Course Name: EMBEDDED SYSTEMS I / III

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

Year: Spring 2021

Lab Report: Final Project -- lightCtrl

Lab Instructor: Philip Southard

Student Name and RUID: Mya Odrick 180004110

Date Submitted: 05/05/2021

Purpose/Objective:

The purpose of this project is to create a controller that will control the LED lights on the zybo board, as well as display a counter of how long the ALS pmod is active. The overall goal of the project is to create a miniature home automation system that can be used to alter and monitor household lights. It communicates to the controller through the UART protocol on the UART pmod and BT2 pmod.

Theory of Operation:

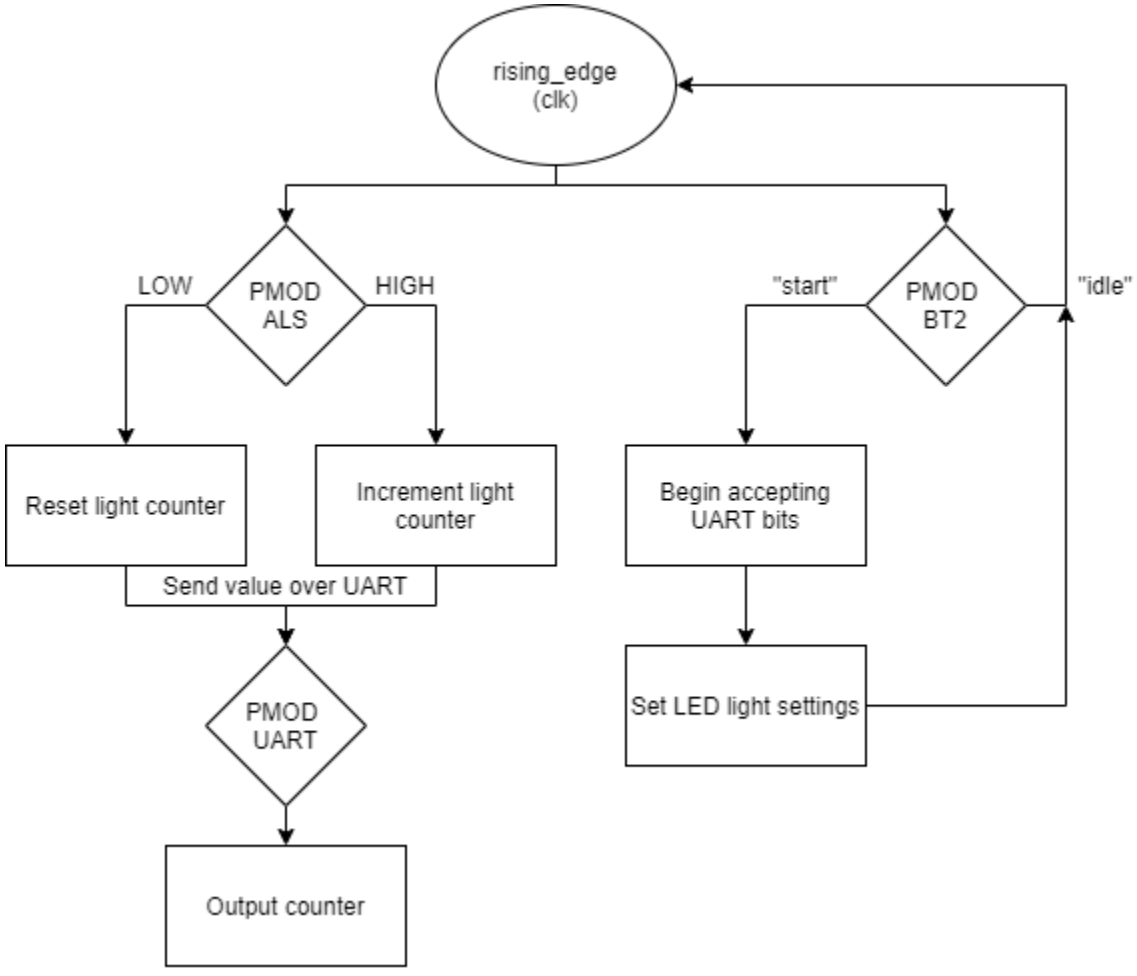


Figure 1 - System Block Diagram

Methodology & Requirements:

a. Python Script

I used a python script to communicate with the COM ports of the UART and BT2 pmods.

