Lab 4

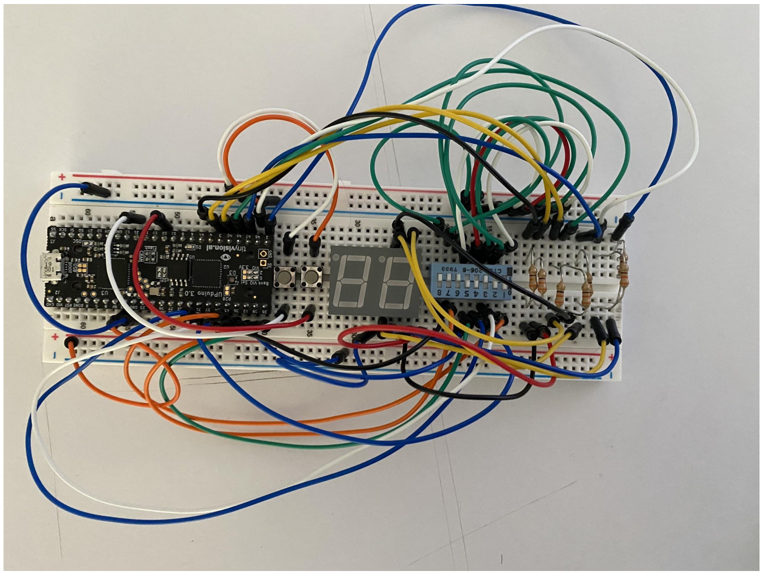
Jake Seidman

3/23/21

Lab Session: Tuesday 1 PM

TA: Zhou Ni

Circuit:



Debug Log

- First I write up the DIP switch and the two switches to the arduino according to the schematic provided in the lab
- I paste my VHDL code for the ALU into radiant, map the correct pins to the input/output, and synthesize and export the design
- I connected the output to 4 LEDS to test if it was correctly working
- No LEDs are lighting up
    - Fix: I had incorrectly wired the schematic, and mistakenly connected the inputs to the switch to the ground
- With multiple test cases, the LEDs seem to light up randomly with their output in no way relating to the input
    - Fix: I had forgotten to flash the FPGA
    - I had trouble flashing the FPGA, because the Numonyx device option was not there
        - Found answer on campuswire (had to use legacy option)
- I was able to flash the FPGA, but my outputs were not showing correct results
    - Rewired pins
- Disconnected the LEDs, and wired the seven segment display
- I looked up the part number on the display to find which input wire corresponds to which LED segment, and wired them accordingly
- After correcting pins, outputs are still not working
    - Only 4th output bit is turning on
    - Solution: I was using the wrong ground pin on the FPGA. After switching to the correct ground pin, the circuit worked perfectly
- I realized the switches had a floating value of 1. Pressing the switch turns them to 0.
    - I inverted the VHDL code to account for this
- The 7 segment display is lighting up were inverted to what they should be
    - I had set active segments to be a 1, instead of a 0.
    - Negated all outputs using “not” in the VHDL code

Testing

- I decided to break up testing into 4 parts, testing each operation separately.
- Two 4-digit numbers have 256 different combinations, so it is unfeasible to test all possible inputs. Instead, I attempted to select a comprehensive subset of the inputs.
- Test cases I picked include the min/max values for inputs, min/max values for the outputs, 0 values in the input and output. In addition to these tests, I attempted to pick test cases that spanned the range of possible inputs and outputs.
    - Note that I did not perform any test cases that result in an output that is negative, or greater than 15. I did this because the output display is not capable of displaying numbers outside of this range, and the lab did not specify that we are required to account for these values.

The values in the table listed below are the expected outcomes of the tests I will perform. I will highlight any values in red that do not match their expected output

Bitwise AND (switches set to 00)

Input 1	**Input 2**	**Output (as shown on display)**
0000	0000	0
0001	0000	0
0001	0001	1
1010	1000	8
1111	1010	A
1100	1110	C
1100	0011	0
1111	0001	0
1111	0000	0
1111	1111	F

Bitwise OR (switches set to 01)

Input 1	**Input 2**	**Output (as shown on display)**
0000	0000	0
0001	0000	1
0001	0001	1
1010	1000	A
1111	1010	F
1100	1110	E
1100	0011	F
1111	0001	F
1111	0000	F
1111	1111	F

Addition (switches set to 10)

Input 1	**Input 2**	**Output (as shown on display)**
0000	0000	0
0001	0000	1
0001	0001	2
0100	1000	F
0001	1010	B
0100	0010	6
1100	0011	F
1000	0011	B
1010	0101	F
1000	0001	9

Subtraction (switches set to 11)

Input 1	**Input 2**	**Output (as shown on display)**
0000	0000	0
0001	0000	1
0001	0001	0
1000	0100	4
1010	0001	9
0100	0010	6
1100	0011	9
1110	0010	A
1010	0101	5
1111	0000	F

All test cases gave the expected output, which leads me to believe that my circuit is working correctly

What was the most valuable thing you learned, and why?

The most valuable thing I learned in this lab is how to use radiant. Using icestudio to engineer circuits is extremely tedious. Especially as circuits get more complicated, it would become more and more implausible to use icestudio. Using radiant and VHDL to engineer circuits allows for much more complex circuits to be created much easier

What skills or concepts are you still struggling with? What will you do to learn or practice these?

Although I was able to complete the VHDL code for this lab, I still struggle a lot in understanding a lot of other VHDL concepts we have covered. I am not confident in my ability to write the code for processes or test benches. Hopefully the future VHDL assignments will bolster my understanding.

How long did it take you to complete the lab? This will help calibrate the workload for future iterations of the course.

Completing the physical lab took me about 3 hours (~1.5 lab sessions). Writing the lab report took me around 1.5 hours.

Code:

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

entity lab4 is
  port(
 a : in unsigned(3 downto 0);
 b : in unsigned(3 downto 0);
 s : in std_logic_vector(1 downto 0);
 output : out unsigned(6 downto 0)
  );
end lab4;

architecture synth of lab4 is

signal displaySignal: unsigned(3 downto 0);

begin
  displaySignal <= (a and b) when (s =  "11") else
    (a or b) when (s =   "10") else
    (a + b) when (s =   "01") else
    (a - b) when (s =  "00");

     output <= not"1111110" when displaySignal = "0000" else
              not"0110000" when displaySignal = "0001" else
              not"1101101" when displaySignal = "0010" else
              not"1111001" when displaySignal = "0011" else
              not"0110011" when displaySignal = "0100" else
              not"1011011" when displaySignal = "0101" else
              not"1011111" when displaySignal = "0110" else
              not"1110000" when displaySignal = "0111" else
              not"1111111" when displaySignal = "1000" else
              not"1110011" when displaySignal = "1001" else
              not"1110111" when displaySignal = "1010" else
              not"0011111" when displaySignal = "1011" else
              not"1001110" when displaySignal = "1100" else
              not"0111101" when displaySignal = "1101" else
              not"1001111" when displaySignal = "1110" else
              not"1000111" when displaySignal = "1111";


end;
```