CE100 Lab Report 2

Adders And Hierarchy

Justin Fortner

Lab 1C

TU/TH 1:30-3:20

10/13/17

**Description:**

The purpose of this lab was to create a ripple-carry adder by creating one full adder and by using hierarchy implement this one adder 3 times. This ripple carry adder that consisted of three full adders would add together two 3-bit inputs designated by the switches on the board. The output of the adder was then converted from from binary to hex and displayed on the board’s 7-segment display.

**Methods:**

**Part 1-Adder:**

First a truth table for a full bit adder needed to be creates. Once obtaining two logic equations from this table, one for the sum and one for the carry out, they were implemented in a full adder module. Once one full adder was completed three of them were attached together in the top level module. This created the ripple-carry adder. Lastly switch inputs were connected to the inputs of the the full adder.

**Part 2-7-Segment Display Conversion & Display:**

This was started this by making a truth table taking the 4 outputs of the ripple-carry adder as inputs for the 7 segment converter. These four inputs were then converted to display on the 7 segment display. This was done by making a truth table with all possible inputs (0000-1111) and how each of these binary numbers would light up the individual segments of the display. From this truth table logic equations for each of the segments was derived. These equations were implemented in a new display module. The outputs of this module were linked up to the 7 segment display on the board so that the hex value would display.

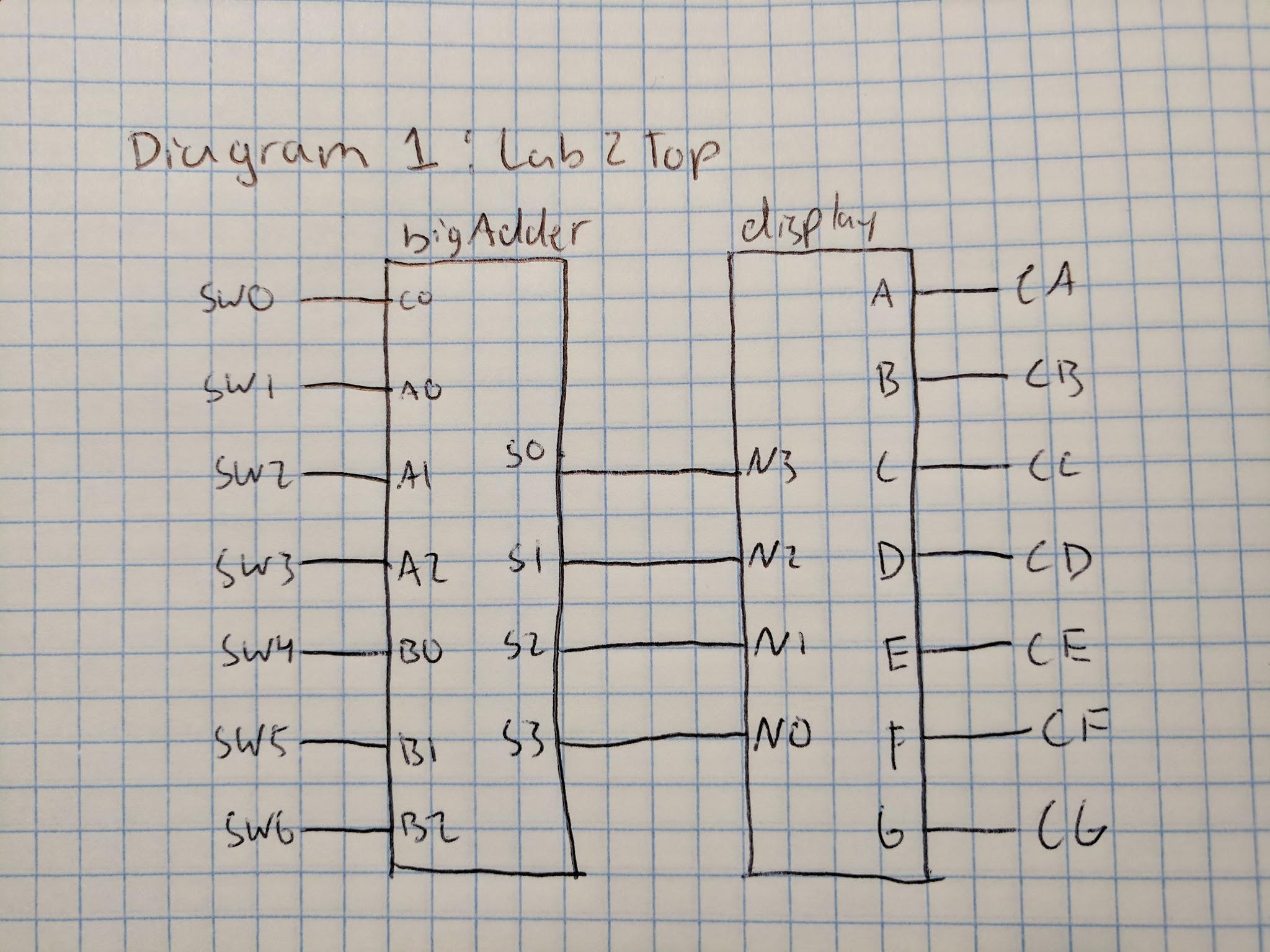
**Results:**

**Design:**

Heiarchy:

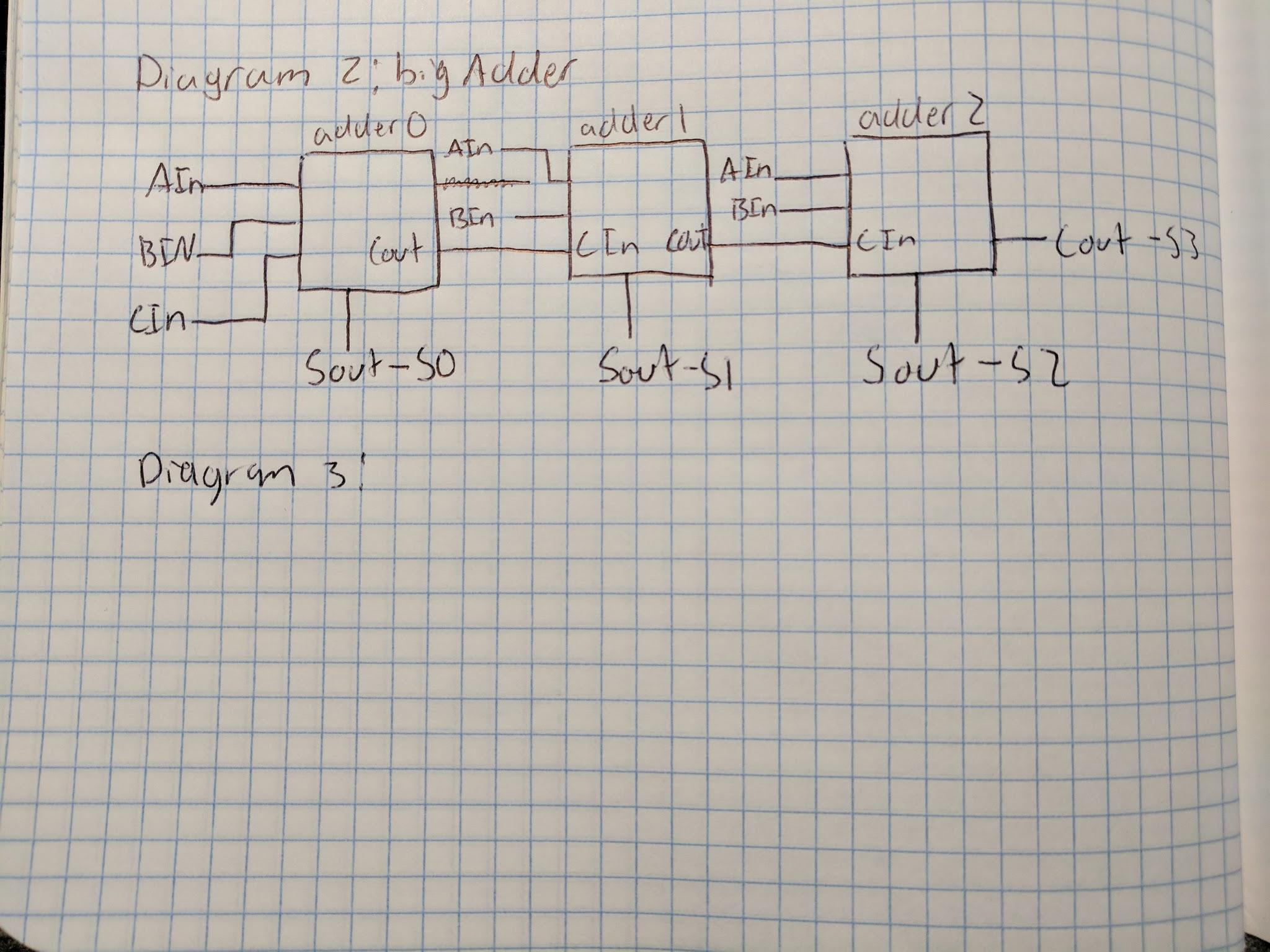
Lab2Top:

My Lab2Top consists of one bigAdder hierarchy and one display hierarchy. In addition to this, the inputs from the switches and outputs for the display can be seen.



bigAdder:

My bigAdder is the ripple-adder consisting of my three full adders. These adders are linked together by the carry out of the first adder to be the carri in of the second and the carry out of the second adder to the carry in of the third. The first adder accepting the rightmost bits of a and b. The second adder accepting the middle bits of a and b. The third adder accepting the leftmost bits of a and b. The outputs of these three adders are then linked to the 7-segment display converter.



FullAdder:

Within my FullAdder is just my implementation of the simplified equations derived from the truth table.

Table 1: Full Adder

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a | b | c | COut | SOut |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

This truth tale can be rewritten as two equations

SOut = (a XOR b) XOR c and COut = a AND b OR c AND (a XOR b)

display:

Within my display module is my implementation of my 7-segment display truth table. These equations allow for specified portions of the board’s LEDs to light up. This causes them to display the proper hexadecimal digit.

Table 2: 7-Segment Display

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | N3 | N2 | N1 | N0 | A | B | C | D | E | F | G |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 5 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 7 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 8 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| A | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| B | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| C | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| D | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| E | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| F | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |

Equations obtained from this truth table:

A = ~N0&~N1&~N2&~N3 | ~N0&~N1&N2&~N3 | ~N0&~N1&N2&N3 | ~N0&N1&~N2&N3 | ~N0&N1&N2&N3 | N0&~N1&~N2&~N3 | N0&~N1&~N2&N3 | N0&~N1&N2&~N3 | N0&N1&~N2&~N3 | N0&N1&N2&~N3 | N0&N1&N2&N3 | ~N0&N1&N2&~N3

B = ~N0&~N1&~N2&~N3 | ~N0&~N1&~N2&N3 | ~N0&~N1&N2&~N3 | ~N0&~N1&N2&N3 | ~N0&N1&~N2&~N3 | ~N0&N1&N2&N3 | N0&~N1&~N2&~N3 | N0&~N1&~N2&N3 | N0&~N1&N2&~N3 | N0&N1&~N2&N3

C = ~N0&~N1&~N2&~N3 | ~N0&~N1&~N2&N3 | ~N0&~N1&N2&N3 | ~N0&N1&~N2&~N3 | ~N0&N1&~N2&N3 | ~N0&N1&N2&~N3 | ~N0&N1&N2&N3 | N0&~N1&~N2&~N3 | N0&~N1&~N2&N3 | N0&~N1&N2&~N3 | N0&~N1&N2&N3 | N0&N1&~N2&N3

D = ~N0&~N1&~N2&~N3 | ~N0&~N1&N2&~N3 | ~N0&~N1&N2&N3 | ~N0&N1&~N2&N3 | ~N0&N1&N2&~N3 | N0&~N1&~N2&~N3 | N0&~N1&N2&N3 | N0&N1&~N2&~N3 | N0&N1&~N2&N3 | N0&N1&N2&~N3

E = ~N0&~N1&~N2&~N3 | ~N0&~N1&N2&~N3 | ~N0&N1&N2&~N3 | N0&~N1&~N2&~N3 | N0&~N1&N2&~N3 | N0&~N1&N2&N3 | N0&N1&~N2&~N3 | N0&N1&~N2&N3 | N0&N1&N2&~N3 | N0&N1&N2&N3

F = ~N0&~N1&~N2&~N3 | ~N0&N1&~N2&~N3 | ~N0&N1&~N2&N3 | ~N0&N1&N2&~N3 | N0&~N1&~N2&~N3 | N0&~N1&~N2&N3 | N0&~N1&N2&~N3 | N0&~N1&N2&N3 | N0&N1&~N2&~N3 | N0&N1&N2&~N3 | N0&N1&N2&N3

G = ~N0&~N1&N2&~N3 | ~N0&~N1&N2&N3 | ~N0&N1&~N2&~N3 | ~N0&N1&~N2&N3 | ~N0&N1&N2&~N3 | N0&~N1&~N2&~N3 | N0&~N1&~N2&N3 | N0&~N1&N2&~N3 | N0&~N1&N2&N3 | N0&N1&~N2&N3 | N0&N1&N2&~N3 | N0&N1&N2&N3

**Testing & Simulation:**

The code was tested using the following inputs and received the following outputs. These inputs were chosen because they thoroughly test my carry in, all three a inputs (S1-S3) and all 3 inputs (S4-S6). No corner cases needed to be considered because the truth table already accounts for minimum and maximum outputs (from 0000000 to 1111111).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **S6** | **S5** | **S4** | **S3** | **S2** | **S1** | **CIn** | **Sout** |
| **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **0** | **0** | **0** | **0** | **0** | **0** | **1** | **1** |
| **0** | **0** | **0** | **0** | **0** | **1** | **1** | **2** |
| **0** | **0** | **1** | **0** | **0** | **1** | **1** | **3** |
| **0** | **1** | **0** | **0** | **1** | **0** | **0** | **4** |
| **0** | **1** | **1** | **0** | **0** | **1** | **1** | **5** |
| **0** | **1** | **1** | **0** | **1** | **0** | **1** | **6** |
| **0** | **1** | **1** | **0** | **1** | **1** | **1** | **7** |
| **1** | **0** | **0** | **0** | **1** | **1** | **1** | **8** |
| **1** | **0** | **1** | **0** | **1** | **1** | **1** | **9** |
| **1** | **0** | **1** | **1** | **0** | **1** | **0** | **A** |
| **1** | **0** | **1** | **1** | **0** | **1** | **1** | **B** |
| **1** | **0** | **1** | **1** | **1** | **0** | **1** | **C** |
| **1** | **0** | **1** | **1** | **1** | **1** | **1** | **D** |
| **1** | **1** | **1** | **1** | **1** | **1** | **0** | **E** |
| **1** | **1** | **1** | **1** | **1** | **1** | **1** | **F** |

