ECE2810 Digital Systems Design Laboratory

Laboratory 3

Combinational Logic, Truth Table, Karnaugh Map

School of Science and Engineering
The Chinese University of Hong Kong, Shenzhen

2025-2026 Term 1

1. Objectives

In Laboratory 3, we will spend the 2-week sessions on the following:

- Learn to use a word generator, logic analyzer, function generator, and four-channel oscilloscope, and wire basic chips in Multisim.
- Practice how to obtain the truth tables and Karnaugh Map for combinational logic circuits.
- Learn the timing hazards in combinational logic circuits, and how to eliminate them.
- Learn to build a combinational logic circuit for a 2-bit by 2-bit circuit.

2. Multisim: Word Generator / Logic Analyzer / Function Generator / Four Channel Oscilloscope / Chips

2.1. Word Generator

The word generator can provide up to 32 channels of digital outputs, facilitating the generation of multi-channel synchronized waveforms. These waveforms can be used as inputs to test whether the circuit behaves consistently with the truth table. In the example below, we will add a word generator that produces 4 channels of digital outputs, ranging from 0b0000 (binary representation of decimal 0) to 0b1111 (binary representation of decimal 15).

- From the instrument column, add a Word generator, as shown in Figure 1.

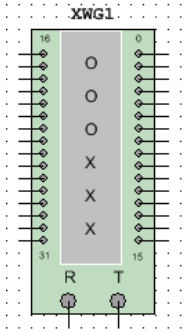

Figure 1. Word generator.

- As shown in Figure 2, double-click it to set the parameters. Choose “Binary”, then click on “Set…”

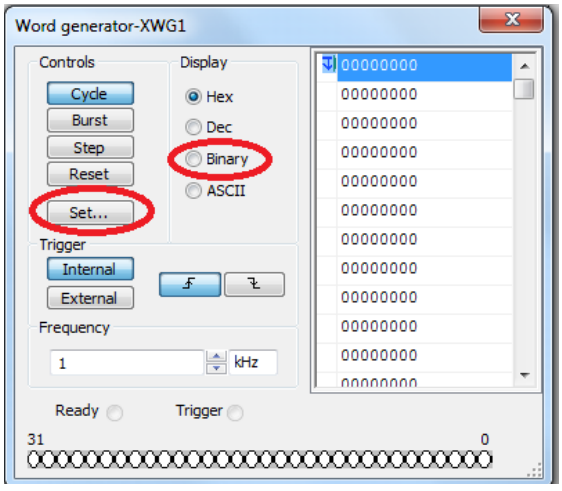

Figure 2. Word generator set dialogue box.

- As indicated in Figure 3, change “Buffer size” from “0400” to “0010” and select “Up counter”. This will generate a 4-bit output from “0b0000” to “0b1111”. Change the output voltage level from 4.5 V to 5.5 V (The 0.5 V can work as the input variance error for your system. You can also observe that the initial low voltage is 500mV, rather than 0V.), as we will use the CMOS6V gates in Multisim. Then click “OK”.

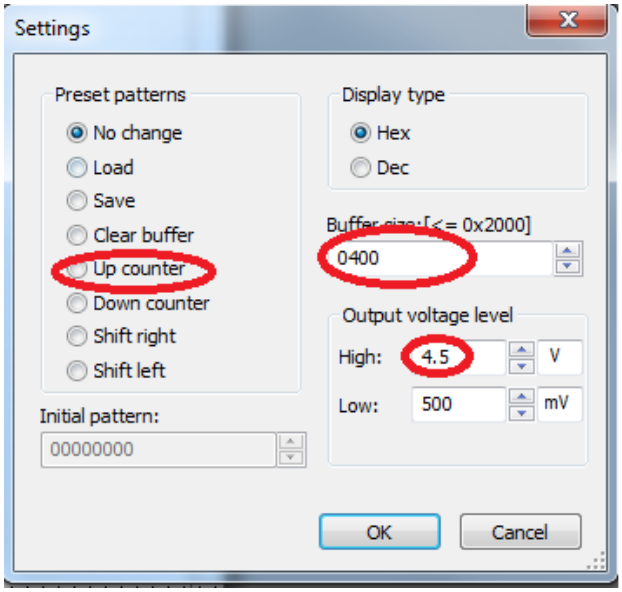

Figure 3. Setting the panel of the word generator.

- Back to the previous dialog box, you will find the sequence changed (Figure 4). Close this dialog box now.

