EIE2810 Digital Systems Design Laboratory

Laboratory Report #3

Name: Student ID:

Date: 2024.3.22

The Chinese University of Hong Kong, Shenzhen

Student ID: Name:

- Experiment A: realize combinational logic design
- Experiment B: learn about timing hazard
- Experiment C: build a logic circuit for 2-bit multiplied by 2-bits

1. Experiment A

1.1 Design

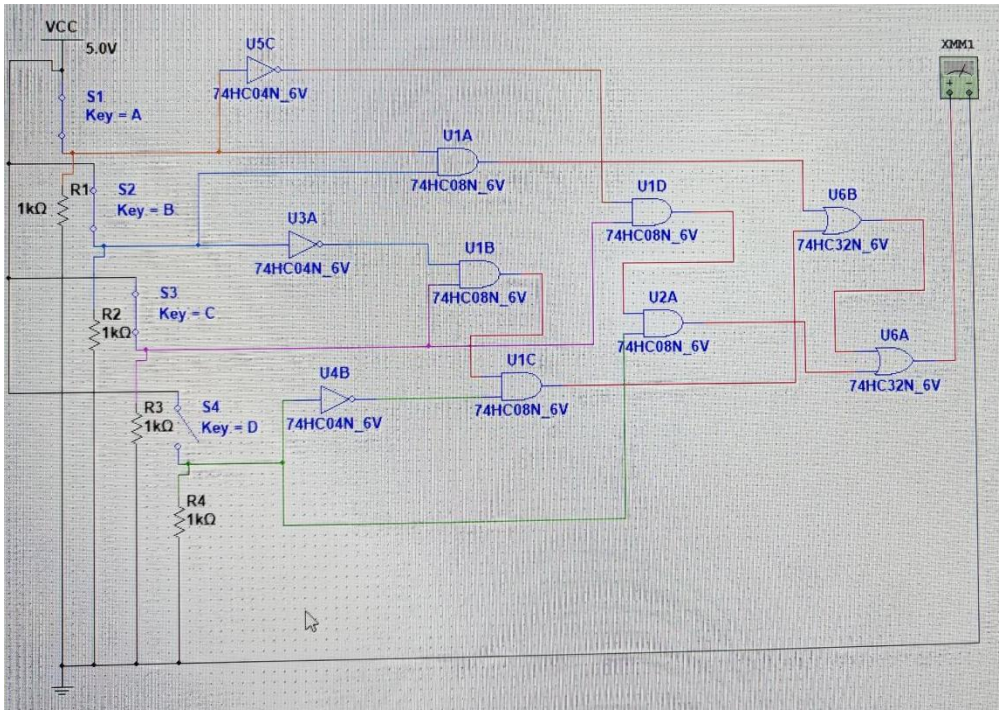

Figure 1. Gate level circuit for Y = AB + $\bar{B}C\bar{D}$ + $\bar{A}CD$

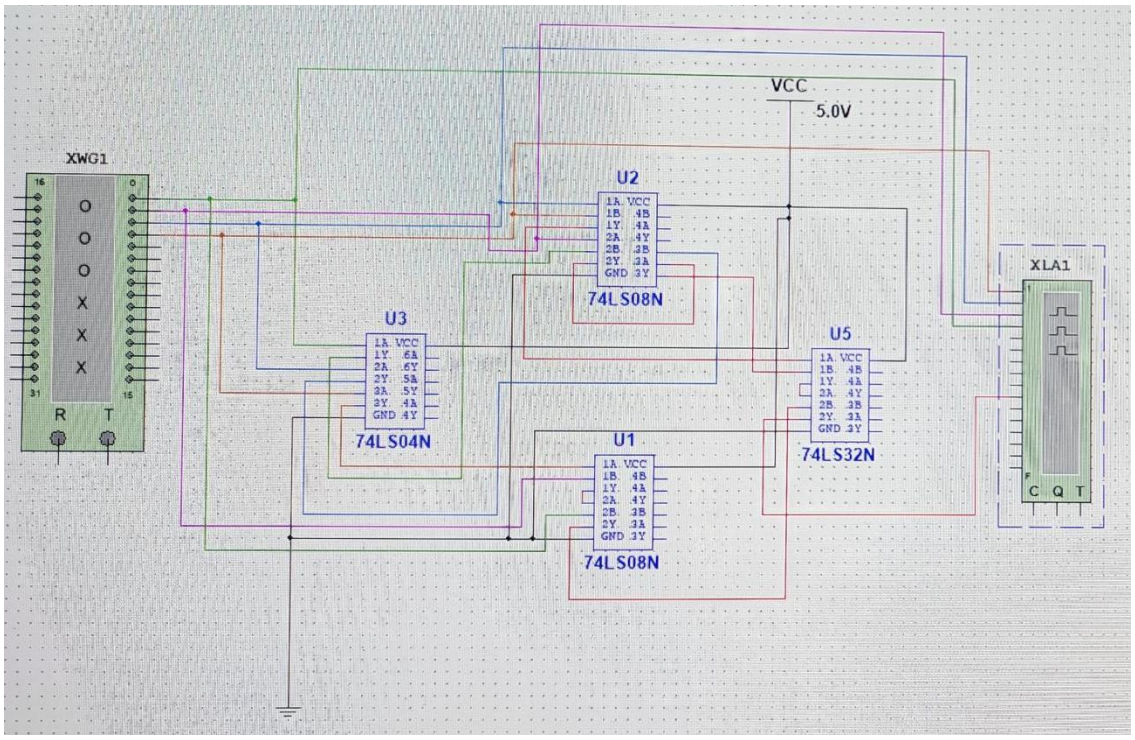

Figure 2. Chip level circuit for Y = AB + $\bar{B}C\bar{D}$ + $\bar{A}CD$ (the outputs of the Word Generator are D, B, C, A, from up to down)

1.2 Result

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Table 1. Truth table for Y = AB + B̄CD̄+ ĀCD

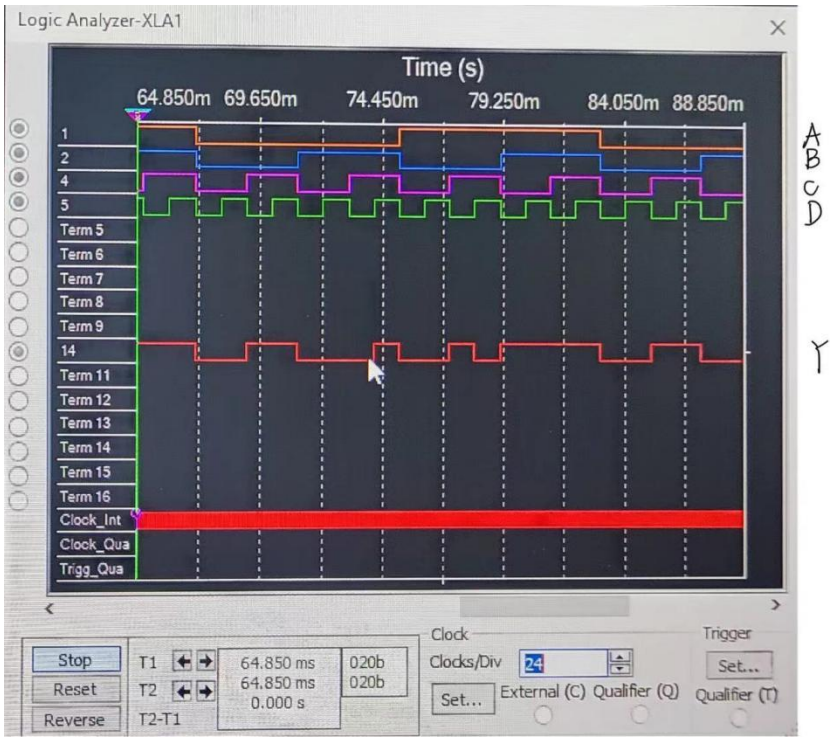

Figure 3. Waveform of circuit in Figure 2

2. Experiment B

2.1 Design

- Multisim designs

In the below two figures, the inputs to the oscilloscope are A, Ā, B (HIGH), Y respectively, from left to right.