**Lab Questions:**

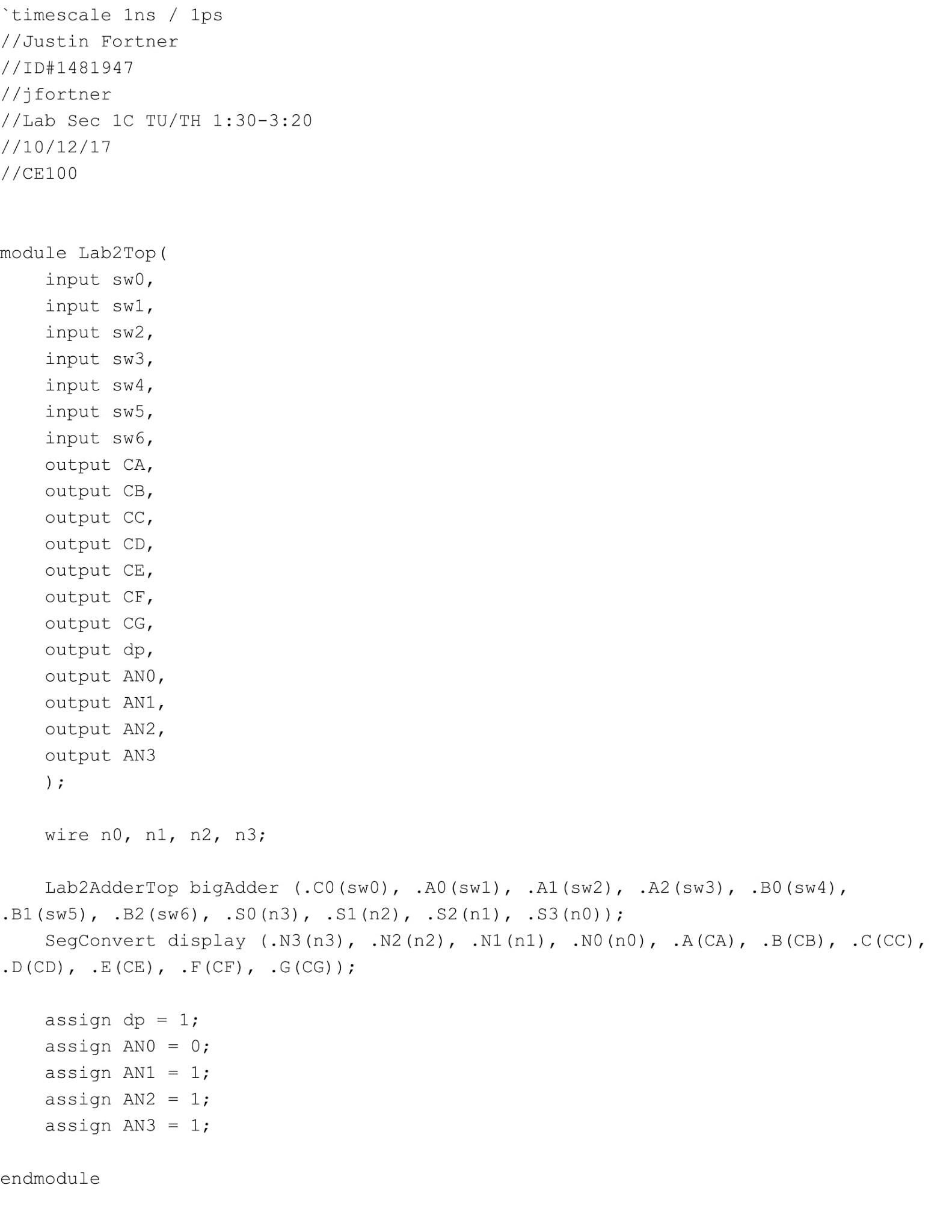
No questions from the lab.

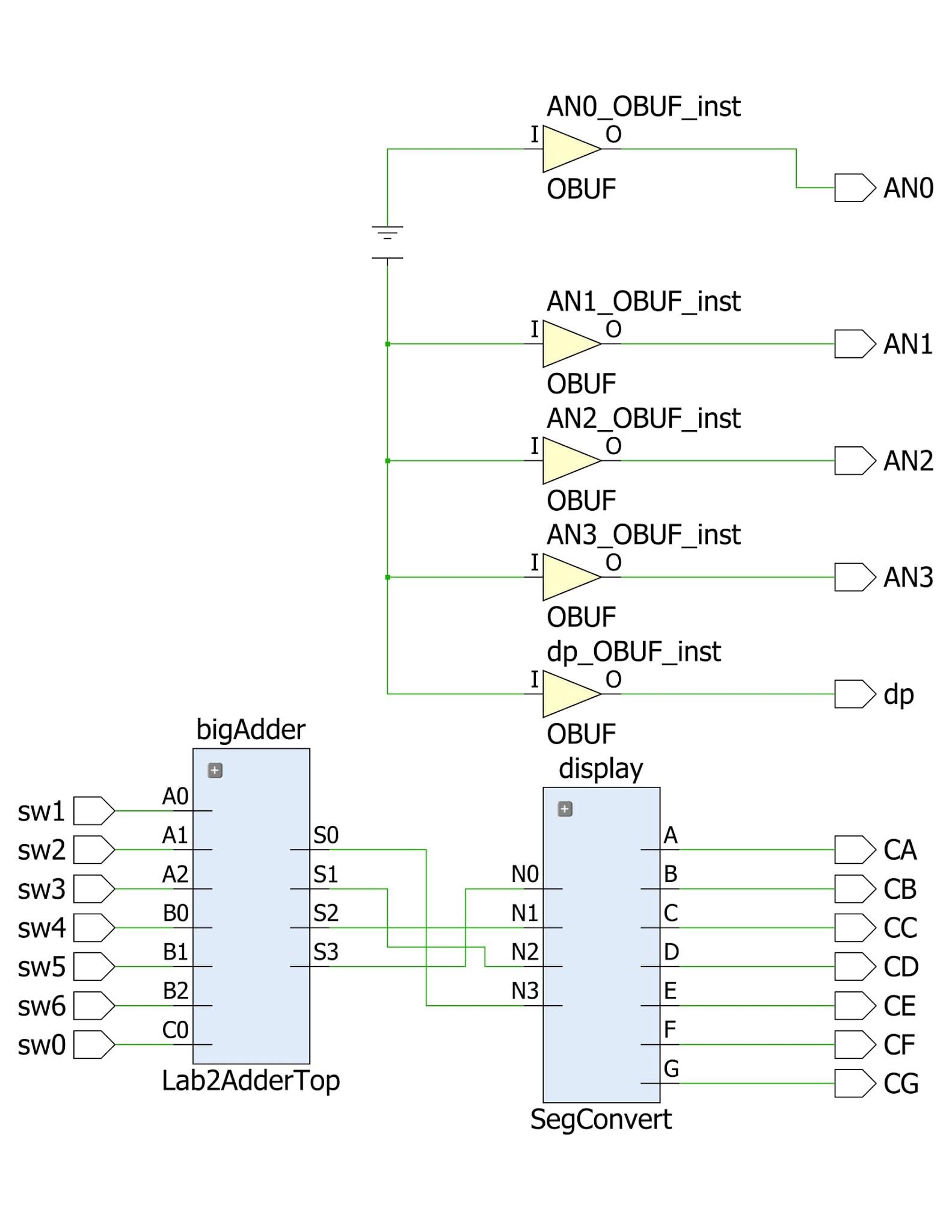
**Conclusion:**

I learned about hierarchy, wires, and the board 7-segment display in this lab. My knowledge of adders was reinforced as well. I did not have any difficulties with this lab. It was time consuming with the large truth tables and equations, but no difficulties. I completed this lab early and went back and added my three full adders to a top level, bigAdder, that was then in the whole labs top level, Lab2Top. Thus there is nothing I would do differently for the next lab. I would like to optimize the equations for the individual segments of the 7-segment LCD display. However, as stated in the lab manual, it would not help very much. Overall I believe this was a very helpful and informative lab.

