Department of Computer Science
School of Computing
UNIVERSITI TEKNOLOGI MALAYSIA

SUBJECT : SECR1013 DIGITAL LOGIC

SESSION/SEM : 2020/2021 - 1

LAB 2 : COMBINATIONAL LOGIC CIRCUIT DESIGN SIMULATION

NAME 1 : Lee Jia Xian A20EC0200

DATE : 10 JAN 2020

Lab # 2
Combinational Digital Circuit Design Simulation Using Deeds Simulator

D. Lab Activities

Part 1 (submitted in pdf file)

Simulating logic circuit, construct truth table and timing diagram with Deeds.

Given Boolean expression as follow:

$$Y=AB+BC+AC$$

1. Convert the non-standard Boolean expression into standard form.

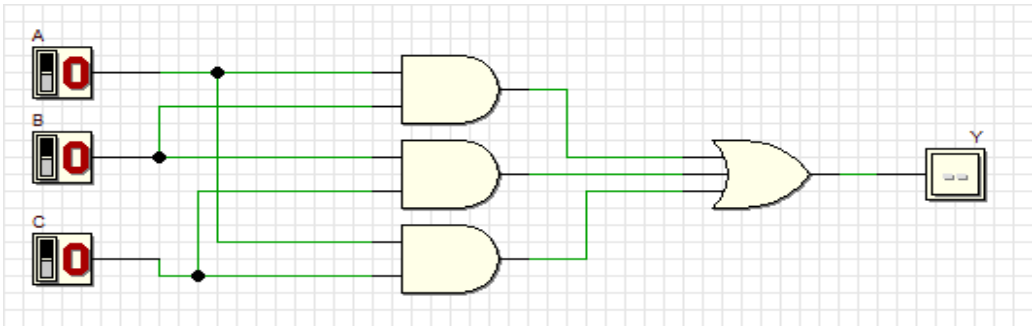
$$\begin{aligned} Y &= AB(C+\bar{C}) + BC(A+\bar{A}) + AC(B+\bar{B}) \\ &= ABC + AB\bar{C} + ABC + \bar{A}BC + ABC + A\bar{B}C \\ &= ABC + AB\bar{C} + \bar{A}BC + A\bar{B}C \end{aligned}$$

2. Based on standard form expression, complete the following truth table.

INPUT			OUTPUT
A	**B**	**C**	**Y**
0	**0**	**0**	**0**
0	**0**	**1**	**0**
0	**1**	**0**	**0**
0	**1**	**1**	**1**
1	**0**	**0**	**0**
1	**0**	**1**	**1**
1	**1**	**0**	**1**
1	**1**	**1**	**1**

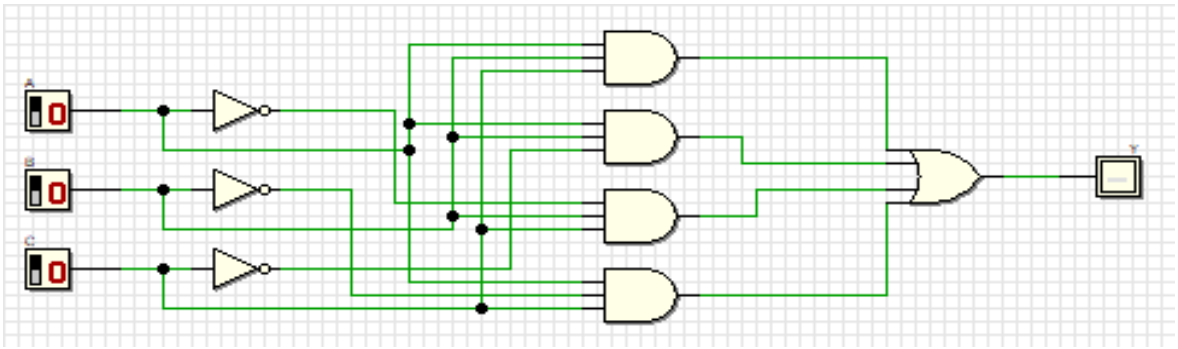
3. Using Deeds Simulator, draw the following circuits and cut & paste your circuit below:

a) Circuit (i) for non-standard form (based on the given expression).



Circuit (i)

b) Circuit (ii) for standard form (from your answer in question (1)).



Circuit (ii)

4. Simulate these two circuits in step (3) and complete their truth table.

(a) Compare the simulation result for these two truth tables. What is your conclusion?

