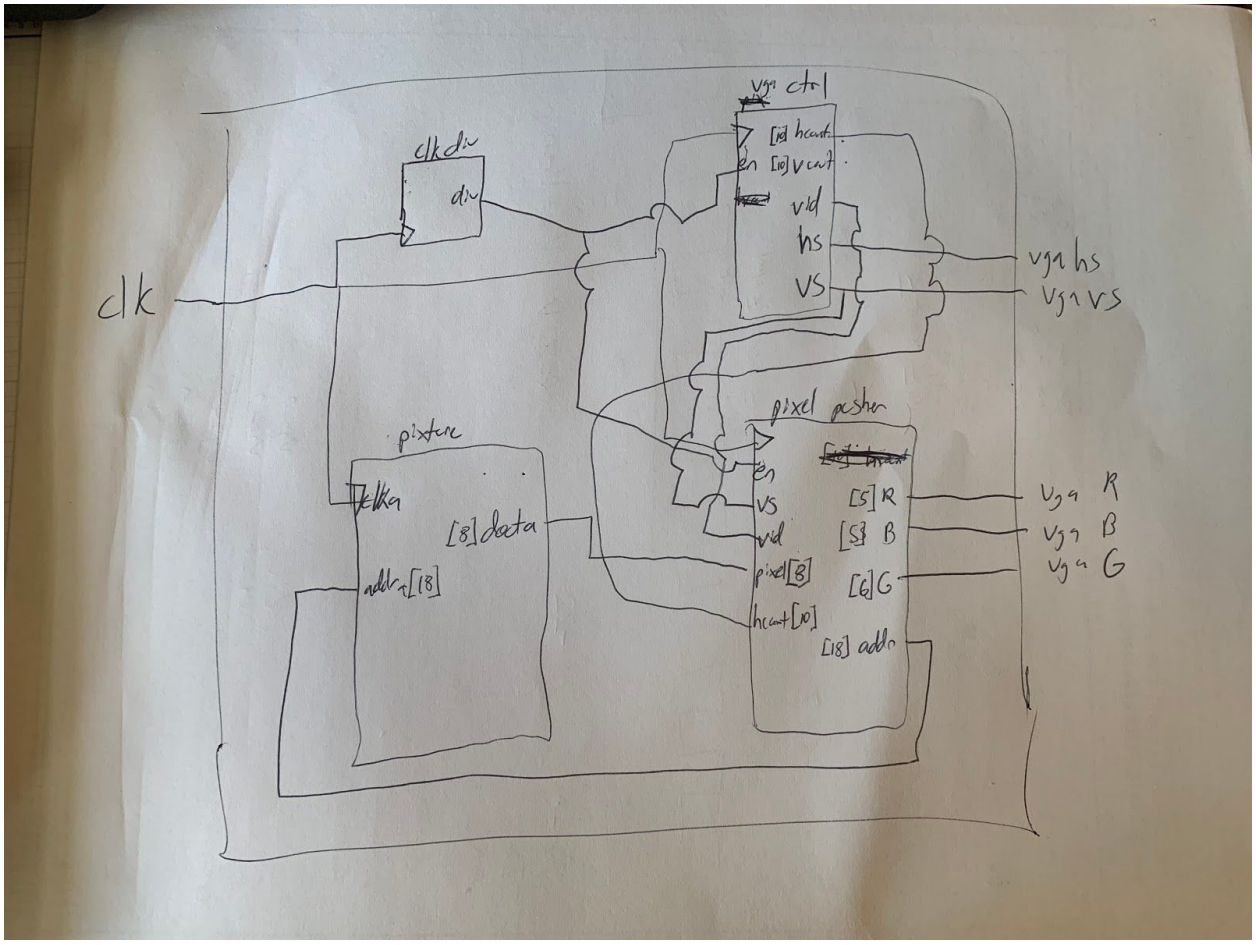
Resource	Estimation	Available	Utilization %
LUT	5	17600	0.03
FF	26	35200	0.07
IO	51	100	51.00
BUFG	1	32	3.13