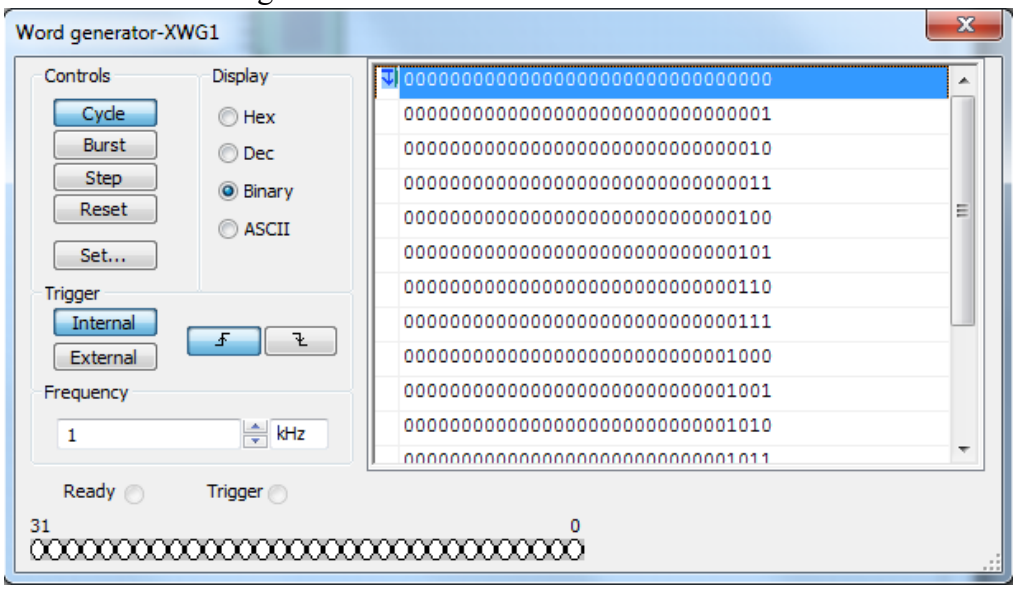

Figure 4. Cycle set in the word generator.

2.2. Logic Analyzer

A logic analyzer measures multiple channels of digital signals. It is similar to the real equipment in the lab. Below, we will try to add the logic analyzer and measure the 4 channels of signals from the word generator. Then construct a simple circuit $Y = ABCD$, and test its input-output.

- Add a logic analyzer from the instrument column, as shown in Figure 5.

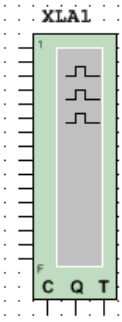

Figure 5. Logic analyzer.

- Connect the 4 channels of output from the word generator to the 4 inputs of the logic analyzer. To make it easy to identify the signals, in the word generator, it is suggested to map the Least Significant Bit (LSB), i.e., the pin marked with 0, to the 4th pin in the logic analyzer, and the Most Significant Bit (MSB) in the word generator to pin 1 in the logic analyzer. Figure 6 shows the connection. (FYI: For example, in the binary number 1010, the LSB is 0 (the rightmost bit), the MSB is 1 (the leftmost bit).)

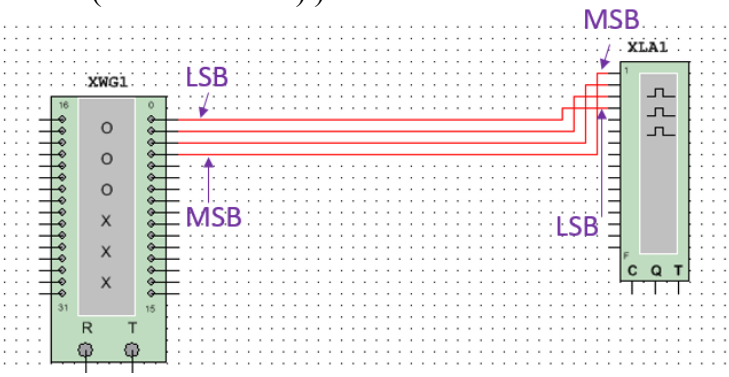

Figure 6. Wiring 1.

- Run the simulation, you can change the "Clocks/Div" to zoom in/out in the time axis, and find the waveform as shown in Figure 7.

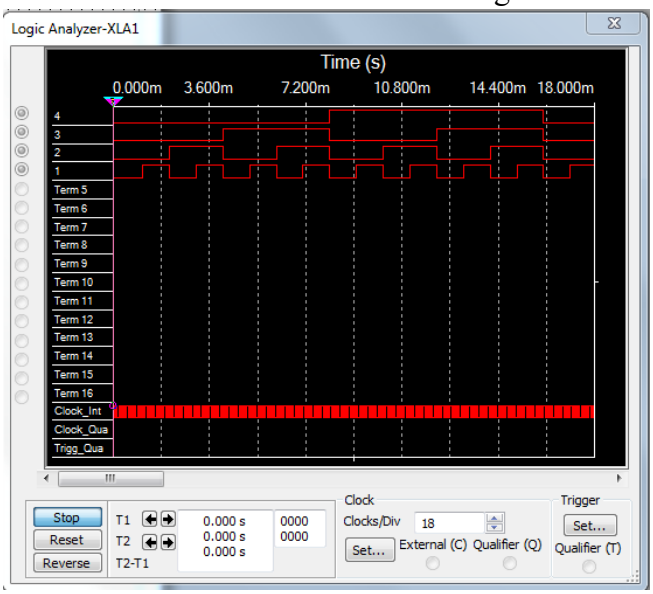

Figure 7. Waveform 1.

- Add AND gates into the circuit, like Figure 8. The combinational logic is $Y = ABCD$.