INPUT			OUTPUT
A	**B**	**C**	**Y**
0	**0**	**0**	**0**
0	**0**	**1**	**0**
0	**1**	**0**	**0**
0	**1**	**1**	**1**
1	**0**	**0**	**0**
1	**0**	**1**	**1**
1	**1**	**0**	**1**
1	**1**	**1**	**1**

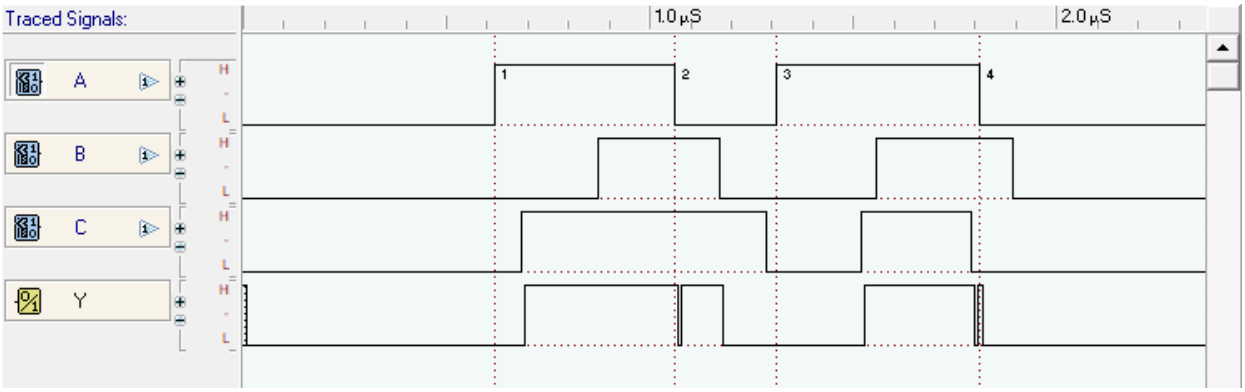
Circuit 1

INPUT			OUTPUT
A	**B**	**C**	**Y**
0	**0**	**0**	**0**
0	**0**	**1**	**0**
0	**1**	**0**	**0**
0	**1**	**1**	**1**
1	**0**	**0**	**0**
1	**0**	**1**	**1**
1	**1**	**0**	**1**
1	**1**	**1**	**1**

Circuit 2

Conclusion:
The truth table for circuit 1 and circuit 2 are the same. So, we can say that circuit 1 is a simplified version of circuit 2. We prefer circuit 1 than circuit 2 because circuit 1 is more effective and can save cost.

5. Simulate output of circuit (ii) with Timing Diagram. Illustrate some examples of different inputs and output. Cut & Paste the timing diagram here.



We can see that when:
A=0, B = 1, C=1, THEN Y = 1
A=1, B=0, C=1, THEN Y = 1
A=1, B=1, C=0, THEN Y = 1
A=1, B=1, C=1, THEN Y = 1

<u>Part 2</u> (video file less than 50MB)

Combinational circuit design process and simulate with Deeds Simulator.

Design Process

i) Determine Parameter Input / Output and their relations.

ii) Construct Truth Table.

iii) Using K-Map, get the SOP optimized form of all Boolean equation outputs.

iv) Draw the circuit and use duality symbol; convert AND-OR circuit to NAND gates ONLY.

v) Simulate the design using Deeds Simulator. Check the results according to Truth Table and Timing Diagram Operation.

Problem Situation

A new digital fault diagnoses circuit is requested to be designed for analysing four bit 2's complement input binary number from sensors A, B, C, and D. Sensor A represents input MSB and sensor D represents input LSB. As shown in the following Figure 5, bit pattern analysis from input sensors A, B, C, and D will trigger four different output errors (active HIGH) of type E1, E2, E3, and E4.

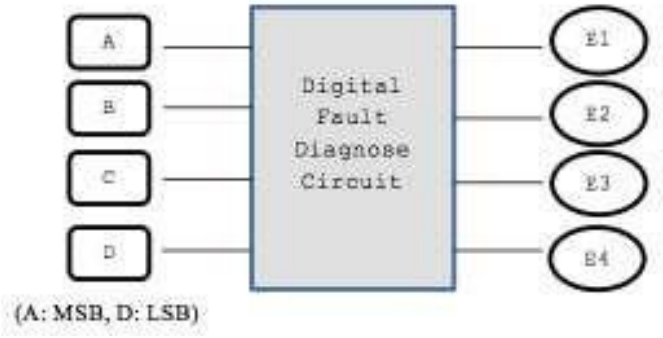


Figure 5

The following rules are used to activate the error's signal type:

RULE 1: E1 is activated if the input number is positive ODD and the majority of the bits is '0'.