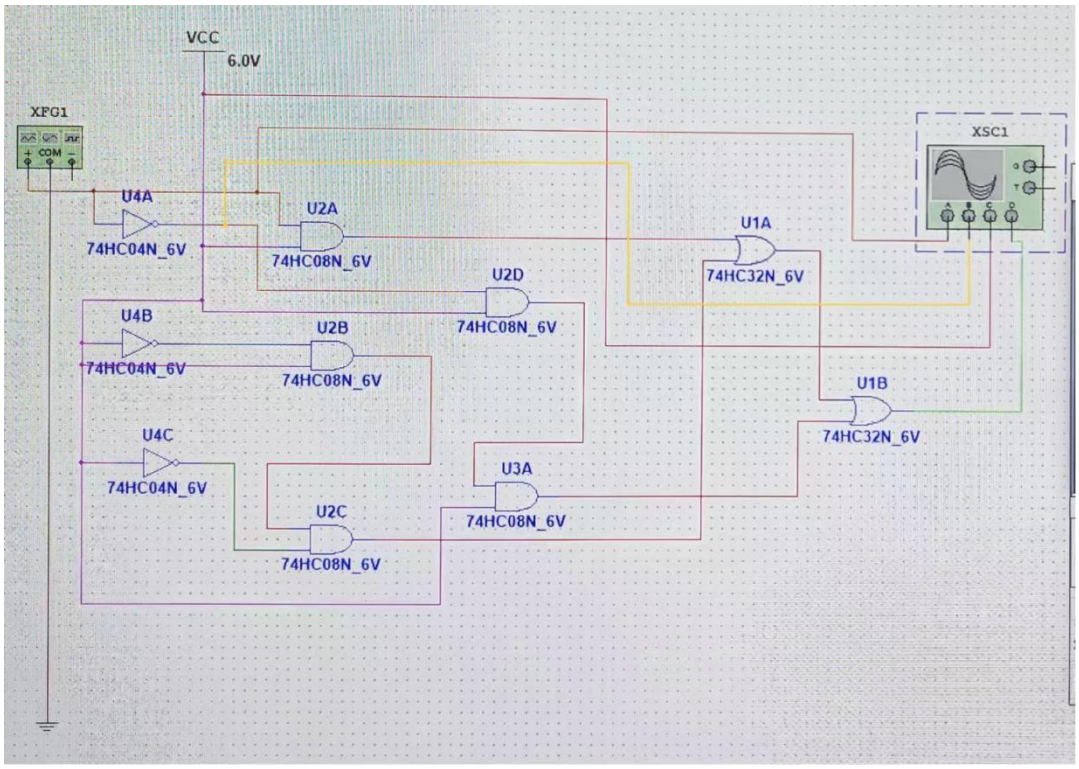

Figure 4. Circuit with high frequency square wave as input to A, other inputs keeping HIGH

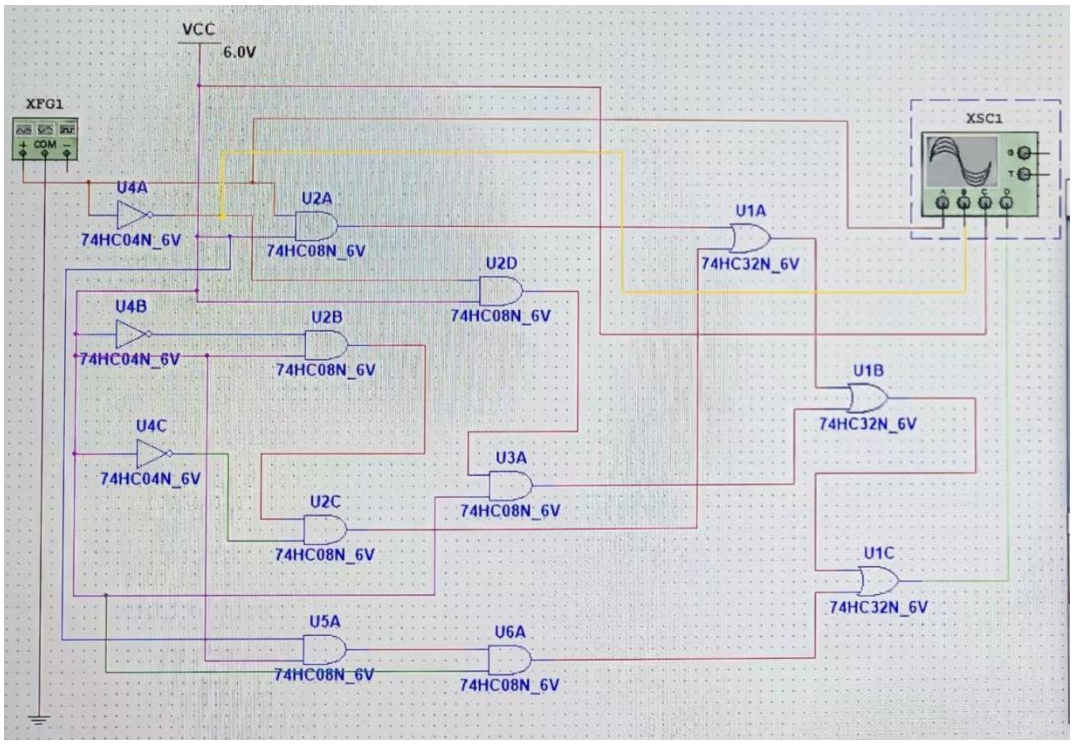

Figure 5. Circuit with additional term BCD

- Hardware designs

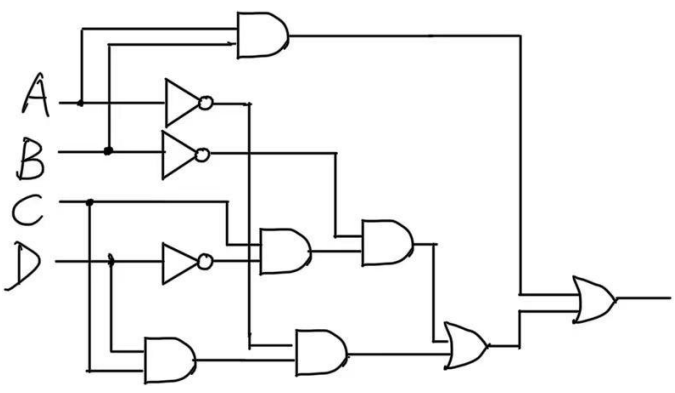

Figure 6. Hardware circuit design of original terms (gate level)

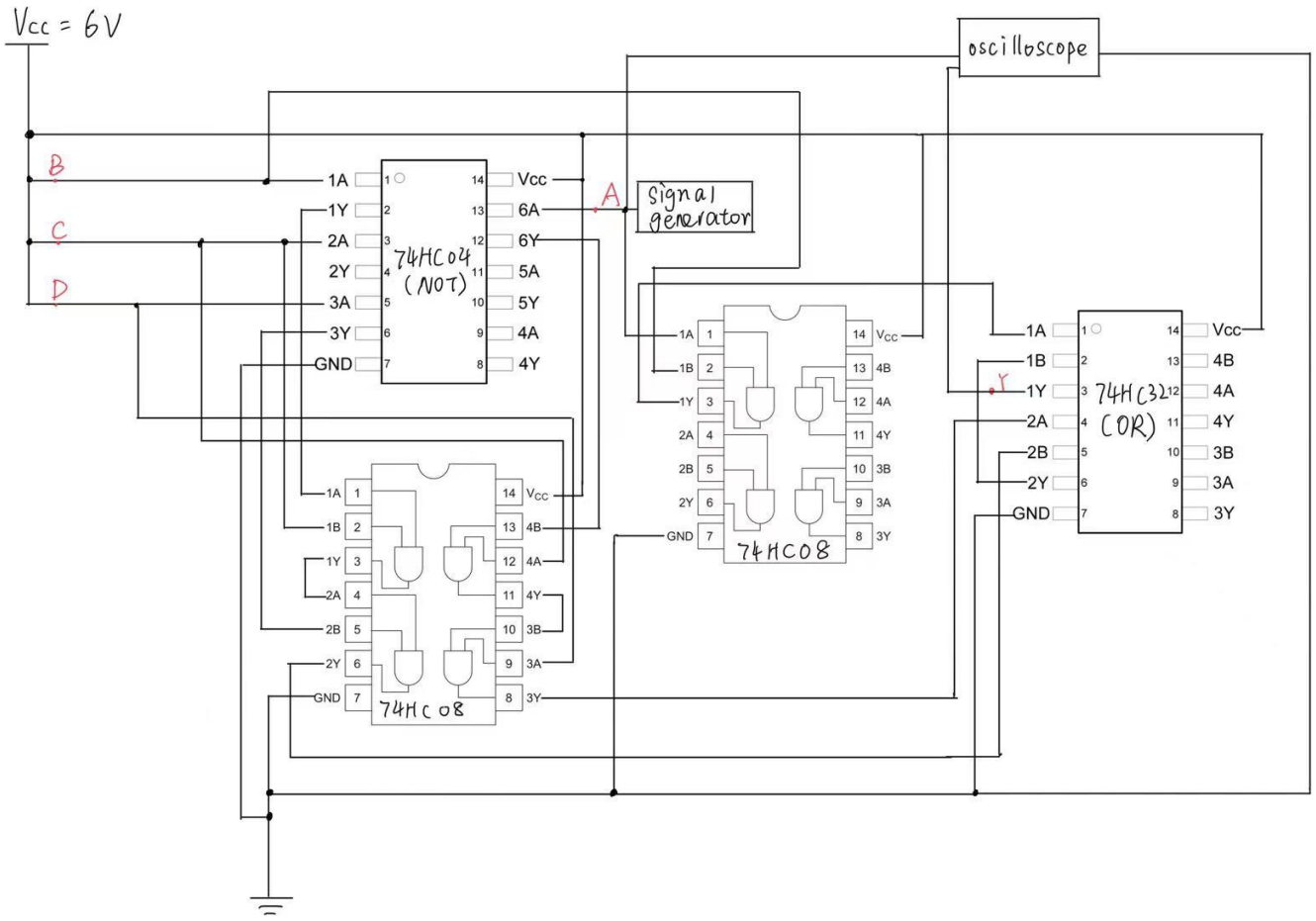

Figure 7. Hardware circuit design of original terms (chip level)