For the UART pmod, when the user clicked the “Light Uptime” button, I made a request for the current value on the receiving end of the UART.

```
ser=serial.Serial(
port ='COM5', # PMOD UART
baudrate=115200,
parity=serial.PARITY_NONE,
stopbits=serial.STOPBITS_ONE,
bytesize=serial.EIGHTBITS,
timeout=1
)
s = ser.read()
#time.sleep(5)
ser.close
```

Figure 2 - Python Script: Connecting to UART COM port

For the BT2 pmod, I sent a request to the transmitting side of the bluetooth. The request contained an 8 bit vector, where the first 4 bits represented the led number, and the last 4 bits represented the state of that led.

```
ser=serial.Serial(
port= 'COM6', #PMOD BT2
baudrate=115200,
parity=serial.PARITY_NONE,
stopbits=serial.STOPBITS_ONE,
bytesize=serial.EIGHTBITS,
timeout=1
)
ser.write(0) # start bit
ser.write(bytes(bitled.encode())) #LED commands
ser.write(0) # stop bit
#time.sleep(5)
ser.close
```

Figure 3 - Python Script: Connecting to BT2 COM port

b. VHDL Components

On the VHDL side, the transmitting side of the UART pmod was constantly updated with a counter value every second. When I used the python script,

the transmission would only occur when the button was pressed, but when I used the termite terminal, there was a constant stream of transmission during the 115200 baud rate.

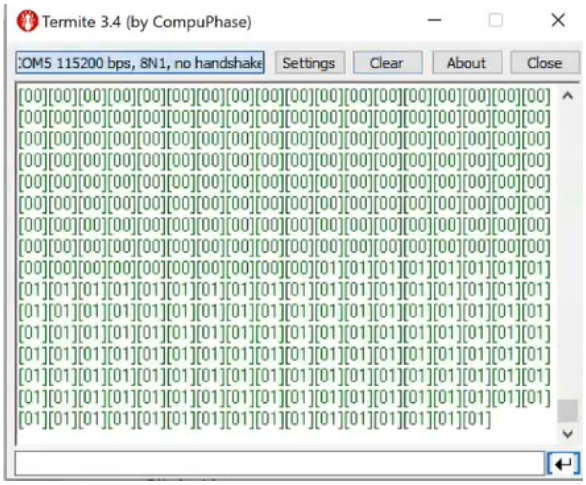


Figure 4 - Termite Terminal

In VHDL, the 8 bit string received by the BT2 pmod is dissected into 4 if statements, where d[7:4] = select [led0, led1, led2, led3] and d[3:0] set[led0, led1, led2, led3].

```
if(d(7) = '1') then
    led(0) <= d(3);
end if;
if(d(6) = '1') then
    led(1) <= d(2);
end if;
if(d(5) = '1') then
    led(2) <= d(1);
end if;
if(d(4) = '1') then
    led(3) <= d(0);
end if;
```