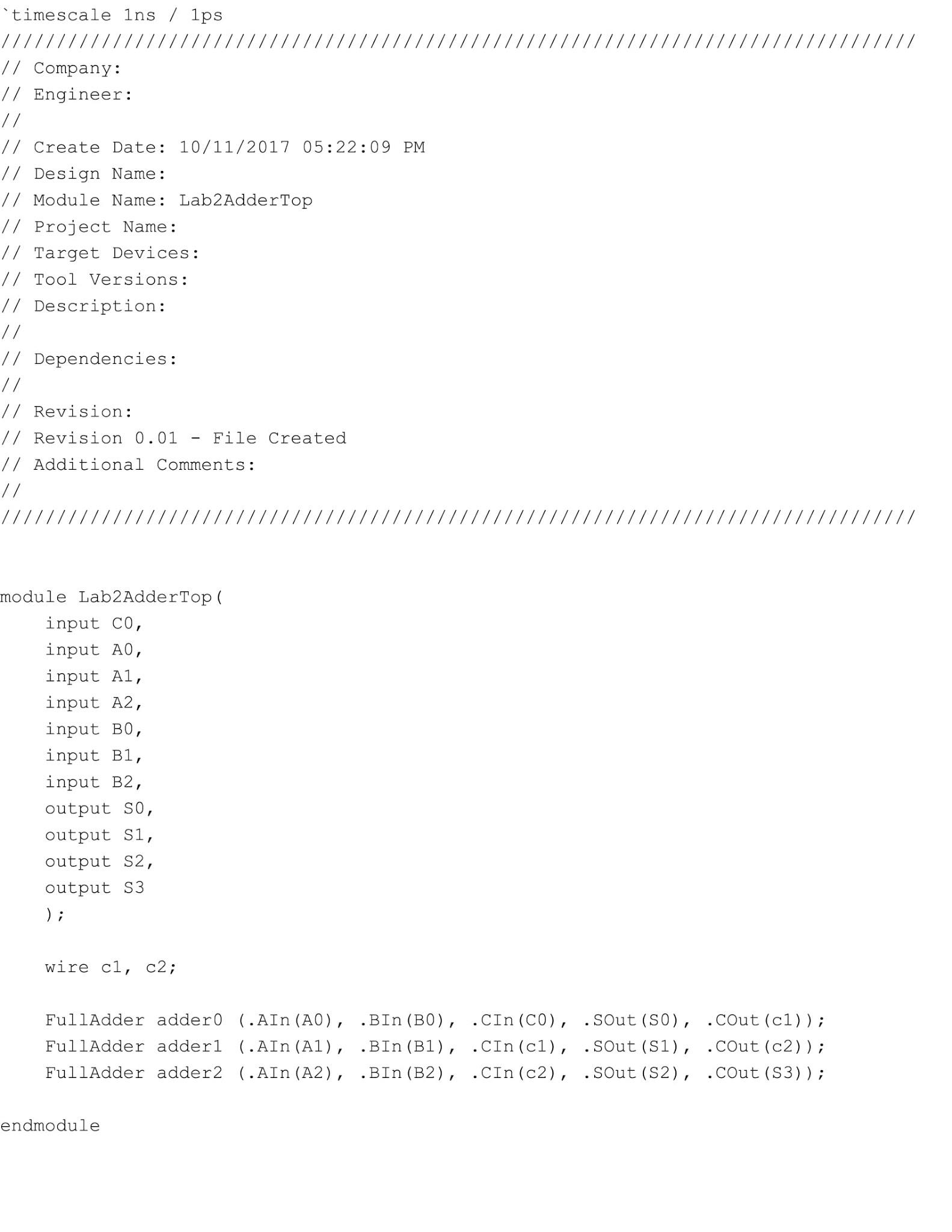
**Appendix: Next Page**

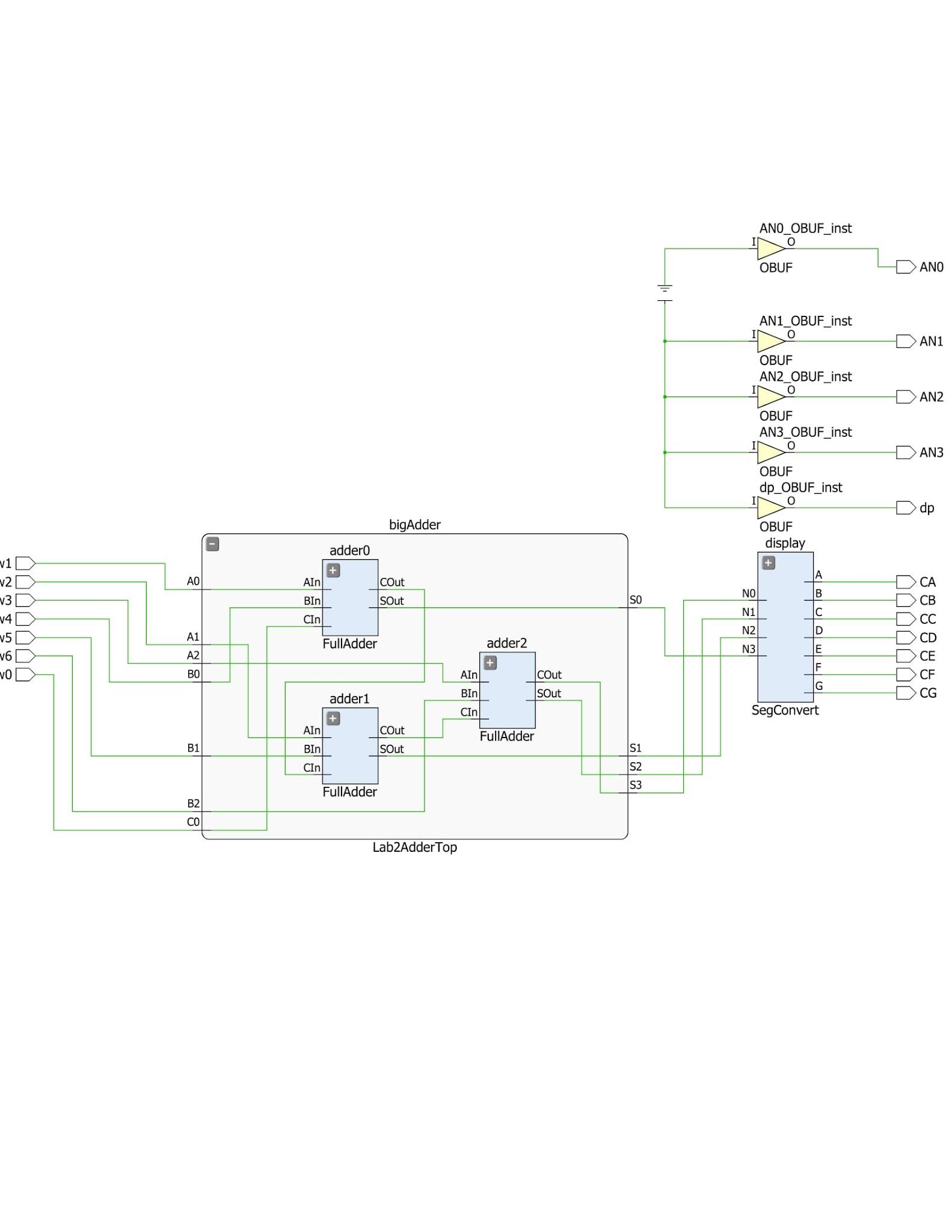
