EIE2810 Digital Systems Design Laboratory

Laboratory 3

Combinational Logic, Truth Table, Karnaugh Map

School of Science and Engineering

The Chinese University of Hong Kong, Shenzhen

2023-2024 Term 2

1. Objectives

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

- Learn to use Word Generator, Logic Analyzer, Function Generator and Four-Channel Oscilloscope, and wire basic chips in Multisim.
- Practice how to obtain truth table and Karnaugh Map for combinational logic circuit.
- Learn the timing hazard in combinational logic circuit, and the method to eliminate it, by both simulation and hardware implementation.
- Learn to build up a combinational logic circuit for 2-bit multiplied by 2-bit.

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

2.1. Word Generator

Word Generator can provide at most 32 channels of digital outputs. It provides convenience to generate multi-channel synchronized waveforms as inputs, to test if the circuit behaves consistently with the truth table. In the example below, we will add a Word Generator, producing 4 channels of digital outputs, from 0b0000 to 0b1111.

- From the instrument column, add a Word Generator, as below.

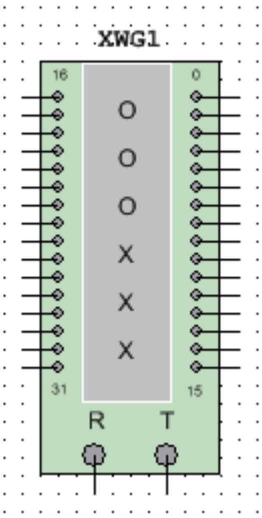


Fig. 1 Word Generator

- Double click it to set the parameters. Choose “Binary”, then click on “Set…”

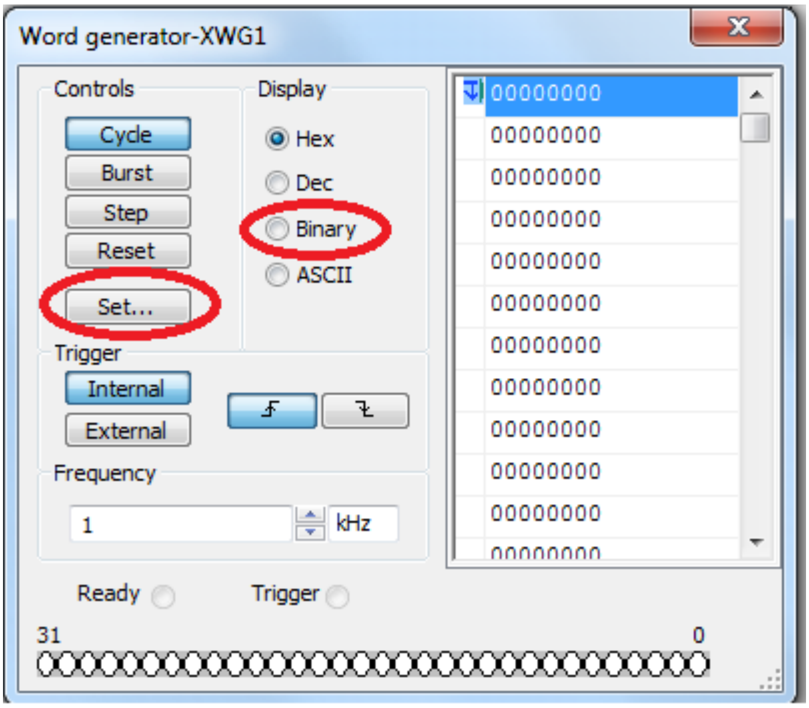


Fig. 2 Word Generator Set Dialogue Box

- Change “Buffer size” from “0400” to “0010”, and select “Up counter”. This will generate a 4-bit output from “0b0000” to “0b1111”. Change 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 the initial low voltage is 500mV, rather than 0V.), as we will use the CMOS6V gates in Multisim. Then click “OK”.

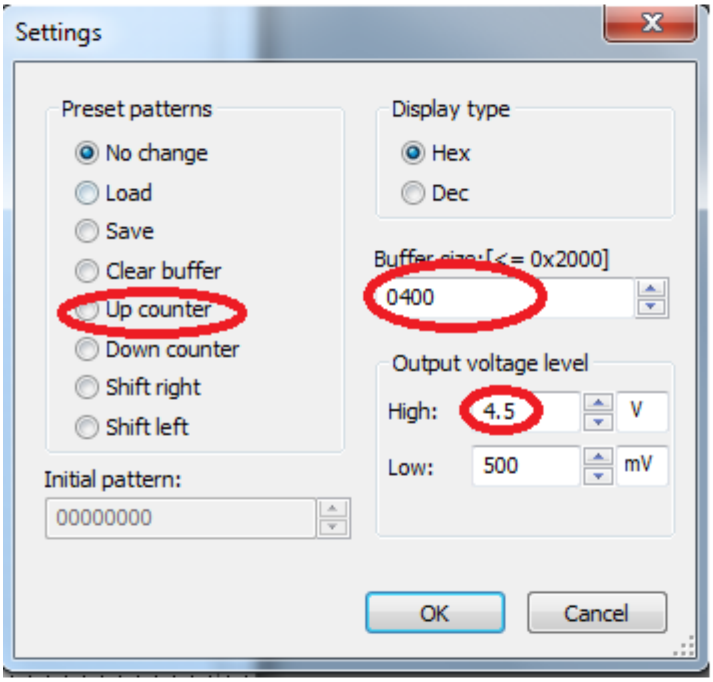
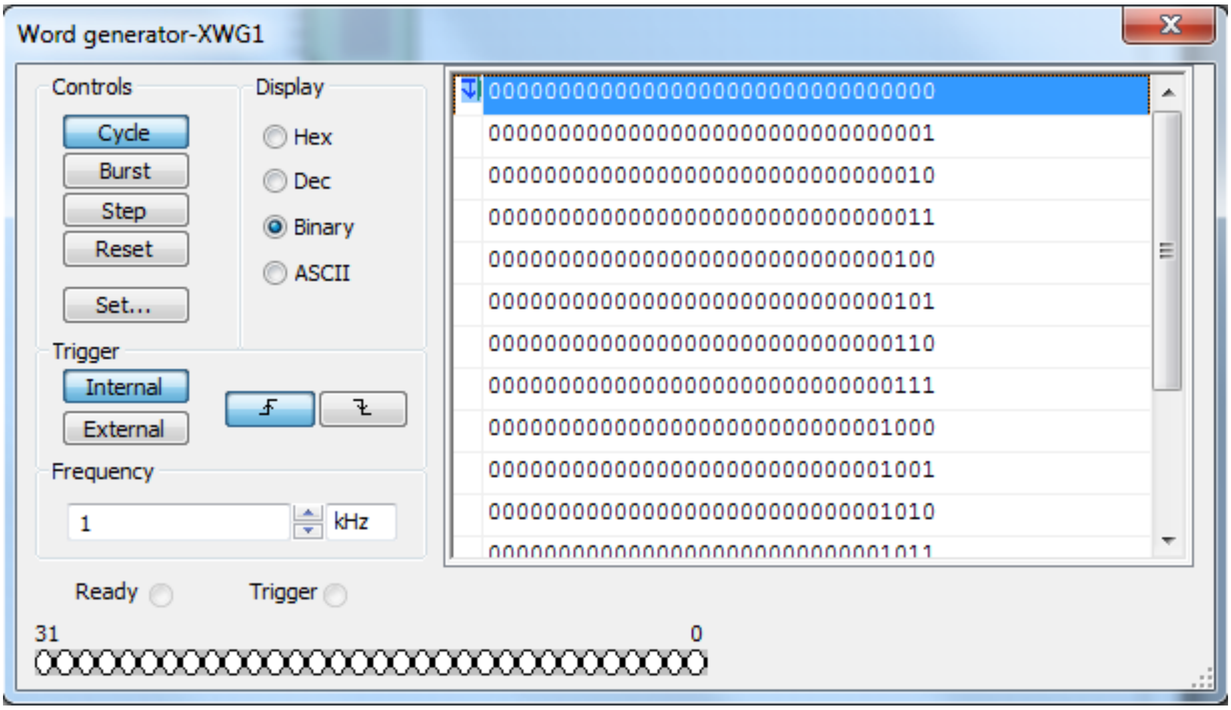