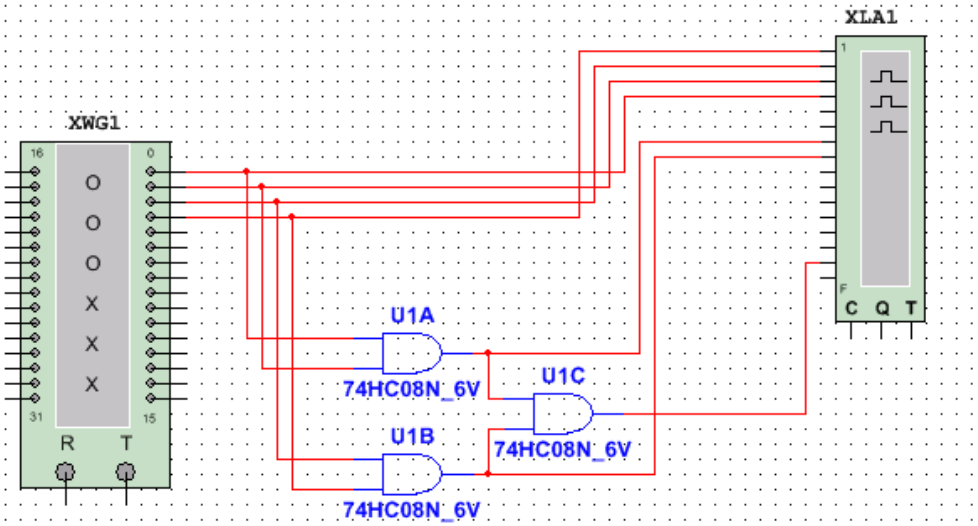

Figure 8. Wiring 2.

- Run the simulation, and you will find the waveform as shown in Figure 9. You can change the "Clocks/Div" to zoom in/out in the time axis.

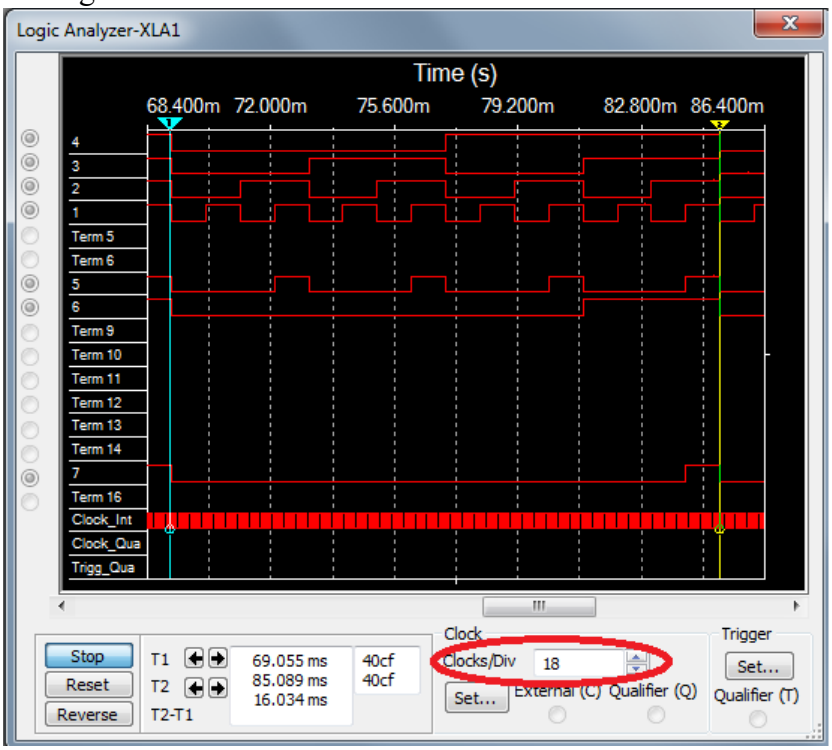

Figure 9. Waveform 2.

2.3. Function Generator

A function generator can generate square/triangular/sine waveforms. Below, we will show an example to generate a 1MHz square waveform, with low voltage at 0V, and high voltage at 6V. The duty cycle is 50%.

- Create a new circuit design, select a function generator from the Instrument column, and add it to the circuit, as shown in Figure 10.

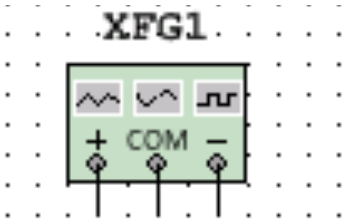

Figure 10. Function generator.

- Double-click the function generator to obtain the dialog box (Figure 11). Change waveform shape, frequency unit, amplitude, and offset. Then, close the dialog box.

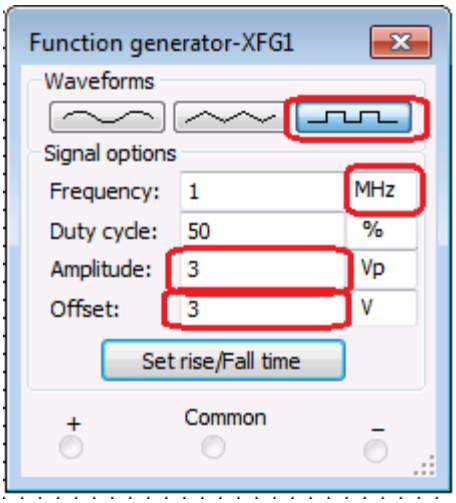

Figure 11. Parameter set panel in the function generator.

- The next step is to ground the COM pin (Figure 12). Then you can use "+" to output the waveform as you set.

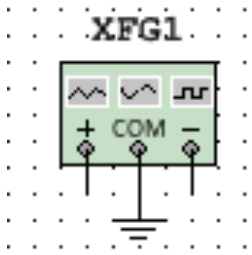

Figure 12. Wiring 3.

2.4. Four-Channel Oscilloscope

We will use a four-channel oscilloscope in the experiment.

