ELEN 153: Digital Integrated Circuit Des 59114

Lab 6: Hierarchical Design Part 1 - Four-Bit Adder

Yutong Li

Wednesday 2:15-5:00pm 2017/11/1

I. OBJECTIVE

- To learn how to create complex circuits using bottom-up hierarchical design.
- To create a four-bit adder schematic and simulate it.

II. LAB PROCEDURE

1. Importing gates

In order to use the provided AND, OR, and XOR gate, we need to open the custom compiler in the Adder file.

2. Half Adder

From the truth table of the half adder, I can out with the equation for outputs C and S, due to inputs X and Y:

$$C = XY$$

$$S = XY' + X'Y = X \oplus Y$$

Based on these equations, I create the schematic for half adder as shown in Figure 2.
Create a symbol of half adder so we can use it in full adder.

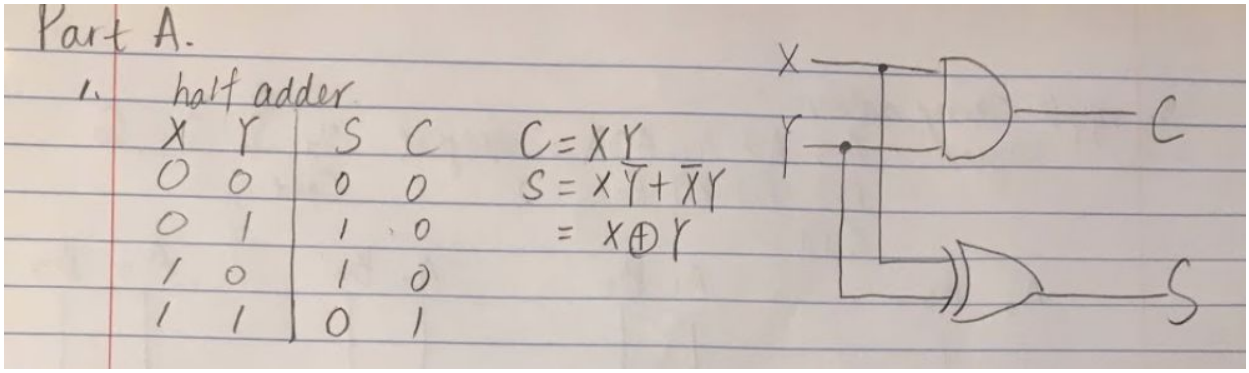

Figure 1. Design of half adder

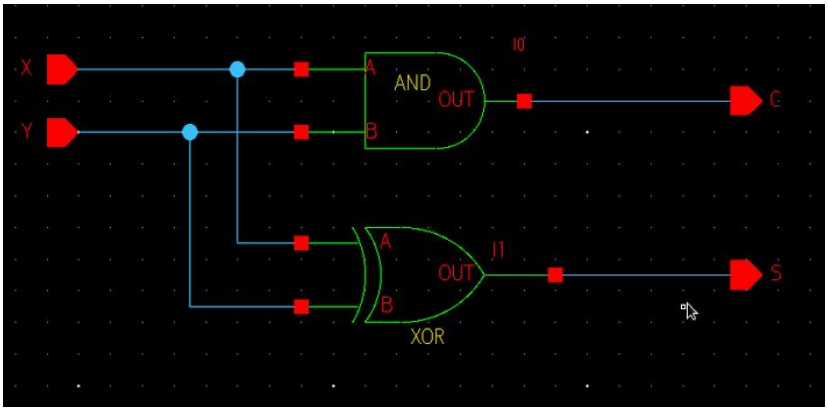

Figure 2. Schematic of half adder

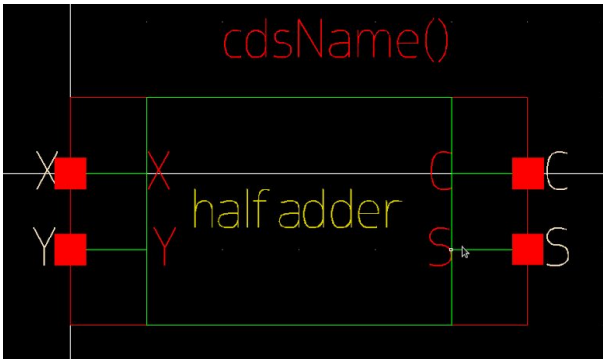

Figure 3. Symbol of half adder

3. Full Adder

From the truth table of full adder, I derive the equations for output Si and Ci+1, due to the inputs Ai, Bi, and Ci:

$$Si = Ai \oplus Bi \oplus Ci$$

$$Ci = AiBi + Ci(Ai \oplus Bi)$$

Since in the half adder, the equation for outputs C and S are:

$$C = XY$$

$$S = XY' + X'Y = X \oplus Y$$

We can use two half adder and one OR gate to create the full adder. The detailed output of each gate is shown in the bottom of Figure 4. However, in Figure 4, I have S on top of C in the half adder symbol, but I have C on top of S in the half adder symbol in Figure 3, which is my actual design, the actual schematic for full adder may look slightly different from that in Figure 4. Create a symbol for the full adder so we can use it for the ripple carry adder.