Fig. 3 Setting Panel of Word Generator

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



Word generator-XWG1

Display: ○ Hex ○ Dec ● Binary ○ ASCII

Frequency: 1 kHz

```
00000000000000000000000000000000
00000000000000000000000000000001
00000000000000000000000000000010
00000000000000000000000000000011
00000000000000000000000000000100
00000000000000000000000000000101
00000000000000000000000000000110
00000000000000000000000000000111
00000000000000000000000000001000
00000000000000000000000000001001
00000000000000000000000000001010
00000000000000000000000000001011
```

Fig. 4 Cycle Set in the Word Generator

2.2. Logic Analyzer

Logic Analyzer measures multiple channels of digital signals. It is similar to the real equipment in the lab. Here below, we will try to add the Logic Analyzer and measure the 4 channels of signals from Word Generator. Then construct a simple circuit Y = ABCD, and test its input-output.

- Add a Logic Analyzer from the instrument column, as below.

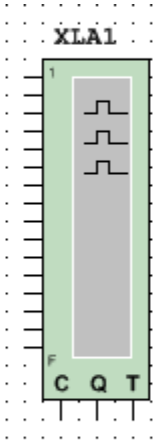


Fig. 5 Logic Analyzer

- Connect the 4 channels of output from Word Generator to 4 inputs of Logic Analyzer. To make it easy to identify the signals, in Work Generator, it is suggested to mapping the LSB, i.e. the pin marked with 0, to the 4th pin in Logic

Analyzer, and MSB in Word Generator to pin 1 in Logic Analyzer. The figure bellow shows the connection.

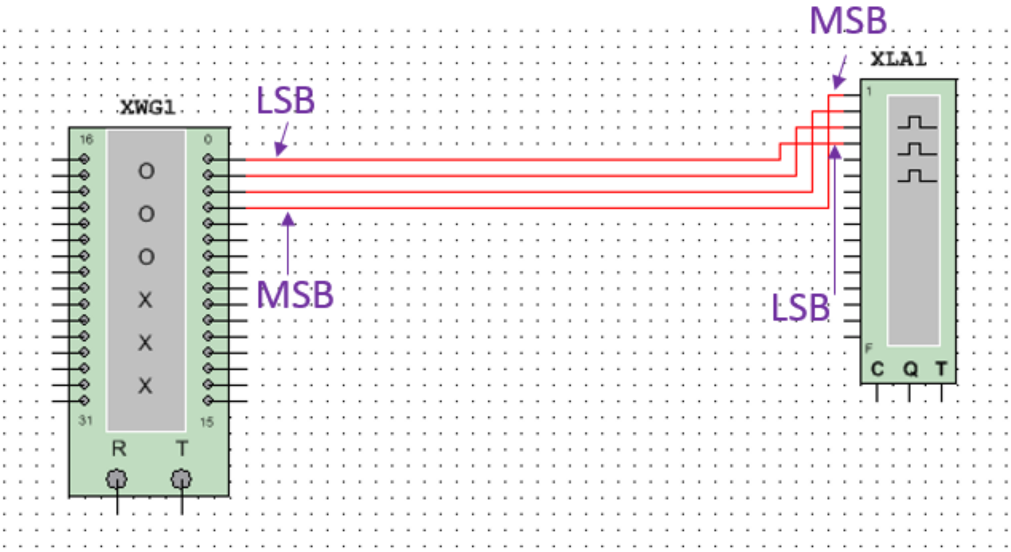
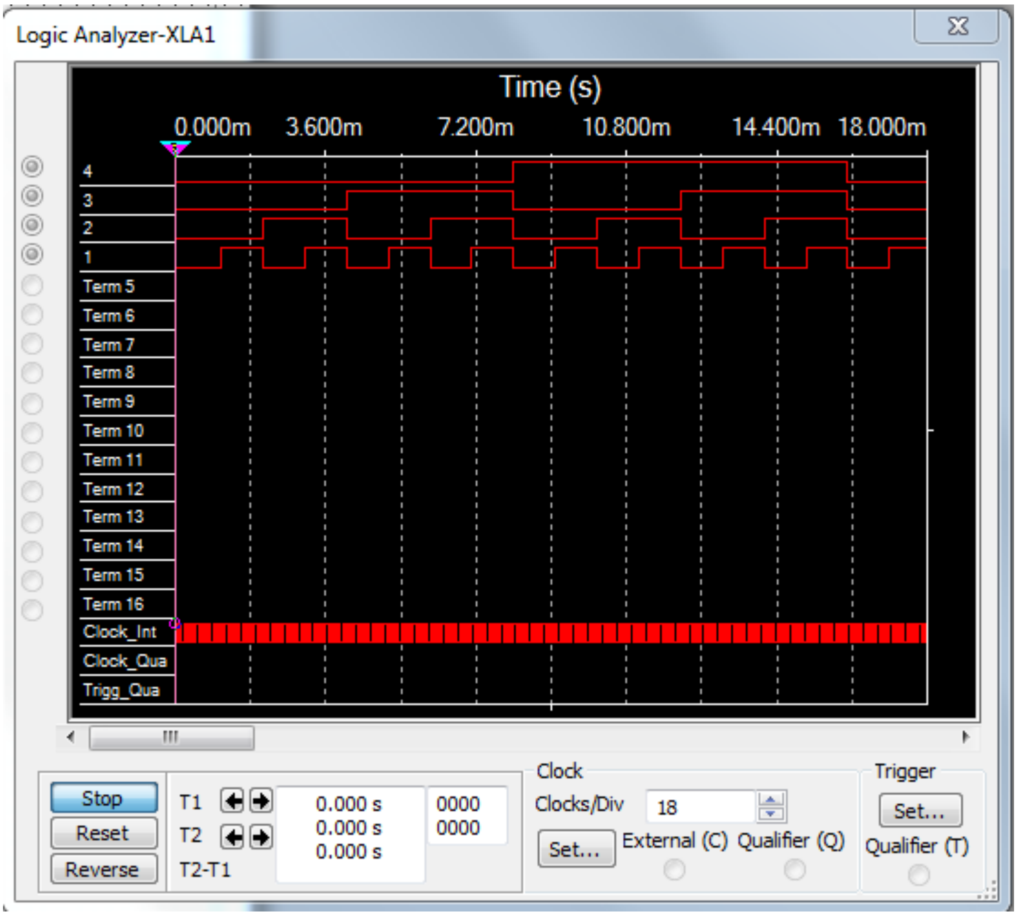