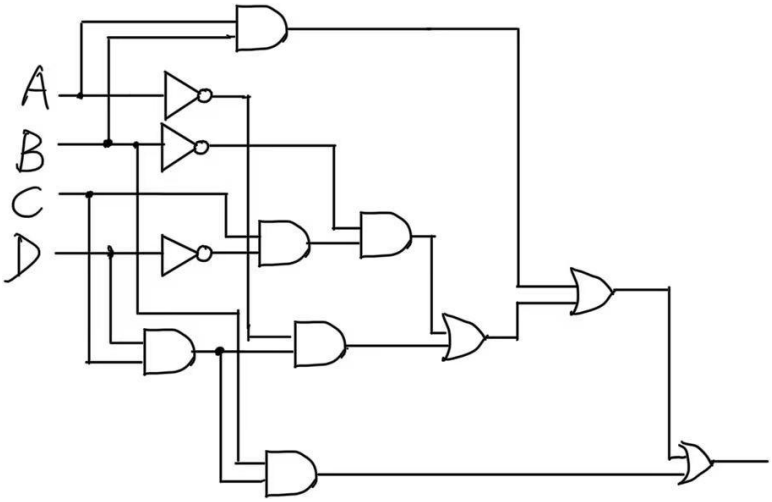

Figure 8. Hardware circuit design with additional term BCD (gate level)

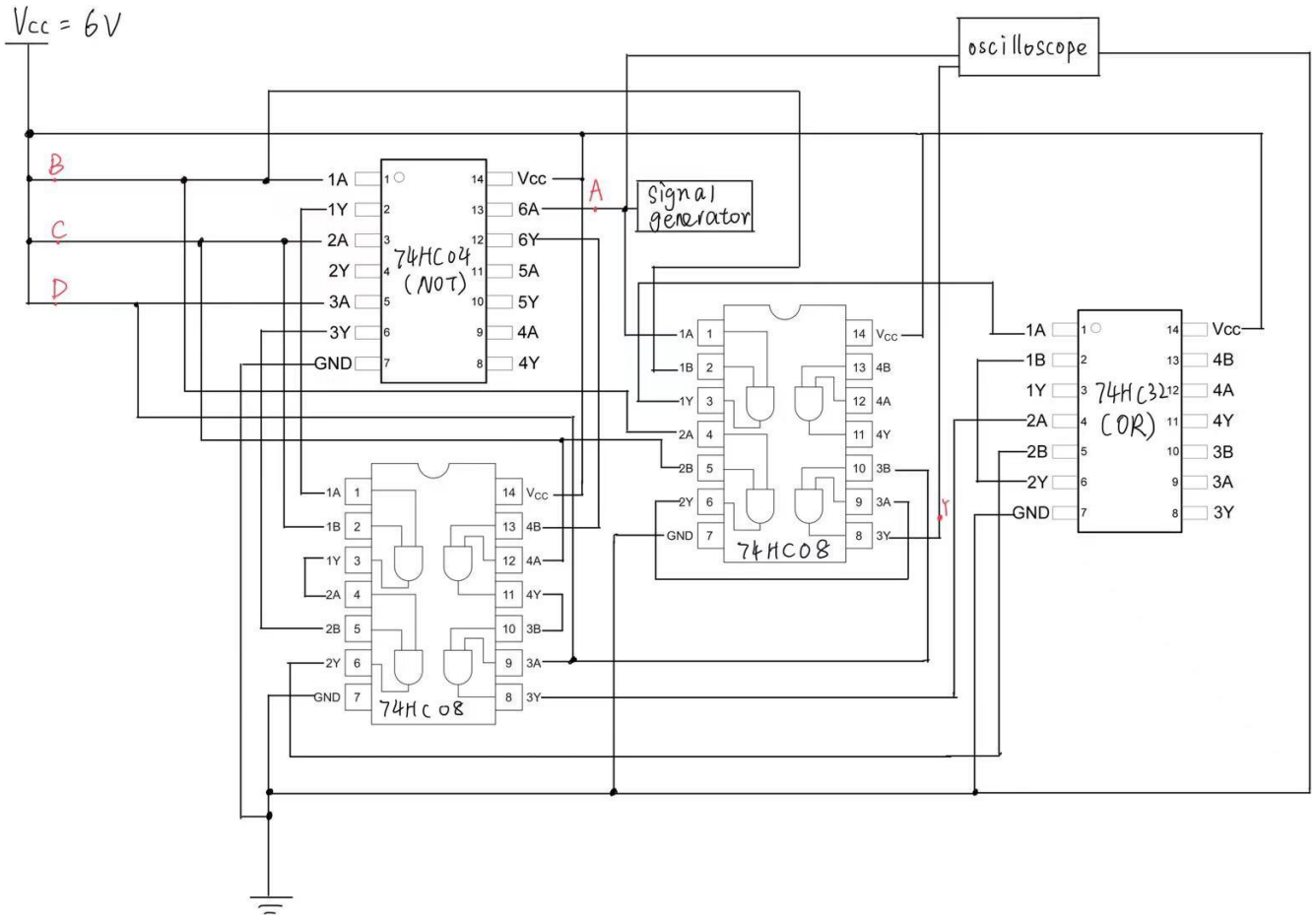

Figure 9. Hardware circuit design with additional term BCD (chip level)

2.2 Result

- Karnaugh map

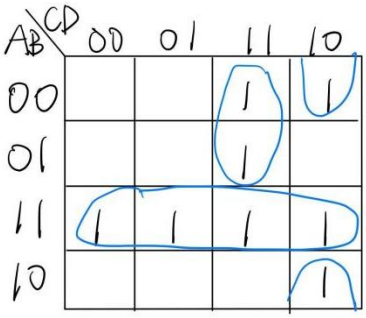

Figure 10. Karnaugh map for Y = AB + $\bar{B}C\bar{D}$ + $\bar{A}CD$

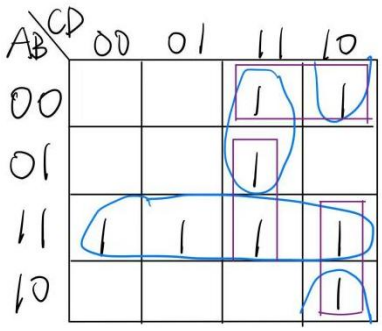

Figure 11. Places where timing hazard may occur (purple rectangle)

Timing hazard may occur at transition between 1111 (ABCD) and 0111 (ABCD); 0011 (ABCD) and 0010 (ABCD); 1110 (ABCD) and 1010 (ABCD).

- Glitch observation and elimination in Multisim

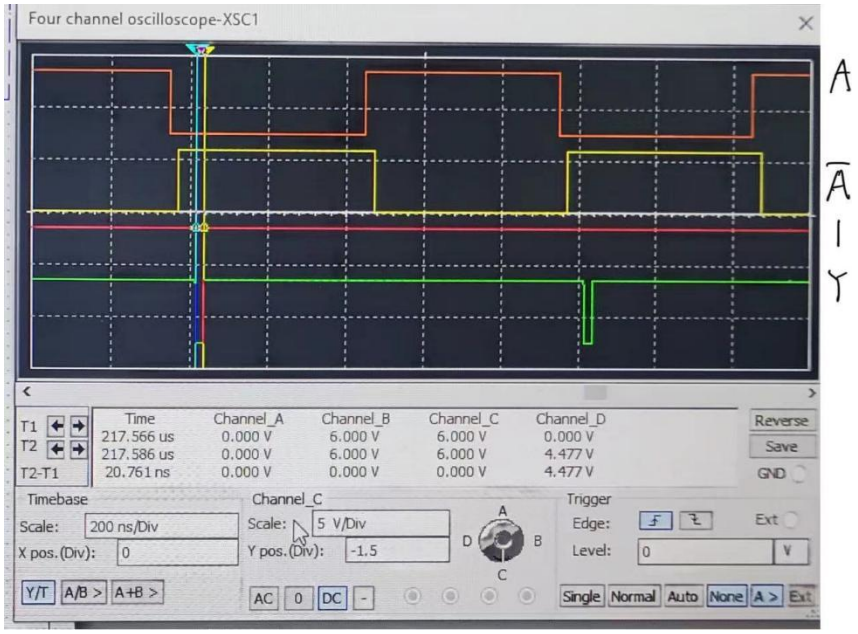

Figure 12. Glitch observed in original circuit (circuit in figure 4)

In Figure 12, it showed the propagation delay of 74HC04 (NOT gate). The timing hazard in the output lasted for about 20.76ns.

