

EIE2810 Digital Systems Design Laboratory

Laboratory Report #4

Name: Student ID:
Date: 2024.3.24

The Chinese University of Hong Kong, Shenzhen

- Experiment A: build up a full-adder in Multisim
- Experiment B: learn to use 7-segment display, and BCD-to-7-segment decoder (74LS47)
- Experiment C: learn to use 4-line-to-16-line decoder (74HC154)
- Experiment D: build an integrated 2-bit x 2-bit multiplier with SIM, 74HC154, 74LS47, and 7-segment indicator

1. Experiment A

1.1 Design

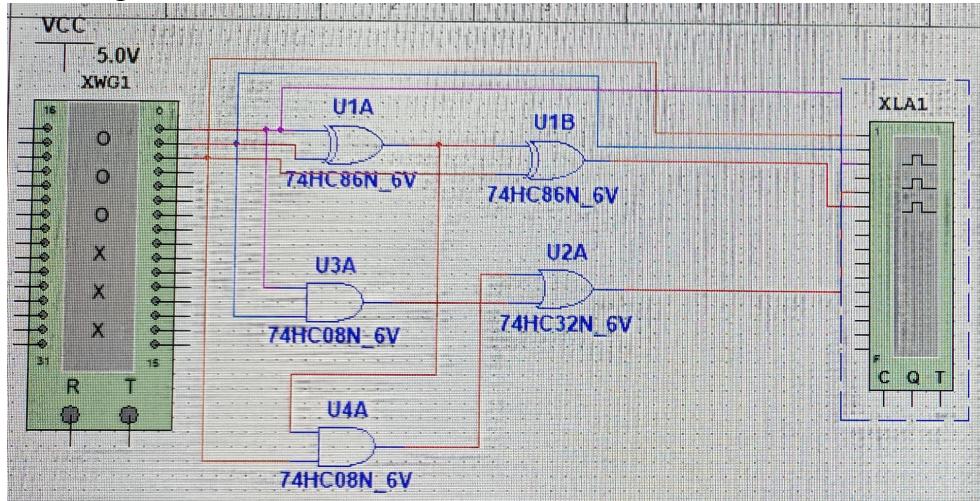


Figure 1. Circuit design (the outputs of the Word Generator are Cin, B, A, from up to down)

1.2 Result

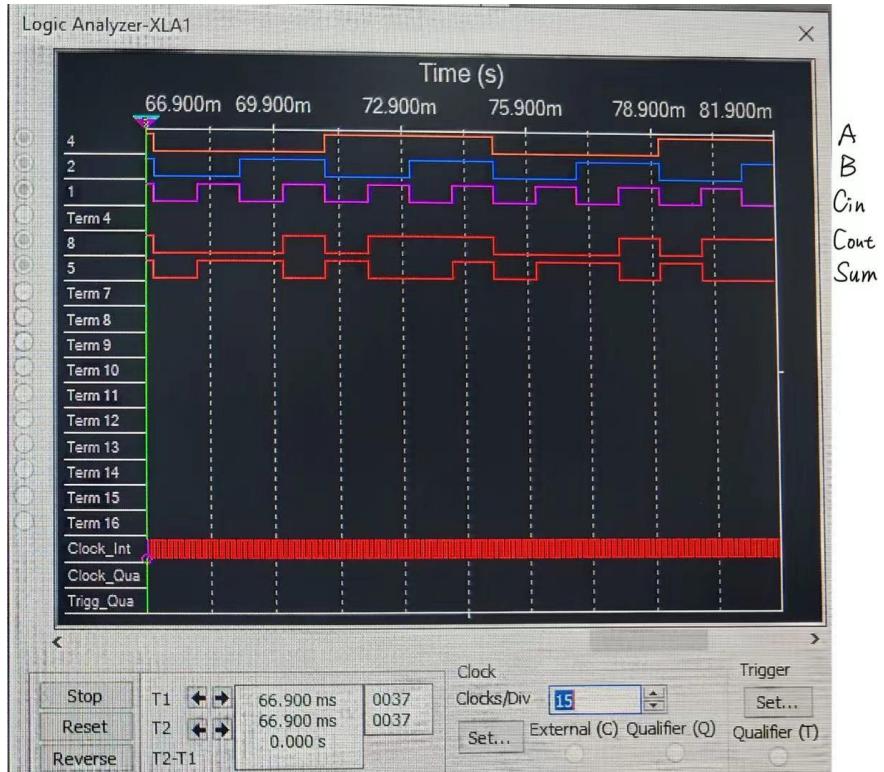


Figure 2. Waveforms of circuit in Figure 1

Analysis:

In Figure 2, when A, B, Cin are all 0, Cout and Sum are both 0;

When (A, B, Cin) is (0, 0, 1) or (0, 1, 0) or (1, 0, 0), Cout is 0, Sum is 1;

When (A, B, Cin) is (0, 1, 1) or (1, 0, 1) or (1, 1, 0), Cout is 1, Sum is 0;

When A, B, Cin are all 1, Cout and Sum are both 1;

The result is consistent with the truth table below:

Input			Output	
A	B	C _{in}	C _{out}	Sum
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

Table 1. Truth table of a full adder

2. Experiment B

2.1 Design

$$V_{CC} = 5V$$

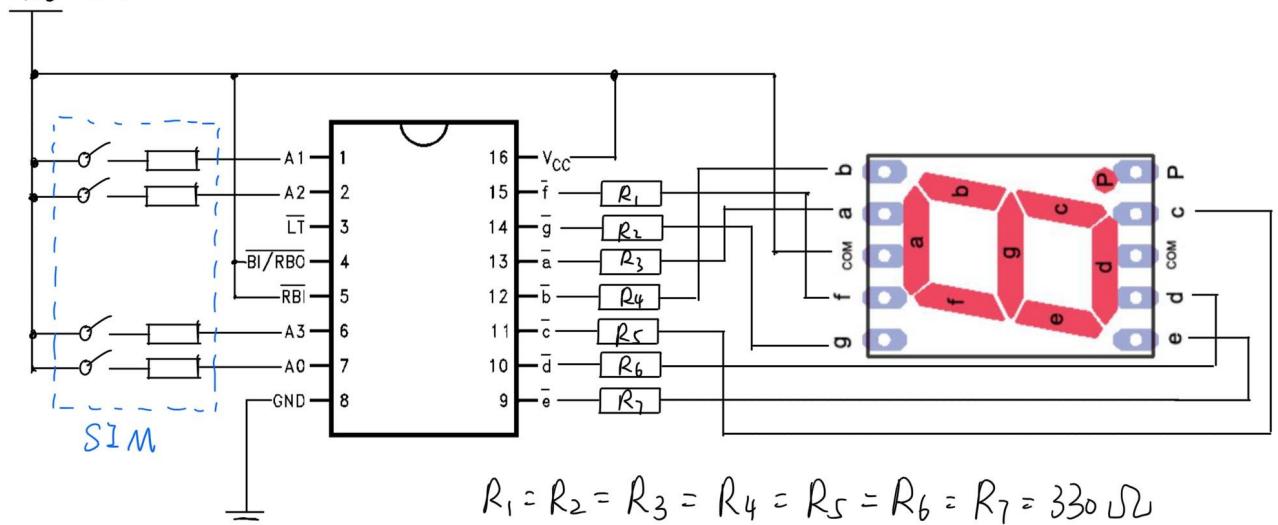


Figure 3. Hardware circuit design (chip level)

2.2 Result

In the below 10 figures, the output of SIM is A3, A2, A1, A0, from left to right.

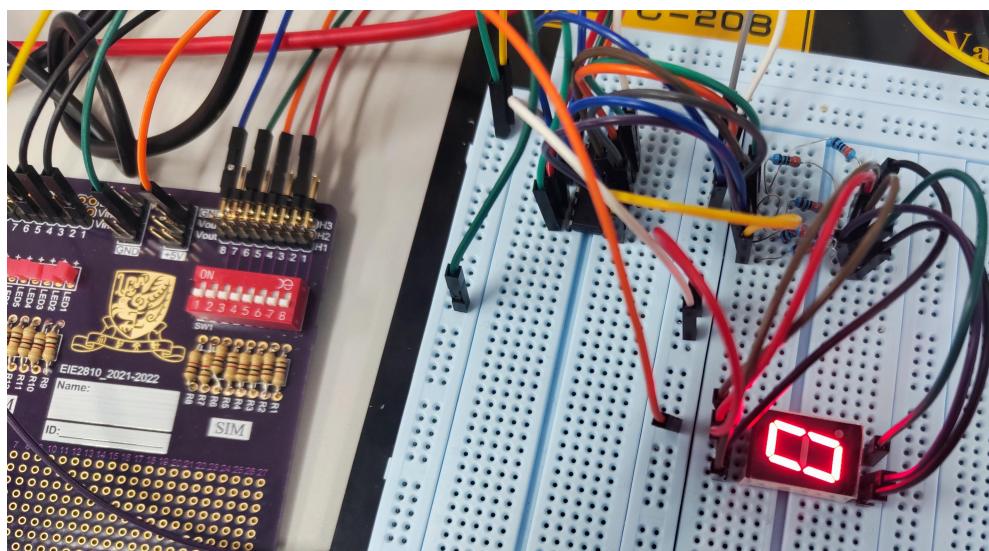


Figure 4. When BCD code is 0000, display "0"

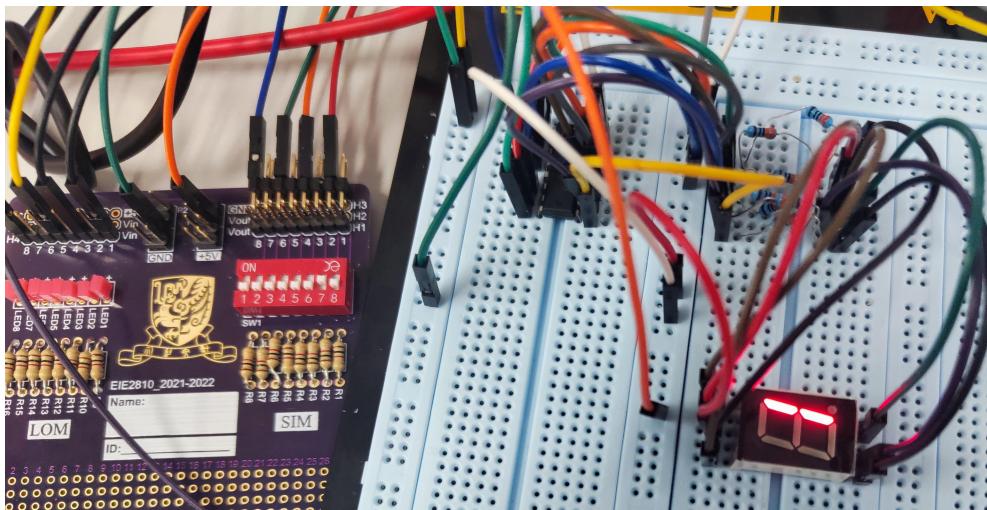


Figure 5. When BCD code is 0001, display “1”