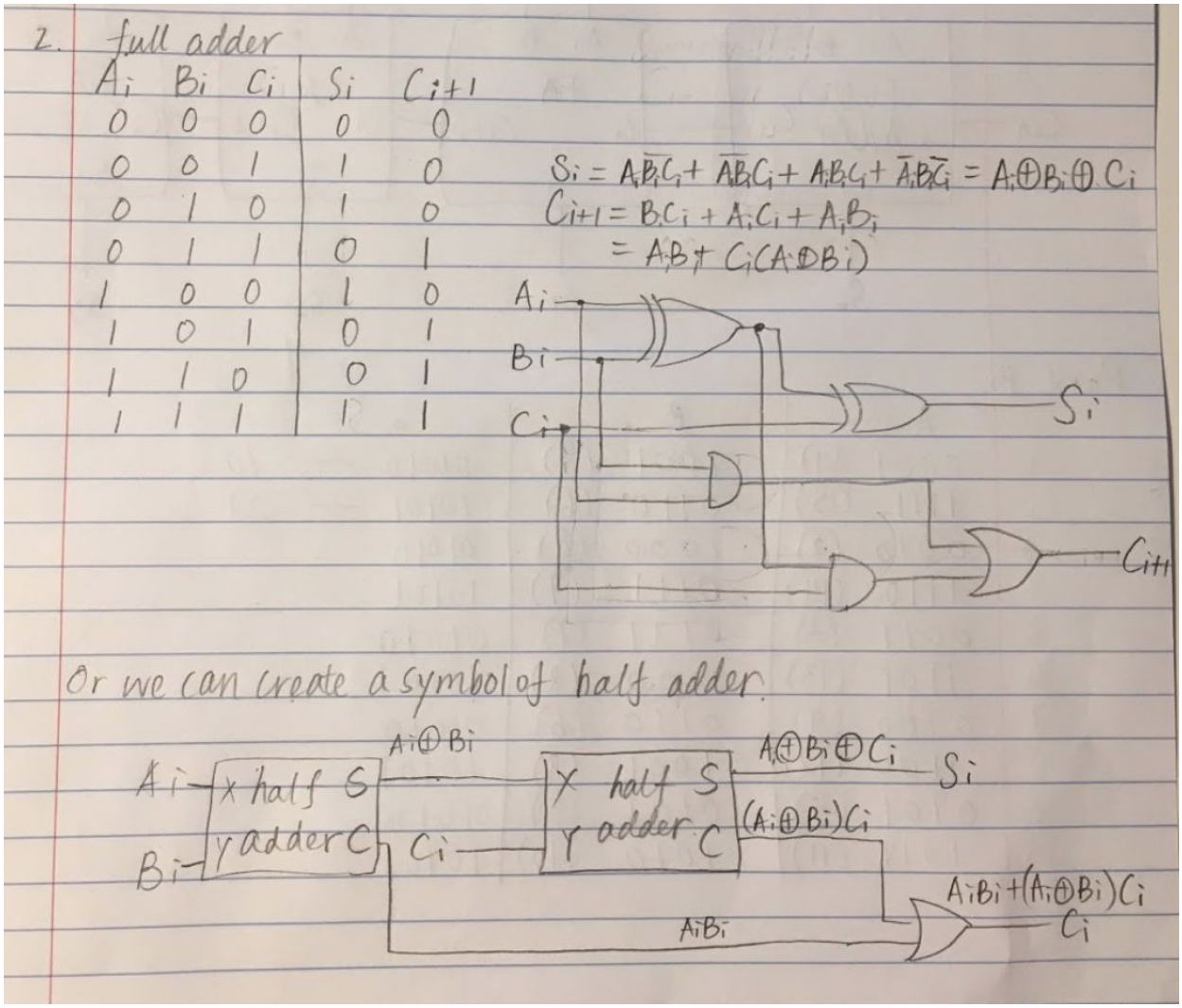

Figure 4. Design of full adder

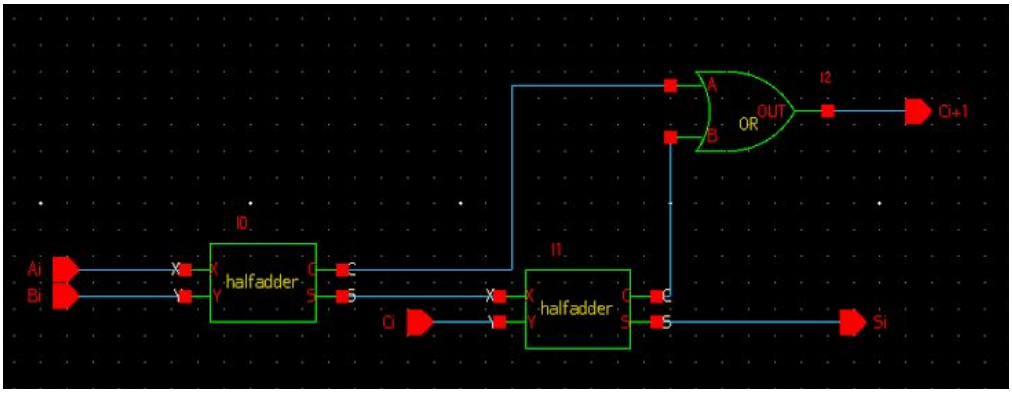

Figure 5. Schematic of full adder

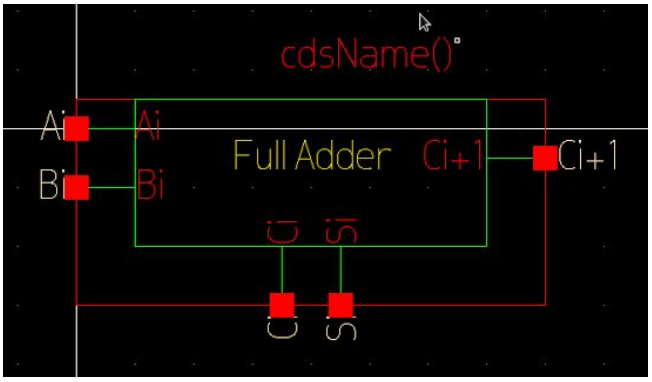

Figure 6. Symbol of full adder

4. Ripple Carry Adder

Create the schematic of ripple carry adder using 4 full adders. Note that the C_{i+1} from the former full adder should be the Ci for the next full adder, and the last C_{i+1} is Cout.