Fig. 6 Wiring 1

- Run the simulation, you can change the “Clocks/Div” to zoom in/out in the time axis and find the waveform as below.



T1 0.000 s 0000
T2 0.000 s 0000
T2-T1 0.000 s

Clocks/Div 18

External (C) ○ Qualifier (Q) ○ Qualifier (T) ○

Fig. 7 Waveform 1

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

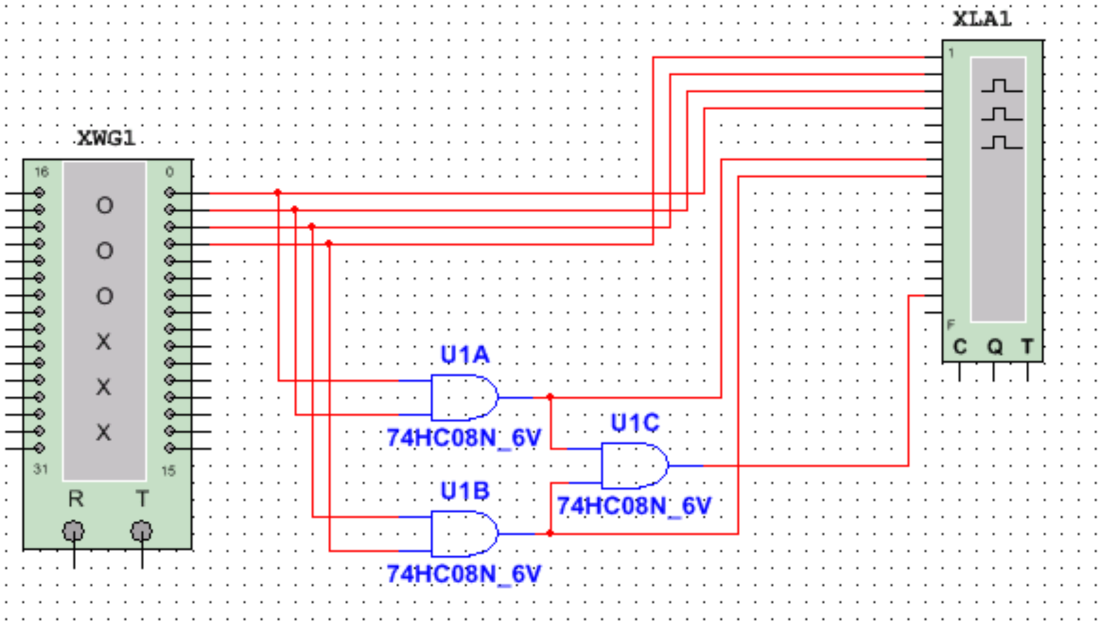
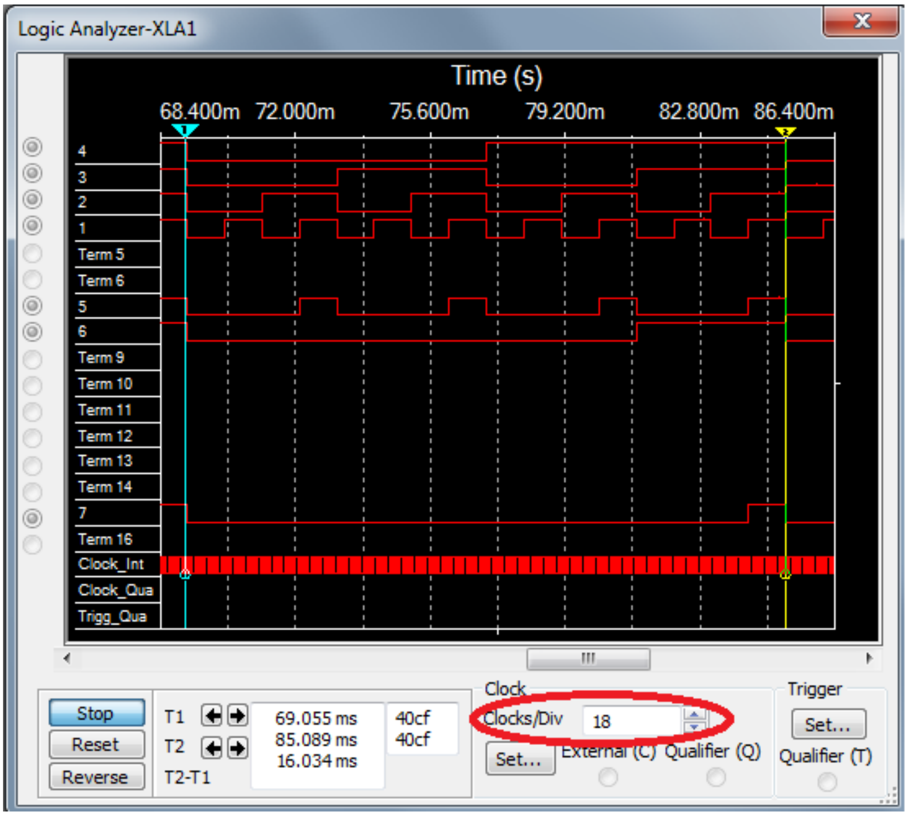