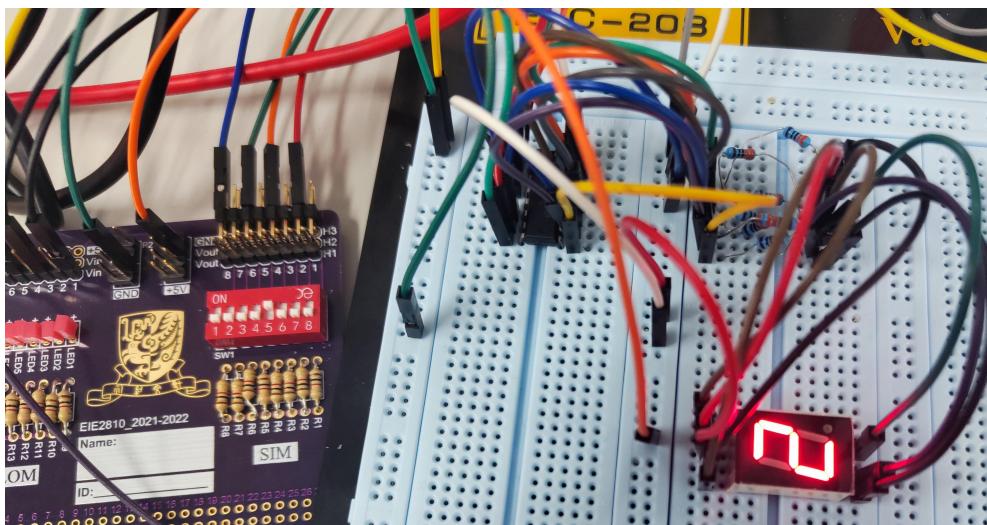


Figure 6. When BCD code is 0010, display “2”

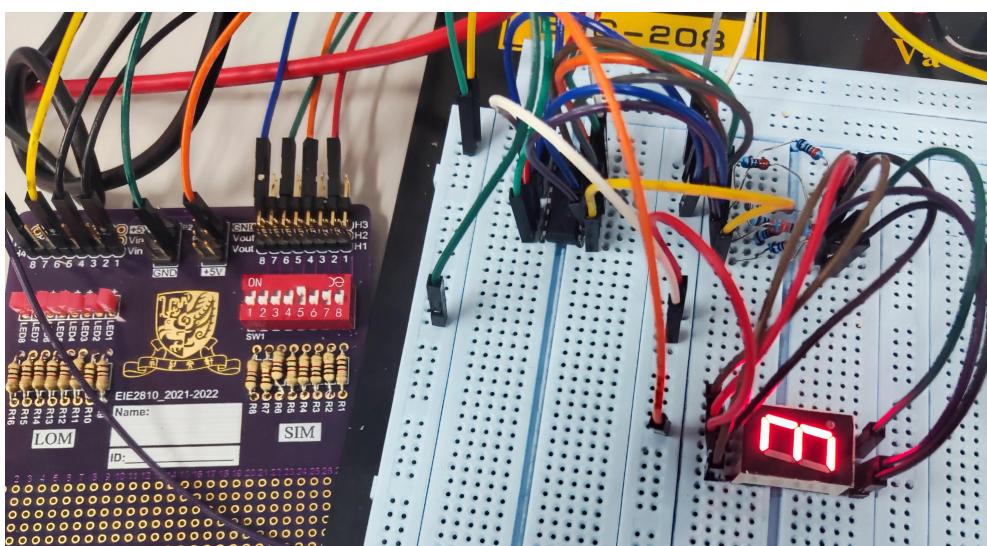


Figure 7. When BCD code is 0011, display “3”

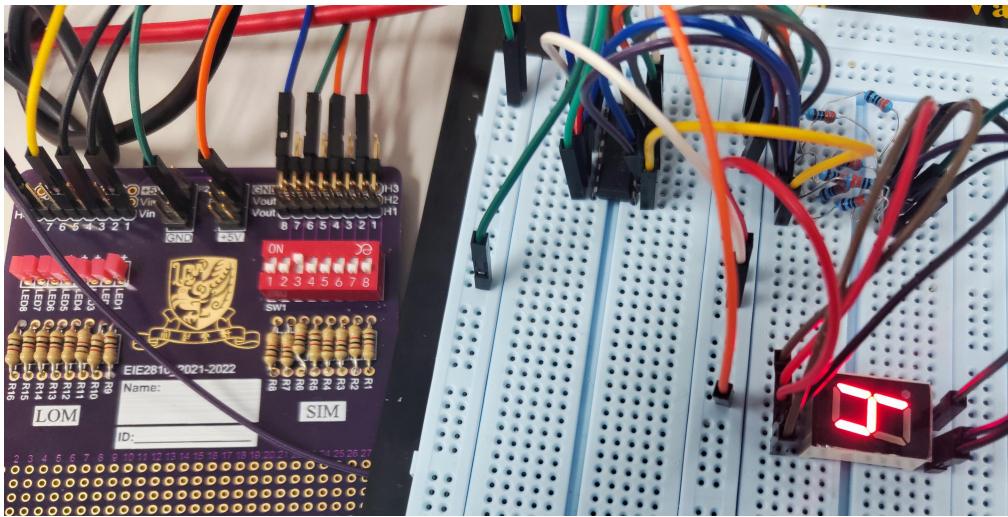


Figure 8. When BCD code is 0100, display “4”

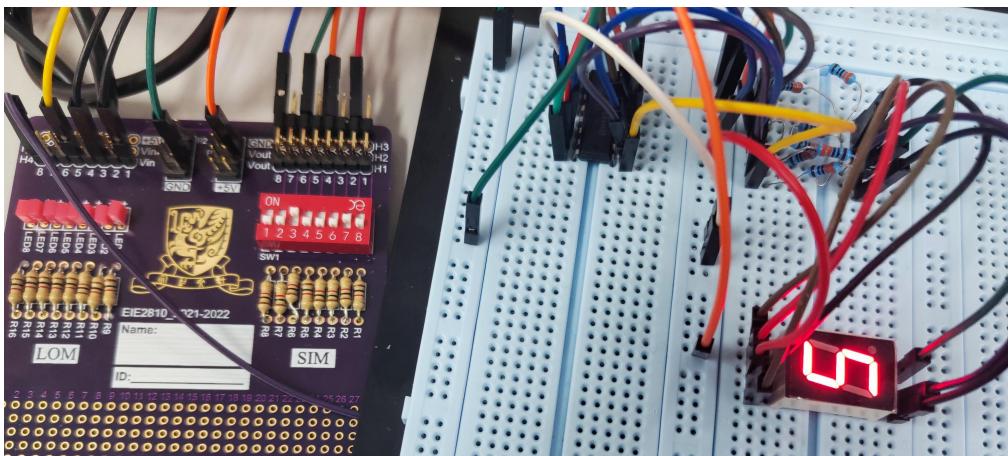


Figure 9. When BCD code is 0101, display “5”

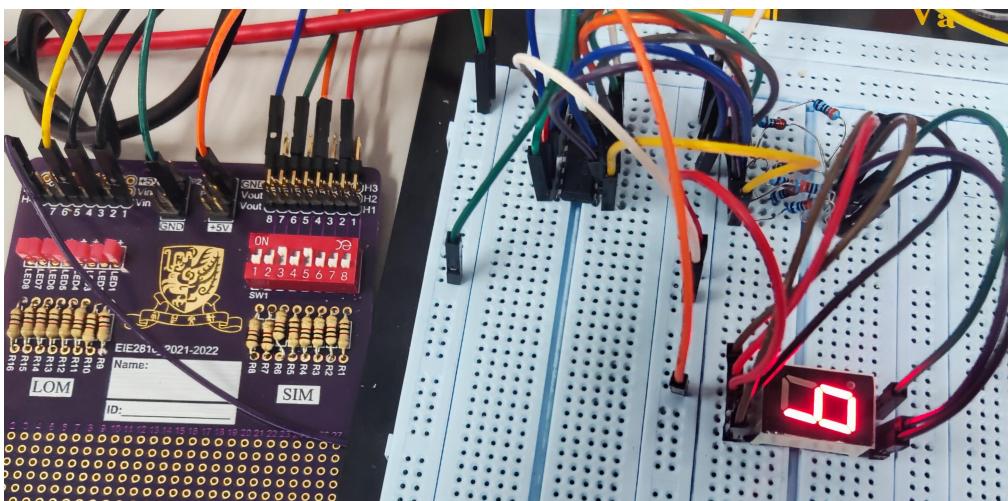


Figure 10. When BCD code is 0110, display “6”

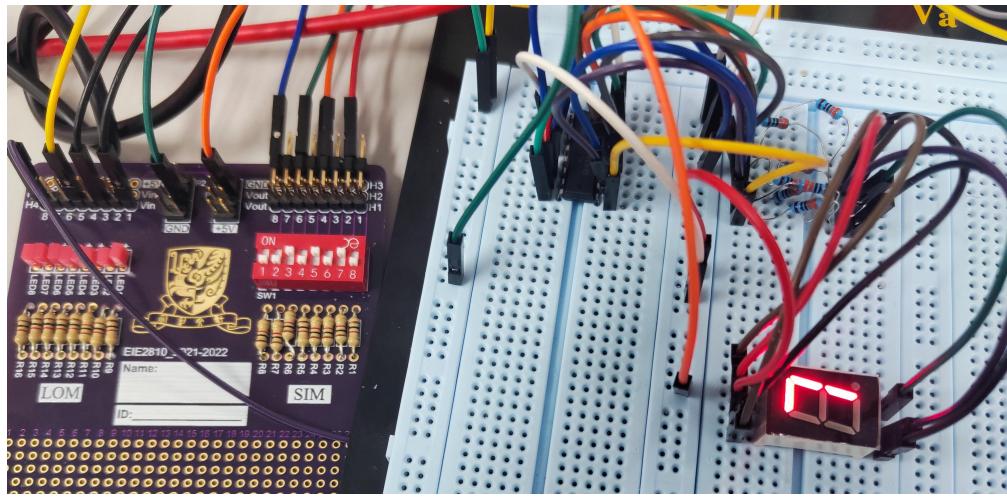


Figure 11. When BCD code is 0111, display “7”