- Select a four-channel oscilloscope (Figure 13) from the Instrument column and add it to the circuit.

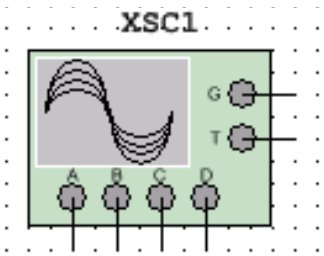

Figure 13. Four-channel oscilloscope.

- Connect Channel A to the output pin "+" of the function generator. For this four-channel oscilloscope, unlike the two-channel oscilloscope you have used, there is no ground in the probe.
- When you run the simulation, double-click the oscilloscope to pop up the dialogue box as shown in Figure 14. You can select the channel, change the time scale, and voltage scale to observe the wave.

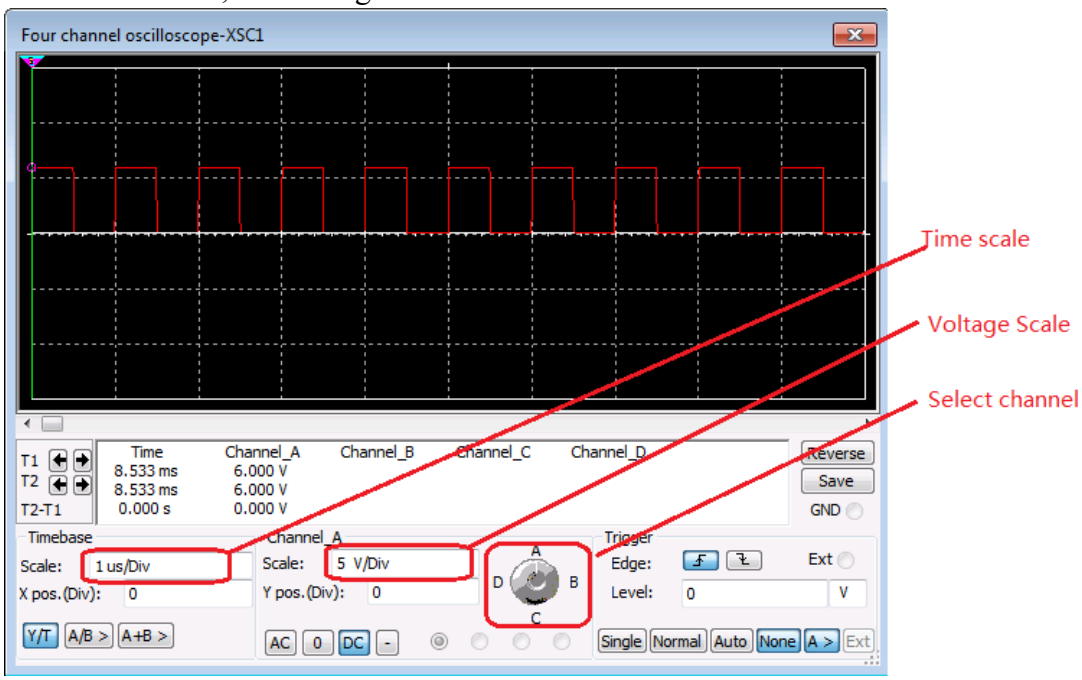

Figure 14. Panel of a four-channel oscilloscope.

2.5. Chips in Multisim

Aside from adding logic gates to the circuit, you can also add chips to the circuit. To build up Y=ABCD, select 74LS08N from TTL=>74LS_IC =>74LS08N. The chips of the 74HC series have not been included in the database yet. So, we will use the 74LS series in the simulation instead. Figure 15 shows the wiring.

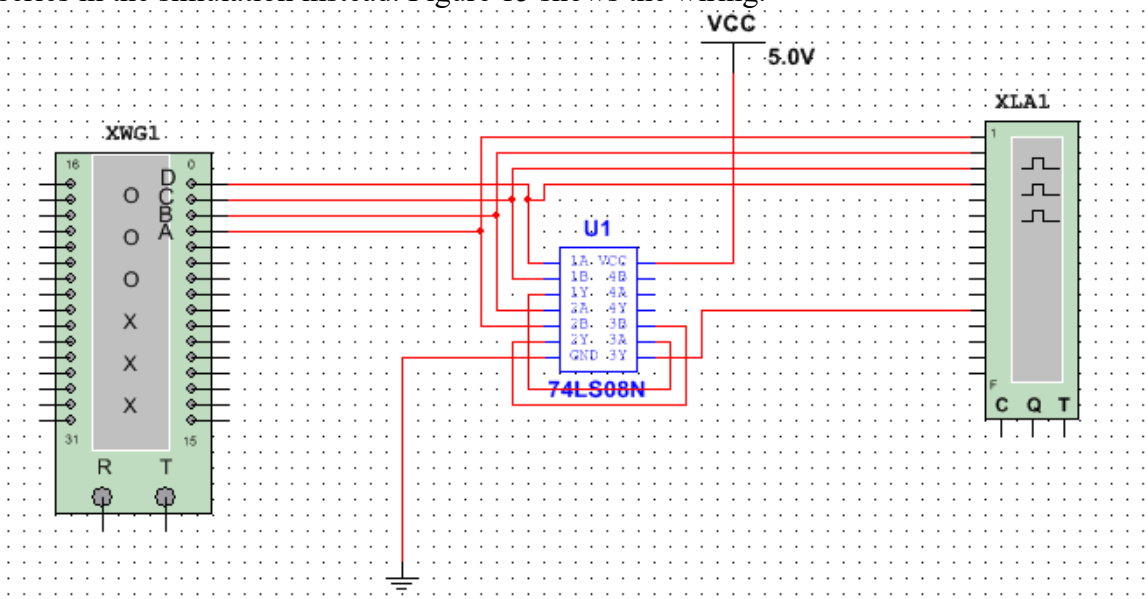

Figure 15. Wiring 4.