RULE 2: E2 is activated if the input number is positive EVEN and the majority of the bits is '0'.

RULE 3: E3 is activated if the input number is negative ODD and the majority of the bits is '1'.

RULE 4: E4 is activated if the input number is negative EVEN and the majority of the bits is '1'.

RULE 5: The output of error signal is invalid if the input has equal bit '0' and bit '1'

(**NOTE:** Positive ODD is positive numbers that are odd and negative.
EVEN is negative numbers that are even).

Experimental Steps

1. Complete Truth Table 1 for Digital Fault Diagnose Circuit. Use variables A, B, C and D as inputs; E1, E2, E3 and E4 as outputs.

Truth Table 1

INPUTS				**OUTPUTS**			
A	**B**	**C**	**D**	**E1**	**E2**	**E3**	**E4**
0	0	0	0	0	1	0	0
0	0	0	1	1	0	0	0
0	0	1	0	0	1	0	0
0	0	1	1	X	X	X	X
0	1	0	0	0	1	0	0
0	1	0	1	X	X	X	X
0	1	1	0	X	X	X	X
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	1	X	X	X	X
1	0	1	0	X	X	X	X
1	0	1	1	0	0	1	0
1	1	0	0	X	X	X	X
1	1	0	1	0	0	1	0
1	1	1	0	0	0	0	1
1	1	1	1	0	0	1	0

2. Using K-MAP, get minimized SOP Boolean expressions for E1, E2, E3 and E4 circuits.

K-map E1:

AB \ CD	00	01	11	10
00		1	X	
01		X		X
11	X			
10		X		X

E1 = $\bar{A}\bar{B}D$

K-map E2:

AB \ CD	00	01	11	10
00	1		X	1
01	1	X		X
11	X			
10		X		X

E2 = $\bar{A}\bar{D}$

K-map E3:

AB \ CD	00	01	11	10
00			X	
01		X		X
11	X	1	1	
10		X	1	X

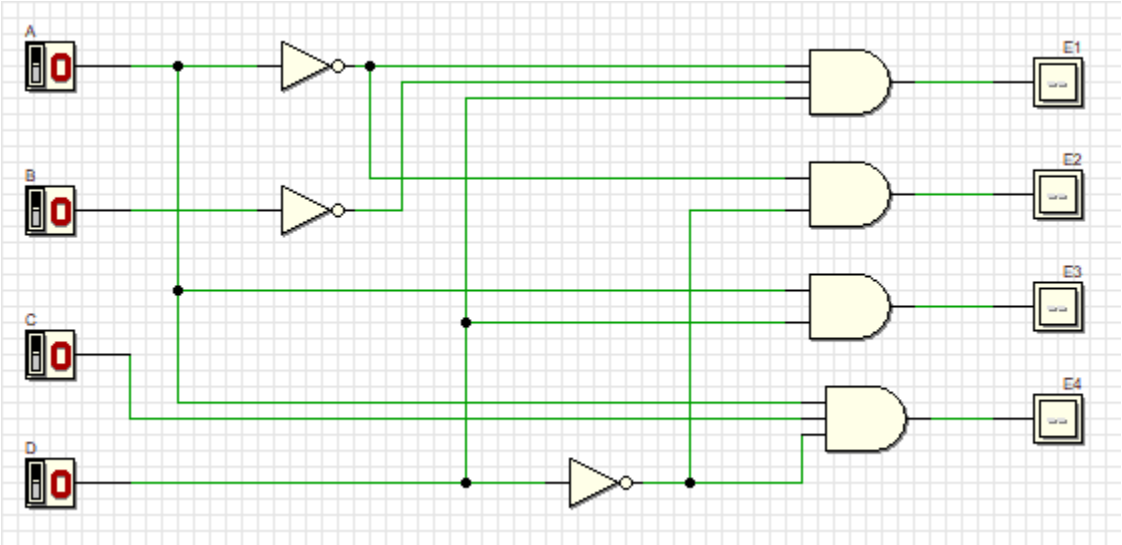
E3 = AD

K-map E4:

AB \ CD	00	01	11	10
00			X	
01		X		X
11	X			1
10		X		X

E4 = $AC\bar{D}$

3. From the Boolean expression in the step (2), draw your final E1, E2, E3 and E4 circuits using 2 input basic gates (AND, OR, NOT). Use Deeds Simulator.