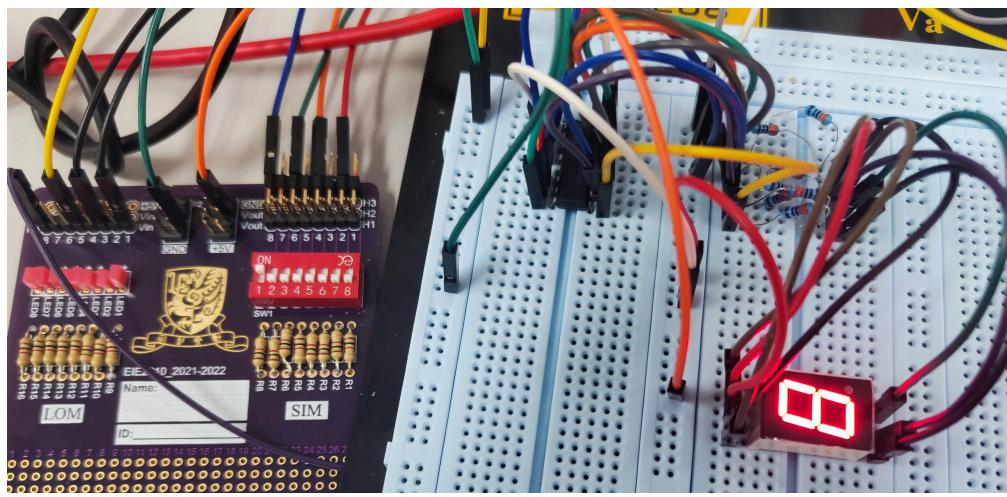


Figure 12. When BCD code is 1000, display “8”

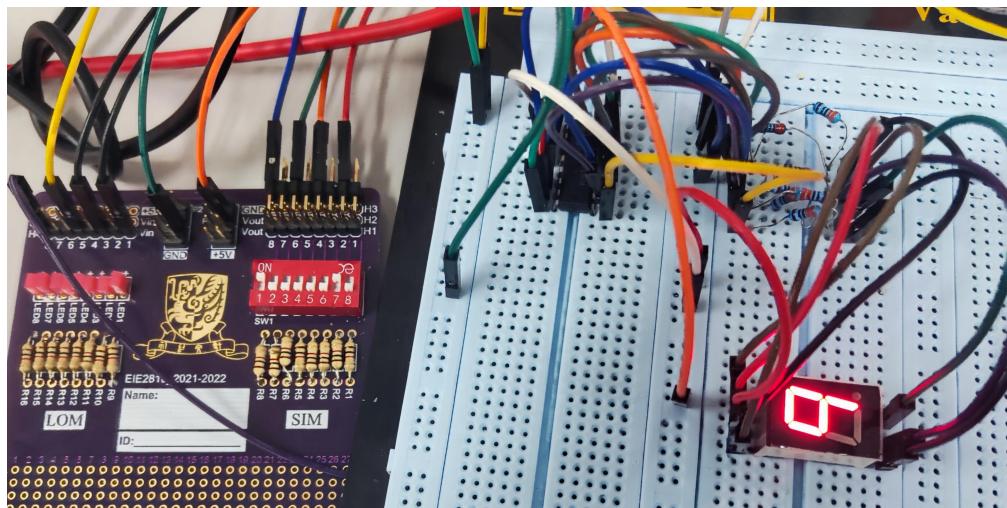


Figure 13. When BCD code is 1001, display “9”

The display of 7-segment is consistent with the input BCD code.

2.3 Question

Q1. Observe the current and power consumption from DC power supply when you change the BCD code. What phenomena can you observe roughly? Try to explain why.

A1.

Observation 1:

BCD code	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001
Current (A)	0.12	0.07	0.11	0.11	0.10	0.11	0.11	0.09	0.13	0.11

Number of lighted segments	6	2	5	5	4	5	5	3	7	5
----------------------------	---	---	---	---	---	---	---	---	---	---

Table 2. Relation between BCD code and total current

Because the voltage remains 5V, the power consumption is $5I$ (W).

When BCD code is 1000 (for digit 8), the current and power consumption are the most significant.

When BCD code is 0000 (for digit 0), the current and power consumption are the second significant.

When BCD code is 0010, 0011, 0101, 0110, 1001 (for digit 2, 3, 5, 6, 9 respectively), the current and power consumption are the third significant.

When the display is 0100 (for digit 4), the current and power consumption are the fourth significant.

When the display is 0111 (for digit 7), the current and power consumption are the fifth significant.

When the display is 0001 (for digit 1), the current and power consumption are the least significant.

This is because different BCD codes will light up different combinations of segments on the 7-segment display. Each segment that is turned on will draw a certain amount of current, so the total current and power consumption will depend on how many segments are illuminated.

The BCD code for the digit 8 will light up all seven segments; the code for the digit 0 will light up six segments; the code for the digits 2, 3, 5, 6, 9 will light up five segments; the code for the digit 4 will light up four segments; the code for the digit 7 will light up three segments; the code for the digit 1 will light up only two segments. More segments are illuminated, more total current and power consumption.

Observation 2:

The relationship between the number of lighted segments and the current/power consumption is not linear.

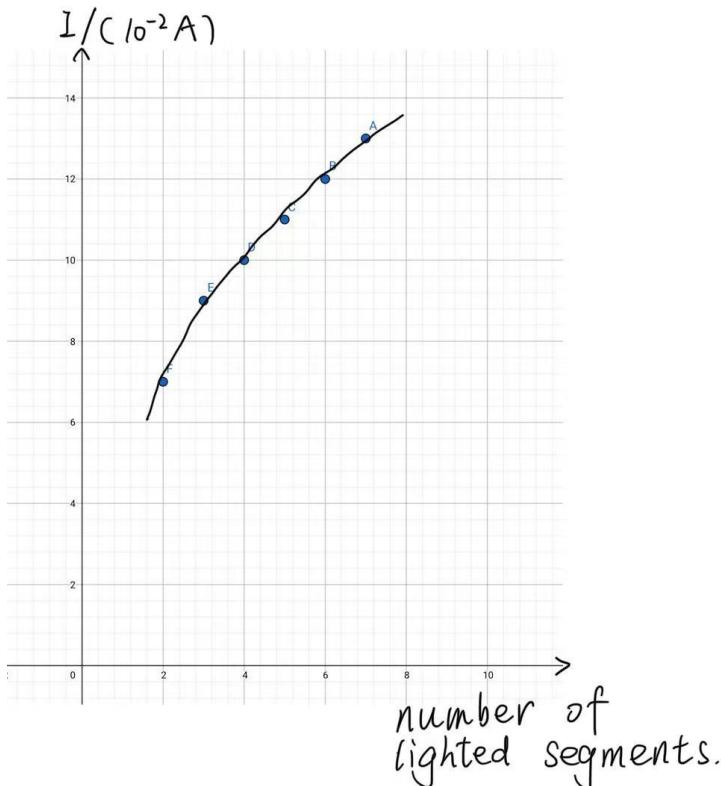


Figure 9. Relationship between total current and number of lighted segments

This is because the total resistance of the circuit, including the 330-ohm resistors and the internal resistance of the SIM, 74LS47 and 7-segment display, will affect the

current flow. The total resistance might change slightly as different segments are turned on, leading to a non-linear relationship between the BCD code and the current/power consumption.

R_o : total internal resistance

$$R_i = 330 \Omega$$

$$R_{\text{total}} = n \cdot R_i + R_o$$

$$I = \frac{V_{cc}}{R_{\text{total}}} = \frac{5}{n \cdot R_i + R_o}$$

$$P = V_{cc} \cdot I = \frac{25}{n \cdot R_i + R_o}$$

Figure 10. Expression of total current and power consumption

3. Experiment C

3.1 Design

A1	A0	B1	B0	Y3	Y2	Y1	Y0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

Table 3. Truth table for A1A0*B1B0

Each output of the decoder (74HC154) corresponds to one row of the truth table, with the decimal conversion of A1A0B1B0 determining the active output. The 74HC154 has

low-activated outputs. For example, when A1A0B1B0 = 1111, the output of \overline{Y}_{15} (at pin 17) is LOW, the others are HIGH.

For the outputs of the multiplier:

Y0 is HIGH when A1A0B1B0 is 0101 or 0111 or 1101 or 1111, which corresponds to the BCD codes for 5, 7, 13, 15 respectively.

Y1 is HIGH when A1A0B1B0 is 0110 or 0111 or 1001 or 1011 or 1101 or 1110, which corresponds to the BCD codes for 6, 7, 9, 11, 13, 14 respectively.

Y2 is HIGH when A1A0B1B0 is 1010 or 1011 or 1110, which corresponds to the BCD codes for 10, 11, 14 respectively.

Y3 is HIGH only when A1A0B1B0 is 1111, which corresponds to the BCD codes for 15.

Because the LOM has high-activated inputs, NAND gates are needed to connect these terms. This inverts the low-activated outputs of the decoder to match the high-activated inputs of the LOM.

The logic expressions for each output can be summarized and simplified as follows:

$$Y_0 = \overline{Y_5 \cdot Y_7 \cdot Y_{13} \cdot Y_{15}} = \overline{Y_7 \cdot Y_{13}} + \overline{Y_5 \cdot Y_{15}}$$

$$Y_1 = \overline{Y_6 \cdot Y_7 \cdot Y_9 \cdot Y_{11} \cdot Y_{13} \cdot Y_{14}} = \overline{Y_7 \cdot Y_{13}} + \overline{Y_{11} \cdot Y_{14}} + \overline{Y_6 \cdot Y_9}$$

$$Y_2 = \overline{Y_{10} \cdot Y_{11} \cdot Y_{14}} = \overline{Y_{10}} + \overline{Y_{11} \cdot Y_{14}}$$

$$Y_3 = \overline{\overline{Y_{15}}}$$

Figure 11. Logic expressions for each output

Using the above simplified expressions, we can reduce the total number of logic gates needed, as the terms that appear multiple times can be reused.