After setting up the word generator similarly to the previous part, you can run the simulation and obtain the result as shown in Figure 16.

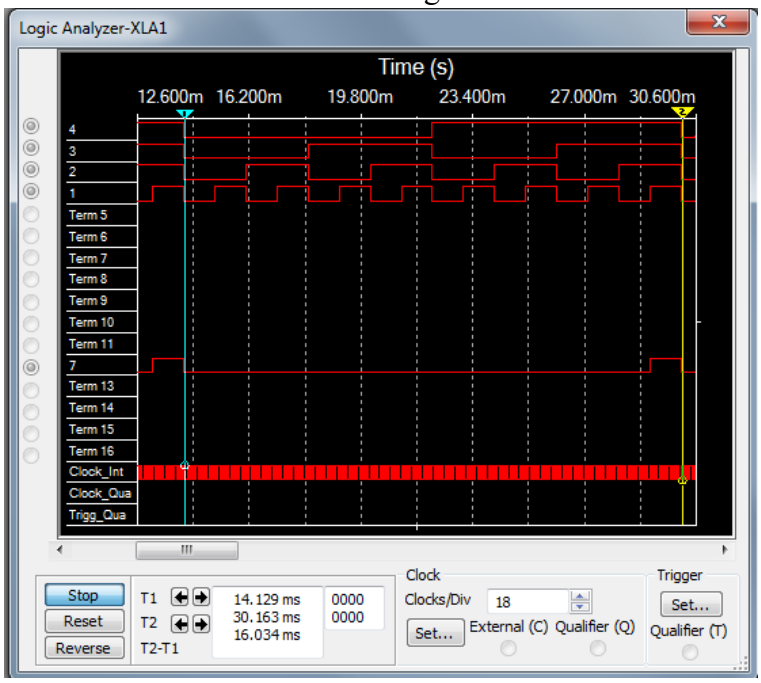

Figure 16. Waveform 3.

3. Combinational Logic

3.1. Truth Table and Karnaugh Map

The digital logic for real applications is more complex than the samples above. With the increase of complexity in combinational logic, to realize the circuit, one general way is (1) write out the truth table, and then (2) use a Karnaugh map to simplify the logic expression. In part 4, you will practice this approach.

3.2 Timing Hazard

When you have simplified the logic expression in the Karnaugh map, be careful. There might be a timing hazard, which is caused due to the delay in logic gates. This will cause undesirable effects.

Let's examine $Y = A + \overline{A}B$ as an example.

We can first build a circuit for the combinational logic in Multisim, like Figure 17. Use a function generator, one NOT, one OR, one AND, and a four-channel oscilloscope.

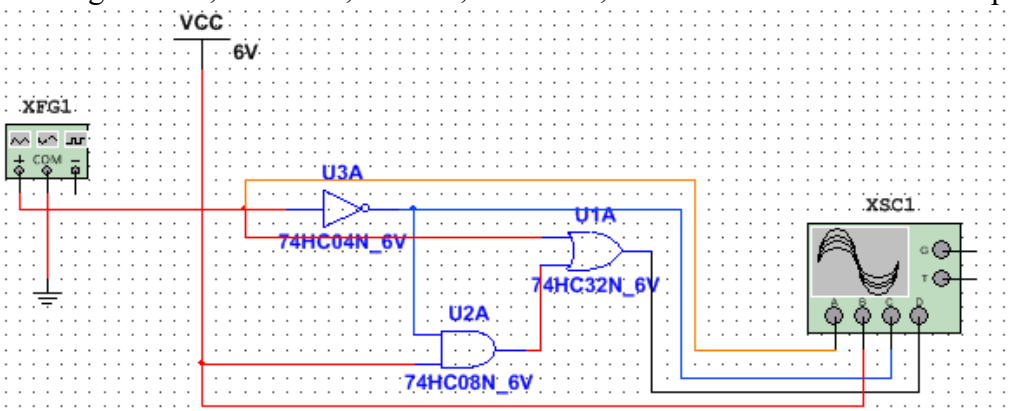

Figure 17. Wiring 5.

For the function generator, set it to output a 1 MHz square wave, with a 50% duty cycle, 6V as high voltage, and 0V as low voltage, as indicated in Figure 18.

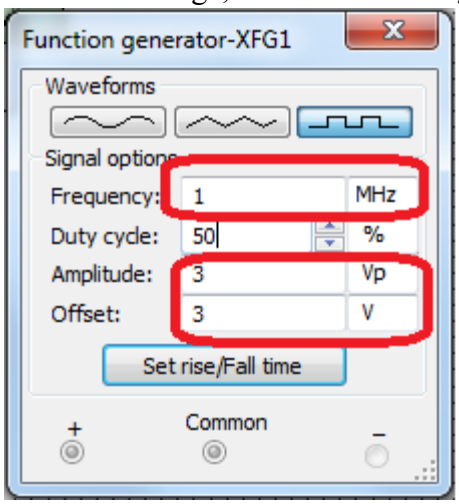

Figure 18. Parameter set panel in the function generator.