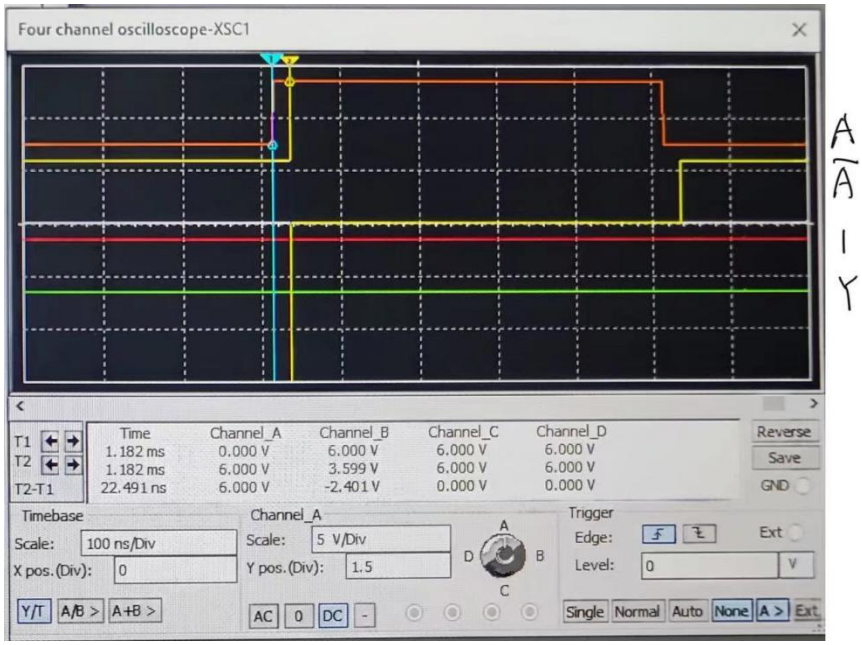

Figure 13. Glitch eliminated (circuit in figure 5)

In Figure 13, the propagation delay of NOT gate remained while the glitch in output disappeared.

- Hardware waveforms

In the below two figures, the upper one (yellow) is the waveform of output Y, the lower one (blue) is the waveform of input A.

Figure 14. Waveform before hazard elimination

Figure 15. Waveform after hazard elimination

Compare Figure 14 and Figure 15, we can see that the hazard disappeared after adding the term BCD.

2.3 Questions

Q1. There are 2 other timing hazards in this circuit. What are they, and how to eliminate them by adding terms?

A1. Between transition 0011 (ABCD) and 0010 (ABCD); 1110 (ABCD) and 1010 (ABCD).

Should add term $\bar{A}\bar{B}C$ to eliminate the hazard between 0011 and 0010, $AC\bar{D}$ to eliminate the hazard between 1110 and 1010.

Q2. In this experiment, we examine the timing hazard in combinational logic in the format $Y = A + \bar{A}$. There is another format, i.e. $Y = A\bar{A}$, which may also have timing hazard. Try to explain why.

A2. These hazards arise due to the different propagation delays in the paths of the circuit.

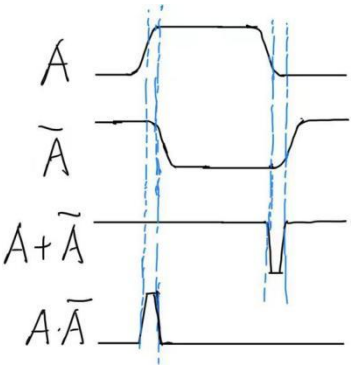

Figure 16. Illustration of propagation delay

1. $Y = A + \bar{A}$

Ideally, the output Y should always be HIGH regardless of the value of A.
However, during the transition of A from HIGH to LOW, there is a propagation delay in the NOT Gate that generates $\bar{A}$. For a brief moment, both A and $\bar{A}$ might be read as LOW by the OR Gate due to this delay, causing Y to momentarily go LOW.

2. $Y = A\bar{A}$

Ideally, the output Y should always be LOW regardless of the value of A.
However, during the transition of A from LOW to HIGH, there is a propagation delay in the NOT Gate that generates $\bar{A}$. For a brief moment, both A and $\bar{A}$ might be read as HIGH by the AND Gate due to this delay, causing Y to momentarily go HIGH.

3. Experiment C

3.1 Design

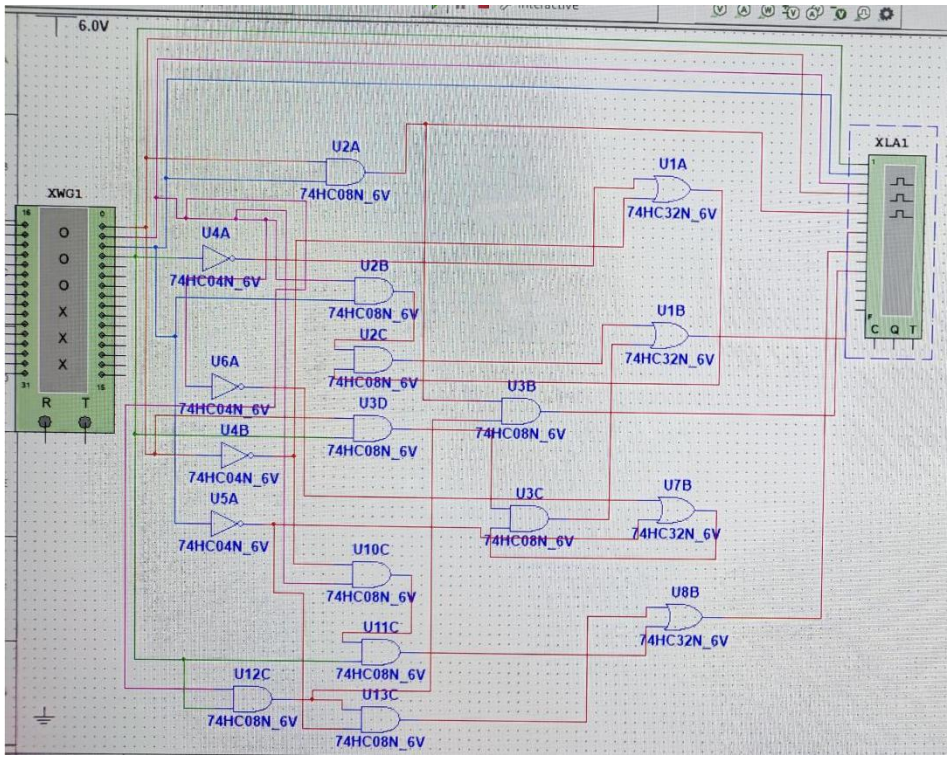

Figure 17. Gate level circuit for A1A0*B1B0 (the outputs of the Word Generator are A0, A1, B0, B1, from up to down)

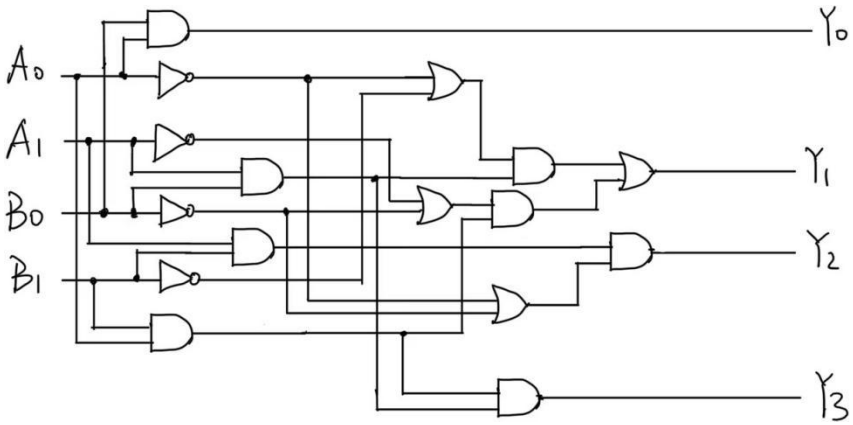

Figure 18. Hardware circuit design (gate level)