Fig. 8 Wiring 2

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



T1 69.055 ms 40cf
T2 85.089 ms 40cf
T2-T1 16.034 ms

Clocks/Div 18

Fig. 9 Waveform 2

2.3. Function Generator

Function Generator can generate square/triangular/sin waveforms. Here 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 Function Generator from the Instrument column, and add to the circuit, as shown below.

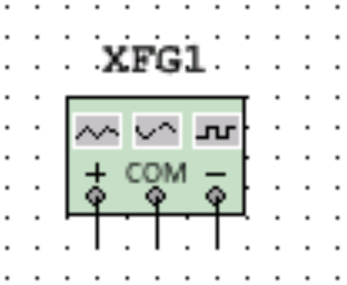
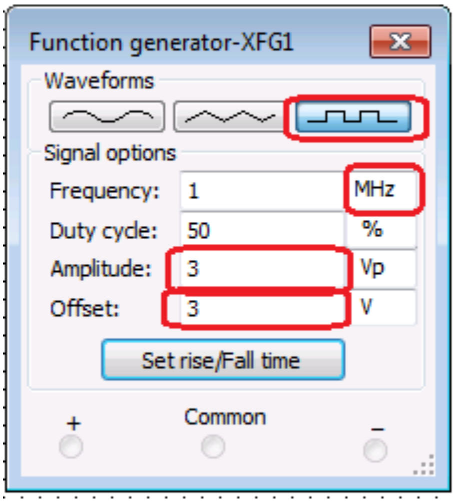


Fig. 10 Function Generator

- Double click the Function Generator, to obtain he dialog box below. Change waveform shape, frequency unit, amplitude and offset. Then, close the dialog box.



Function generator-XFG1

Waveforms

Signal options

Frequency: 1 MHz

Duty cycle: 50 %

Amplitude: 3 Vp

Offset: 3 V

Fig. 11 Parameter Set Panel in Function Generator

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

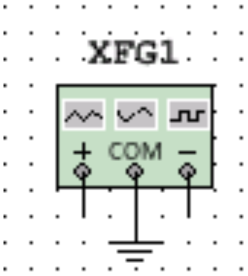


Fig. 12 Wiring 3

2.4. Four Channel Oscilloscope

We will use a Four Channel Oscilloscope in the experiment.

- Select a Four-Channel Oscilloscope from the Instrument column and add it to the circuit.

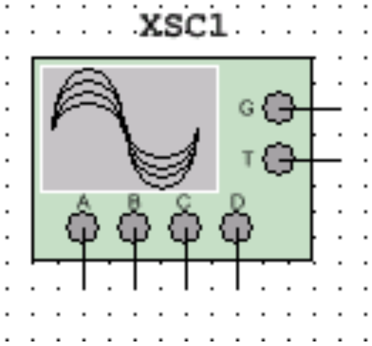
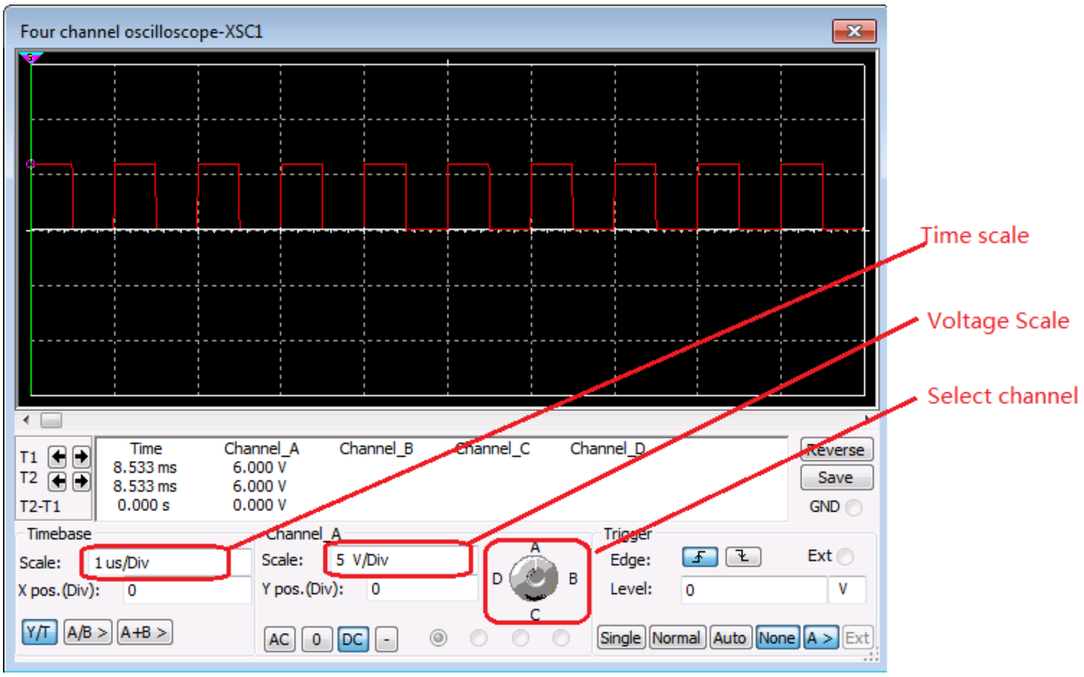


Fig. 13 Four Channel Oscilloscope

- 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 prop out the dialogue box as below. You can select the channel, change time scale and voltage scale to observe the wave.


