Final Project: ES4 Spring 2021

Ibrahima Barry, WIlly Lin
Zach Osman, James Eidson
ECE Tufts University

5/11/2021

1 Overview

In this lab, a dual digit display decoder(dddd) was designed using VHDL and it's output was displayed on a 7-segment display decoder (with two digit places). First, VHDL code (device.vhd) to alternately flash two LEDs was made. The purpose which was to eventually do a similar thing with the two digit slots of the 7-seg. display decoder. The idea was to have both LEDs alternate with great enough frequency so they appeared to be on simultaneously. . Then, vhdl code (dddd.vhd) to produce the output of the one's place and ten's place on the 7-seg. was made. Finally, using a DIP switch with 6 bits wired up, the values $[0, 2^6 - 1]$ were displayed on the 7-seg.

2 Technical Description and Design

Flashing LEDs:
The HSOC built-in module with a 26-bit counter was used to drive two LEDs. By changing the bit of the counter used to drive the LEDs the ON-OFF frequency of the LEDs was changed. It turned out at using the 18th bit of the counter made it so that the two LEDs seemed to be ON at the same time:

at $48MHz/2^{20}$ => still seems to blinking

at $48MHz/2^{19}$ => still seems to blinking just a bit

at $48MHz/2^{18}$ => steady not blinking

Two-Digit Display:
By understanding how to make the two LEDs look they're on simultaneously, two NMOS transistors were used as "switches" turning on and off each digit. Here is the schematic:
Then in the top module (device.vhd) the code to switch between which LED output is on was written. The logic for the left and right digits from (dddd.vhd) were used as inputs to a MUX with the select input being the msb of the counter driver. Then based on whether or not that bit was on or off one 7-bit output was (for the ones-place or tens-place) was outputted. Here is the block diagram for this process:
A zip file of all the source code will be provided.

This is the final implementation on the breadboard:

3 Results and Testing

The behavior of the circuit was farily straightforward to test. One could manually use the DIP switches to do all 2^6 possible number combinations which would take very long (but doable).

Doing this for all possible digits I was sure the circuit functioned correctly. Also a lab TA looked over my work and agreed it worked as intended.

4 Debugging Log

1. A few LEDs on the 7-seg weren't lighting up or they were very dim.

- Manually setting the LEDs on would work but not otherwise.
- some possible causes included something wrong with VHDL code, the wiring of the breadboard, and a hardware component not working (not likely).
- Resolution: I needed to ground both GND pins of the Upduino. After that the dim effect went away.
- Lesson: Always check ground connections because even if other parts of the circut make sense you can get unexpected results.

2. problem: The transistors weren't acting as switches for the LED on the 7-seg. properly.

    - I knew there was a problem because both the digits on the 7-seg would always have the same numerical value. Both digits were displaying the value of the ones place.
    - possible causes included wiring of the transistor, confusion of what is the gate, drain and source.
    - The issue was that while the transistors were wired correctly they weren't acting as switches because the power pins of the 7-seg for both digits were also connected. I.e the transistors weren't controlling the flow of current in a meaningful way since both the digits were always on regardless of the what the transistors were doing.
    - the lesson I learned was to really understand what each component does and not just wire it up mindlessly. If I understood the role of the transistors, I would have recognized that the LED's on the 7-seg do not need to be connected to power while the source of the transistor was already connected to power.

3. problem: The LEDs weren't giving values as expected from the input given by the DIP switches.

    - There was a problem because when I toggled the switches the LEDs weren't producing the correct values.

- Possible causes included: FPGA pin's not connected properly, the logic in the VHDL code.
- It turned out that the reset signal on my counter was not set to any value. After making this change the outputs seen on the LEDs of the 7-seg were correct.

5 Reflection

1. What was the most valuable thing you learned, and why?
   - I learned how to debug my VHDL code in a way different to how one debugs code in a language like Java or Python. Many times the code can compile but the hardware implementation may not behave the way you want it.
2. What skills or concepts are you still struggling with? What will you do to learn or practice these?
   - I'm still struggling with how to write testbenches for very large complicated code. To practice I will review the textbook and try writing testbenches for other VHDL code I've written.
3. This assignment took about 72 hours (not including several hours tyring to write a testbench) during all of the days in which I worked on it.

6 Work Divison

7 source code