Lab 3
Simon Carballo
1618309
10/30/2020
Section D

Write Up

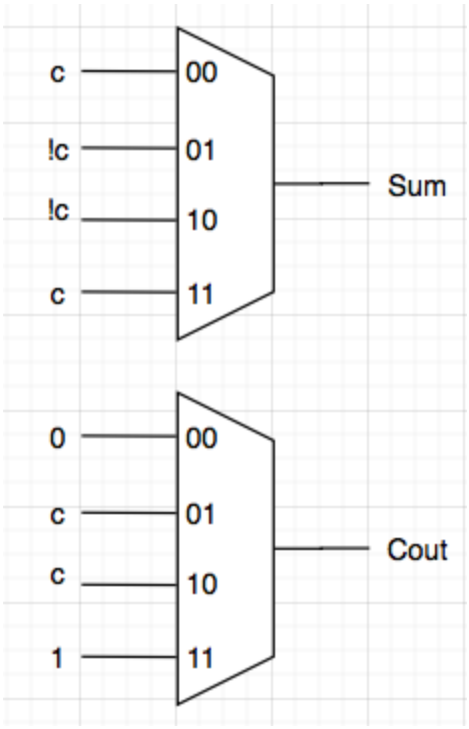
- **Description:** In this lab we were asked to design and implement a combinational circuit which adds 0-3 to an 8-bit binary number. This displays the sum on the two rightmost 7-segment LED displays on the Basys 3 Board. This will be operated using the 8 switches (sw[0] - sw[7]) and two extra buttons (btnU, btnD) that represent a 2-bit addition. The challenge of this lab is to build the adders and converters using MUX's.

- **Design:** For this design we have the Top Level that calls to 5 subsections which includes: Incrementer, Upper/Lower Segment Converter, Digital Selector, and a 2to1 MUX. As we go over each subsection we will look more into depth of how each section is designed.
    - <u>Incrementer:</u>
      First let's look at the incrementer that acts as the 8-bit adder in this design:
        - Truth Table of a Full Adder:

a	b	c	Sum	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

        - As we see in the truth table we add the sum of a, b, and cin, and when there is a value over 1, there would be an overflow representing Cout.
        - Now that we are refreshed on how an Adder works. This time we will create a 4-1MUX with the truth table by designating a, and b as the selector bits(in this case). This can be seen by the color coding of each selector state that will be used in the MUX. This creates the following boolean equation with for the two outputs:
            - Sum = c!a!b + !c!ab + !ca!b + cab

    - Cout = c!ab + ca!b + ab
- Visually these MUX would look like this for the two outputs:



- By putting the two MUXs together we create a full adder!

- Now we have to make it into the 8-bit incrementer by calling the Adder 8 times and wiring them together just like how we would wire adders together. In this design we also need to be aware that there are two extra buttons (btnU, btnD) that act as a 2-bit adder which would be added into the rest of the 8-bit address. In order to do this we set the first two adders as the second input while the rest would be set to 0. This would create the incrementer design that would be implemented into the TopLevel Design.

- <u>Upper/Lower Segment Converter:</u>
  Now we will take a look at the seven segment converter for this design
    - Like the adder we will by using MUXs to design our seven segment converter. In this design we will be using a 8-1 MUX to find the outputs for each segment.
    - First let's take a look at the truth table:

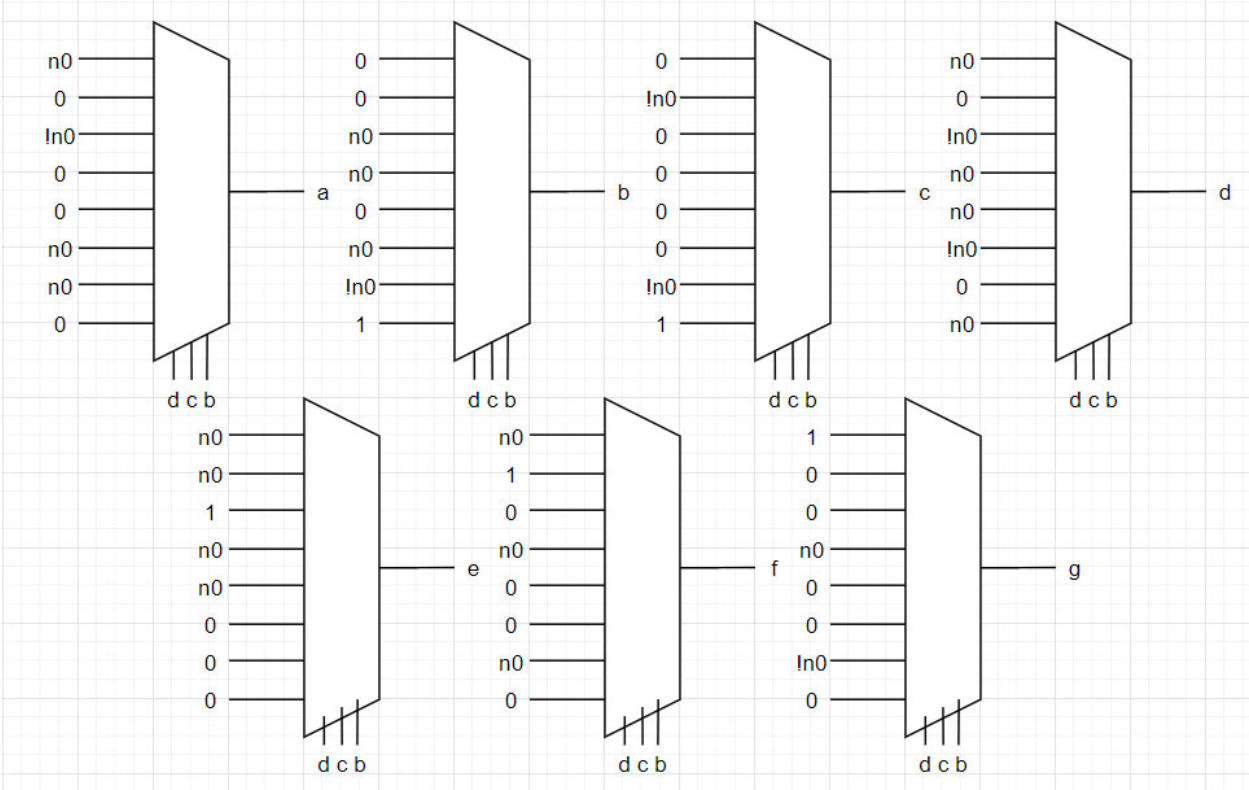
n3	n2	n1	n0	a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	1	0	0	1	1	1	1

0	0	1	0	0	0	1	0	0	1	0
0	0	1	1	0	0	0	0	1	1	0
0	1	0	0	1	0	0	1	1	0	0
0	1	0	1	0	1	0	0	1	0	0
0	1	1	0	0	1	0	0	0	0	0
0	1	1	1	0	0	0	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1	1	0	0
1	0	1	0	0	0	0	1	0	0	0
1	0	1	1	1	1	0	0	0	0	0
1	1	0	0	0	1	1	0	0	0	1
1	1	0	1	1	0	0	0	0	1	0
1	1	1	0	0	1	1	0	0	0	0
1	1	1	1	0	1	1	1	0	0	0