Four channel oscilloscope-XSC1

	Time	Channel_A	Channel_B	Channel_C	Channel_D
T1	8.533 ms	6.000 V			
T2	8.533 ms	6.000 V			
T2-T1	0.000 s	0.000 V			

Timebase
Scale: 1 us/Div
X pos.(Div): 0

Channel_A
Scale: 5 V/Div
Y pos.(Div): 0

Trigger
Edge:
Level: 0 V

Fig. 14 Panel of Four Channel Oscilloscope

2.5. Chips in Multisim

Aside of adding logic gates into the circuit, you can also add chips into the circuit.

To build up Y=ABCD, select 74LS08N from TTL=>74LS_IC =>74LS08N. The chips of 74HC series have not been included in the database yet. So, we will use 74LS series in the simulation instead.

The figure below shows the wiring.

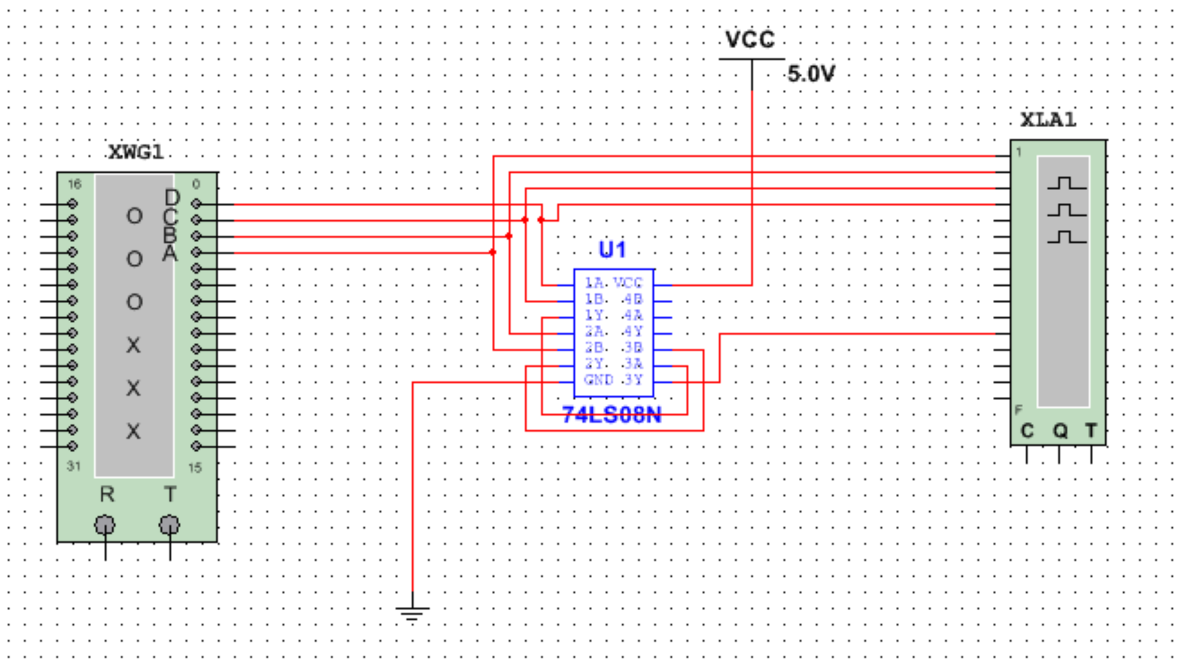
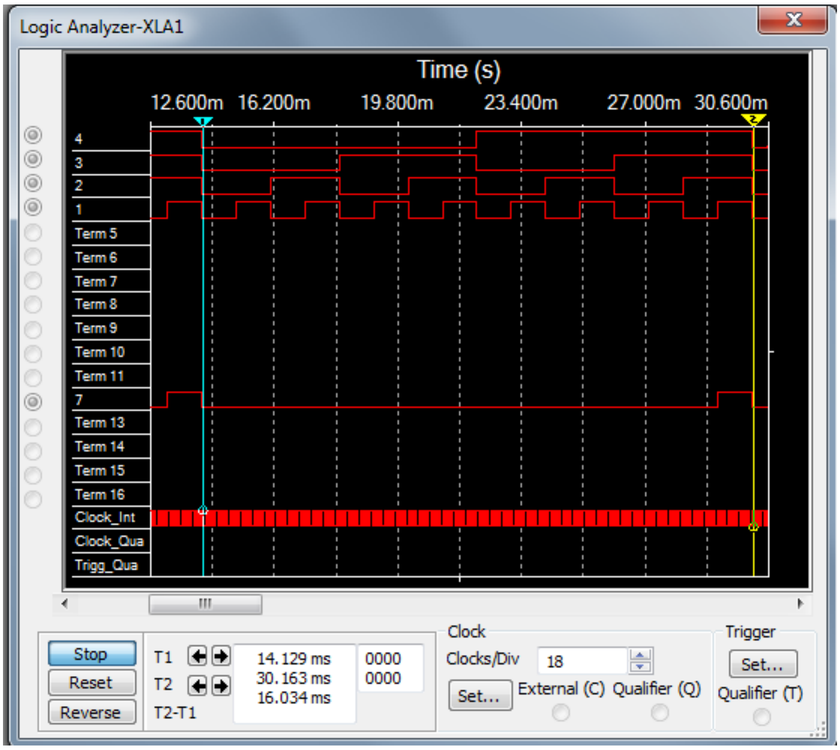


Fig. 15 Wiring 4

After setting up the Word Generator similarly as in the previous part, you can run the simulation and obtain the result as below.



T1 14.129 ms 0000

T2 30.163 ms 0000

T2-T1 16.034 ms

Clocks/Div 18

Fig. 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 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 Karnaugh map, be careful. There might be 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 firstly build up a circuit for the combinational logic in Multisim like below. Use a Function Generator, one NOT, one OR, one AND, and a Four Channel Oscilloscope.