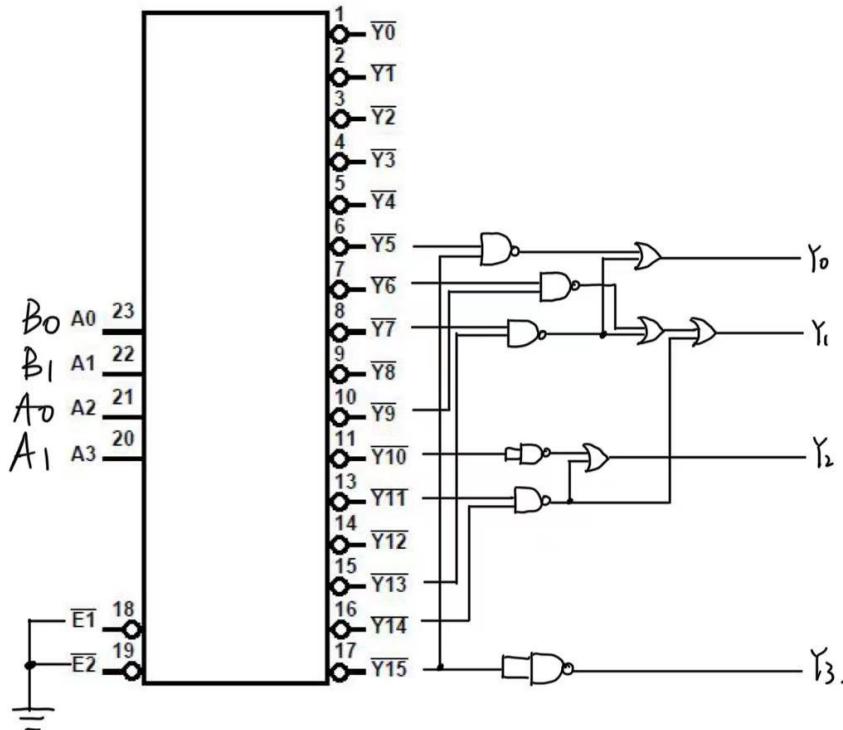


Figure 12. Circuit diagram (logic symbol level)

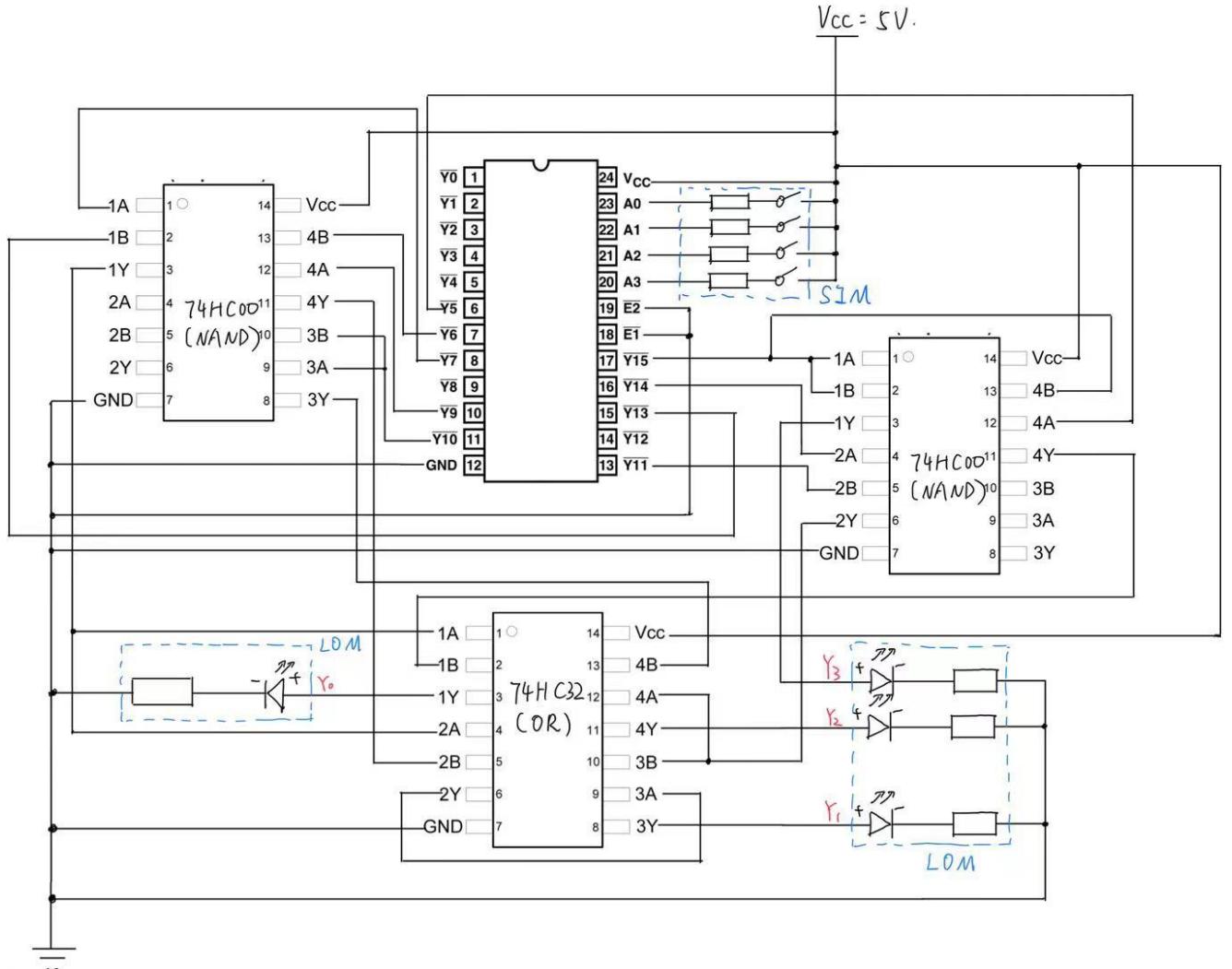


Figure 13. Circuit diagram (chip level)

3.2 Result

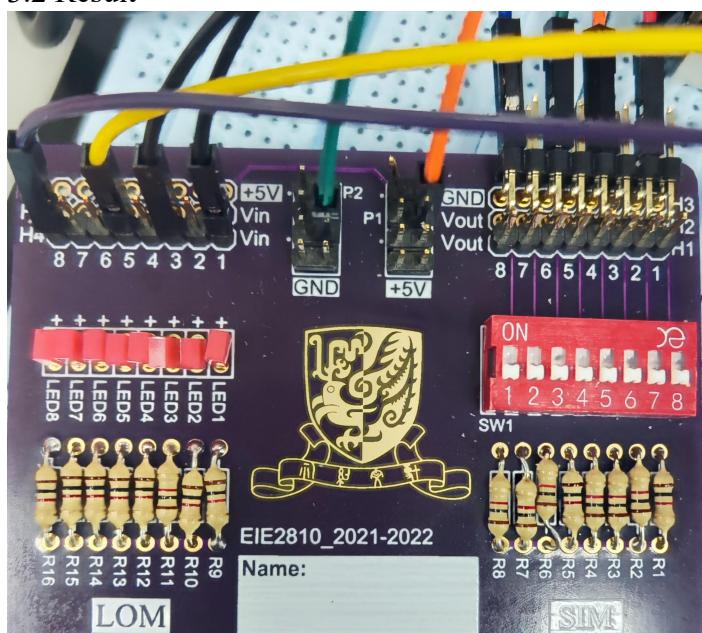


Figure 14. When input is (0, 0, 0, 0) (A1, A0, B1, B0), output is (0, 0, 0, 0) (Y3, Y2, Y1, Y0)

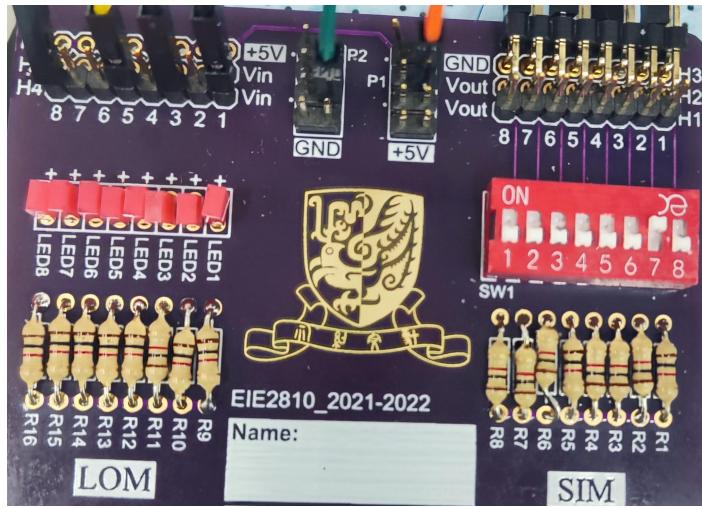


Figure 15. When input is (0, 0, 0, 1) (A1, A0, B1, B0), output is (0, 0, 0, 0) (Y3, Y2, Y1, Y0)

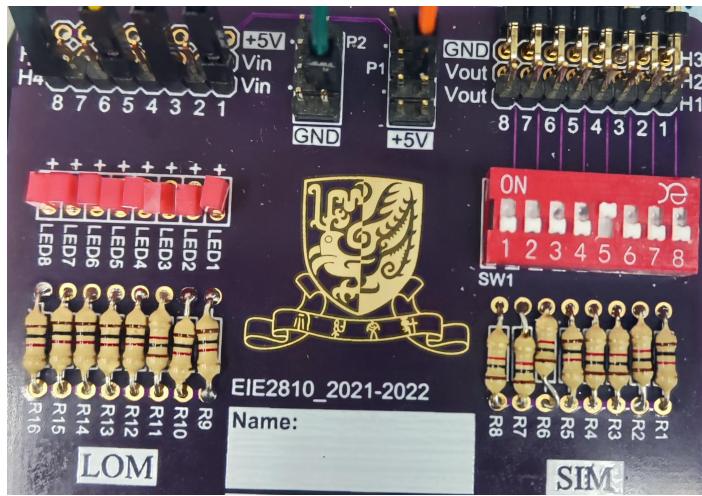


Figure 16. When input is (0, 0, 1, 0) (A1, A0, B1, B0), output is (0, 0, 0, 0) (Y3, Y2, Y1, Y0)

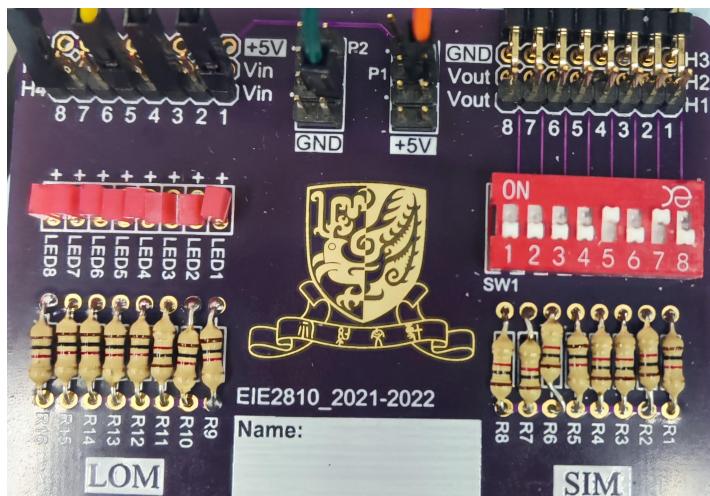


Figure 17. When input is (0, 0, 1, 1) (A1, A0, B1, B0), output is (0, 0, 0, 0) (Y3, Y2, Y1, Y0)

