Mazen Abid
ENEL 384 - 094
Dave Duguid
Dec 4th, 2023

ENEL 384 Report Design with VHDL and FPGA - LED Intensity Control

Background:

Designing a digital circuit with VIVADO for the BASYS 3 FPGA Board, I chose this project to learn VHDL through a practical and simple implementation.

Requirements:

The design must incorporate:

- A synchronous circuit of reasonable complexity (>1 stage, <100 stages). This means that a project may not be comprised solely of combinational logic
- At least one internal module created with VHDL
- User I/O (switches, push-buttons, LED's, 7 segment display, etc)

Code Explanation for the main VHDL file

It consists of 4 logics broken down bit by bit below and a constraint file

1. **INPUT COUNTER**
   This determines the brightness of the green led through by using (clk) a clock
   This happens by using two switches:
   - one to increase
   - one to decrease.
   - counter (D flip-flops)
   - Pressing the up button increases the level of intensity and shows it on the 7 segment display

.

2. **COUNTER CLOCK**
   A clock was needed for each input so I created a clock divider that determines how fast the buttons can be pressed between each level of brightness.

3. **CLOCK DIVIDER**
   Clock divider for the LED bulb to determine 15 different levels of intensity.
   - With 0 being off to 15 displaying maximum brightness.
   - The clock divider increments each button press by what we set to be the levels of brightness.
   - Each increasing level meant an increase in the clock for the LED bulb.
     - Brightness doesn't increase linearly, the clock in the highest intensity could go and be decreased by clocks accordingly.

4. **Controlling LED:**

Finally combining all the files into the LED Controller file, to summarize, the components used are the following:

- Input counter
- Clock divider
- LED clock divider

I borrowed a Seven-segment display driver code, the “sseg” from Dr. Bryan Mealy from CalTech. It drives the button inputs to the seven-segment display on the Basys 3 board so that we know what level of brightness is on. Moving forward, the LED Controller uses flip flops to increase or decrease the count which controls both the seven segment display and the level of brightness of the LED bulb.

Constraints

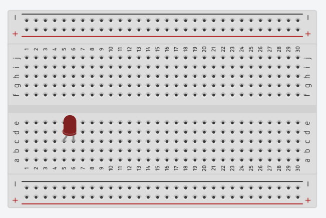
- Buttons
  - Changed T18 to up_enable in code to increase brightness
  - Changed U17 to down_enable in code to decrease brightness

- 7 segment display
  - W7, W6, U8, V8, U5, V5, U7, V7 represent each segment of one display
  - U2, U4, V4, W4 represent each anode displayed, only 2 are active because the highest number is 15

- PMOD Header JC
  - JC7 is where we connect one of the wires of the LED bulb, and the other wire leads to GND.

Diagram of the simple Breadboard Setup using TinkerCad is provided below:

Note: You can set it up however you like. The short end should always be connected to ground, and the long end should be connected to pin JC7 on the BASYS 3 FPGA.



Summary:

In summary, the project initial project requirements were met by having 15 stages or different intensity levels, with 2 modules built in VHDL that interact with the user in I/O manner using push-buttons, and the 7 segment display to control the LED.