**Lab2Top:**

****

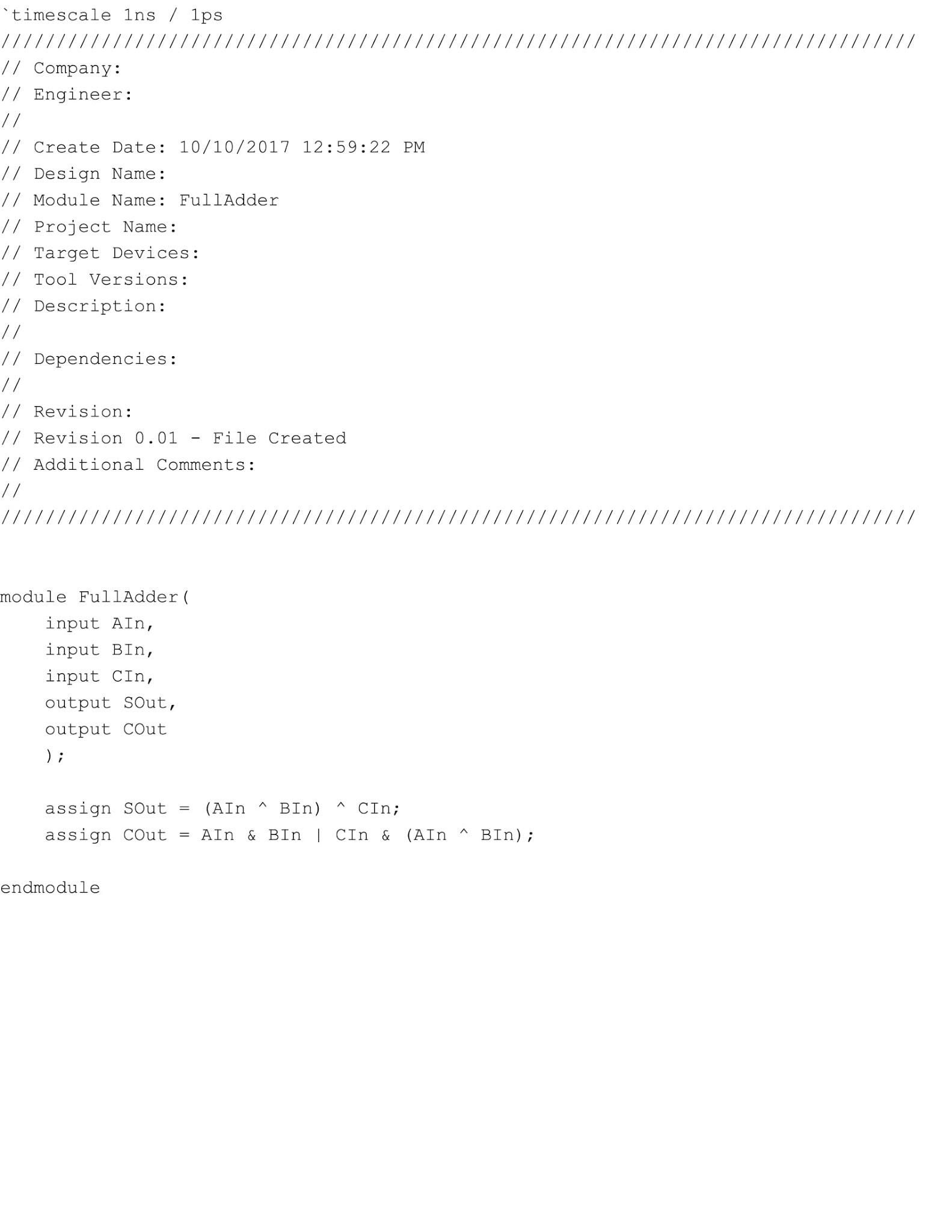
****

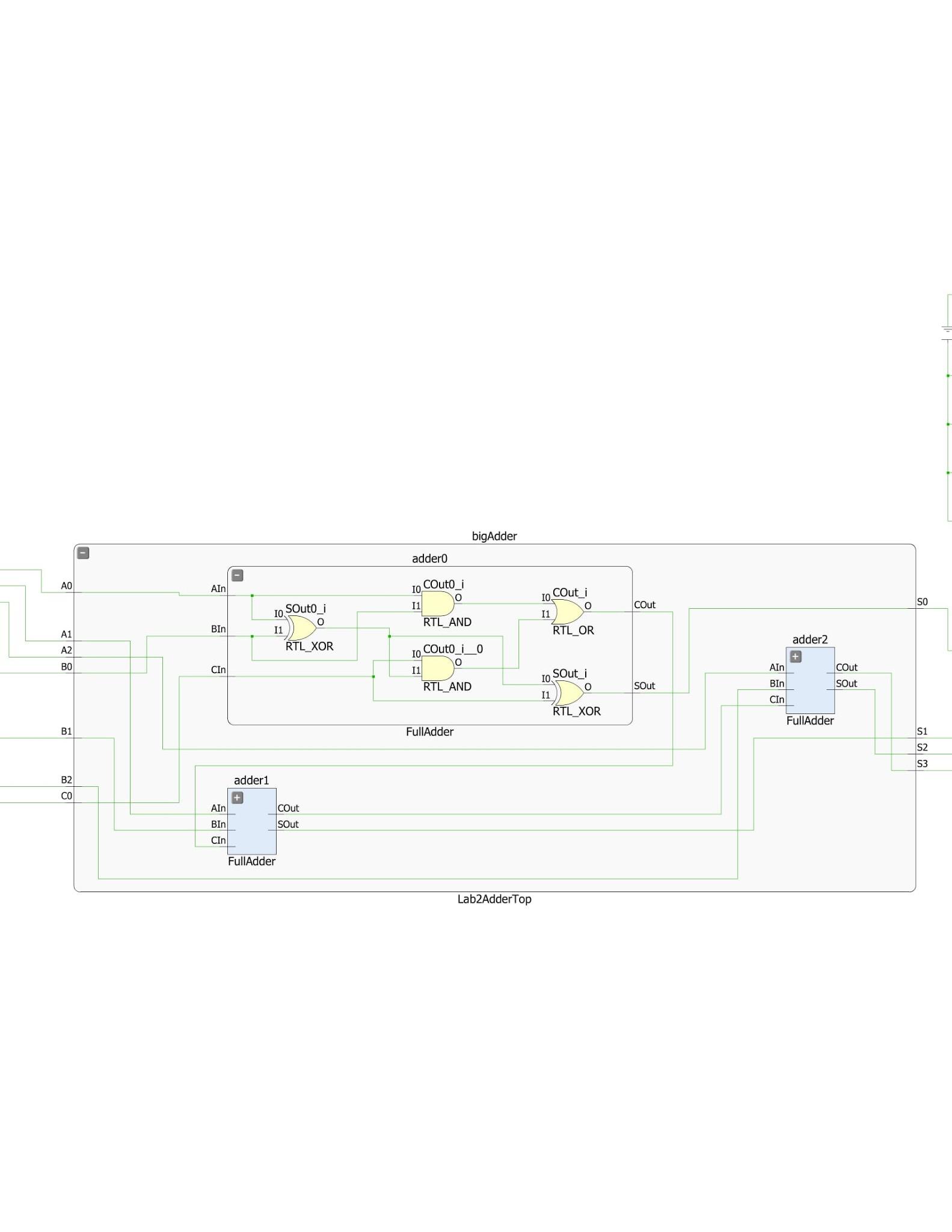
**bigAdder:**

****