Run the simulation, and you will observe that the output is not consistent at 1 (as indicated by the white line in Figure 19).

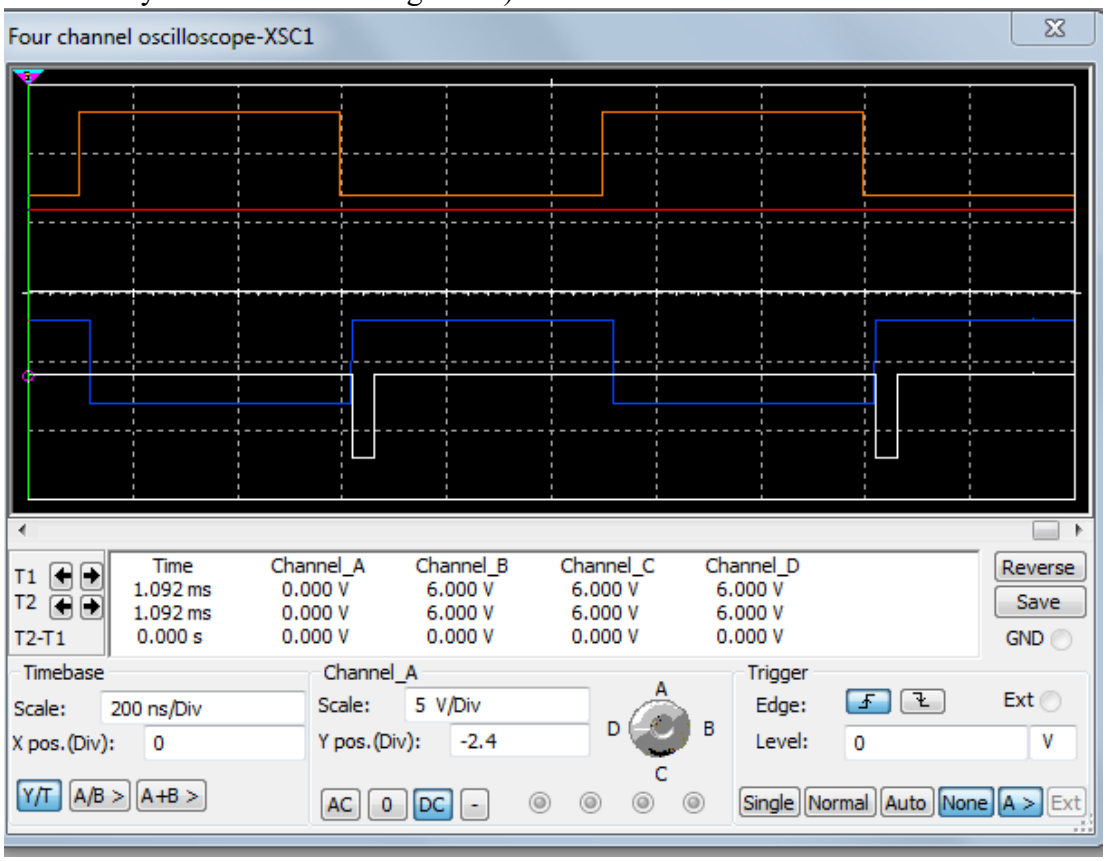

Figure 19. Glitch.

Due to the delay in the NOT gate, there is a brief period when A=0 and $\overline{A} = 0$, as shown in Figure 20. This is the cause of the glitch.

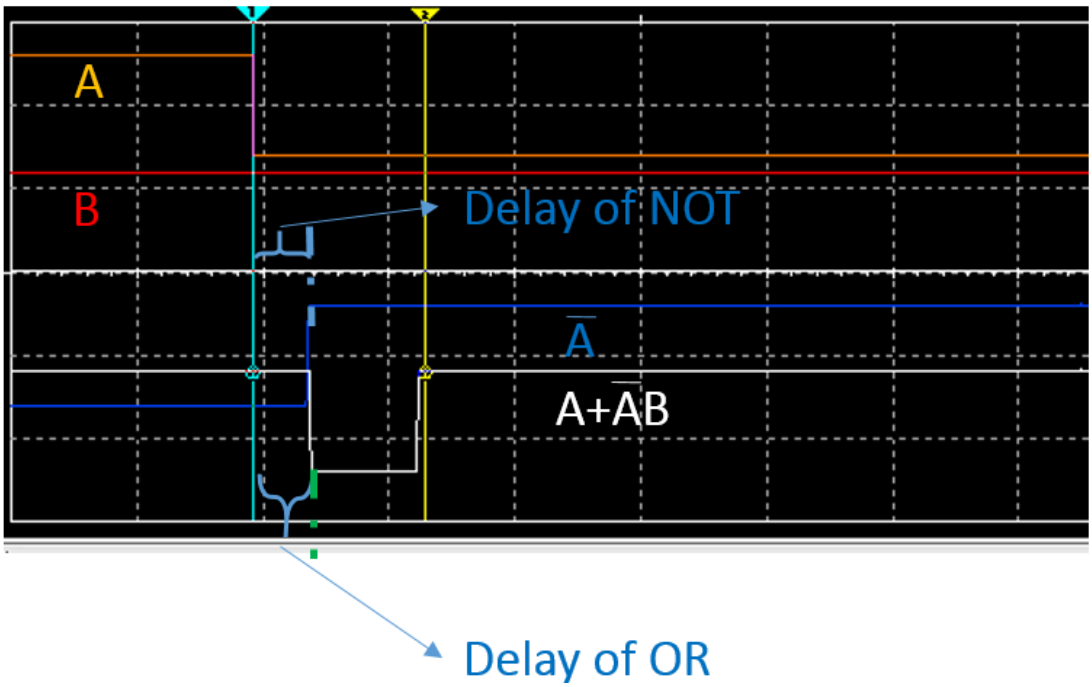

Figure 20. Delay analysis.