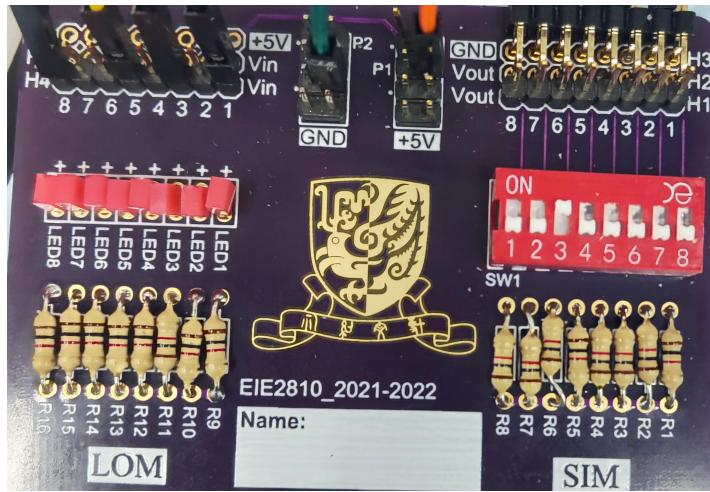


Figure 18. When input is (0, 1, 0, 0) (A1, A0, B1, B0), output is (0, 0, 0, 0) (Y3, Y2, Y1, Y0)

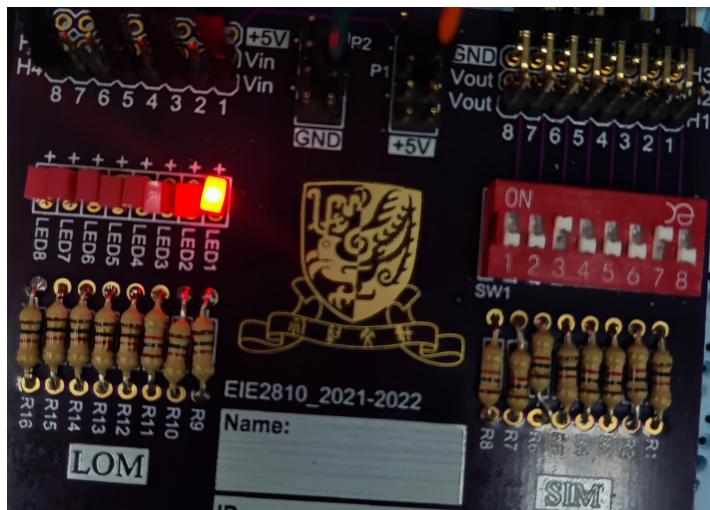


Figure 19. When input is (0, 1, 0, 1) (A1, A0, B1, B0), output is (0, 0, 0, 1) (Y3, Y2, Y1, Y0)

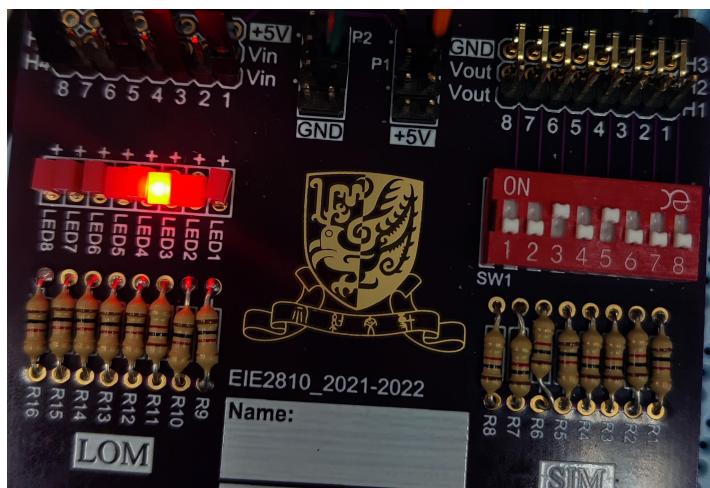


Figure 20. When input is (0, 1, 1, 0) (A1, A0, B1, B0), output is (0, 0, 1, 0) (Y0, Y1, Y2, Y3)

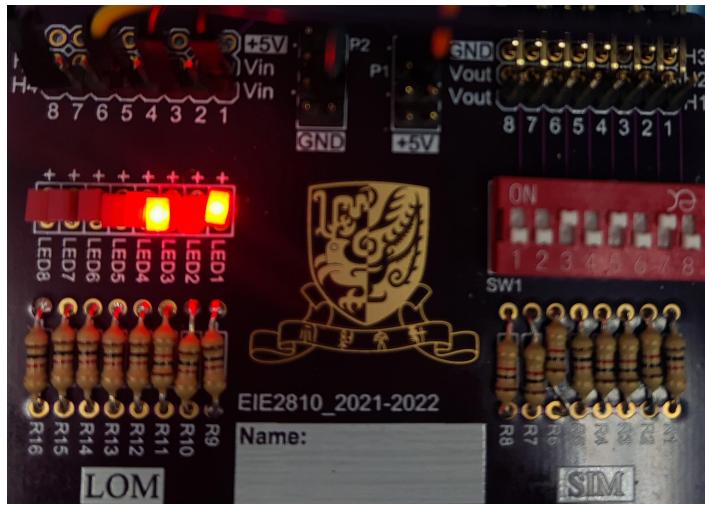


Figure 21. When input is (0, 1, 1, 1) (A1, A0, B1, B0), output is (0, 0, 1, 1) (Y3, Y2, Y1, Y0)

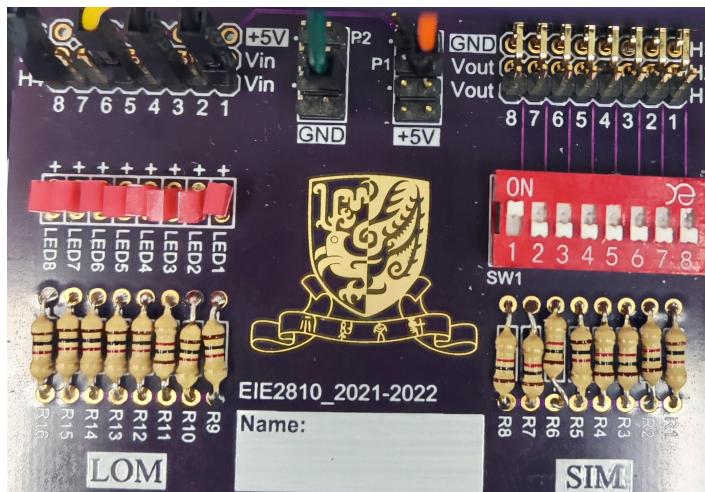


Figure 22. When input is (1, 0, 0, 0) (A1, A0, B1, B0), output is (0, 0, 0, 0) (Y3, Y2, Y1, Y0)

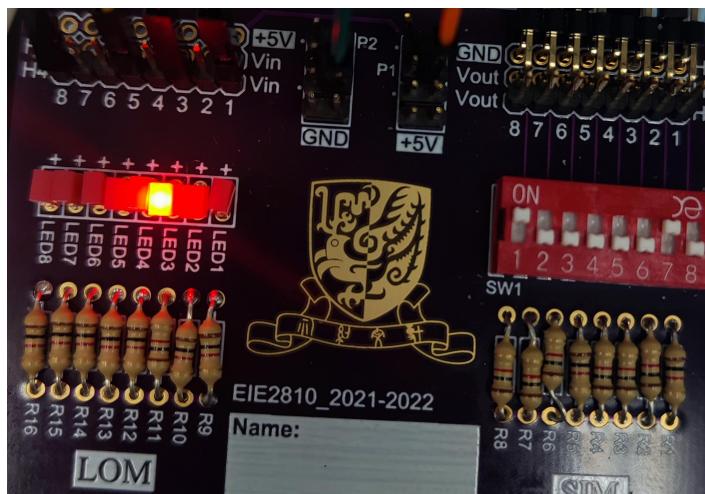


Figure 23. When input is (1, 0, 0, 1) (A1, A0, B1, B0), output is (0, 0, 1, 0) (Y3, Y2, Y1, Y0)

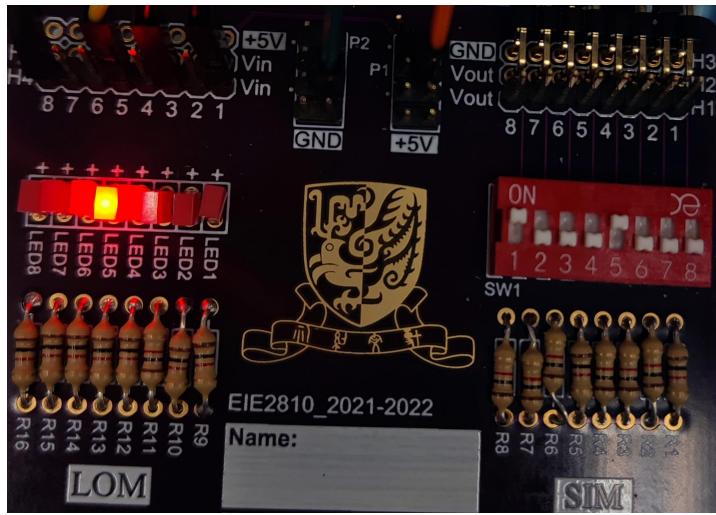


Figure 24. When input is (1, 0, 1, 0) (A1, A0, B1, B0), output is (0, 1, 0, 0) (Y3, Y2, Y1, Y0)

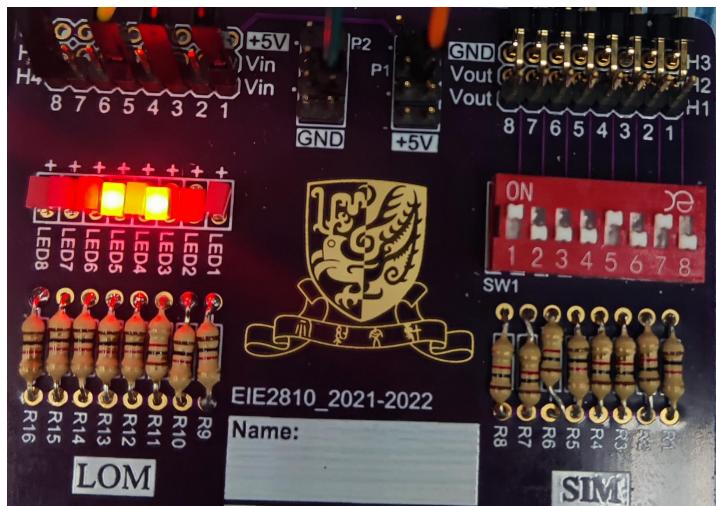


Figure 25. When input is (1, 0, 1, 1) (A1, A0, B1, B0), output is (0, 1, 1, 0) (Y3, Y2, Y1, Y0)