Figure 5 - VHDL code snippet - bt2_rx.vhd

c. Visuals

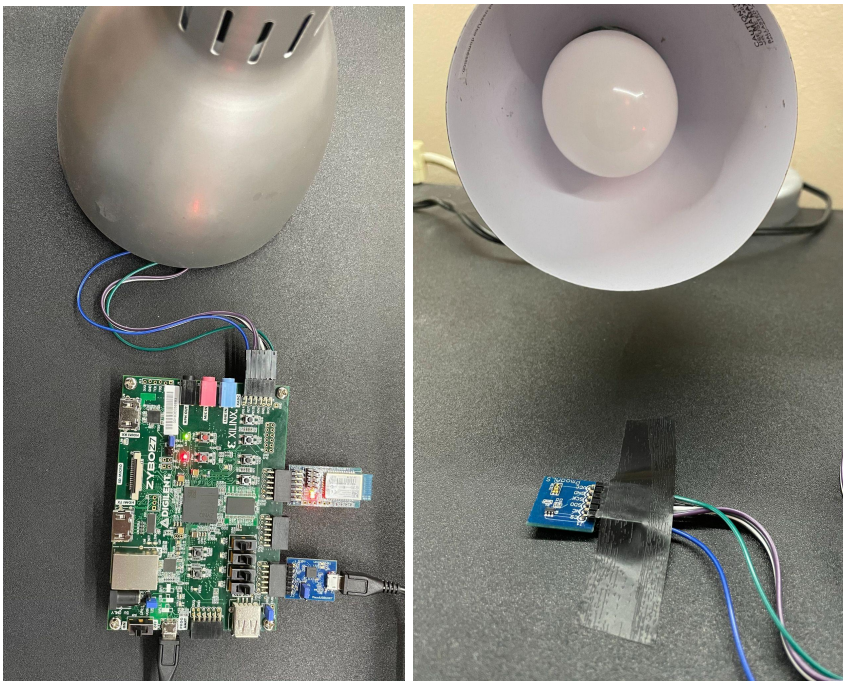
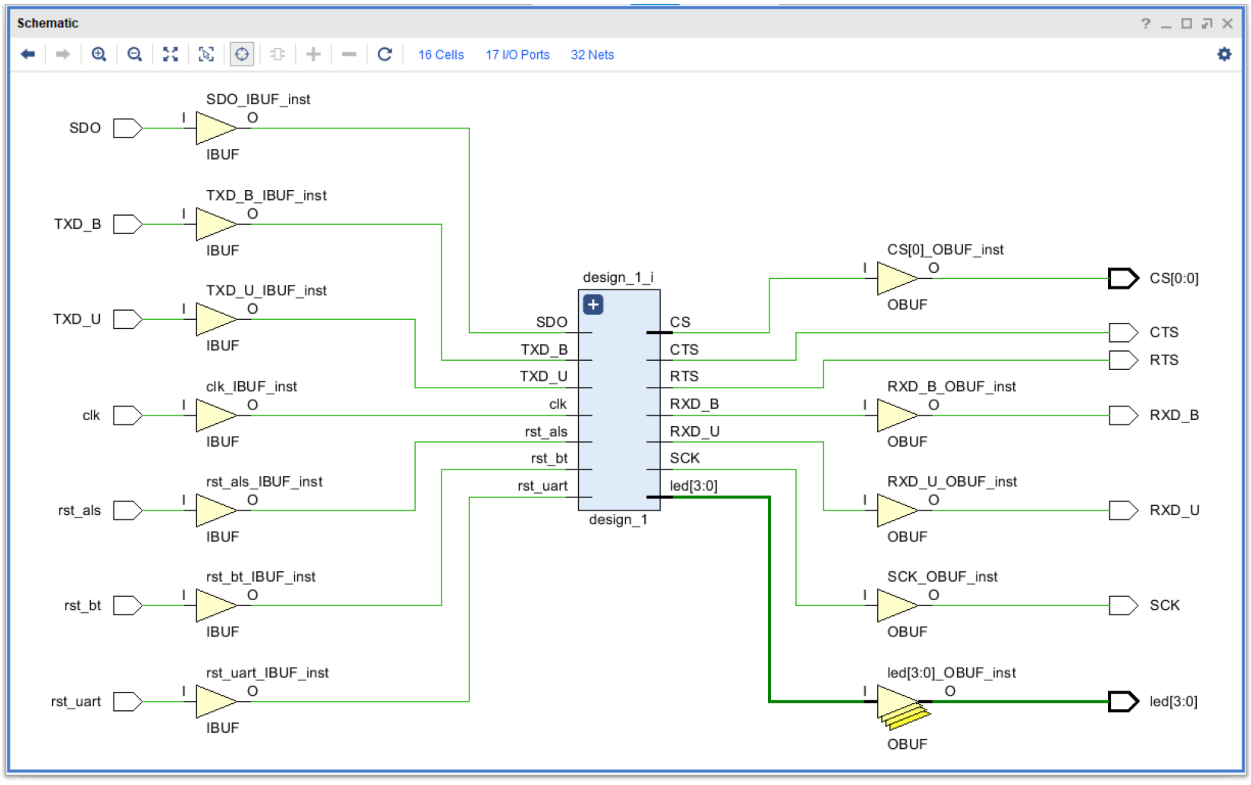


