Frank I Gomez
4/13/21
Lab 2, Section D.
ID: 1650550
figomez
CSE 100L

Lab 2 Write Up

1. Description

The purpose of this lab was to create an 8 bit full adder, which would then be displayed on a seven-segment display as a hex value from 0 to 15/F.

2. Design

- Top Level

Diagram 1

As can be seen in Diagram 1, the 8 switches enter full adder, providing a 4 bit output, which is then taken to the seven segment display converter, which converts the 4 bits into logic functions, which then determine which segments of the display are on or off. Something to note is that the display is driven by low values, which makes the logic equations easier to compute. Finally, as only the least significant digit was powered, the display's DP, AN1, 2, and 3 values were always set to 1, to keep them turned off.

- Full Adder and Mini Adders

The adder was controlled by the 8 switches, which would act as values of 0, 1, 2, or 4. The full adder itself consisted of 3 smaller adders, which carried the output into the next respective adder, and the final output of the system.

Each mini-adder consisted of 2 functions, setting values for the carry out variable, and the sum variable.

a	b	c_in		s	c_out
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

Table 1

It can be seen by Table 1 that the sum output is actually an XOR gate, while the c_out variable is a AND b, OR'd with a XOR b AND c_in.

- Seven Segment Display

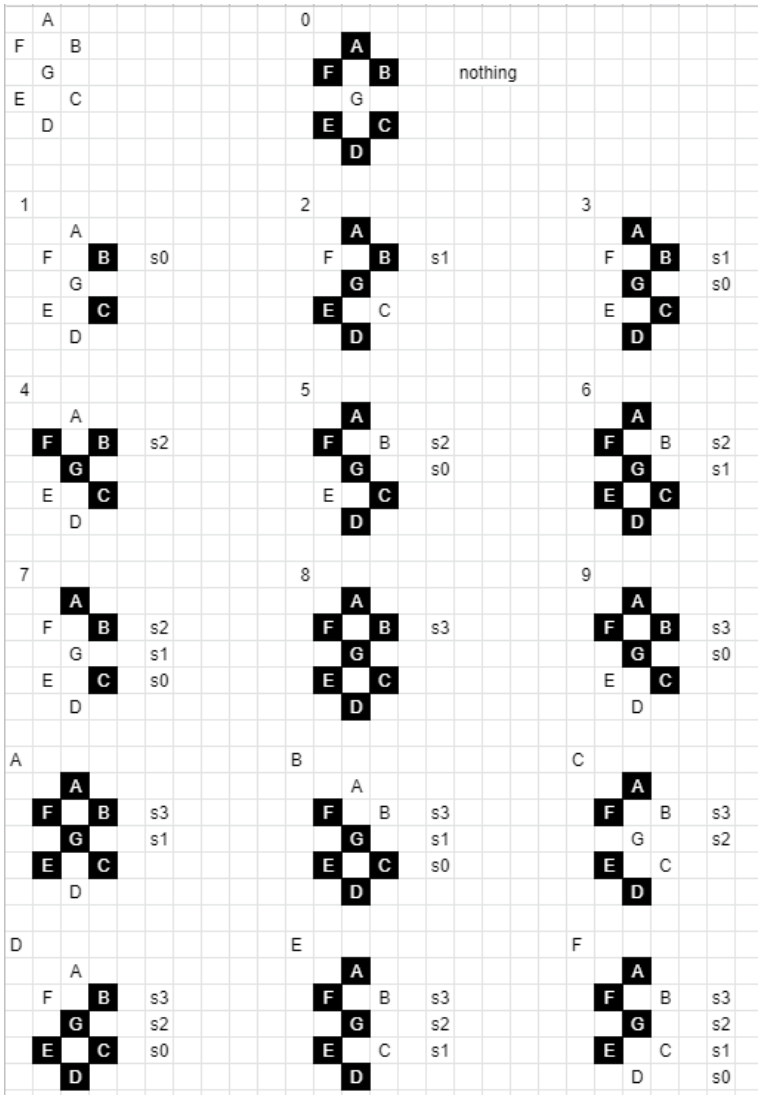

Diagram 2

The interesting segment of the display was creating logic functions for each possible value, to determine whether a light should be on or off. First, all 16 values were designed in a spreadsheet, displayed here as Diagram 2, with the black cells representing values that needed to be lit, or, in other words, set to low. As such, each of the 7 outputs consisted of a logic function OR'ing all the values for which said segment would be off, or otherwise, set to high. Take, for example, segment g in value 0. As g needed to be driven there, its logic function consisted of $s0\&s1'\&s2'\&s3'$. This would, in turn, power the g when given that input, disabling it on the display. Segment g then had this statement OR'd with the other 3 inputs for which it would need to be disabled. This method was used for all 7 outputs.