Create a symbol for the ripple carry adder so we can use it for four bit adder.

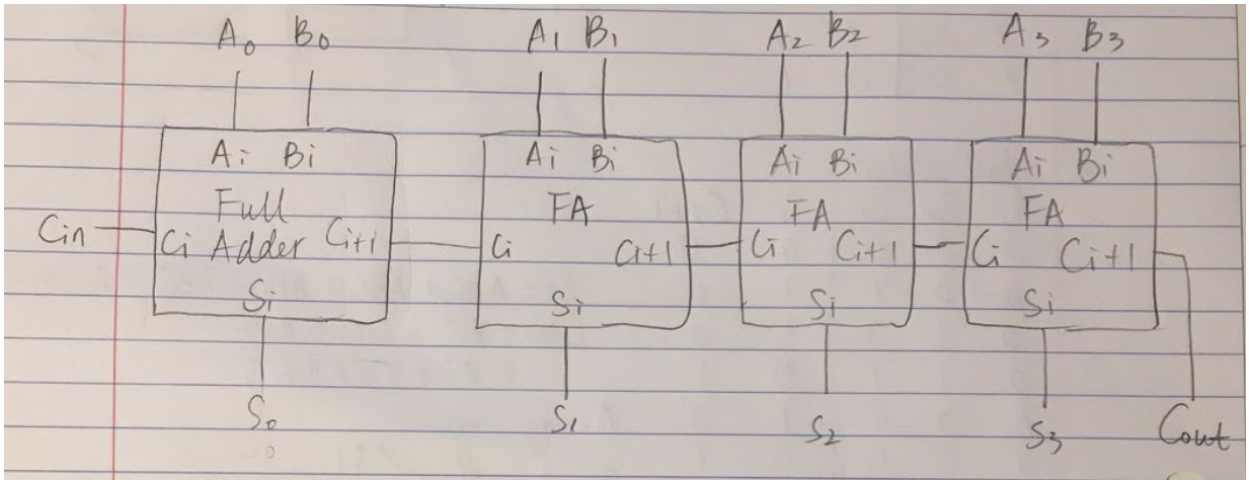

Figure 7. Design of ripple carry adder

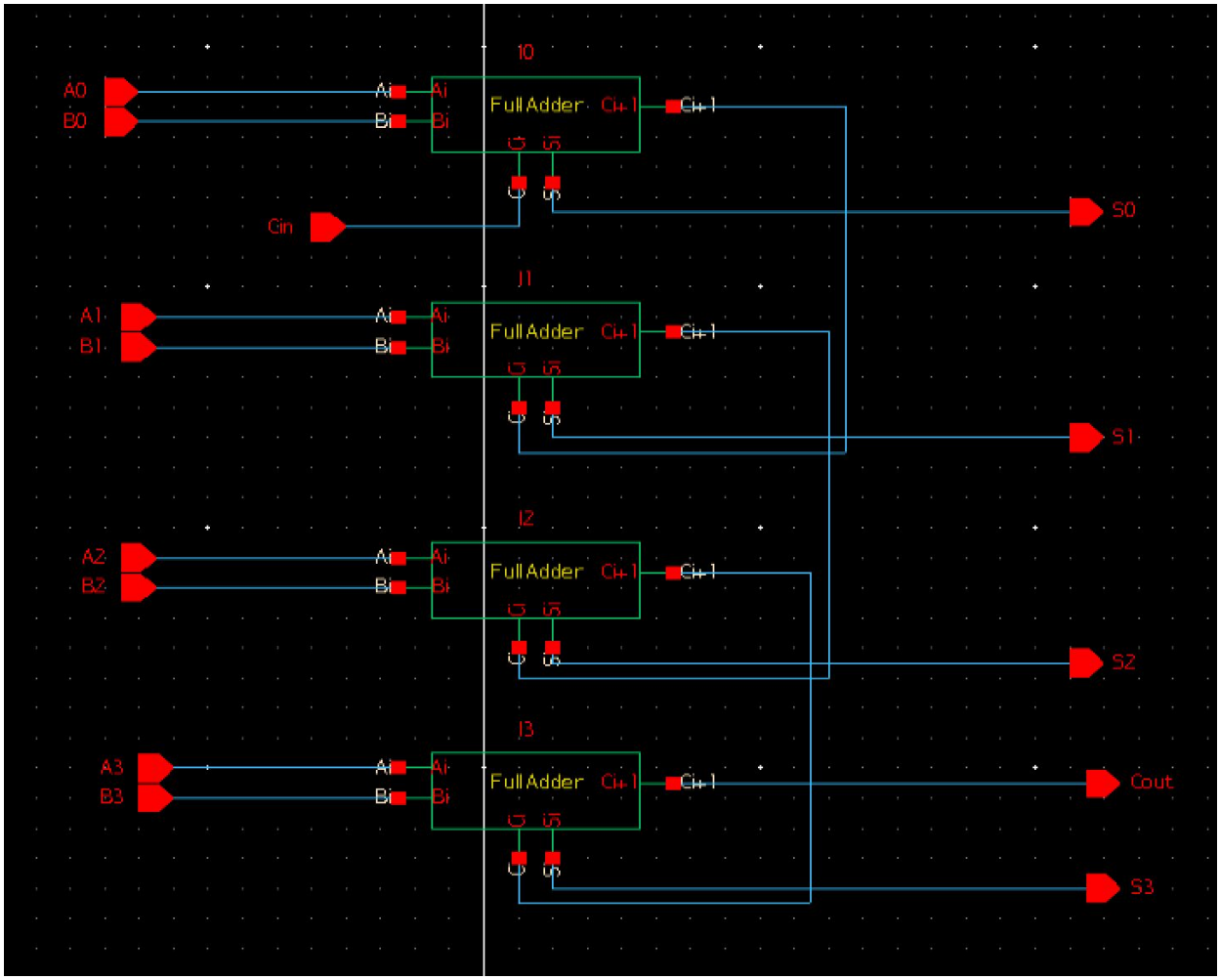

Figure 8.Schematic of ripple carry adder