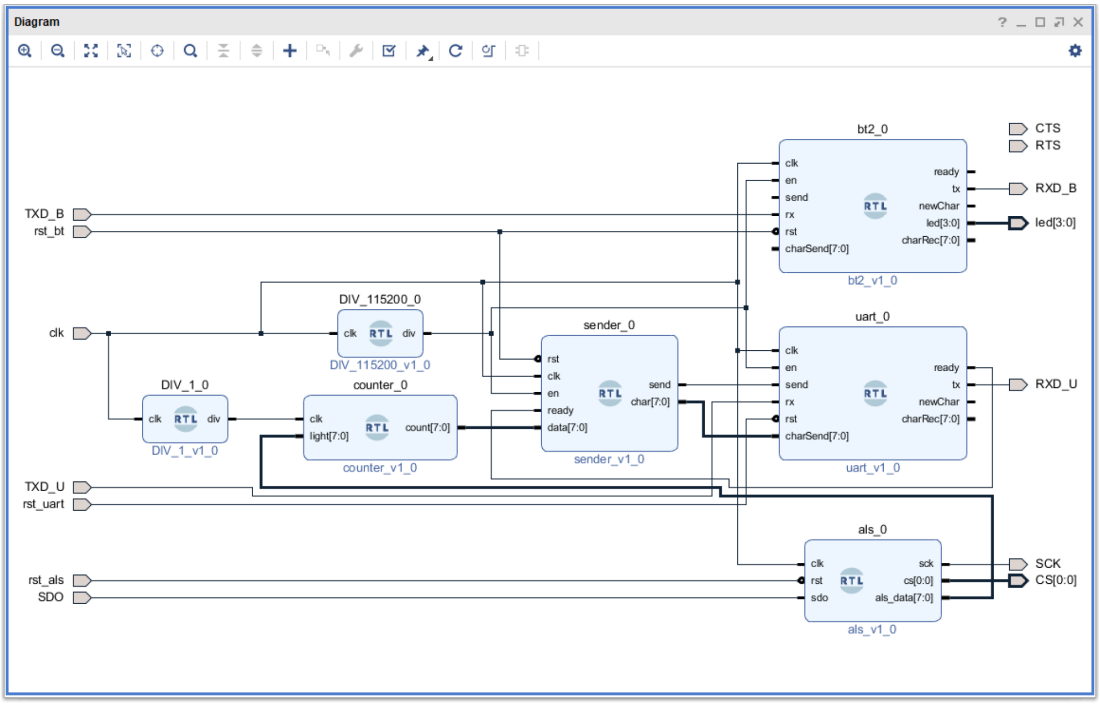
Figure 6 & 7 - Actual Board connections

Vivado Schematics:

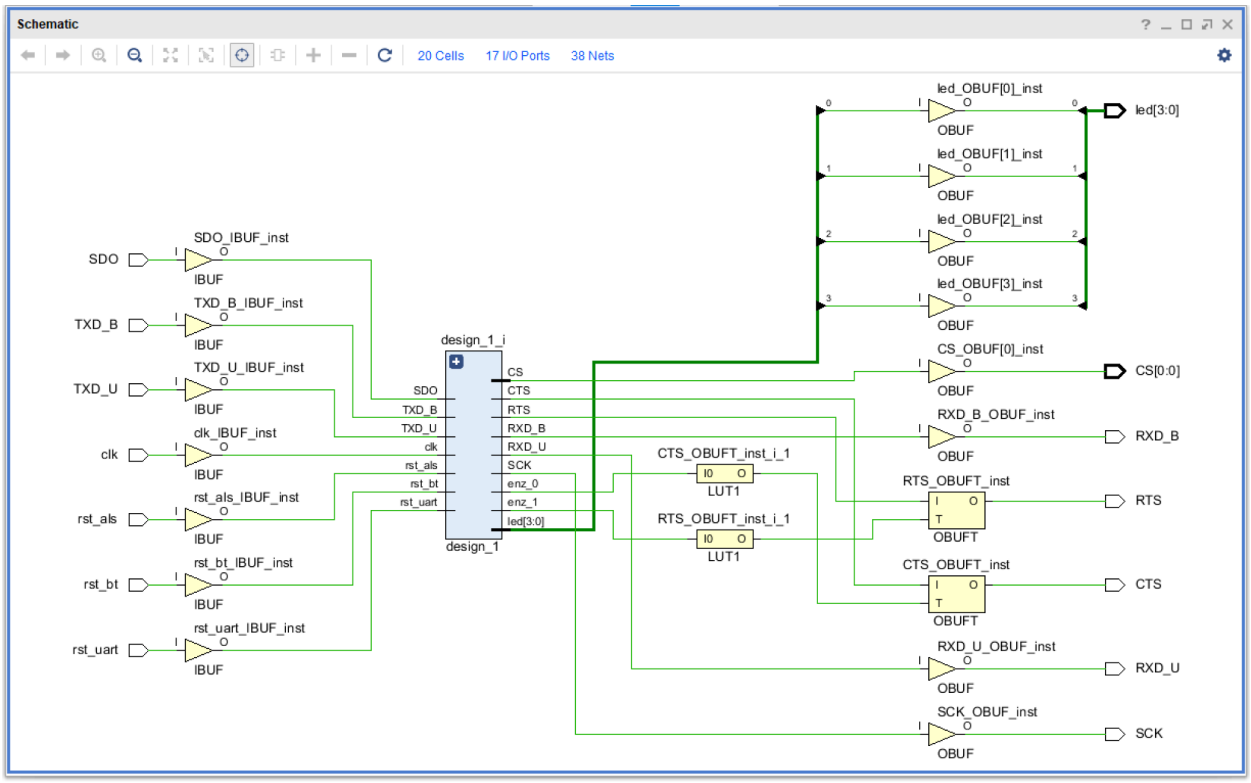
a) Vivado Elaboration Schematic



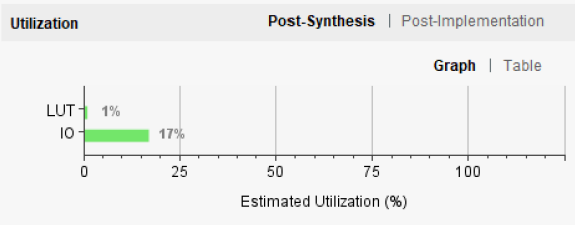
b) Vivado Block Design



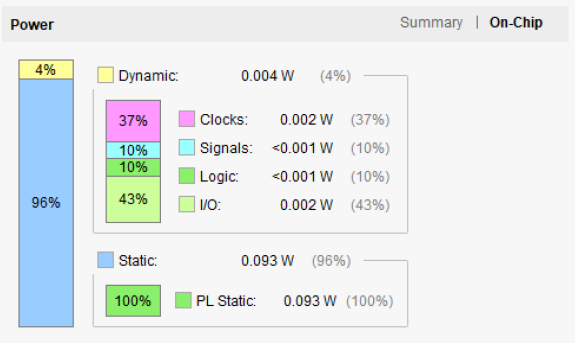
c) Vivado Synthesis Schematic



d) Post- Synthesis Utilization Table



e) On-Chip Power Graphs



Constraint File:

1. I used the UART pmod in the second row of JE, BTS pmod in JC, ALS pmod in JA, and used switches 2, 1, 0 as the reset pins for the pmods respectively.

Conclusion:

During this project, I became familiar with reading the Diligent reference manuals, designing a block diagram, and building and executing my own project scope. Throughout the weeks of working on this project, I had to redefine my project multiple times due to not being able to figure out the logic of certain components. However, I was able to readjust and use what I've learned during the semester in VHDL as well as my own skills with Python and complete the project.

Follow Up:

As the semester comes to an end, I believe I learned a lot this semester about FPGAs and working as an engineer. A lot of this class was based on independent work and finding resources on our own. I believe that helped me learn how to approach engineering problems in the workplace rather than how we would typically approach them in a classroom setting. With that independent mindset, I was able to fully grasp an understanding of how the Zybo board works and debug

common issues on my own. However, I also appreciated our TA's who helped me figure out other issues within my work that I couldn't discern on my own.