3. Testing and Simulation

Running a verilog simulation was all the testing that was required for this lab, as it provided all 16 switch combinations, and displayed their ASCII outputs, in the way that would be displayed on a physical screen. The only difficulties during testing was actually from the test harness itself, as there were a few errors (such as running for twice as long in one case, and not running the final case) that needed to be fixed prior to final testing. Initial testing revealed one of the logic statements for segment b of the seven segment display was entered incorrectly, and after fixing it, the program functioned as expected.

4. Results

- Document which pins of the FPGA you used and how they were connected to the switches and 7-segment displays.

The following FPGA pins were used: CA, CB, CC, CD, CE, CF, CG, DP, AN0, AN1, AN2, and AN3, renamed from Verilog's seg[0], seg[1], seg[2], seg[3], seg[4], seg[5], seg[6], dp, an[0], an[1], an[2],and an[3] pins. All the "C" pins were outputs from the seven-segment display module, each corresponding to a specific segment of the display. AN0 corresponded to the least significant display, and as such, was powered off. The remaining pins, DP, AN1, AN2, and AN3 were driven high, to keep those segments of the display powered off.

- Determine the longest path from any input to any output in your 3-bit adder. (i.e. the highest number of gates that any input goes through before reaching an output.)

The longest path from any input to any output in the 3 bit adder was 3, taken from either of the direct switches' inputs.

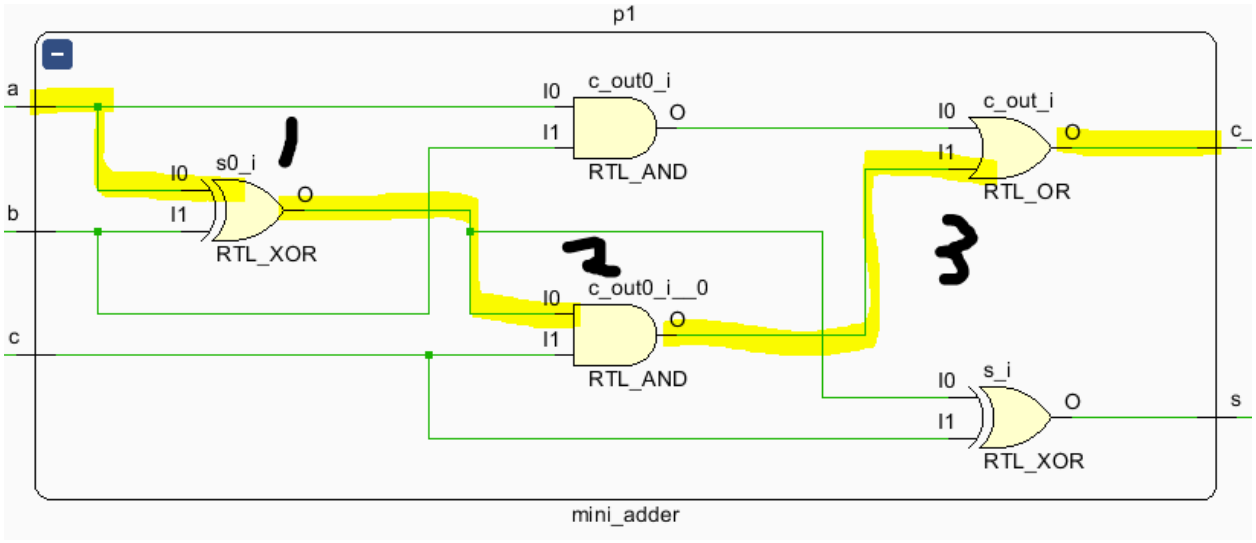

Diagram 3

Seen above in Diagram 3, input a travels through an XOR gate with input b, which is then AND'd with input c, which is then OR'd against the result of a AND b, resulting in c_out.

- For an n-bit adder determine the length of the longest path from any input to any output (in terms of n).

The longest path an input can take in an n-bit adder is n gates, traveling from the input to the xor of the carry-out.

5. Conclusion

Through this lab, I learned how to transfer an input of 8 bits into a single, seven segment display, by the use of a full adder. The only real difficulties came in the form of Verilog syntax, as the logic behind the lab was relatively simple in comparison. Nothing would be changed on a subsequent attempt, as the lab provides sufficient reading to understand without needing to ask for assistance.

6. Appendix

(Note: Waveform Viewer cannot be printed to PDF, and as such, it is added here as a screenshot. Additionally, as the testbench was modified as stated above, it will also be attached.)

The bottom 4 values, the x variables, are the top level wires.