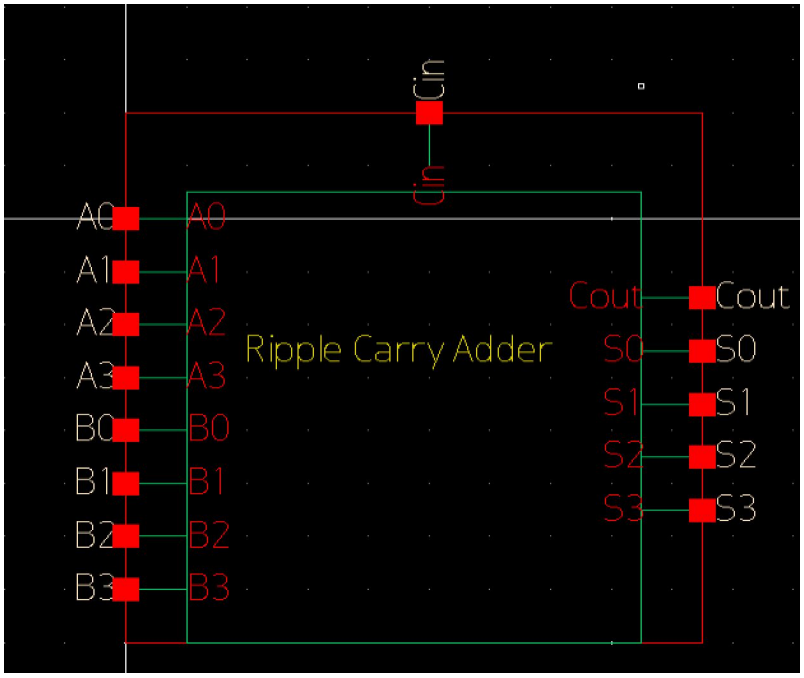

Figure 9. Symbol of ripple carry adder

5. Simulation

Connect all 8 inputs with V_{PAT}, and given each inputs the patterns shown in Figure 11 and table 1. Cin should be connected to ground so we can have 0 as Cin all the time.