- This is a 4-variable truth table for the 16hex values available on a seven segment display. a - g represents each segment of the Cathode Segment Display. There is a DP segment but that will be unused for this design. In order to figure out active high and lows of each segment we go through each hex value and determine whether the segment will be on or off. For example, when we want the value “1” we would need to have the segments ‘b’ and ‘c’ on therefore making them the lows(0s) for a cathode diode.
- Now that we refresh on how a converter works, we will be using 7 8-1 MUXs to create our converter. In order to do this we will use n1, n2, and n3 as the selector bits shown by the color coding in the truth table. By finding the highs and lows of n0 at each selector state we can design the converter we get these boolean expressions for each 8to1 MUX.
    - Selector : n1, n2, n3
    - a = n0(!n1!n2!n3) + !n1(!n1n2!n3) + n0(n1!n2n3) + n0(!n1n2n3)
    - b = n0(!n1n2!n3) + !n0(n1n2!n3) + n0(n1!n2n3) + !n0(!n1n2n3) + (n1n2n3)
    - c = !n0(n1!n2!n3) + !n0(!n1n2n3) + (n1n2n3)

- d = n0(!n1!n2!n3) + !n0(!n1n2!n3) + n0(n1n2!n3) + n0(!n1!n2n3) + !n0(n1!n2n3) + n0(n1n2n3)
- e = n0(!n1!n2!n3) + n0(n1!n2!n3) + (n1n2n3) + n0(n1n2!n3) + n0(n1!n2n3)
- f = n0(!n1!n2!n3) + (n1n2n3) + n0(n1n2!n3) + n0(!n1n2n3)
- g = (!n1!n2!n3) + n0(n1n2!n3) + !n0(!n1n2n3)

- By following the boolean expression we obtain a 8to1 MUX that looks like this:



- Just like a regular converter, when called by the top level each input will go through the 7 MUXs for their respective segments. Therefore in order to have two seven segment displays to work, we would need two groups of 8to1 MUXs for the lower and upper values of the 8-bit adder.

- <u>Digsel</u>:
  - This module/subsection is provided to us via the lab manual. It's primary function is to output a low frequency of highs and lows for the two seven segment displays to work.