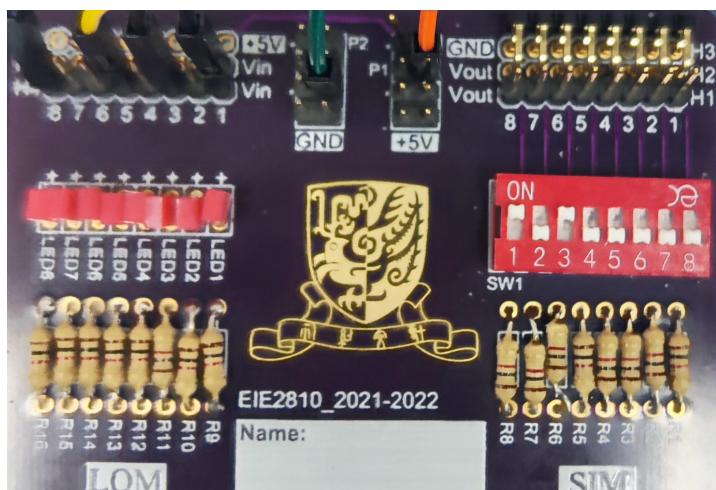


Figure 26. When input is (1, 1, 0, 0) (A1, A0, B1, B0), output is (0, 0, 0, 0) (Y3, Y2, Y1, Y0)

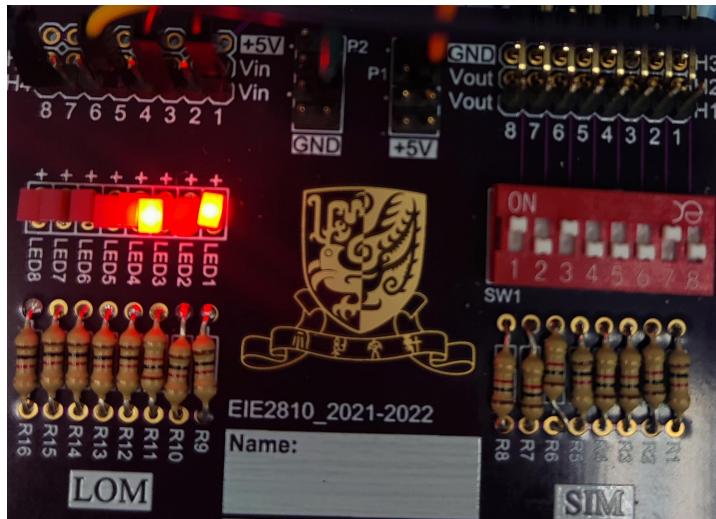


Figure 27. When input is (1, 1, 0, 1) (A1, A0, B1, B0), output is (0, 0, 1, 1) (Y3, Y2, Y1, Y0)

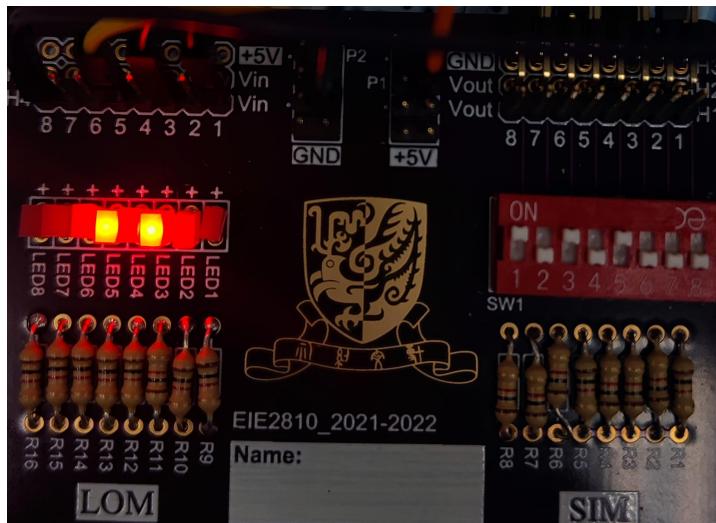


Figure 28. When input is (1, 1, 1, 0) (A1, A0, B1, B0), output is (0, 1, 1, 0) (Y3, Y2, Y1, Y0)

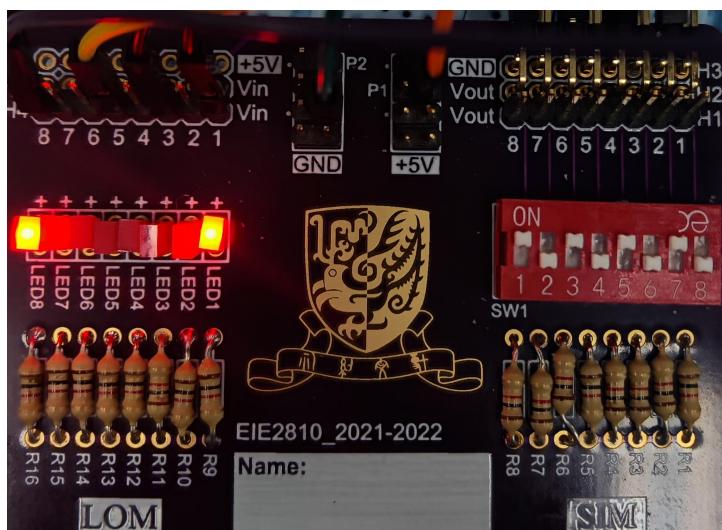


Figure 29. When input is (1, 1, 1, 1) (A1, A0, B1, B0), output is (1, 0, 0, 1) (Y3, Y2, Y1, Y0)

The result is consistent with the truth table.

3.3 Question

Q1. If you are required to build into a 2-bit + 2-bit summation, how to design the circuit? Put down the intermediate steps of your design, and the final designed logic symbol level circuit diagram. You will need to use NAND to connect the output pins of 74HC154. At this time, you do not have limit of using only 74HC00. You can use NAND with any number of inputs.

A1.

A1	A0	B1	B0	Y2	Y1	Y0
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	1
1	0	1	0	1	0	0
1	0	1	1	1	0	1
1	1	0	0	0	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	1
1	1	1	1	1	1	0

Table 4. Truth table for A1A0 + B1B0

Each output of the 74HC154 corresponds to one row of the truth table, with the decimal conversion of A1A0B1B0 determining the active output.

For the outputs of the multiplier:

Y0 is HIGH when A1A0B1B0 is 0001 or 0011 or 0100 or 0110 or 1001 or 1011 or 1100 or 1110, which corresponds to the BCD codes for 1, 3, 4, 6, 9, 11, 12, 14 respectively.

Y1 is HIGH when A1A0B1B0 is 0010 or 0011 or 0101 or 0110 or 1000 or 1001 or 1100 or 1111, which corresponds to the BCD codes for 2, 3, 5, 6, 8, 9, 12, 15 respectively.

Y2 is HIGH when A1A0B1B0 is 0111 or 1010 or 1011 or 1101 or 1110 or 1111, which corresponds to the BCD codes for 7, 10, 11, 13, 14, 15 respectively.

The logic expressions for each output can be summarized as follows:

$$Y_0 = \overline{Y_1} \cdot \overline{Y_3} \cdot \overline{Y_4} \cdot \overline{Y_6} \cdot \overline{Y_9} \cdot \overline{Y_{11}} \cdot \overline{Y_{12}} \cdot \overline{Y_{14}}$$

$$Y_1 = \overline{Y_2} \cdot \overline{Y_3} \cdot \overline{Y_5} \cdot \overline{Y_6} \cdot \overline{Y_8} \cdot \overline{Y_9} \cdot \overline{Y_{12}} \cdot \overline{Y_{15}}$$

$$Y_2 = \overline{Y_7} \cdot \overline{Y_{10}} \cdot \overline{Y_{11}} \cdot \overline{Y_{13}} \cdot \overline{Y_{14}} \cdot \overline{Y_{15}}$$

Figure 30. Logic expressions for each output

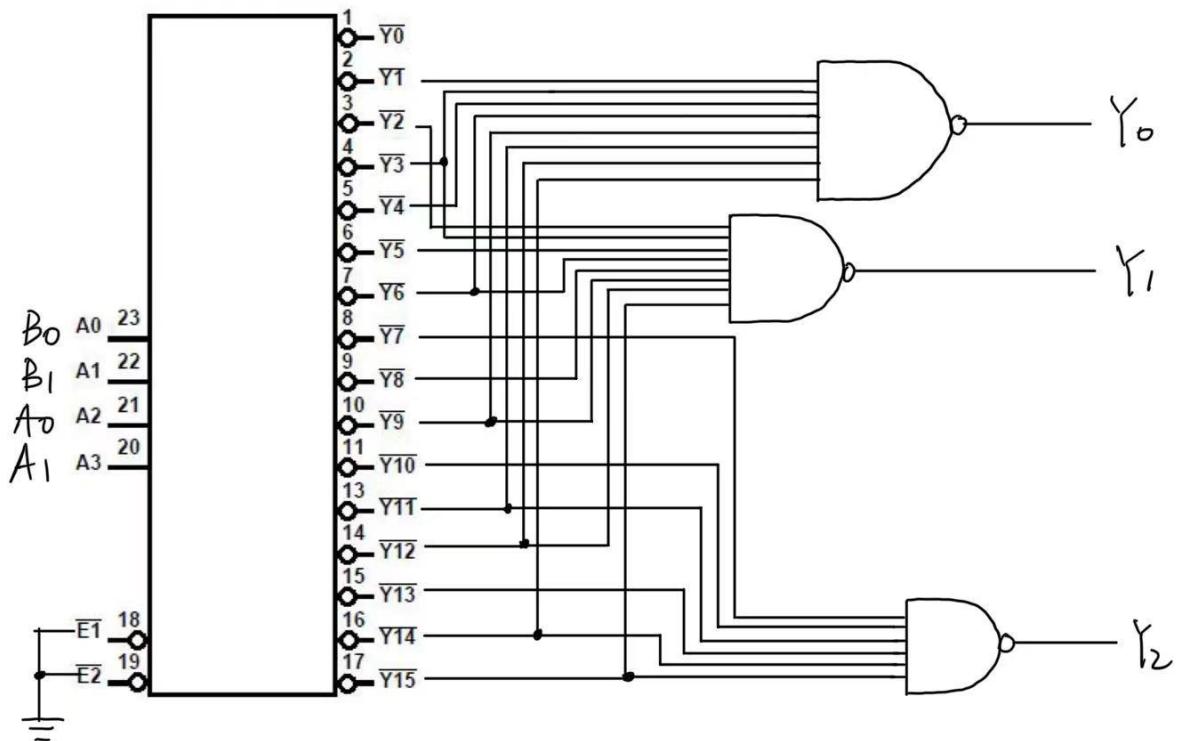


Figure 31. Circuit diagram (logic symbol level)