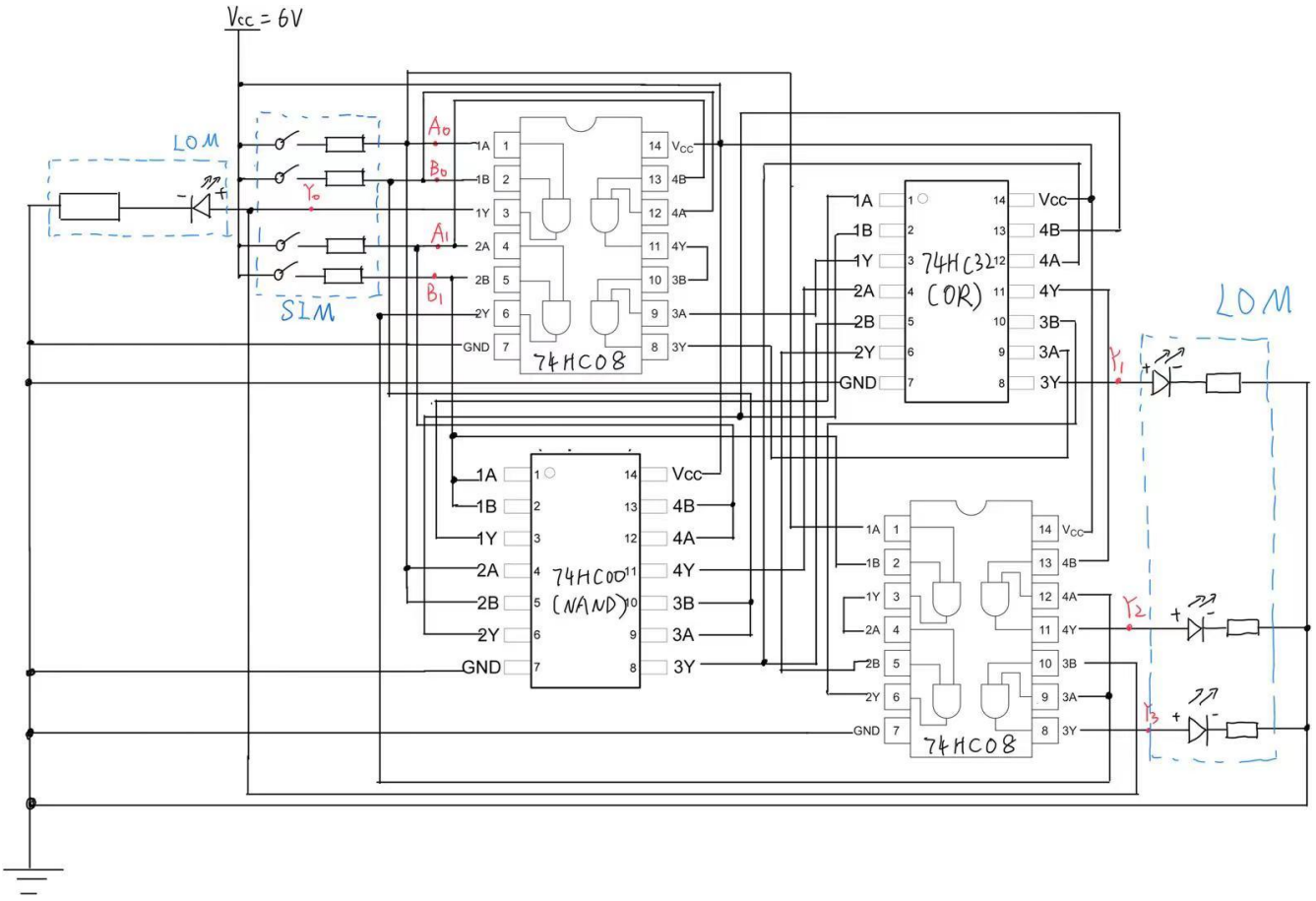

Figure 19. Hardware circuit design (chip level)

3.2 Result

- Truth table and Karnaugh map

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 2. Truth table for A1A0*B1B0

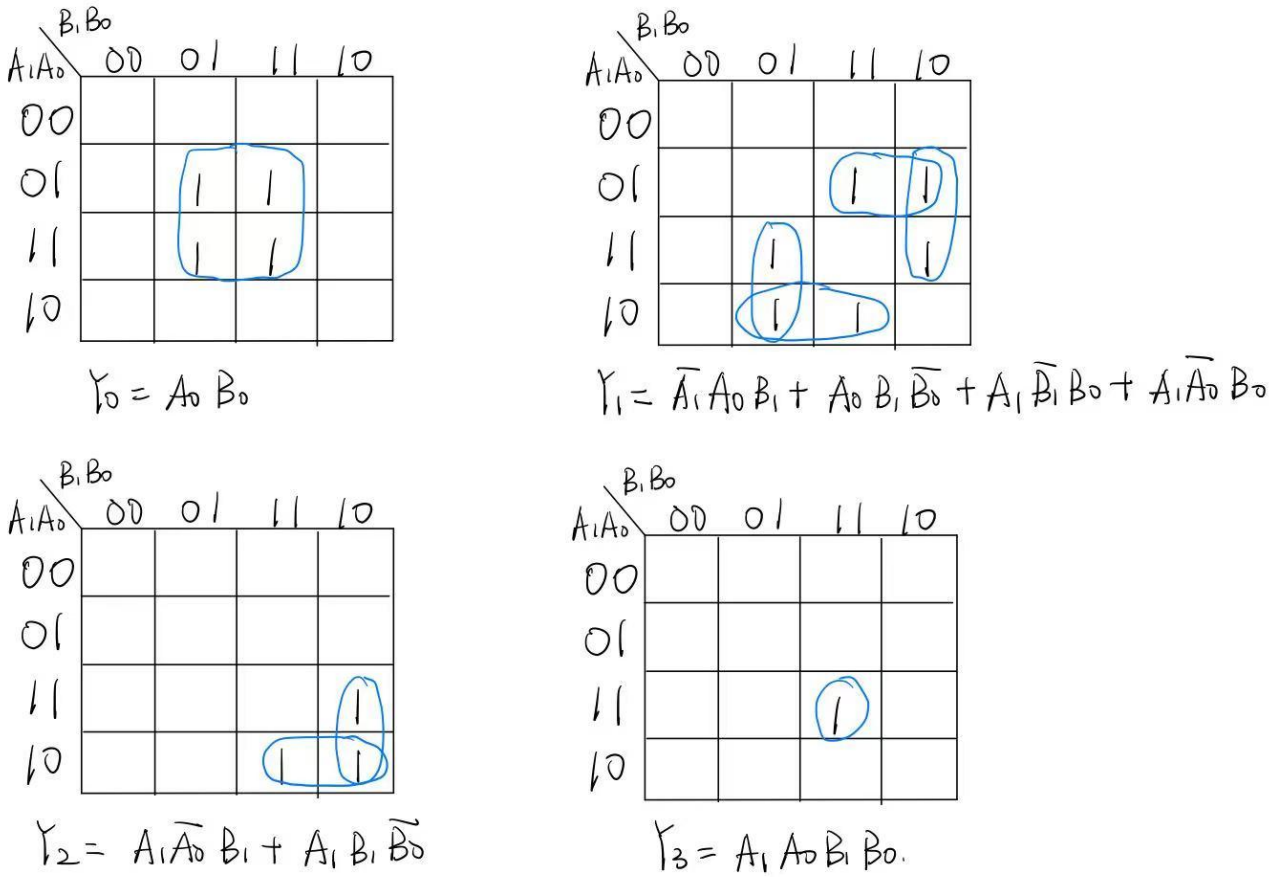

Figure 20. Karnaugh map and minimized SOP Boolean expression for Y0, Y1, Y2, Y3 each

- Multisim test result

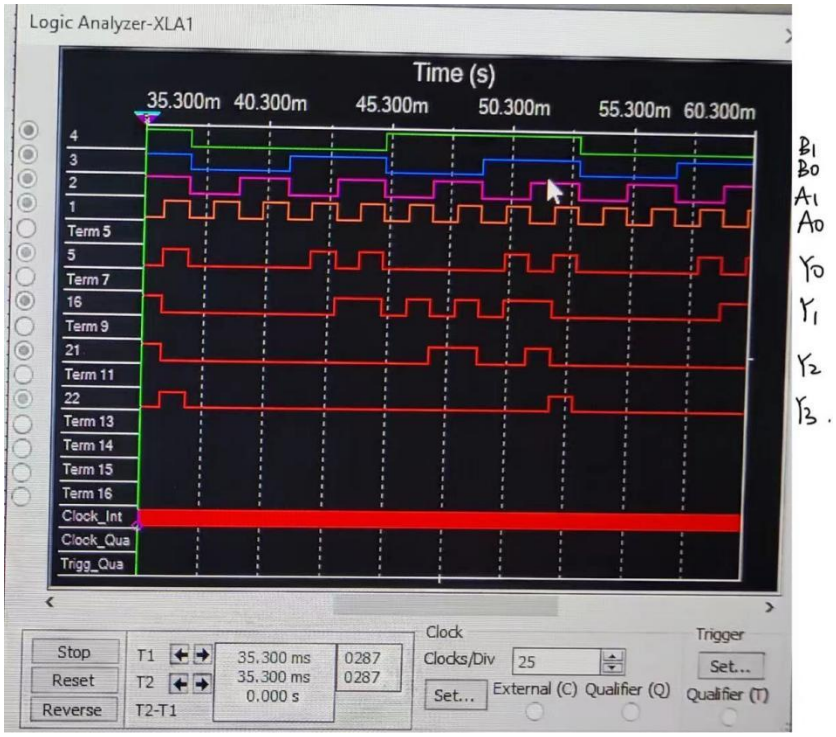

Figure 21. Waveform of the circuit in Figure 17

This result is consistent with the truth table.

- Hardware experiment result

In bellow figures, the SIM inputs are A1, A0, B1, B0 respectively (from right to left) , LOM outputs are Y0, Y1, Y2, Y3 respectively (from right to left).