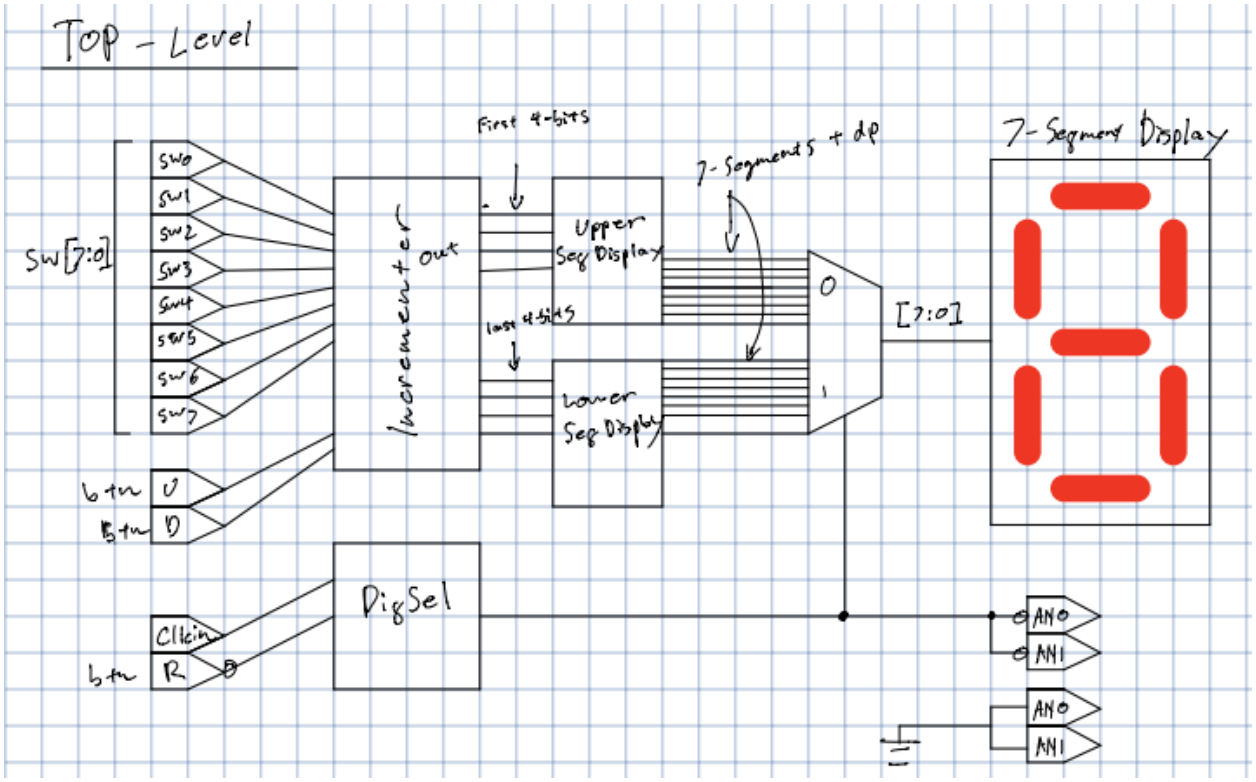
- <u>2to1 MUX:</u>

- Finally the final piece is the MUX that takes inputs from the two converters we designed above to be selected by the Digsel for display on the seven segment converter. We need to make sure that 8-bits gets through the MUX. To summarize, we need this MUX to be able to display two seven segments at the same time in human eyes that would work in unison with the 8-bit adder or incrementer.

- Top Level:
-

- With this truth table we can acquire a boolean expression for each segment to make the 16Hex values (n0,n1, n2, n3 == a, b, c, d):
  - a = a!b!c!d + !abc!d + ab!cd + a!bcd
  - b = a!bc!d + !abc!d + ab!cd + !a!bcd + !abcd + abcd
  - c = !ab!c!d + !a!bcd + !abcd + abcd
  - d = a!b!c!d + !a!bc!d + abc!d + a!b!cd + !ab!cd + abcd
  - e = a!b!c!d + ab!c!d + !a!bc!d + a!bc!d + abc!d + a!b!cd
  - f = a!b!c!d + !ab!c!d + ab!c!d + abc!d + a!bcd
  - g = !a!b!c!d + a!b!c!d + abc!d + !a!bcd
- We can input this boolean expression into the converter by taking in the 4 outputs and wiring each output to its respective segment. We will do this in the Top_Level Diagram as shown below.

- Top-Level:



- As you can see in the diagram above these are all the subsections put together to create the design. The 8 inputs are directed into the increment which outputs a summation of 8bits. The first 4-bits are connected to the lower segment converter and the last 4-bits are connected to the upper segment converter. We connect these are the two inputs for the 2x1 MUX with the selector being Digsel operated with a reset and internal clock. This MUX will then frequently output the two inputs of the MUX to two separate seven segment displays as shown with the wire to AN0 and AN1.

- **Testing & Simulation:**
  In order to test my design I used two different resources like the previous lab. The first resource is by running a simulation directly from the Vivado application. Using the provided testbench file, I entered the test cases spaced between each 100ns. These tests consisted of displaying the 16HEX values on both displays, mismatching them, and finally overloading them with the buttons that add 2-bit values. Another source was by programming the bitstream to the Basys3 board. This method is a more efficient method of testing since I can visually see if I wired the wrong switches or swapped the buttons.

- **Results:**
  - How fast the signal **dig_sel** is oscillating (in simulation).

        - I cannot find the exact number that the dig_sel is oscillating at, but I know for a fact that the oscillations can only be detected in simulation when observing microseconds.
    - Whether you observed any flickering in the 7-segment display
        - I did not see any flickering on the 7-segment display, this is because the digsel is oscillating so fast that a human eye can not detect the on and off of the seven segment display.

- **Conclusion**
    - From this lab I learned how to implement MUXs to the previous modules we have created (Adder, Seven Segment Converter). By using this previous knowledge I was able to create a 8-bit adder and separate the outputs into two different displays. I also learned how to use busses to keep wiring tidy in a bigger design. The most interesting part of this design is how the digsel oscillation works to provide power for two displays using a MUX. It is definitely very fascinating to see how fast an oscillation is, and undetectable to the human eye.