Assignment 1.2

Thursday, January 25, 2021 8:36 PM

a_3	a_2	a_1	a_0	S_0	S_1	S_2	S_3	S_4	S_5	S_6
0	0	0	0	1	0	1	1	1	1	0
0	0	0	1	1	1	1	0	1	1	1
0	0	1	0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1	1	1	1
0	1	0	0	1	1	0	1	0	0	1
0	1	0	1	0	0	1	1	0	0	1
0	1	1	0	1	1	0	1	0	0	1
0	1	1	1	0	1	1	1	1	1	1
1	0	0	0	0	1	1	1	0	0	1

★ Did not include other 8 rows as we don't care about numbers greater than $1000_2 = 8_{10}$

I began by building the Logisim circuit first, then building the truth table. For each number between 0 and 8, there is a submodule of NOR and AND gates to ensure that the buttons that needs to be on are pressed (passed through regular AND) and the ones that need to be off are off (passed through NOR and then connected to the previously mentioned AND). These result in 8 different output streams, which each need to turn on certain segments. However, I needed to do this such that I wouldn't cross wires between any of the submodule outputs. Thus, I assigned each segment its own 8-input OR gate so that if it detected a signal from any of the 8 streams, it would turn on the segment and not cross wires with the other streams. Overall, no such algebra was really involved, more of a brute force to ensure no unwanted signals turned on. This also has the added effect of not producing any segment lights when a binary number past "1000" is turned on with button combos.