4. Simulate the Deeds circuit in step (3):

a) Update Truth Table 2 based on the simulation result.

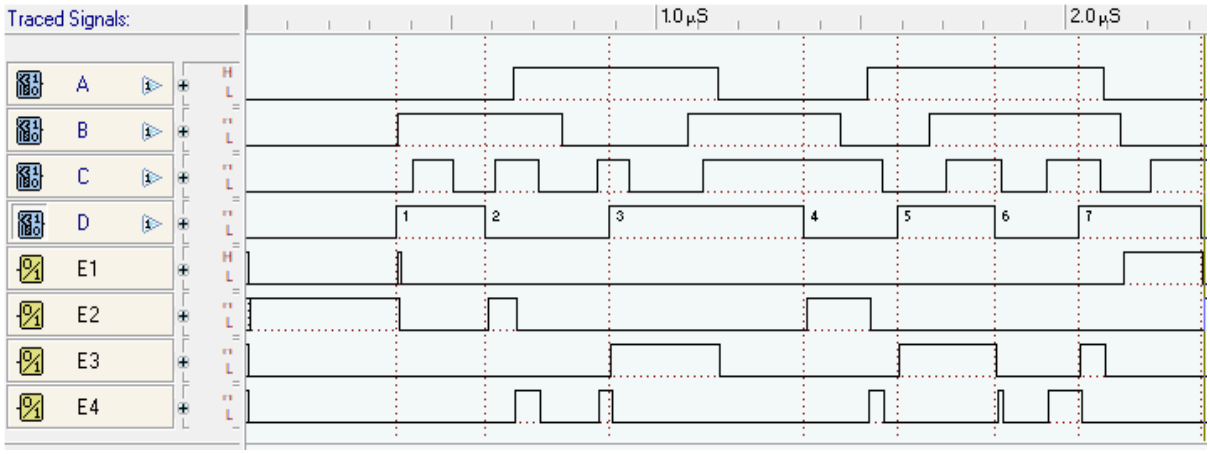
Truth Table 2

INPUTS				OUTPUTS			
A	**B**	**C**	**D**	**E1**	**E2**	**E3**	**E4**
0	0	0	0	0	1	0	0
0	0	0	1	1	0	0	0
0	0	1	0	0	1	0	0
0	0	1	1	1	0	0	0
0	1	0	0	0	1	0	0
0	1	0	1	0	0	0	0
0	1	1	0	0	1	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	0	0	1
1	0	1	1	0	0	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	0
1	1	1	0	0	0	0	1
1	1	1	1	0	0	1	0

b) Compare the output results in Truth Table 2 with Truth Table 1. What is your conclusion?

Since we did not ignore the invalid input during the simulation, we still get the output when the input has equal bit '0' and bit '1'. But still why the output is not 0 for all E1, E2, E3, and E4 when there is an invalid input? This is because we did not ignore the invalid value during the simplification using k-map to get the most minimized SOP Boolean expressions. However, we still can ignore these outputs because the inputs are invalid. Besides that, the truth table 2 is same with the truth table 1 and we can conclude that the circuit we created is a simplified circuit for the problem.

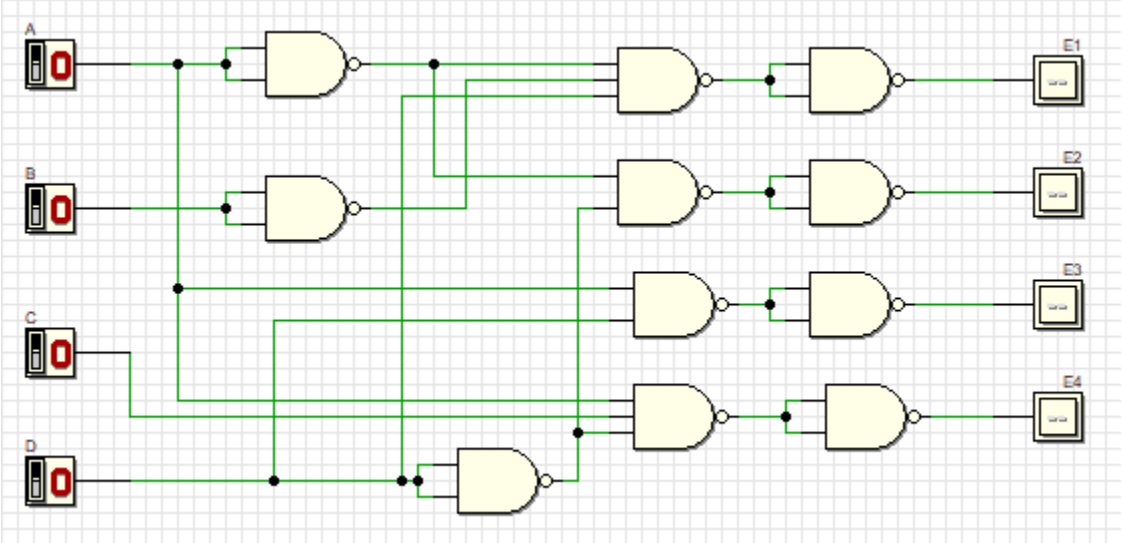
c) Produce the Timing Diagram (you can create your own signals)