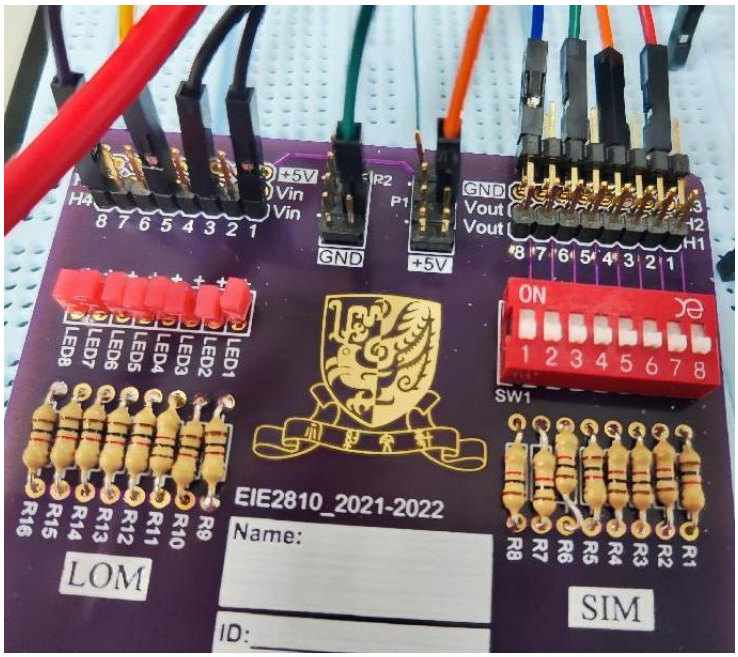

Figure 22. When input is (0, 0, 0, 0) (A1, A0, B1, B0), output is (0, 0, 0, 0) (Y3, Y2, Y1, Y0)

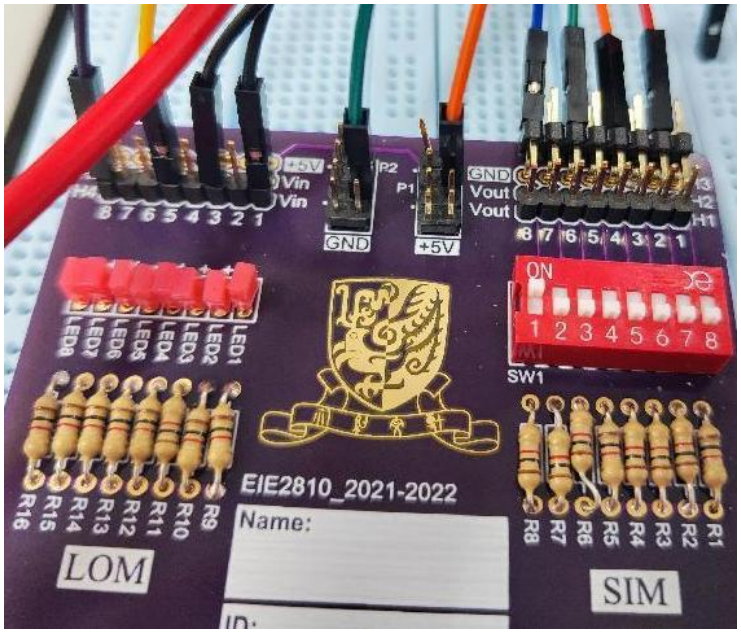

Figure 23. When input is (0, 0, 0, 1) (A1, A0, B1, B0), output is (0, 0, 0, 0) (Y3, Y2, Y1, Y0)

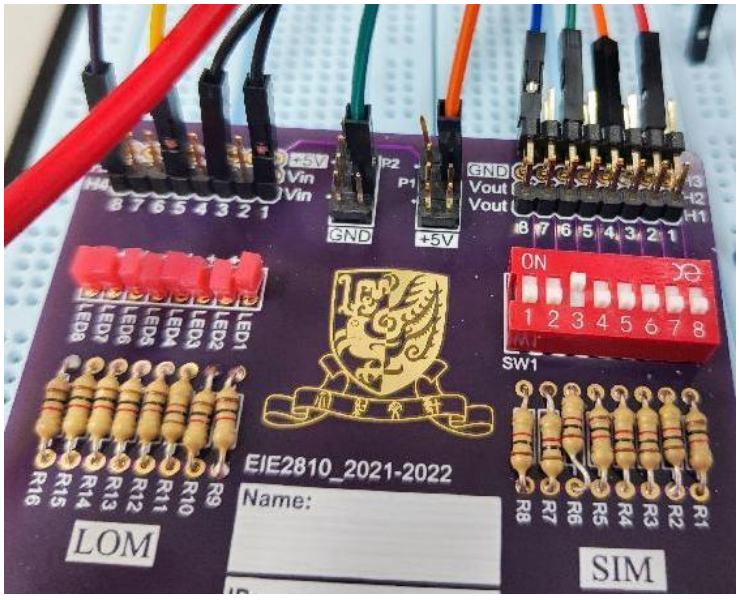

Figure 24. When input is (0, 0, 1, 0) (A1, A0, B1, B0), output is (0, 0, 0, 0) (Y3, Y2, Y1, Y0)

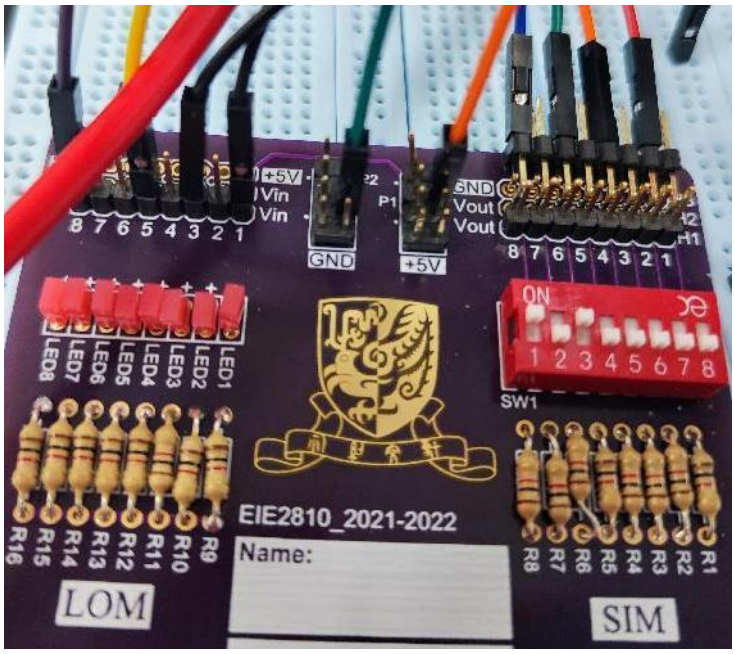

Figure 25. When input is (0, 0, 1, 1) (A1, A0, B1, B0), output is (0, 0, 0, 0) (Y3, Y2, Y1, Y0)

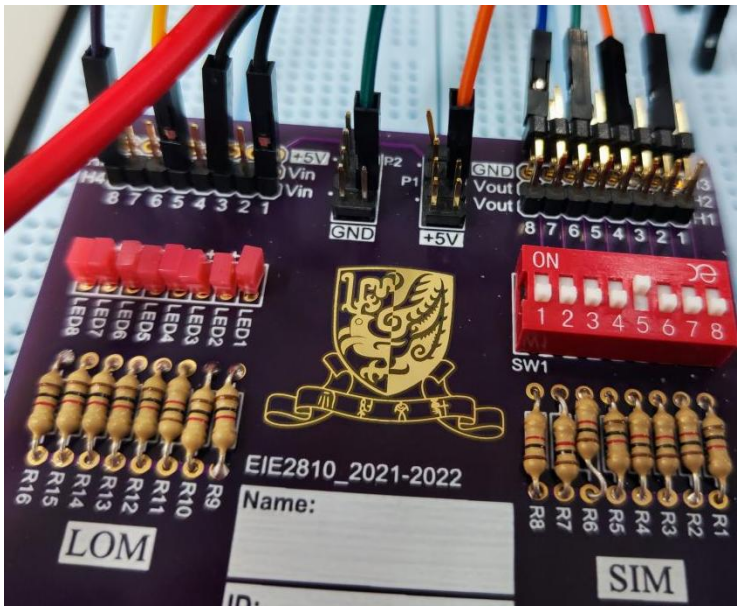

Figure 26. When input is (0, 1, 0, 0) (A1, A0, B1, B0), output is (0, 0, 0, 0) (Y3, Y2, Y1, Y0)

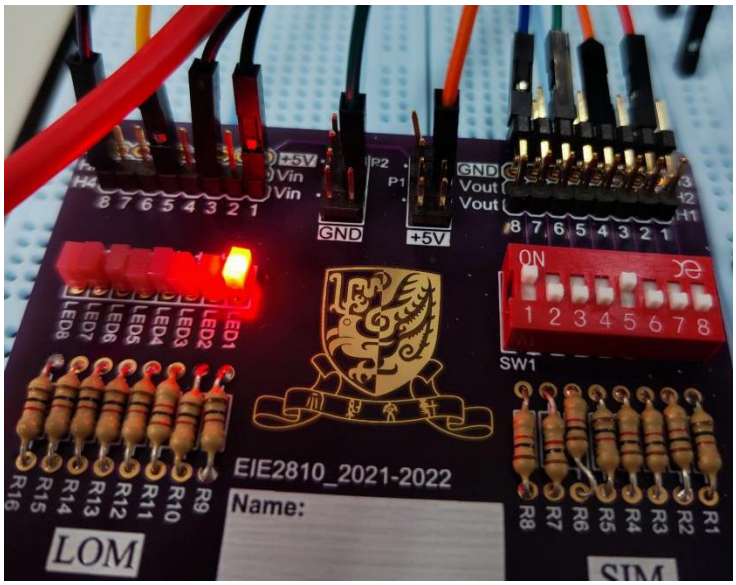

Figure 27. When input is (0, 1, 0, 1) (A1, A0, B1, B0), output is (0, 0, 0, 1) (Y3, Y2, Y1, Y0)