4. Experiment D

4.1 Result

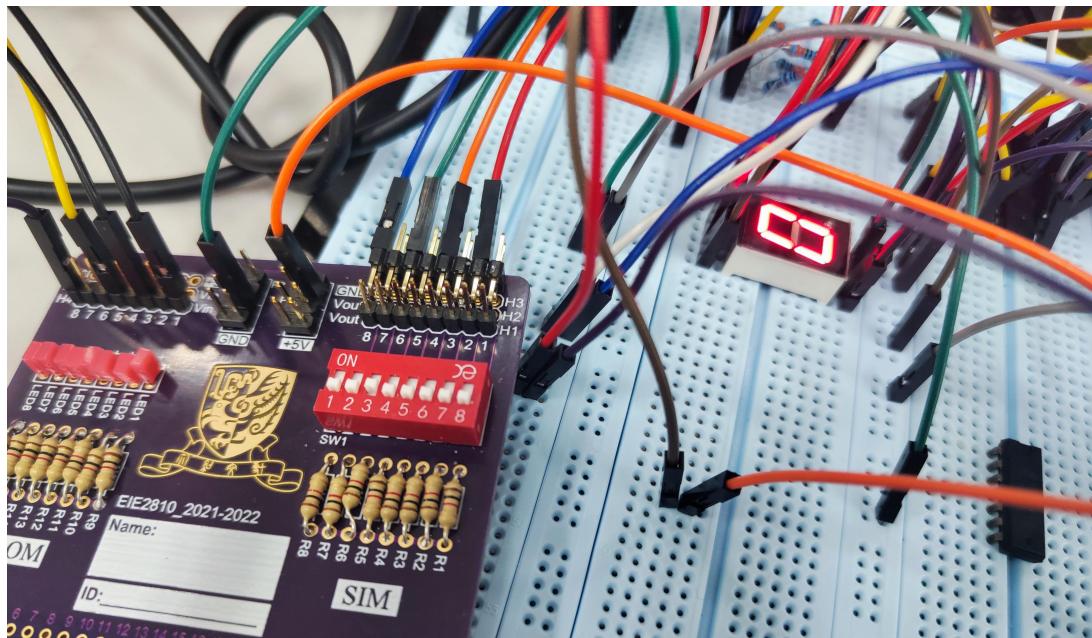


Figure 32. When input is (0, 0, 0, 0) (A1, A0, B1, B0), display “0”

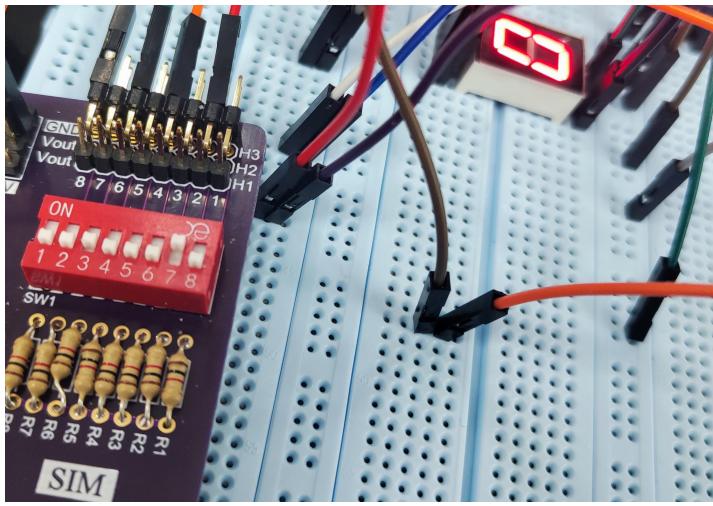


Figure 33. When input is (0, 0, 0, 1) (A1, A0, B1, B0), display “0”

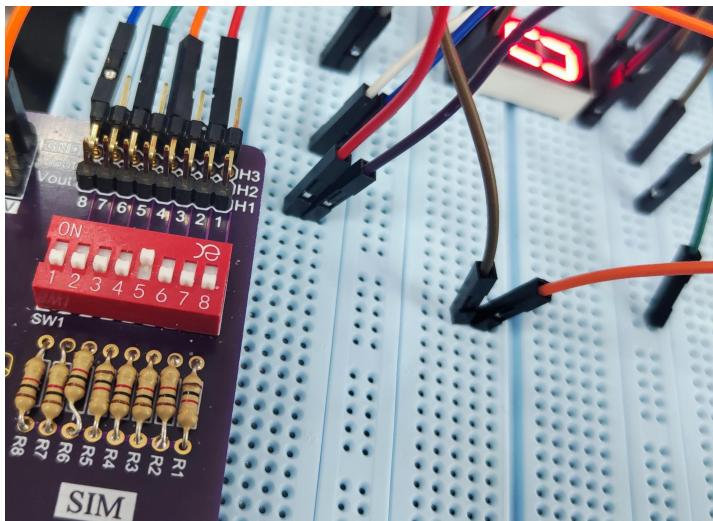


Figure 34. When input is (0, 0, 1, 0) (A1, A0, B1, B0), display “0”

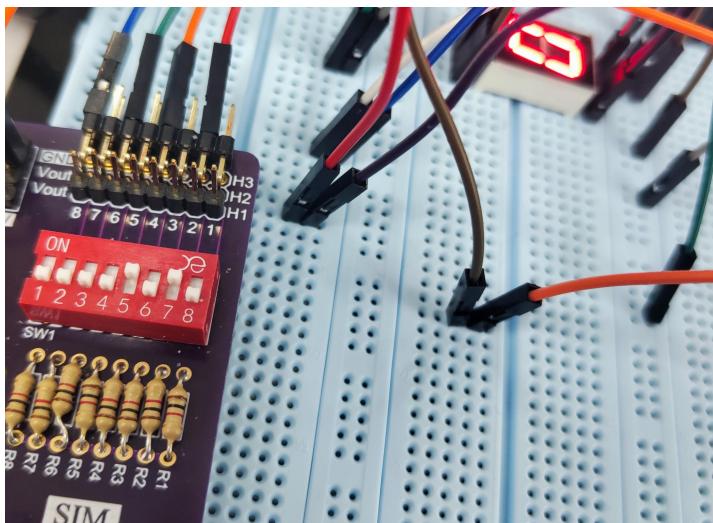


Figure 35. When input is (0, 0, 1, 1) (A1, A0, B1, B0), display “0”

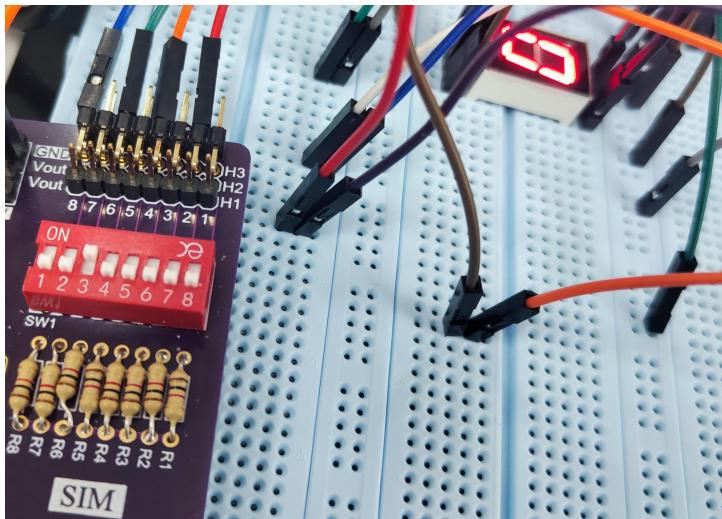


Figure 36. When input is (0, 1, 0, 0) (A1, A0, B1, B0), display “0”

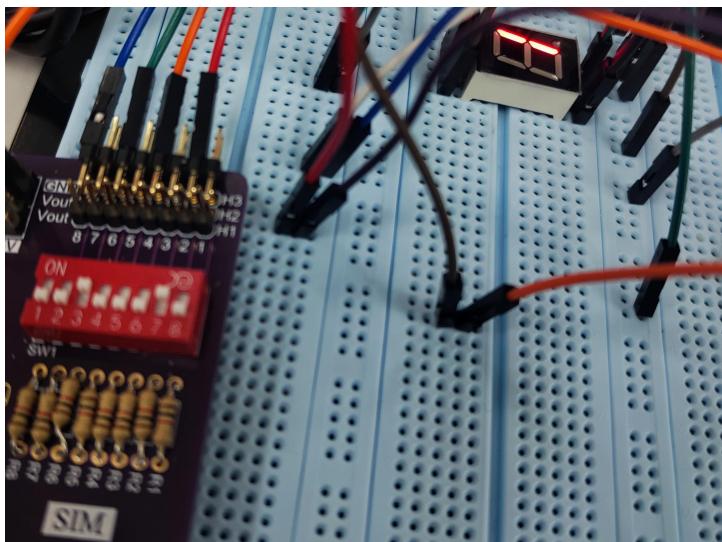


Figure 37. When input is (0, 1, 0, 1) (A1, A0, B1, B0), display “1”

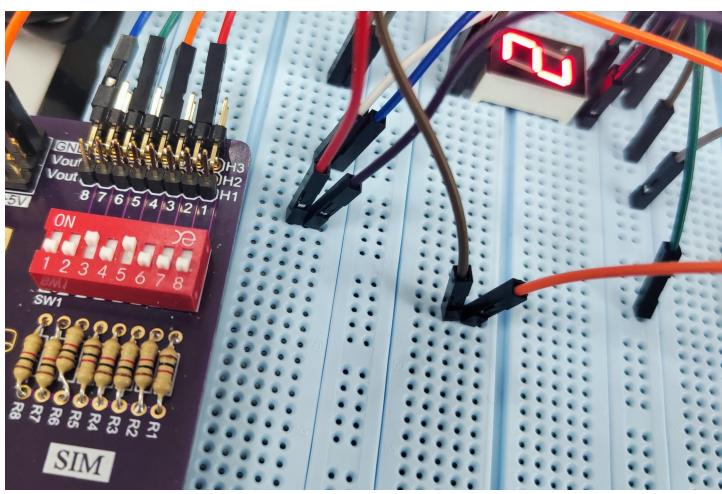


Figure 38. When input is (0, 1, 1, 0) (A1, A0, B1, B0), display “2”

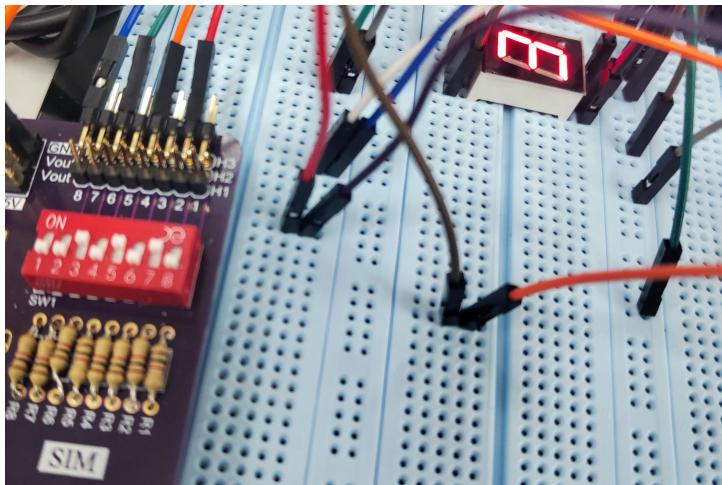


Figure 39. When input is (0, 1, 1, 1) (A1, A0, B1, B0), display “3”