In the Karnaugh map (Figure 21), the 2 circles simplify the logic expression as two terms in $Y = A + \overline{A}B$.

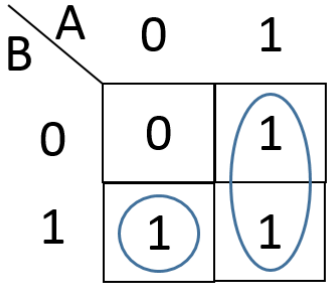

Figure 21. Karnaugh map.

When B=1, it is simplified to $Y = A + \overline{A}$. The delay in $\overline{A}$ causes the problem. Moreover, it can be observed that if there are any adjacent circles in the simplifications of the Karnaugh map, the problem will occur. To eliminate the timing hazard, one approach is to add an additional term that does not cause a logic conflict. In this example (Figures 22 and 23), you can add the additional circle, resulting in a term B. It will make the logic expression as $Y = A + \overline{A}B + B$.

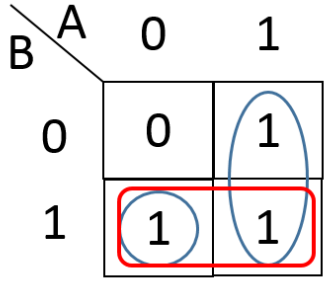

Figure 22. Term added in the Karnaugh map.

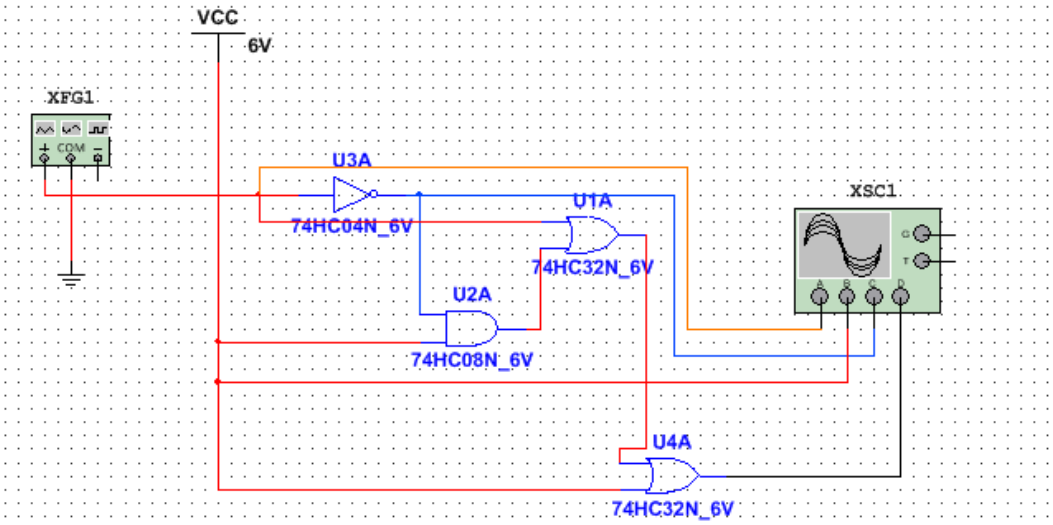

Figure 23. Term added in Multisim.

Though the delay in NOT still exists, the glitch has been eliminated (Figure 24).

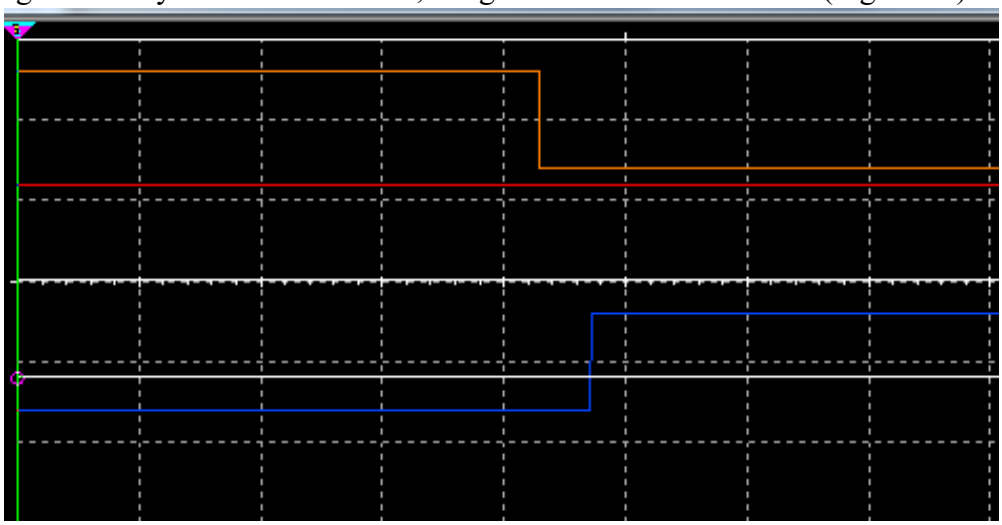

Figure 24. Glitch eliminated.