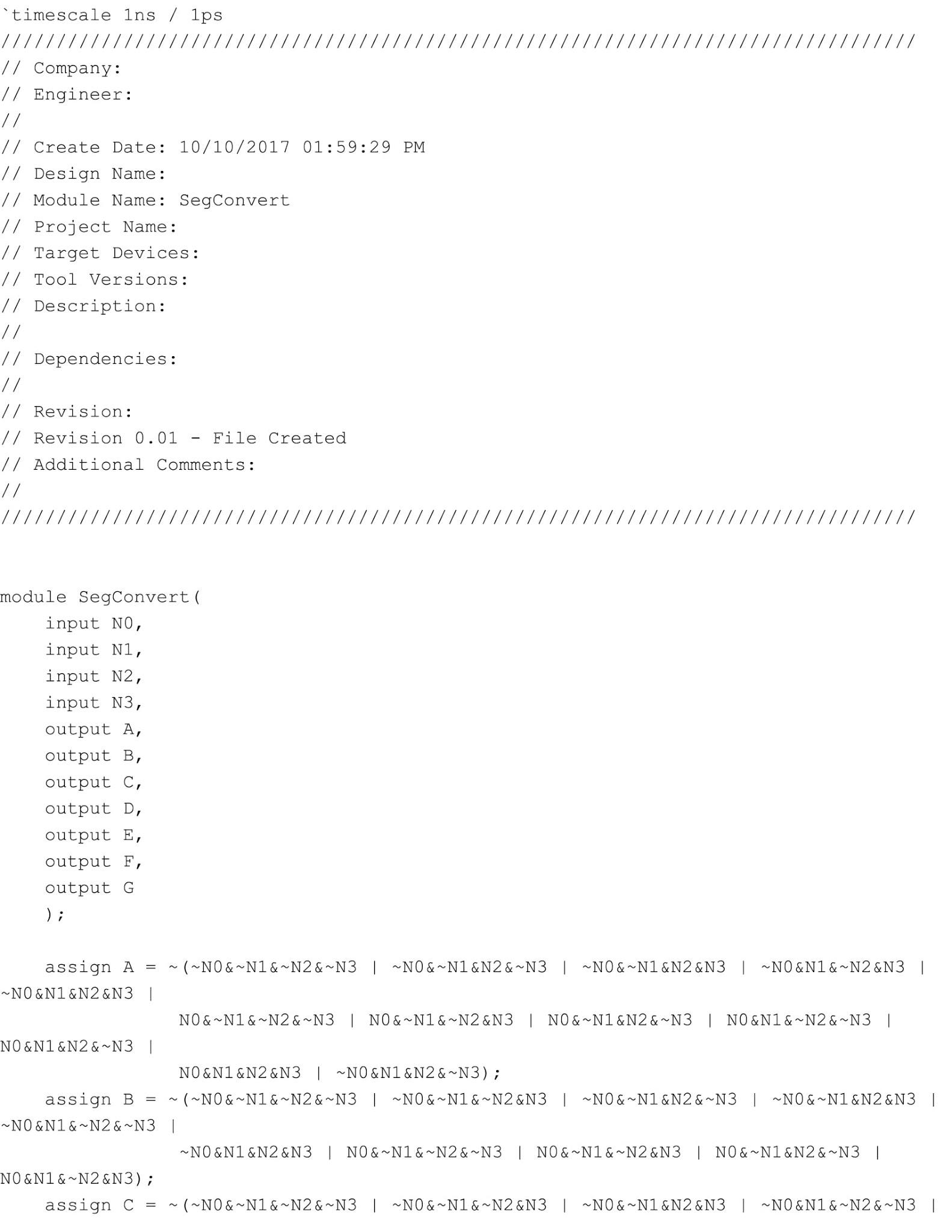
****

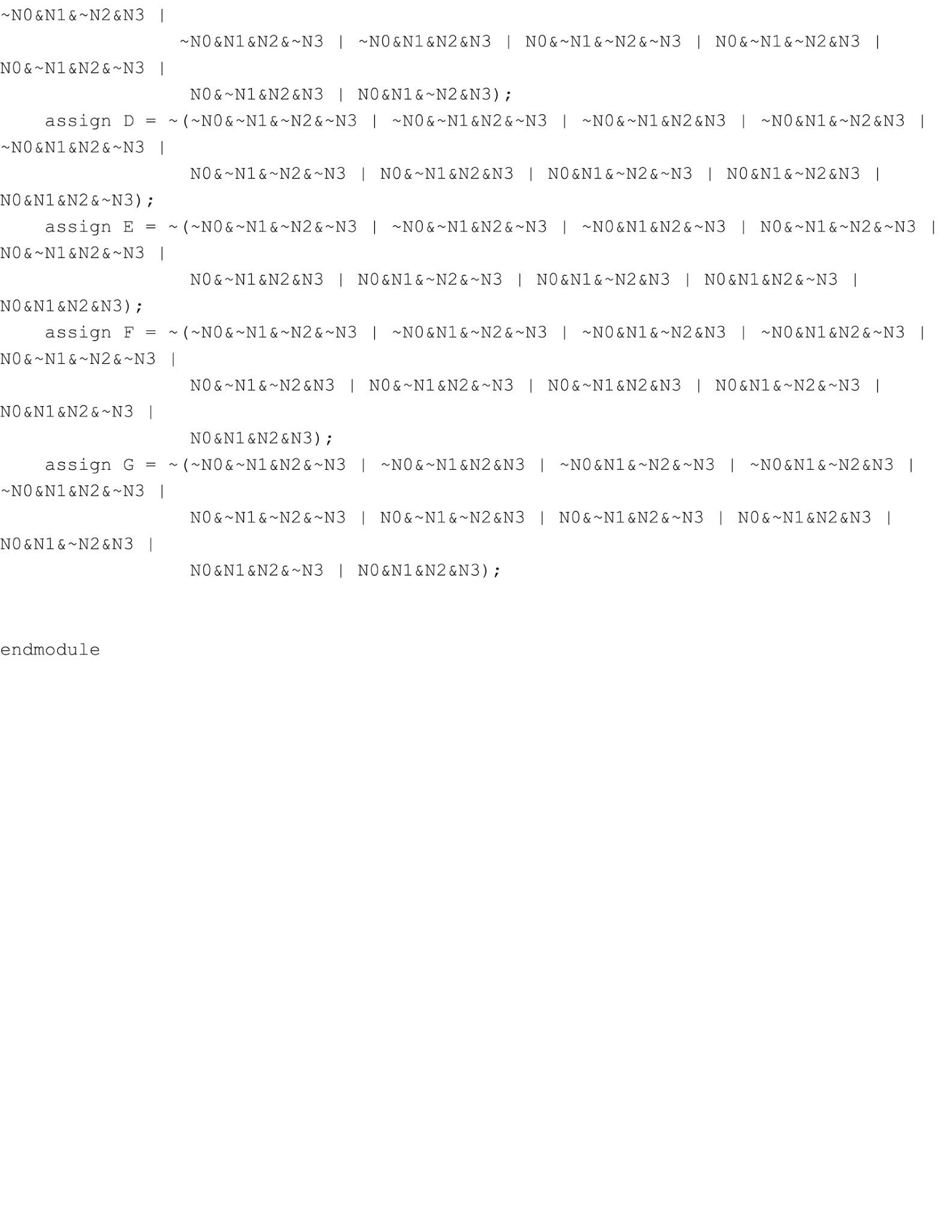
**FullAdder:**