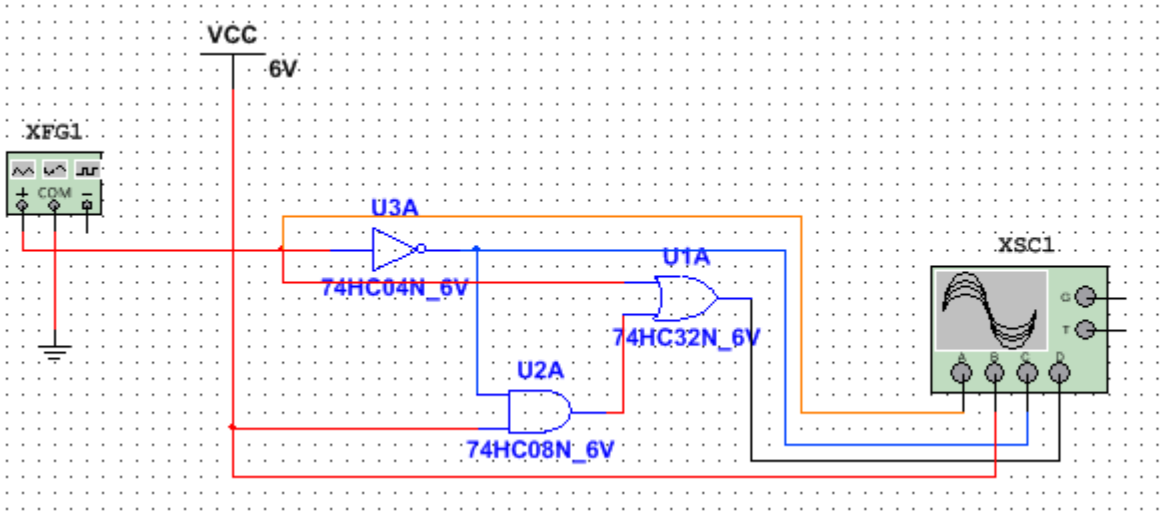
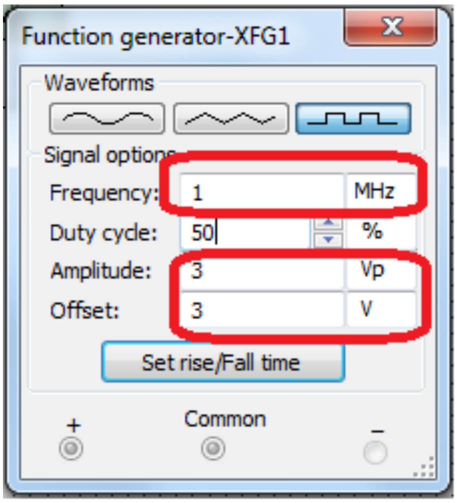


Fig. 17 Wiring 5

Set a 1 MHz square wave, with 50% duty cycle, 6V as high voltage, and 0V as low voltage.



Frequency: 1 MHz

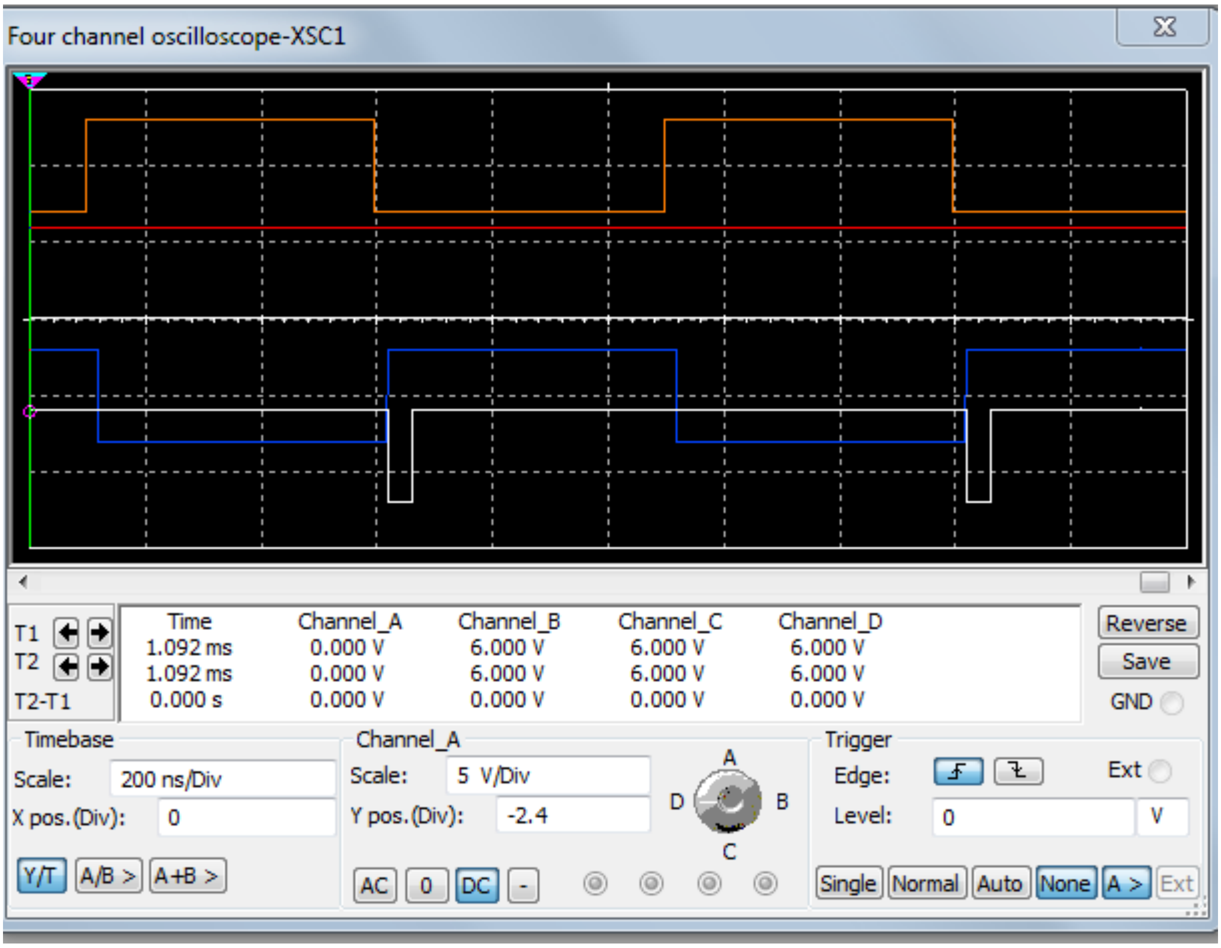
Duty cycle: 50 %

Amplitude: 3 Vp

Offset: 3 V

Fig. 18 Parameter Set Panel in Function Generator

Run the simulation, you will observe the output not constantly 1 (as shown in the white line below).