In SAE simulation, setup transient simulation and choose all inputs (A0-A3, B0-B3) and all outputs (Cout, S0-S3) as shown in Figure 12. In the waveform, if the order of the output is Cout on the top, then S0, S1, S2, S3, we should see the pattern of output shown in Table 1.

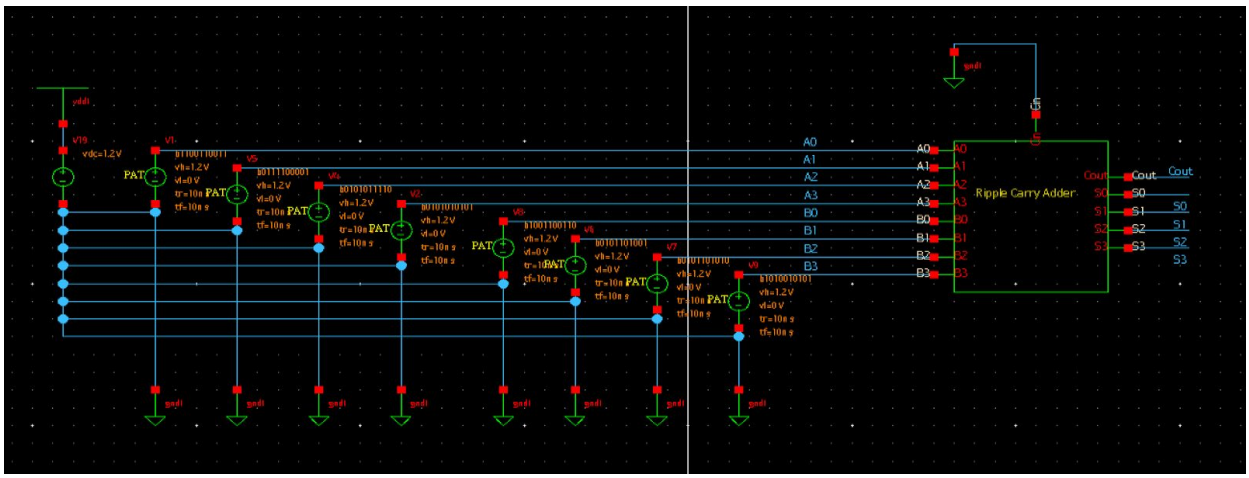

Figure 10. Schematic of four-bit ripple carry adder

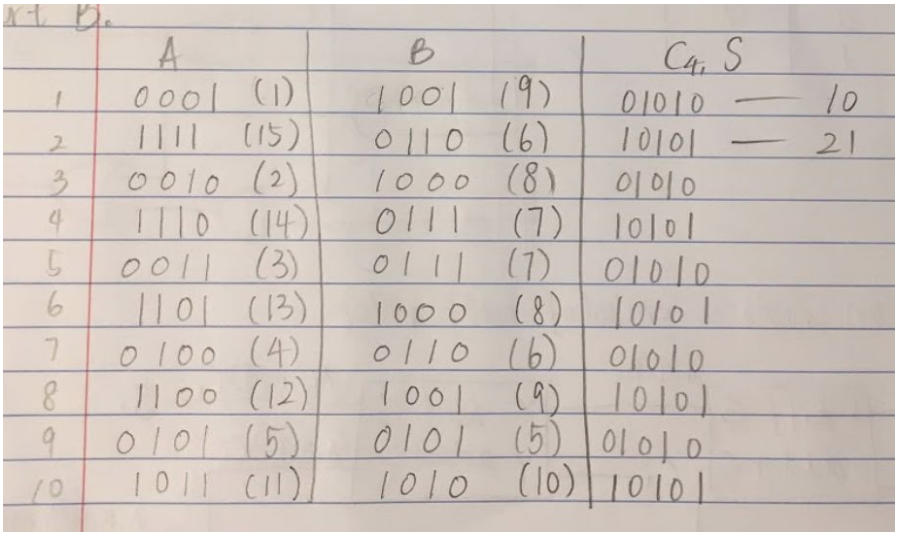

Figure 11. Table of inputs for desired outputs (10 and 21)

Table 1. Table of inputs for desired output (10 and 21)

input	pattern	Cout S3 S2 S1 S0
A0	1100110011	01010
A1	0111100001	10101
A2	0101011110	01010
A3	0101010101	10101
B0	1001100110	01010
B1	0101101001	10101
B2	0101101010	01010
B3	1010010101	10101