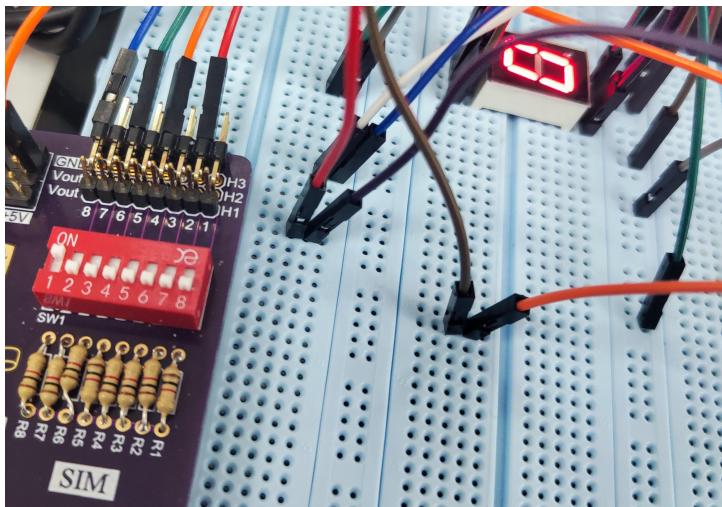


Figure 40. When input is (1, 0, 0, 0) (A1, A0, B1, B0), display “0”

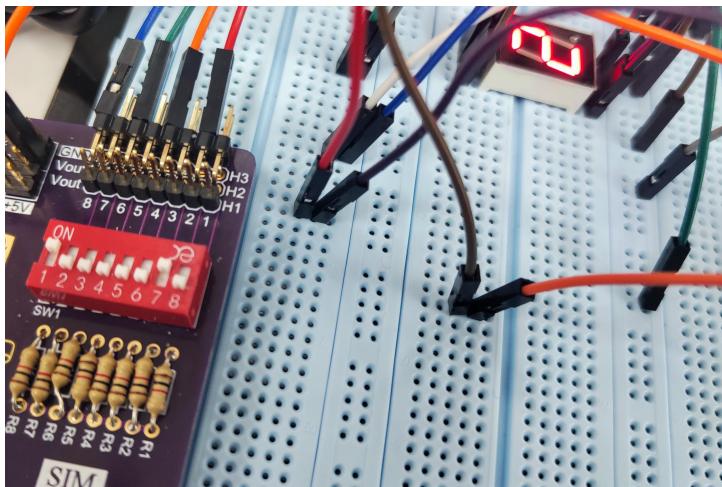


Figure 41. When input is (1, 0, 0, 1) (A1, A0, B1, B0), display “2”

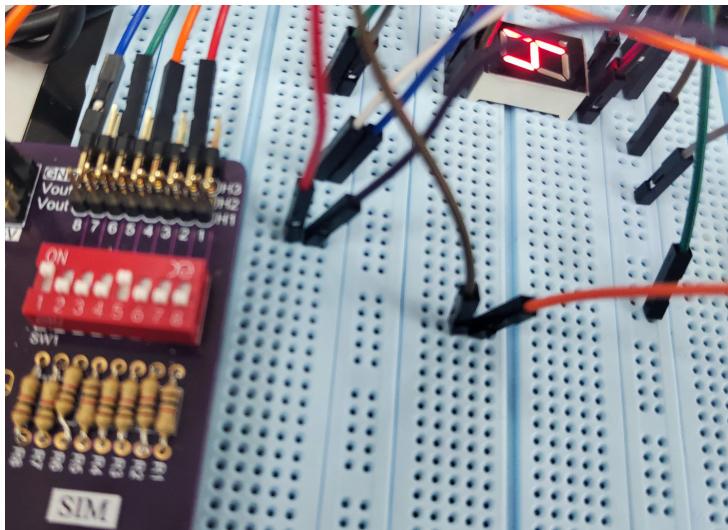


Figure 42. When input is (1, 0, 1, 0) (A1, A0, B1, B0), display “4”

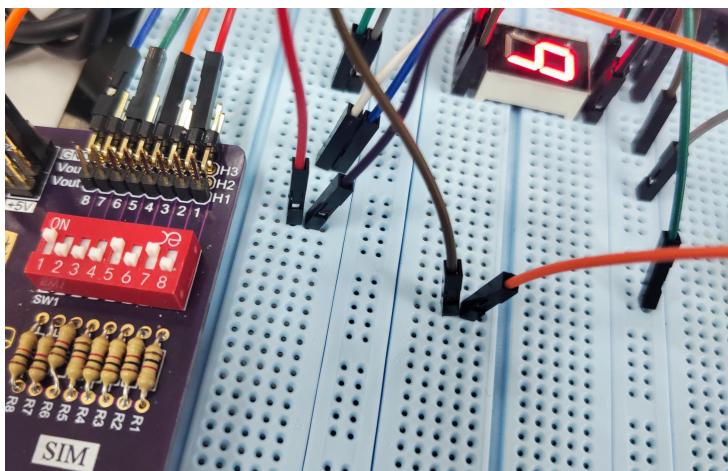


Figure 43. When input is (1, 0, 1, 1) (A1, A0, B1, B0), display “6”

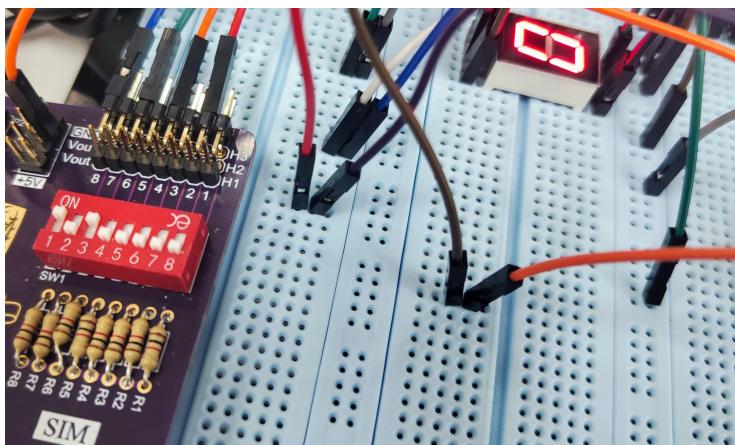


Figure 44. When input is (1, 1, 0, 0) (A1, A0, B1, B0), display “0”

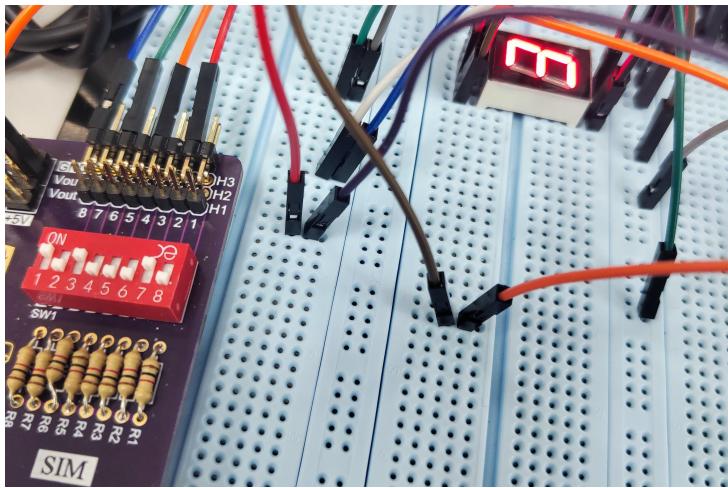


Figure 45. When input is (1, 1, 0, 1) (A1, A0, B1, B0), display “3”

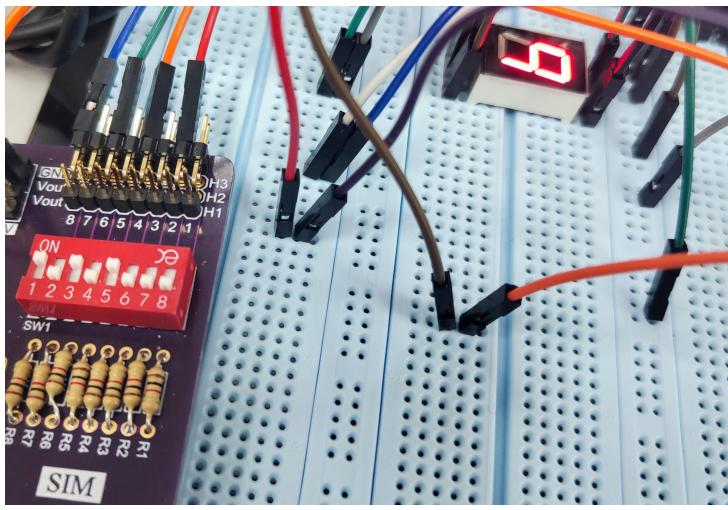


Figure 46. When input is (1, 1, 1, 0) (A1, A0, B1, B0), display “6”

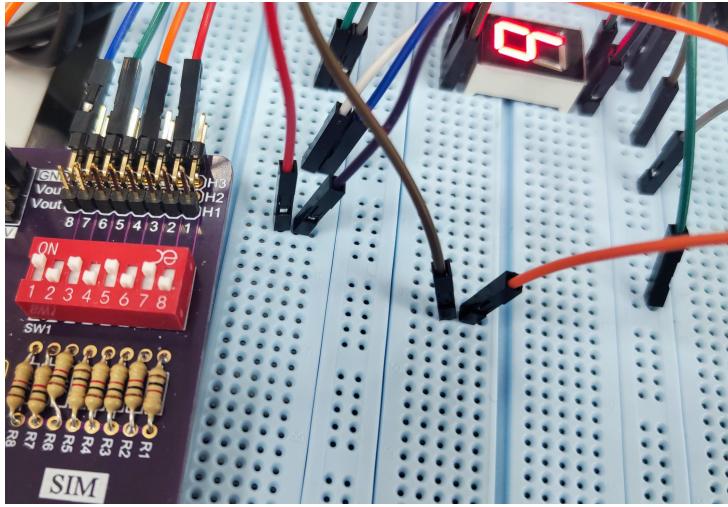


Figure 47. When input is (1, 1, 1, 1) (A1, A0, B1, B0), display “9”

The result is consistent with the multiplication of two 2-bit number.

5. Conclusion

In this lab, I

- Used a 4-line-to-16-line decoder, a 7-segment display, and a BCD-to-7-segement decoder.
- Built a full-adder in Multisim.
- Built a combinational logic circuit for a 2-bit by 2-bit multiplication operation, using a 4-line-to-16-line decoder, a BCD-to-7-segement decoder, and a 7-segment

indicator.

I learned:

- When LT is low and BI/RBO is high, all segments of 74LS47 will be turned on, which can be used to test if any segment is burnt.
- When using a 7-segement display, it is important to wire resistors into the circuit to prevent damage to the display.
- When designing hardware combinational logic circuits, we can factor out common terms and reuse them to reduce the total number of gates needed, leading to a more efficient and cost-effective design.