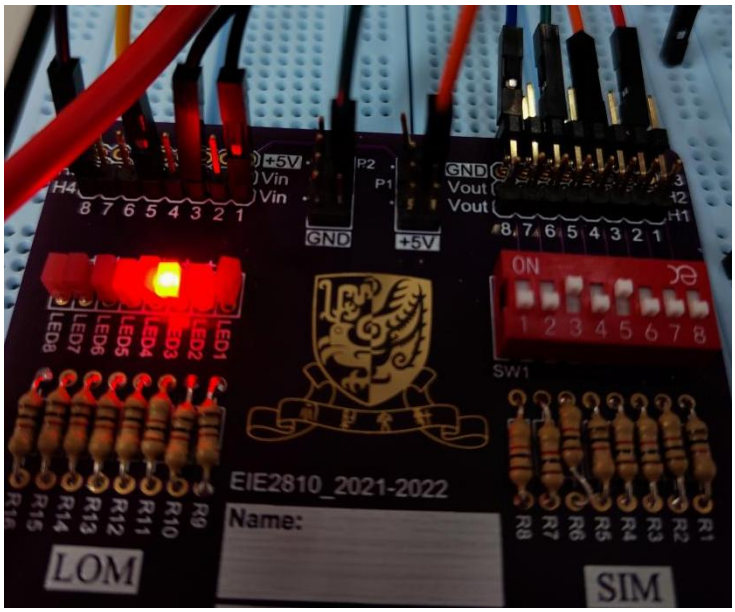

Figure 28. When input is (0, 1, 1, 0) (A1, A0, B1, B0), output is (0, 0, 1, 0) (Y0, Y1, Y2, Y3)

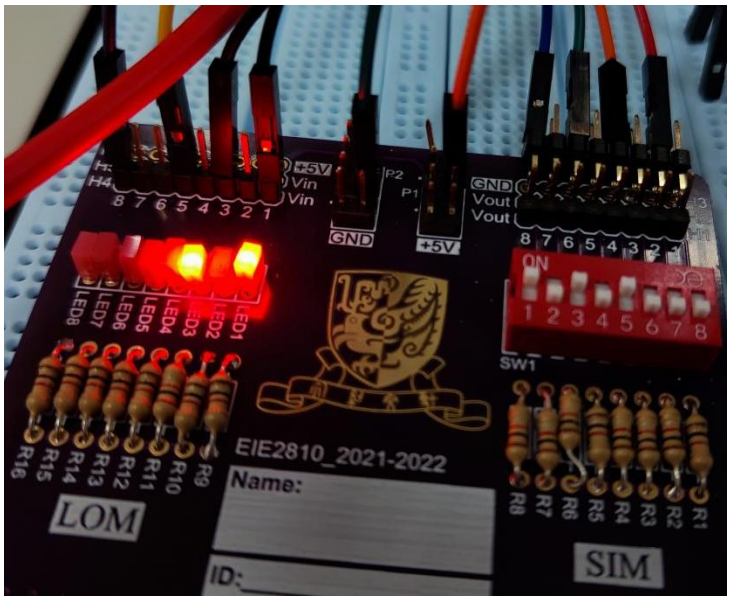

Figure 29. When input is (0, 1, 1, 1) (A1, A0, B1, B0), output is (0, 0, 1, 1) (Y3, Y2, Y1, Y0)

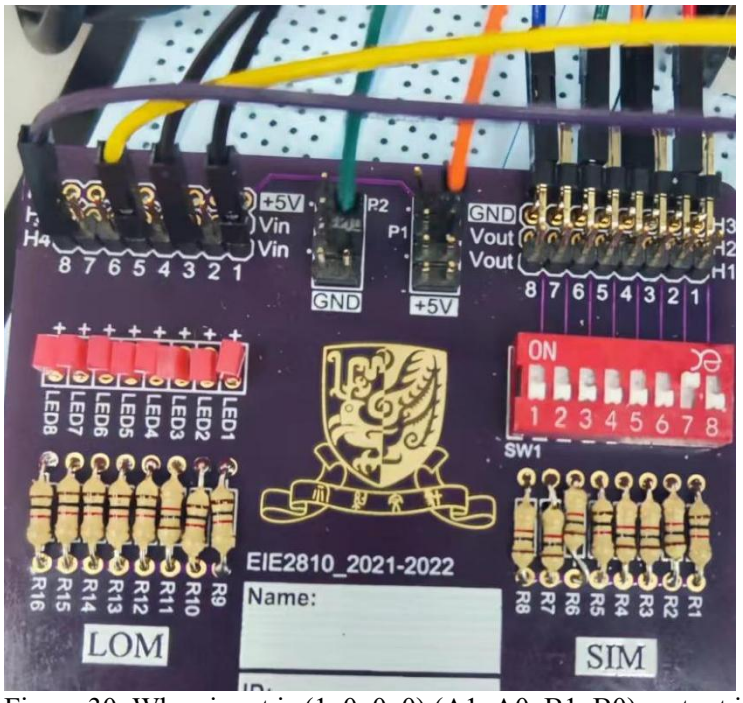

Figure 30. When input is (1, 0, 0, 0) (A1, A0, B1, B0), output is (0, 0, 0, 0) (Y3, Y2, Y1, Y0)

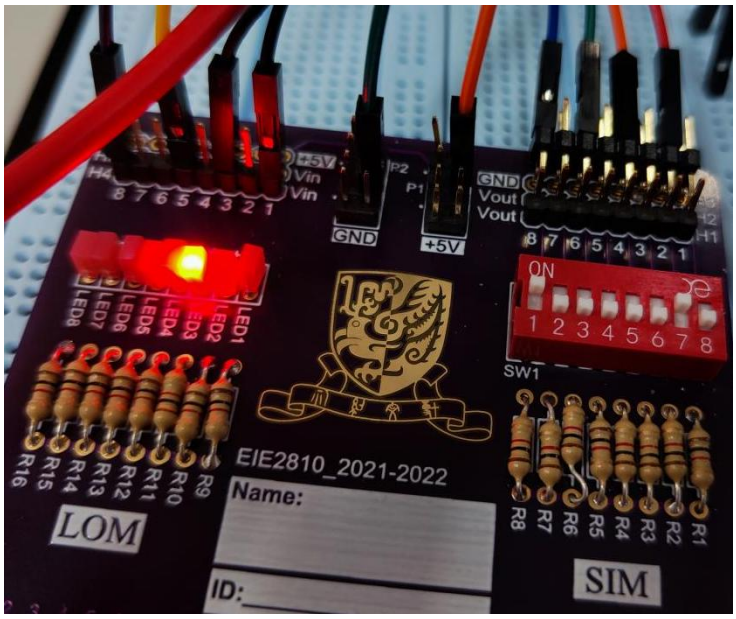

Figure 31. When input is (1, 0, 0, 1) (A1, A0, B1, B0), output is (0, 0, 1, 0) (Y3, Y2, Y1, Y0)

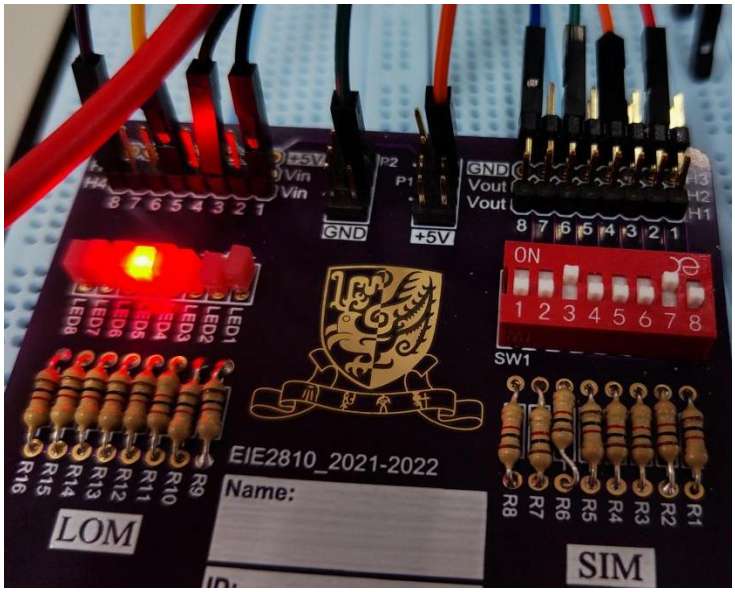

Figure 32. When input is (1, 0, 1, 0) (A1, A0, B1, B0), output is (0, 1, 0, 0) (Y3, Y2, Y1, Y0)