	Time	Channel_A	Channel_B	Channel_C	Channel_D
T1	1.092 ms	0.000 V	6.000 V	6.000 V	6.000 V
T2	1.092 ms	0.000 V	6.000 V	6.000 V	6.000 V
T2-T1	0.000 s	0.000 V	0.000 V	0.000 V	0.000 V

Timebase Scale: 200 ns/Div

X pos.(Div): 0

Channel_A Scale: 5 V/Div

Y pos.(Div): -2.4

Edge:

Level: 0 V

Fig. 19 Glitch

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

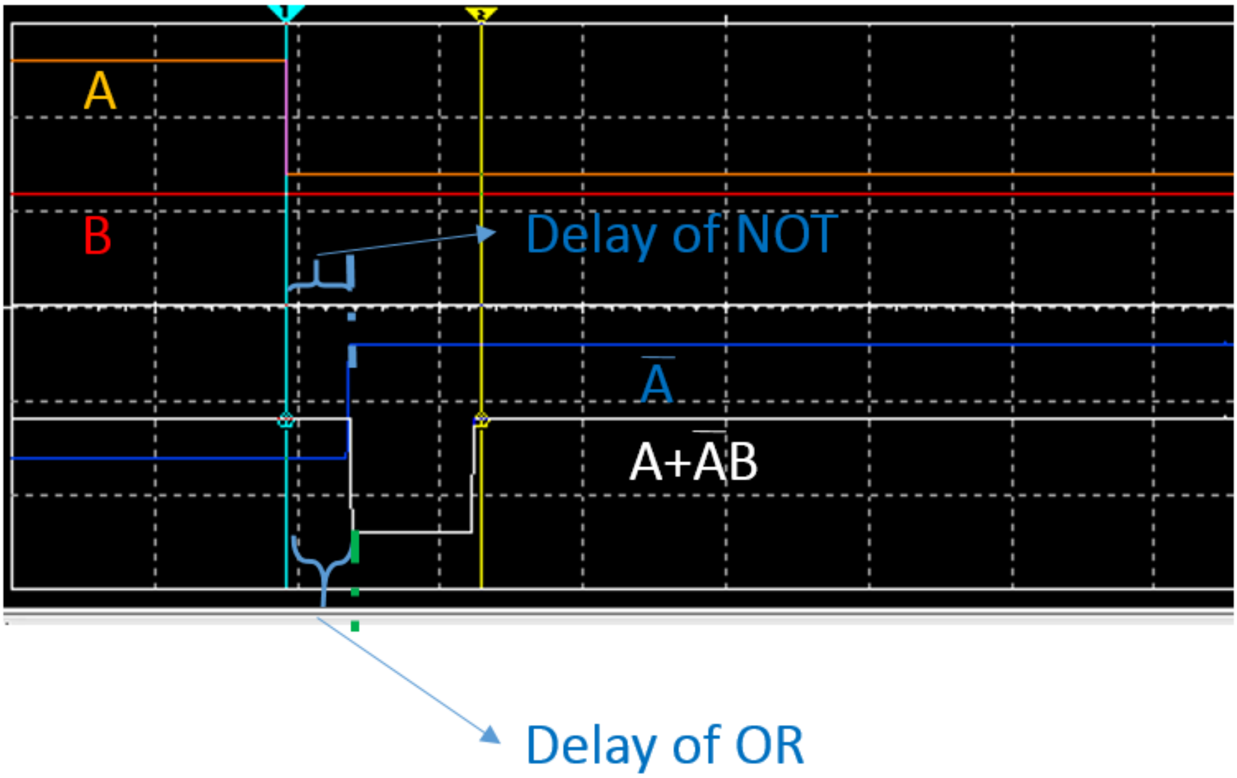
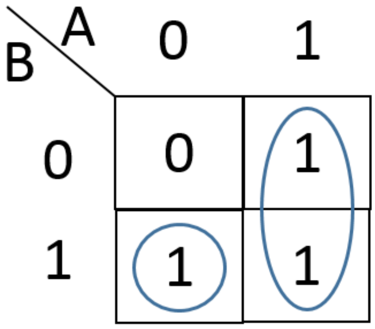


Fig. 20 Delay Analysis

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



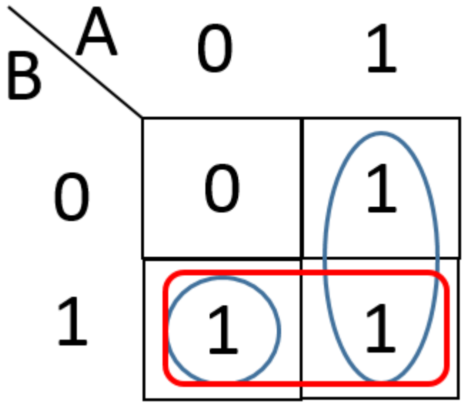
B \ A	0	1
0	0	1
1	1	1

Fig. 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 Karnaugh map, the problem will occur.

To eliminate the timing hazard, one way is to add an additional term, which does not cause logic conflict. In this example, you can add the additional circle, resulting in a term B. It will make the logic expression as $Y = A + \overline{A}B + B$.