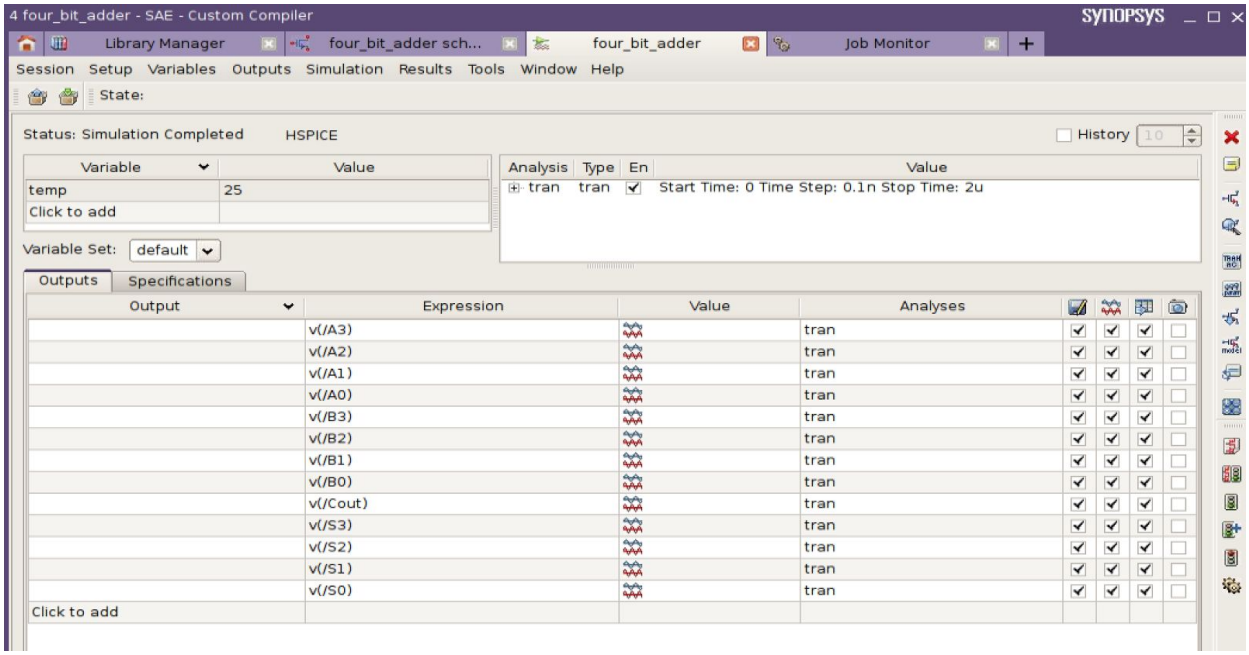

Figure 12. SAE simulation setup

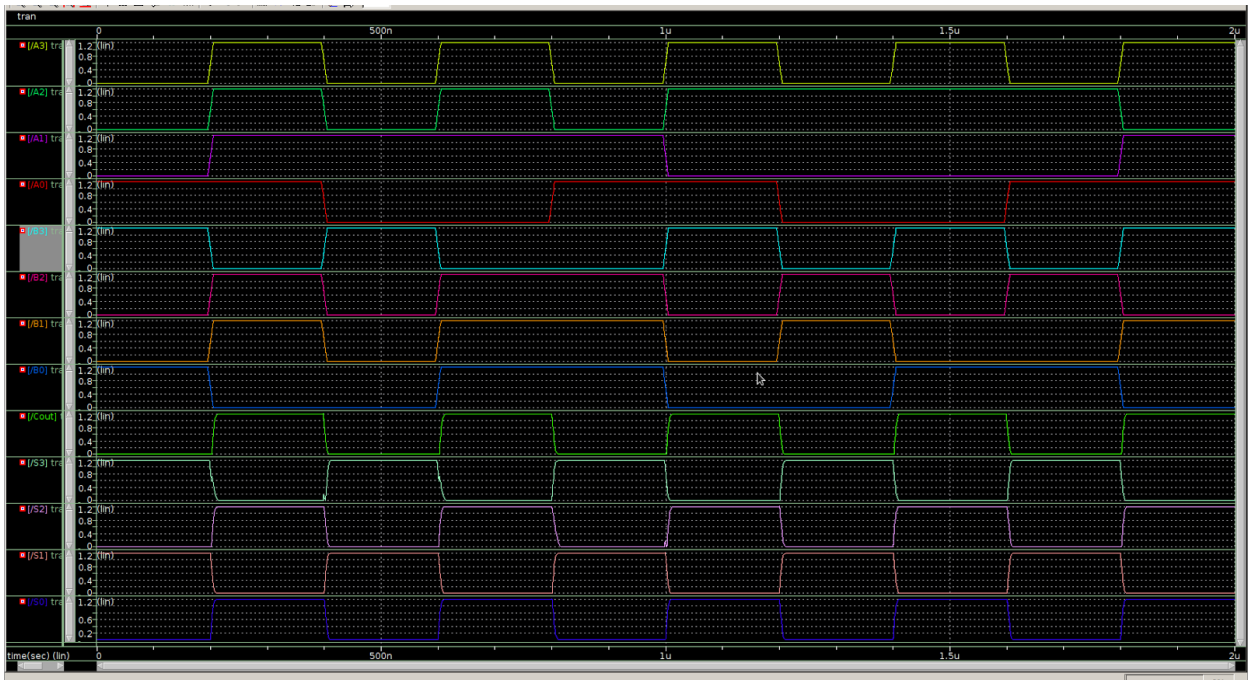

Figure 13. Waveform of four bit adder simulation

Question: How can you make an 8-bit adder using your 4-bit ripple carry adder?
For example, we have A0-A7 and B0-B7 as inputs.

1. Have A3-A0 and B3-B0 as the inputs of a 4-bit ripple carry adder, and Cin is 0. Cout goes into the next adder as Cin. S3-S0 will be the least significant half of the final output.

2. Have A7-A4 and B7-B4 as the inputs of a 4-bit ripple carry adder, and Cin is the Cout of the former adder. Cout and S3-S0 will be the most significant half of the final output.

III. CONCLUSION

In this lab, I learn how to create complex circuit using hierarchy design. But because we are using hierarchy design, it is important to know the connection between the designs (like how can we use half adder to create full adder), and make sure the symbol that we use for the next circuit is correct. Also, the arrangement of the inputs and outputs in the schematic will determine how the symbol will look like, so in order for the final schematic to look neat, having inputs and outputs on different sides of the symbol will help. Moreover, the order of the inputs and outputs is important, make sure whether it is from the least to the most significant bit or the other way.