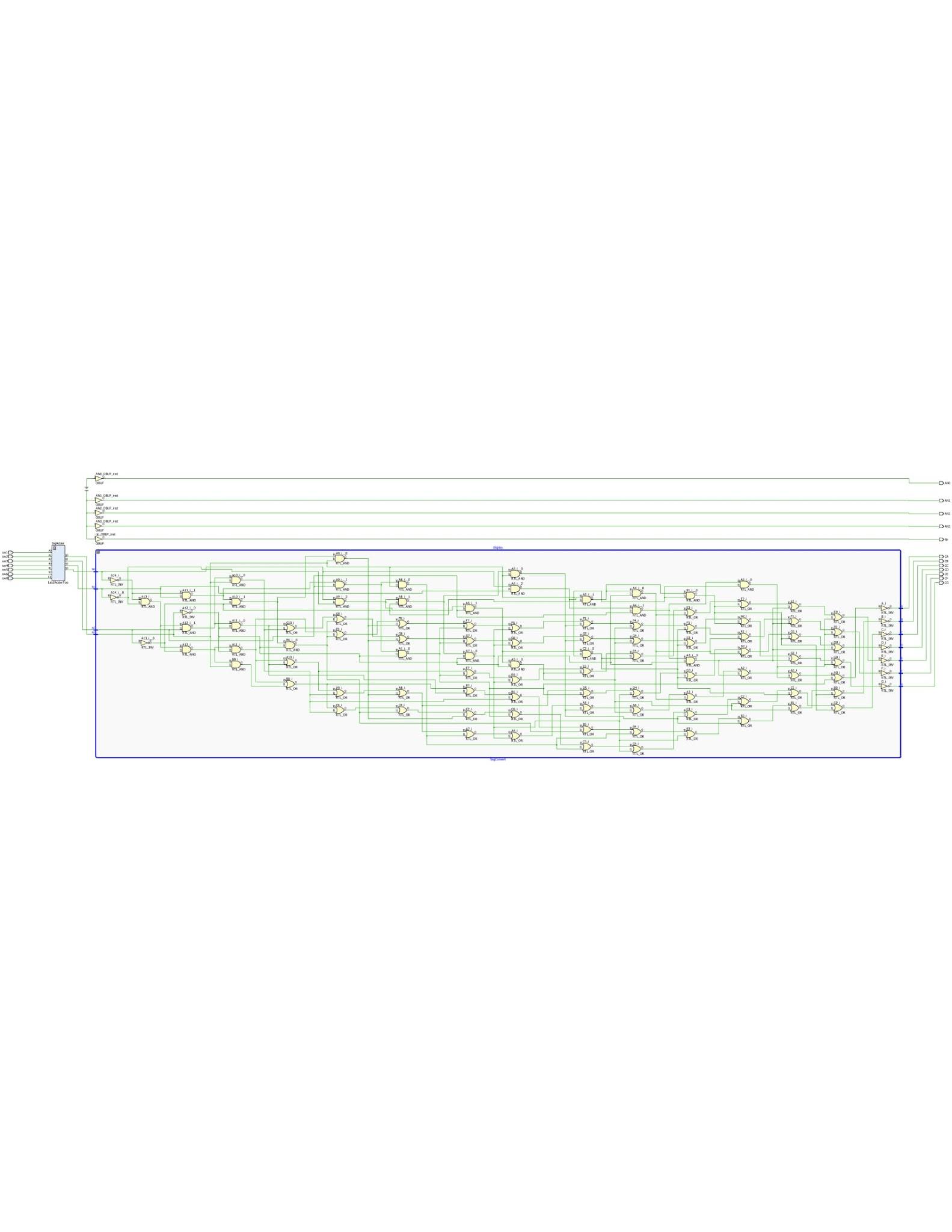
****

****

**Display:**

****

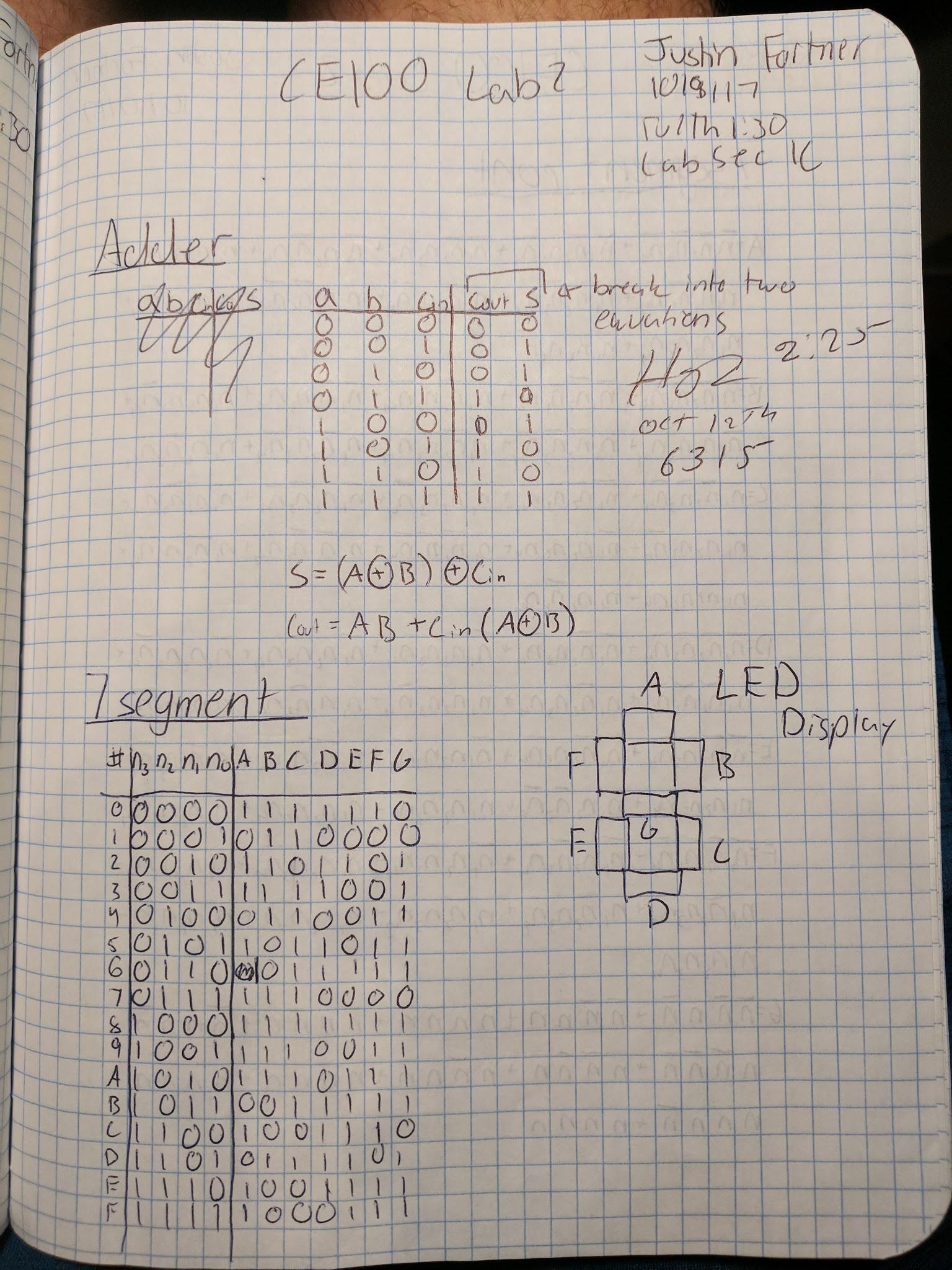
****