d) Explain some analysis values based on your timing diagram:

From the time diagram we can see that:
For E1 = 1, A=0, B=0, D=1
For E2 = 1, A=0, D=0
For E3 = 1, A=1, D=1
For E4 = 1, A=1, C=1, D=0

5. Using dual symbol concept, convert your circuit in step (3) to NAND gates only. Use Deeds Simulator.



6. Simulate the Deeds circuit in step (5):

a) Update Truth Table 3 based on the simulation result.

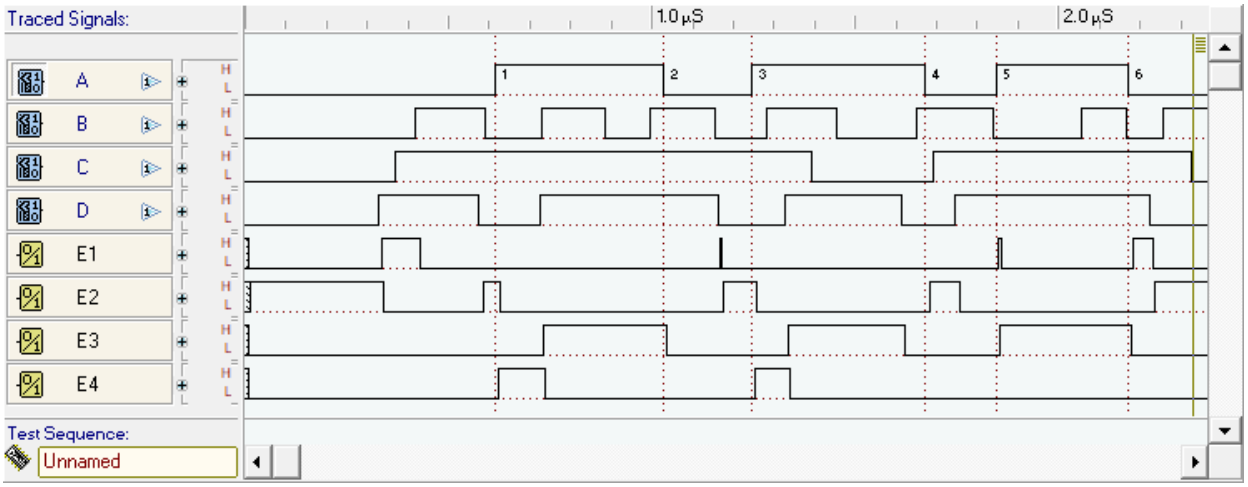
Truth Table 3

INPUTS				OUTPUTS			
A	**B**	**C**	**D**	**E1**	**E2**	**E3**	**E4**
0	0	0	0	0	1	0	0
0	0	0	1	1	0	0	0
0	0	1	0	0	1	0	0
0	0	1	1	1	0	0	0
0	1	0	0	0	1	0	0
0	1	0	1	0	0	0	0
0	1	1	0	0	1	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	0	0	1
1	0	1	1	0	0	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	0
1	1	1	0	0	0	0	1
1	1	1	1	0	0	1	0

b) Compare the output results in Truth Table 3 with Truth Table 2. What is your conclusion?

The Truth Table 3 is totally same with the Truth Table 2. We can see that NAND gate is a universal gate because we can use NAND gate to convert and perform the function of a basic gates(NOT and AND gates in this circuit).

c) Produce the Timing Diagram (you can create your own signals)



d) Explain some analysis values based on your timing diagram:

From the time diagram we can see that:
For E1 = 1, A=0, B=0, D=1
For E2 = 1, A=0, D=0
For E3 = 1, A=1, D=1
For E4 = 1, A=1, C=1, D=0