4. Experiments

4.1 Experiment 1: Combinational Logic Circuit

Realize and test the combinational logic of $Y = AB + \overline{B}C\overline{D} + \overline{A}CD$.

Procedures:

(1) Write the truth table for the inputs and output.

(2) In Multisim, build up the circuit with AND (74HC08N_6V), OR (74HC32_6V), and NOT (74HC04N_6V) gates. Use the word generator to produce input signals and use the logic analyzer to observe the input and output. Save the design file and record the waveform.

(3) After step (2), keep this circuit. You will still need it for the next experiment.

[DEMONSTRATION - 1] When you have completed step (2), please demonstrate it to the instructor or TA.

[IN REPORT] Include your truth table, designed circuits (in gate level) in Multisim, and waveform in step (2).

4.2 Experiment 2: Timing Hazard

There is a timing hazard if you implement the combinational logic of $Y = AB + \overline{B}C\overline{D} + \overline{A}CD$ directly by the sum of 3 products. This experiment will enable you to test the timing hazard and eliminate it.

There are three state transitions that may have timing hazards. One transition is between 1111(ABCD) and 0111 (ABCD). Note: Whether a timing hazard exists from 1111 to 0111, or from 0111 to 1111, depends on which signal combination (A or $\overline{A}$) arrives at the final OR gate faster. The number of gates for A and $\overline{A}$ to travel through is different. If A is ahead of $\overline{A}$, timing hazard occurs when the input changes from 1111 to 0111. Otherwise, it is from 0111 to 1111.

Procedures include **simulation** and **hardware** implementation:

(1) Build up the Karnaugh map, and find the places where timing hazards will occur. (Hint: If there are adjacent circles that group all "1"s, there will be a timing hazard.)

(2) Continue with the simulation circuit in step (2) of Experiment 1. Remove the word generator and replace it with the function generator in Multisim to generate a high-frequency square wave (e.g., 1 MHz) as input to A. Set the inputs of B, C, and D to VCC. Use the oscilloscope (not the logic analyzer) to observe the input A and output Y. You should expect to see a glitch.

(3) By adding additional terms in $Y = AB + \overline{B}C\overline{D} + \overline{A}CD$, in Multisim, try to eliminate the timing hazard between 1111(ABCD) and 0111 (ABCD).

(4) Build the circuit on the breadboard for steps (2) and (3). You are allowed to use four chips: 74HC04 x1, 74HC08 x2, and 74HC32 x1. **These four chips will be enough to complete the design.** Use the oscilloscope to observe the output before and after eliminating the glitch.

[DEMONSTRATION - 2] Demonstrate to the instructor or TA when you have:

(1) Identify the glitches in step (2) in Multisim and eliminate the timing hazard in Multisim in step (3).

(2) Build the circuit on the breadboard and observe the elimination of the glitch.

[IN REPORT] Include all the necessary designs and observations in this experiment.
[QUESTIONS IN THE REPORT] (1) There are 2 other timing hazards in this circuit. What are they, and how can we eliminate them by adding terms?

4.3 Experiment 3: Combinational Logic Circuit for Multiplying Two 2-Bit Numbers

Design and build a combinational logic circuit with four inputs and four outputs. Its function is to multiply two 2-bit numbers, labeled A1, A0, and B1, B0. The outputs are labeled Y3, Y2, Y1, and Y0. Use AND, OR, and NOT gates.
Procedures of this experiment include:

(1) Create a truth table with A1, A0, B1, B0 as inputs, and Y3, Y2, Y1, Y0 as outputs.

(2) Construct the Karnaugh map and derive the SOP Boolean expression.

(3) Use Multisim to build the circuit at the chip level. You are allowed to use four chips: 74LS04 x1, 74LS08 x2, 74LS32 x1. **These four chips will be enough to complete the design.** Use the word generator to produce input signals and use the logic analyzer to observe the input and output. Validate your circuit is functioning as expected.

[DEMONSTRATION - 3] When you have realized the circuit in simulation (step (3)), please demonstrate it to the instructor or TA.
[IN REPORT] Include all the necessary designs and observations in this experiment.

5. Lab Report

Write the lab report comprehensively. A template has been provided on Blackboard. You can find it in the folder named ***Digital Systems Design Lab/Report Template***.
Submit the report of Lab 3 in ***PDF*** to the folder ***Digital Systems Design Lab/Report Submission/Lab 3*** on Blackboard by the deadline below:

- ***LAB02 (Thursday session) 23:59, Saturday, November 1, 2025***
- ***LAB01 (Friday session) 23:59, Sunday, November 2, 2025***

Each day of late submission will result in a 10% deduction from the report's raw marks.

6. Appendix

IC needed for this lab:

1. 74HC04 x1
2. 74HC08 x2
3. 74HC32 x1

Remember to sort and return items 1-4 back to the storeroom after lab.

For any malfunctioning component, report to the instructor or TA, and DO NOT put it back.