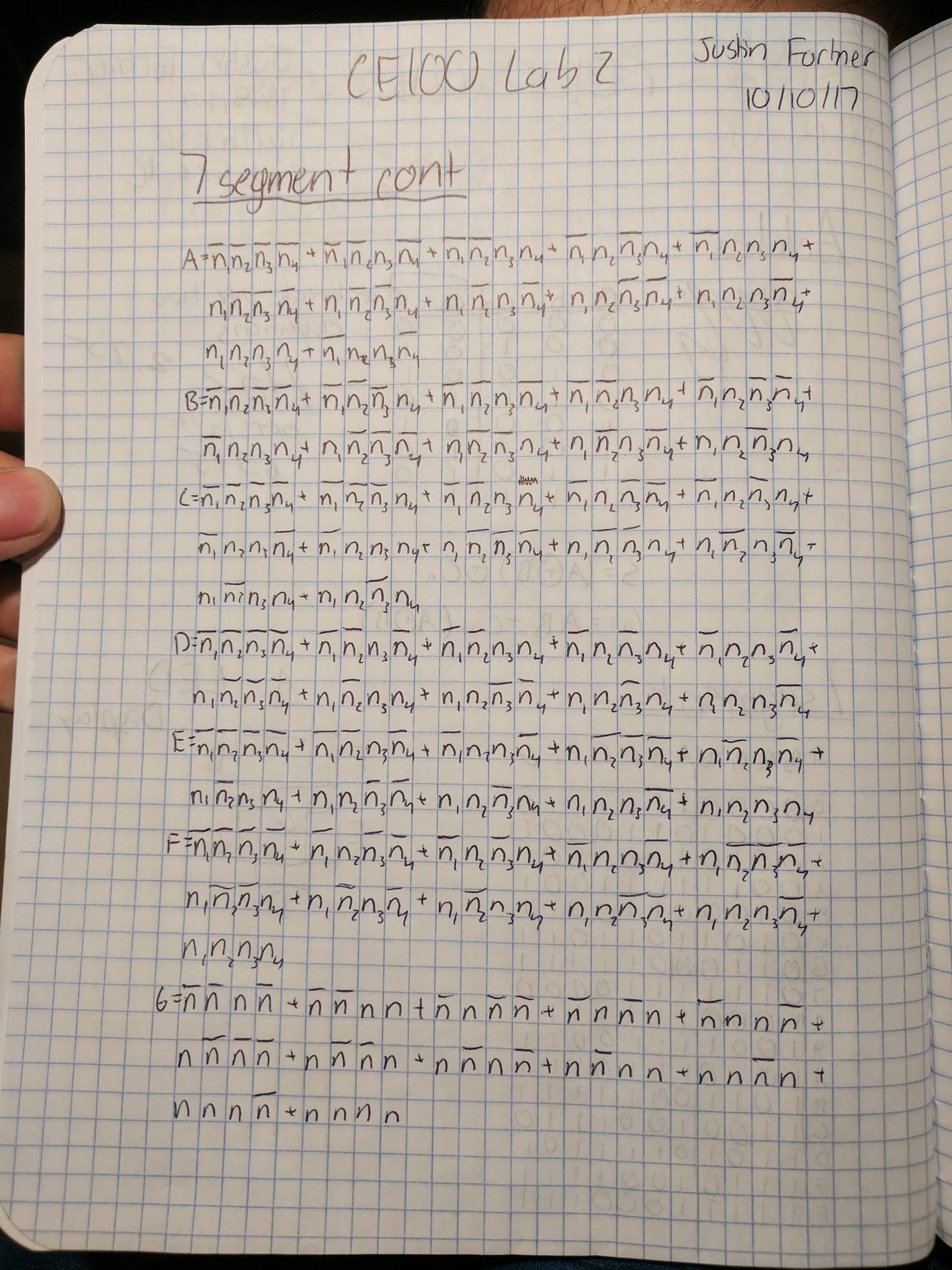
****

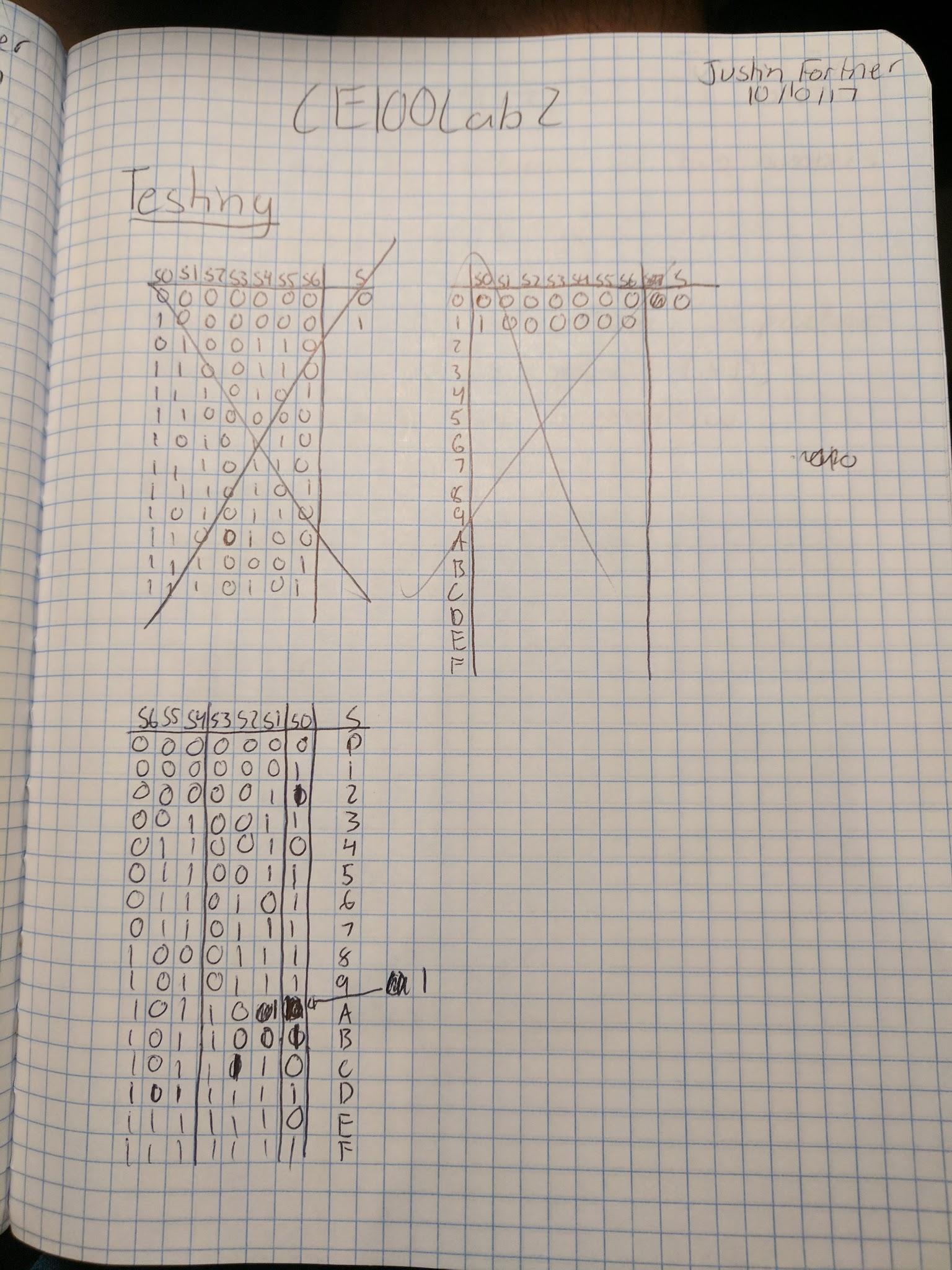
**Waveform:**

****

**Notebook:**

****

****

****