On Chip Power Graph

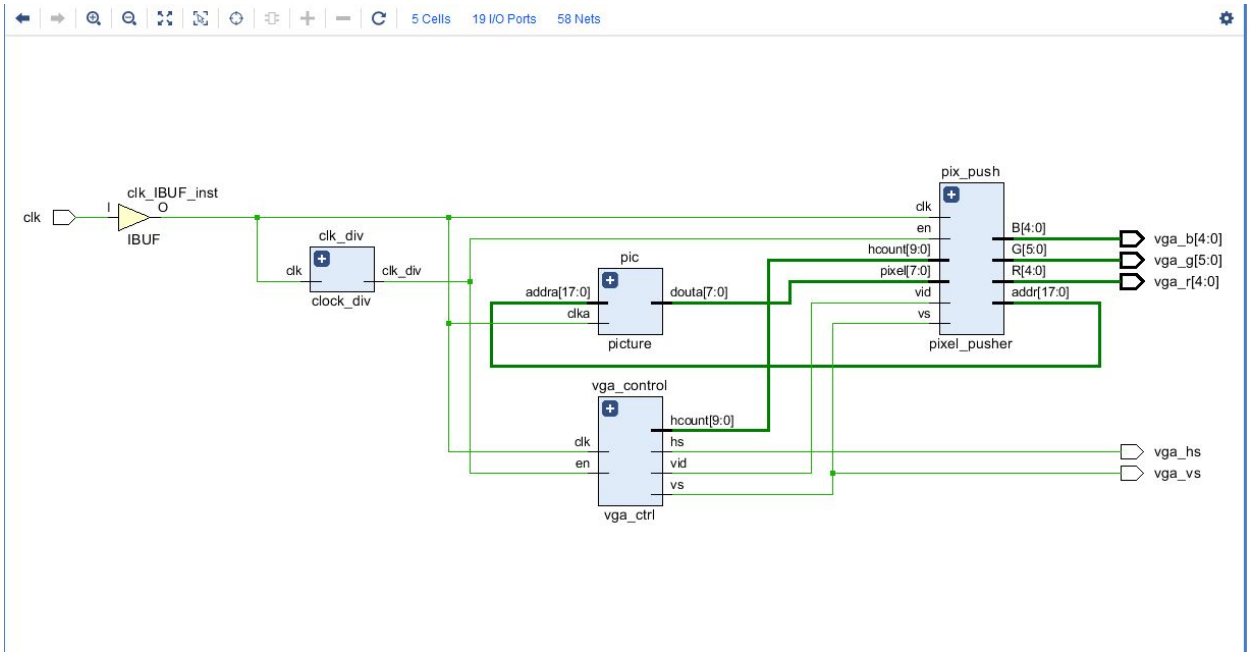


Image_top

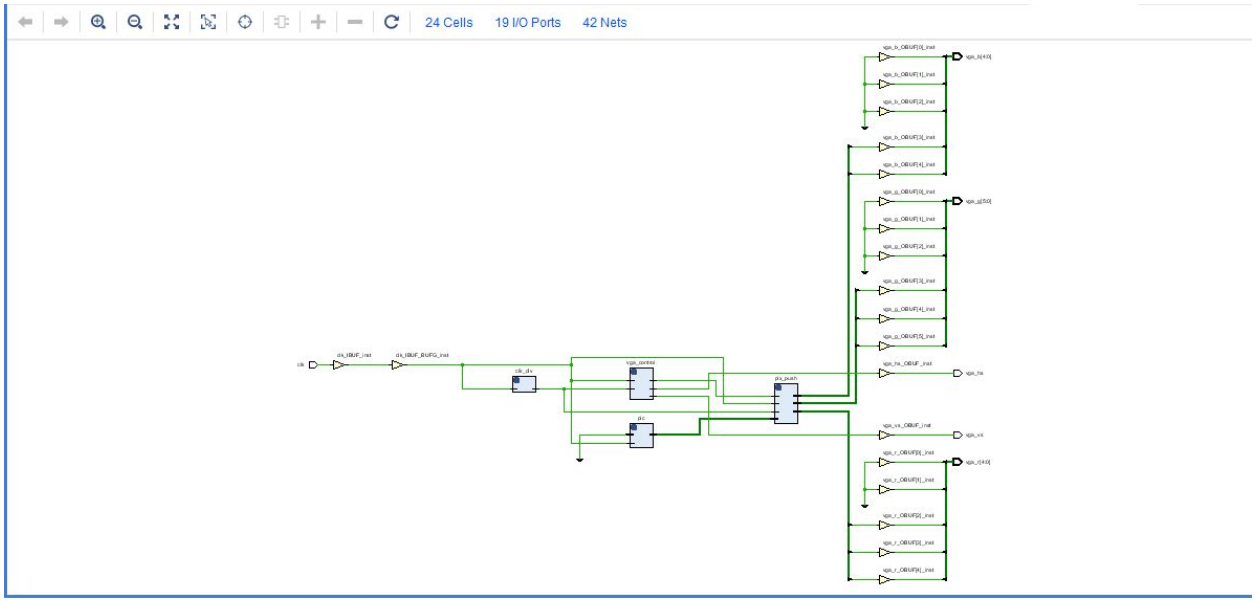
Hand Drawn schematic:



RTL Schematic:



Synthesis Schematic:



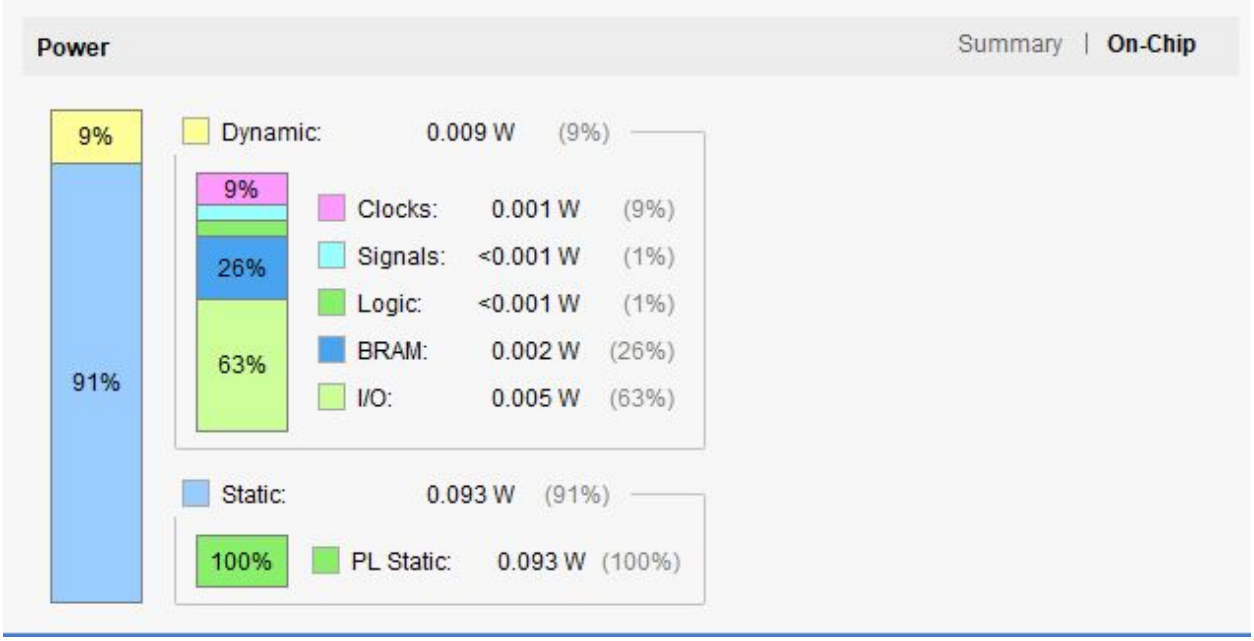
Post Synthesis Utilization Table:

Utilization — **Post-Synthesis** | Post-Implementation

Graph | **Table**

Resource	Estimation	Available	Utilization %
LUT	34	17600	0.19
FF	35	35200	0.10
IO	19	100	19.00
BUFG	1	32	3.13

On Chip Power Graph



I learned how to implement VGA. There were many troubles throughout the lab, including figuring out which wires port mapped where, and that vcount was not utilized outside of vga_ctrl. After over 10 hours of tireless troubleshooting including not putting the necessary space before a “then” in an if statement, and forgetting the ‘<’ sign in a less than or equal to, I was finally able to see the knight in the monitor, as shown below.


TM
DELL