B \ A	0	1
0	0	1
1	1	1

Fig. 22 Term Added in Karnaugh Map

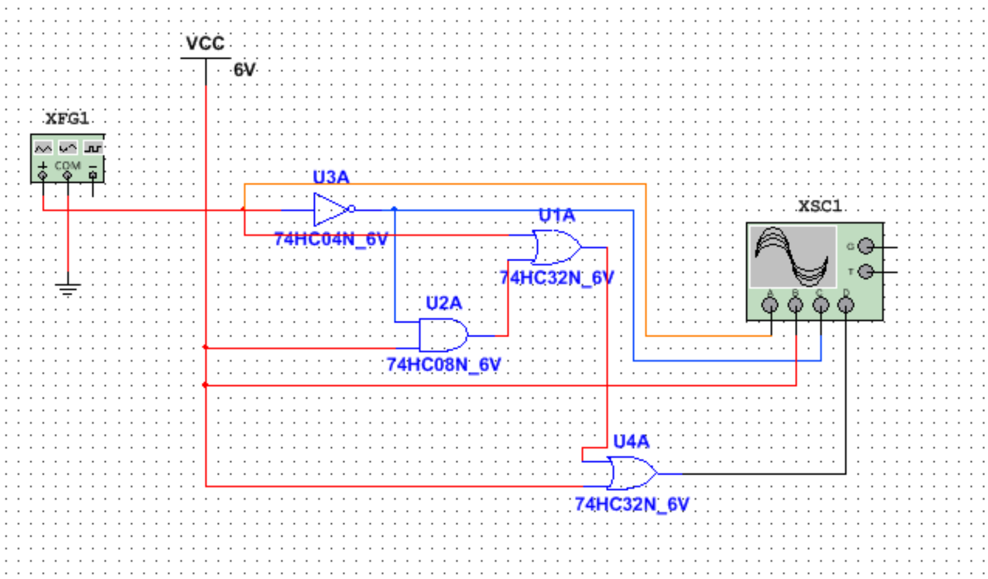


Fig. 23 Term Added in Multisim

Though the delay in NOT still exists, the glitch has been eliminated.

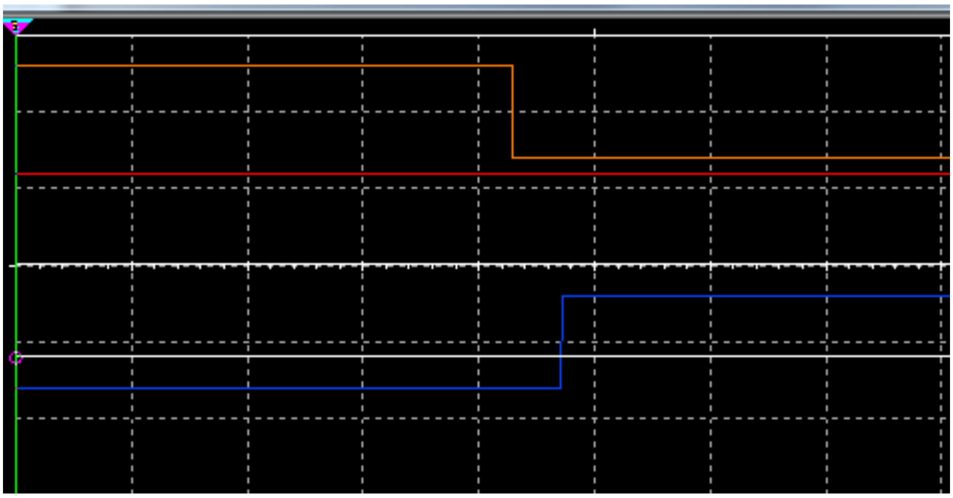


Fig. 24 Glitch Eliminated

4. Experiments

4.1 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 inputs and output.
2. In Multisim, build up the circuit with AND (2-input), OR (2-input), NOT (1-input), and test the result. Save this design file.
3. Start another design in Multisim, wire the chips (74LS04 x1, 74LS08 x2, 74LS32 x1, **NOT logic gate symbols**), and conduct the simulation. Use Word Generator and Logic Analyzer, and record the waveform. Save this design file.
4. Implement the circuit by hardware, including 74HC04 x1, 74HC08 x2, 74HC32 x1. To prevent from the situation that any of your chips malfunctions, it is suggested that you check each chip with SIM and LOM first. Use SIM to generate all 16 inputs, and LOM to indicate the output.
5. After step 4, keep this circuit. You will still need it for the next experiment.

[DEMONSTRATION - 1] When you have completed steps 3 and 4, demonstrate to instructor or TA.

[IN REPORT] Include your truth table, designed circuits (both in gate level, and chip level) in Multisim, and waveform in step 3.

4.2 Timing hazard

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

At this moment, you may still not know when the timing hazard will occur. There are three state-transitions which may have timing hazards. One transition is between 1111(ABCD) and 0111 (ABCD)[1].

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 which group all "1"s, there will be a timing hazard.)
2. Based on the simulation circuit built by gate symbols (not chip-level circuit), use function generator in Multisim to generate a high frequency square wave (e.g. 1 MHz) as input to A. Set B, C and D as high voltage. Try to observe the glitch with oscilloscope in Multisim.
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. Implement the **timing hazard free** circuit by hardware. The chips, which you can use, include 74HC04 x1, 74HC08 x2, and 75HC32 x1.

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

(1) Discovered the glitches in step 2 in Multisim;
(2) Eliminated the timing hazard in Multisim;
(3) Successfully eliminate the timing hazard by hardware.

[IN REPORT] Include all the necessary designs and observations in this experiment.

[QUESTIONS] (1) There are 2 other timing hazards in this circuit. What are they, and how to eliminate them by adding terms? (2) In this experiment, we examine the timing hazard in combinational logic in the format $Y = A + \overline{A}$. There is another format, i.e. $Y = A\overline{A}$, which may also have timing hazard. Try to explain why.

4.3 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 NAND 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 minimize the SOP Boolean expression.
3. Use Multisim to build up the circuit logic symbols (**NOT by chips**), and validate its functionality.
4. Use chips (74HC00 x1, 74HC08 x2, 74HC32 x1), SIM and LOM to implement the circuit by hardware, and check the truth table. First of all, you still need to check that your chips all work well. In this step, you have a lot of cables, be careful in

[1] Whether timing hazard exists from 1111 to 0111, or from 1111 to 0111, depends on which signal combination (A or $\overline{A}$) arrives at the final OR gate faster. The numbers of gates for A and $\overline{A}$ to travel through are different. If A is ahead of $\overline{A}$, timing hazard occurs from when input changes from 1111 to 0111. Otherwise, it is from 0111 to 1111.

wiring. (Hint: you can draw a logic circuit diagram by hand, and mark logic gates with numbers, so that you can identify each gate when you still need its output later.) **These four chips will be enough to complete the design. If you cannot make it, you may ask for more chips, but your grades in this experiment 4.3 will have 30% deduction.**

[DEMONSTRATION - 3] Demonstrate to instructor or TA, when you have:

(1) Realized the circuit in simulation
(2) Realized the circuit by hardware

[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:

- **23:59, Friday, March 22nd, 2024**

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

Appendix:

IC needed for this lab:

1. 74HC00 x1
2. 74HC04 x1
3. 74HC08 x2
4. 74HC32 x1

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

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