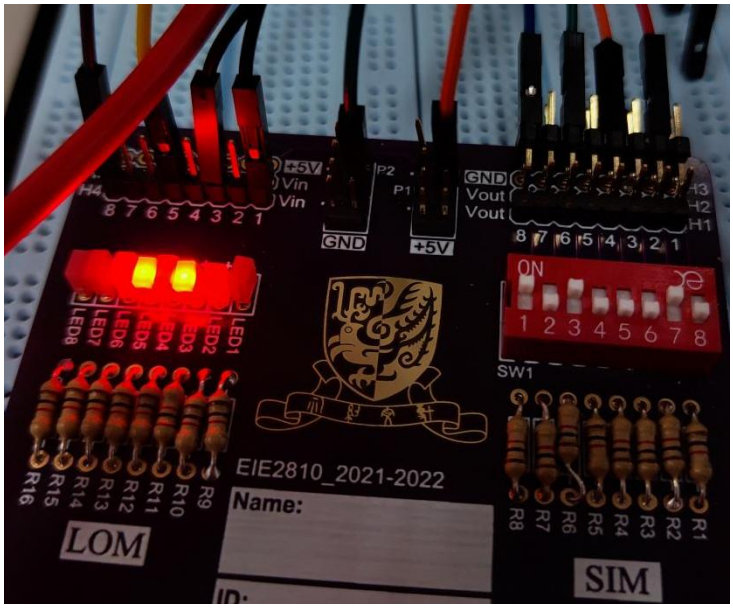

Figure 33. When input is (1, 0, 1, 1) (A1, A0, B1, B0), output is (0, 1, 1, 0) (Y3, Y2, Y1, Y0)

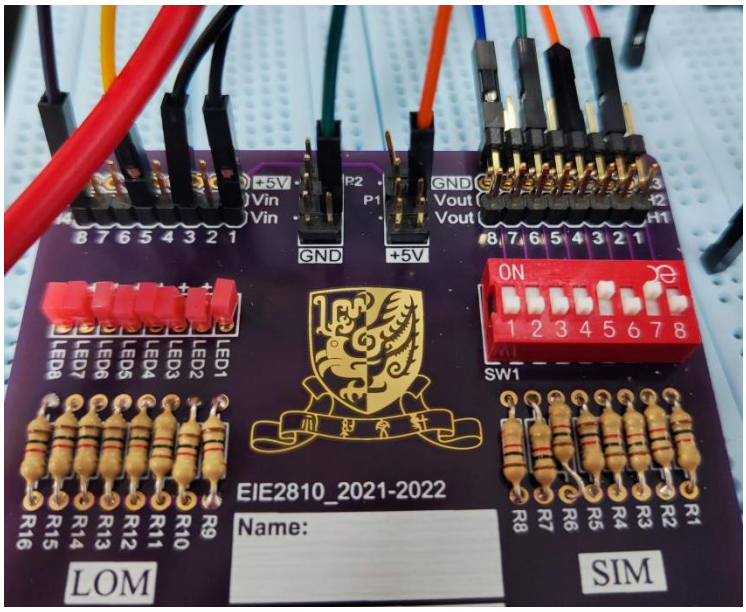

Figure 34. When input is (1, 1, 0, 0) (A1, A0, B1, B0), output is (0, 0, 0, 0) (Y3, Y2, Y1, Y0)

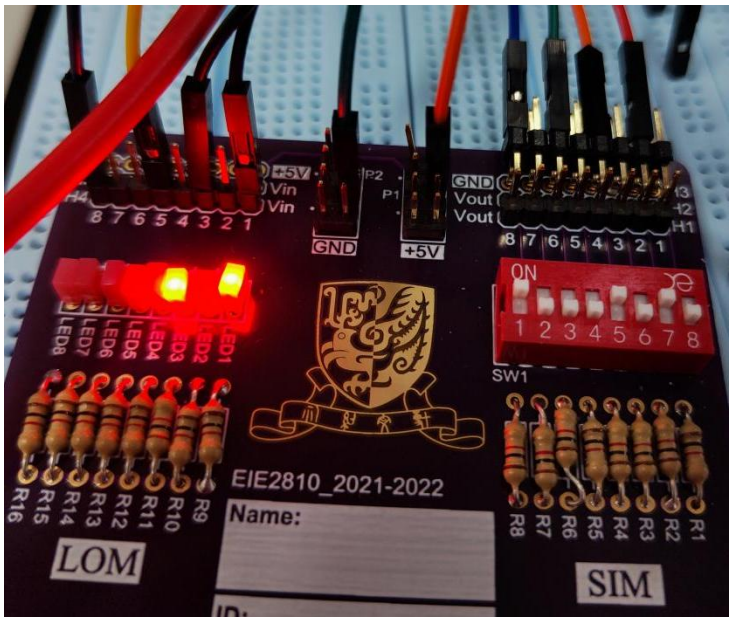

Figure 35. When input is (1, 1, 0, 1) (A1, A0, B1, B0), output is (0, 0, 1, 1) (Y3, Y2, Y1, Y0)

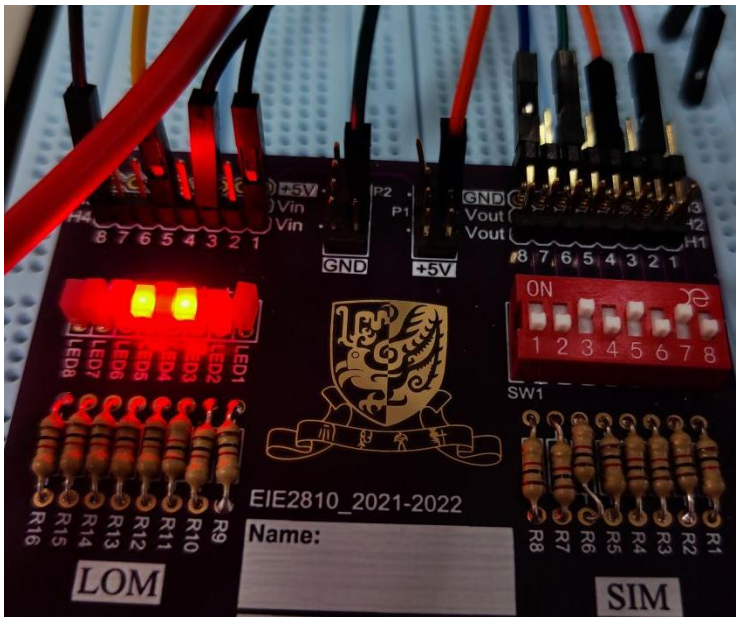

Figure 36. When input is (1, 1, 1, 0) (A1, A0, B1, B0), output is (0, 1, 1, 0) (Y3, Y2, Y1, Y0)

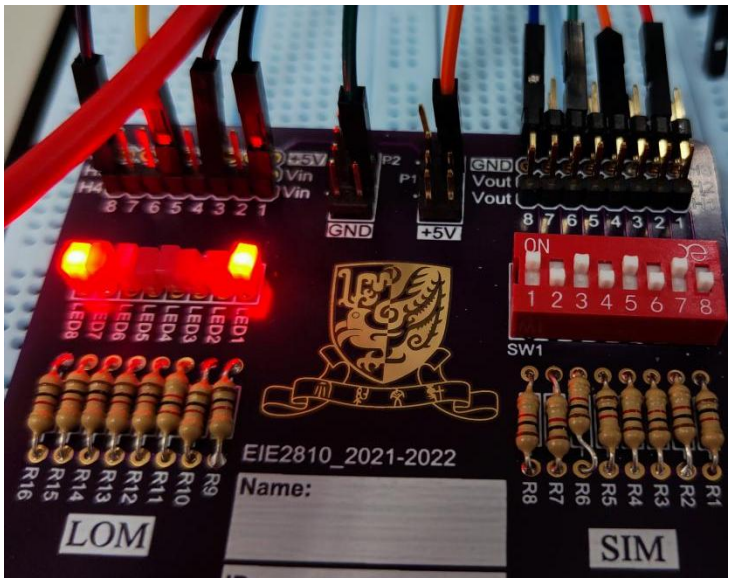

Figure 37. When input is (1, 1, 1, 1) (A1, A0, B1, B0), output is (1, 0, 0, 1) (Y3, Y2, Y1, Y0)

The result is consistent with the truth table.

4. Conclusion

In this lab, I

- Used Word Generator, Logic Analyzer, Function Generator and Four-Channel Oscilloscope in Multisim.
- Understood and eliminated timing hazard in combinational logic circuit.
- Built a combinational logic circuit for a 2-bit multiplied by 2-bit operation.

I learned:

- It's better to connect the output of the Word Generator and the input of the Logic Analyzer in reverse order (i.e. the LSB of the Word Generator to pin 4 in the Logic Analyzer, and the MSB in the Word Generator to pin 1 in the Logic Analyzer). In this way, the waveforms in the Logic Analyzer are displayed in order.
- When running a simulation in Multisim, the waveforms displayed in the Logic Analyzer update rapidly, making it difficult to analyze specific details. Should pause or stop the simulation to observe.
- In reality, logic gates can cause propagation delay, which can lead to glitches in the circuit. To eliminate them, we can add redundant terms that connect each pair of adjacent circles in the Karnaugh map. These terms do not change the overall function of the circuit but ensure that the output remains stable during input transitions.
- When observing the timing hazard in $Y = AB + \bar{B}C\bar{D} + \bar{A}CD$, to magnify the glitch and make it easier to observe, we can add the terms $\bar{B}C\bar{D}$, $\bar{A}CD$ first and then add AB; we can also use multiple NOT Gates in series (a "buffer chain") to produce $\bar{A}$ to introduce additional delays.
- 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. For example, to build A1$\bar{A}$0B1 + A1$\bar{B}$0B1, changing it into A1B1($\bar{A}$0 + $\bar{B}$0) can save one AND Gate.