



ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

ภาคการศึกษาที่ 1.....ปีการศึกษา 2564

รหัสวิชา 010113026 ชื่อวิชา Digital Laboratory

ตอนเรียน 7.....หมายเลขโต๊ะ.....

รหัสนักศึกษา.....6201011631188.....

ชื่อ-นามสกุล.....หนทีลีลาณ.....สังสมบุรณ

อาจารย์ผู้สอน.....CSP.....

เวลาที่ทำการทดลอง 13:00-16:00 วันที่ 29/ ก.ค./ 64

## การทดลองที่ 2

### Combinational and Adder circuit

#### วัตถุประสงค์

1. เพื่อให้สามารถใช้โปรแกรมคอมพิวเตอร์จำลองการทำงานของวงจรลอจิกเกตได้
2. เพื่อให้เข้าใจความสัมพันธ์ของ วงจรลอจิก ลอจิกฟังก์ชัน ตารางความจริง และแผนภาพทางเวลา
3. เพื่อให้เข้าใจพื้นฐานการทำงานของวงจรแบบ Combination-circuit
4. เพื่อให้เข้าใจพื้นฐานการทำงานของวงจร Adder circuit

#### อุปกรณ์

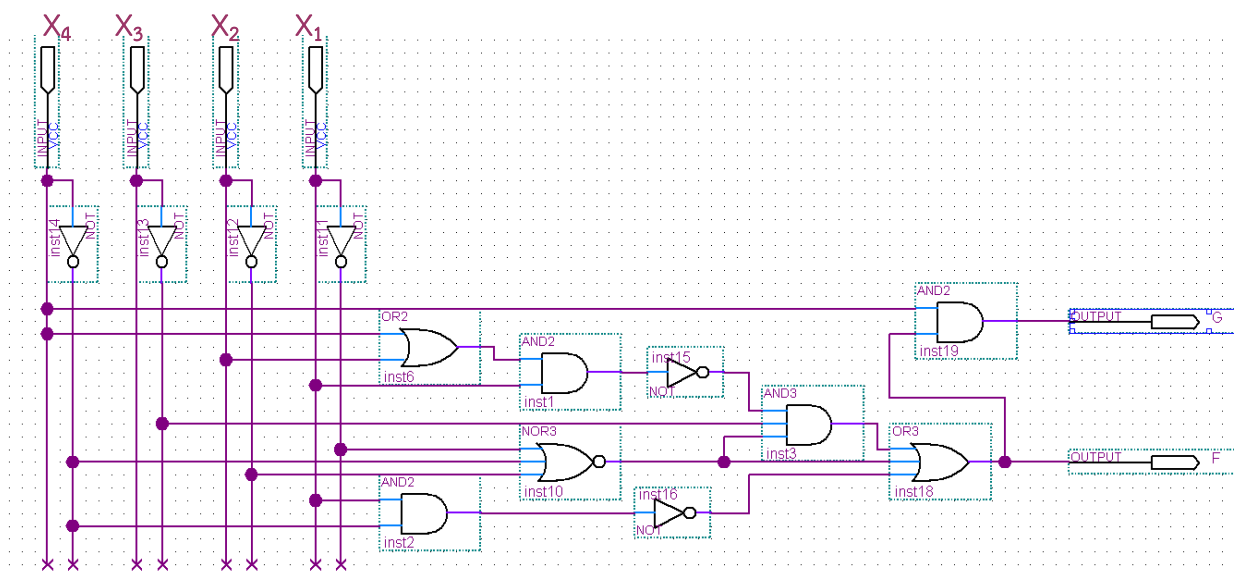
1. ระบบคอมพิวเตอร์ 1 เครื่อง พร้อมติดตั้งโปรแกรม Quartus II เวอร์ชัน 8.0 (Student Edition) ขึ้นไป

### การทดลองตอนที่ 1

วงจร Combinational: Timing, Truth-table and Logic Function

#### คำสั่งการทดลอง

1. ให้ น.ศ. สร้างไฟล์เตอร์สำหรับเก็บงานโปรเจกต์ไฟล์ ขึ้นใหม่เพื่อเก็บงานที่จะทดลองในการทดลองนี้ จากนั้นเขียนวงจรในรูปที่ 1 เพื่อจำลองการทำงานด้วยโปรแกรม Quartus II และให้ใช้ชิพ EP3C10E144C8



รูปที่ 1



2. จากรูปที่ 1 ให้เขียนสมการ ลอจิกฟังก์ชันของเอาต์พุต F และ G

$$F(X_4, X_3, X_2, X_1) = \overline{(X_4 + X_2)} \overline{X_1} \overline{X_3} (\overline{X_4} + \overline{X_2} + \overline{X_1}) + (\overline{X_4} \overline{X_2} + \overline{X_1}) + \overline{X_4} X_1$$

$$G(X_4, X_3, X_2, X_1) = X_4 [(\overline{X_4} + X_2) \overline{X_1} \overline{X_3} (\overline{X_4} + \overline{X_2} + \overline{X_1}) + (\overline{X_4} \overline{X_2} + \overline{X_1}) + \overline{X_4} X_1]$$

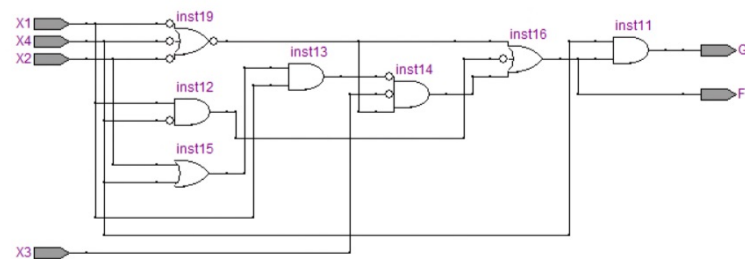
ลดรูปสมการ (ทำ minimization สมการลอจิกฟังก์ชัน) จะได้เป็น

$$F(X_4, X_3, X_2, X_1) = \overline{X_4} X_1 = \overline{X_1} + X_4$$

$$G(X_4, X_3, X_2, X_1) = X_4 (\overline{X_4} X_1) = X_4 \overline{X_1} + X_4 = X_4 (\overline{X_1} + 1) = X_4$$

รูปวงจรลอจิกในระดับ RTL ที่ได้ เมื่อทำการคอมไพล์วงจรในรูปที่ 1 บน Quartus II

(ดูในเมนู Tools >> Netlist Viewer >> RTL Viewer)



สมการลอจิกของวงจร (RTL)

$$F(X_4, X_3, X_2, X_1) = \overline{X_1} + \overline{X_2} + \overline{X_4} + X_1 \overline{X_4} + (X_2 + X_4) X_1 X_3 (\overline{X_1} + \overline{X_2} + \overline{X_4})$$

$$G(X_4, X_3, X_2, X_1) = X_4 [\overline{X_1} + \overline{X_2} + \overline{X_4} + X_1 \overline{X_4} + (X_2 + X_4) X_1 X_3 (\overline{X_1} + \overline{X_2} + \overline{X_4})]$$

ลายเซ็นอาจารย์ผู้ควบคุม..... /...../.....

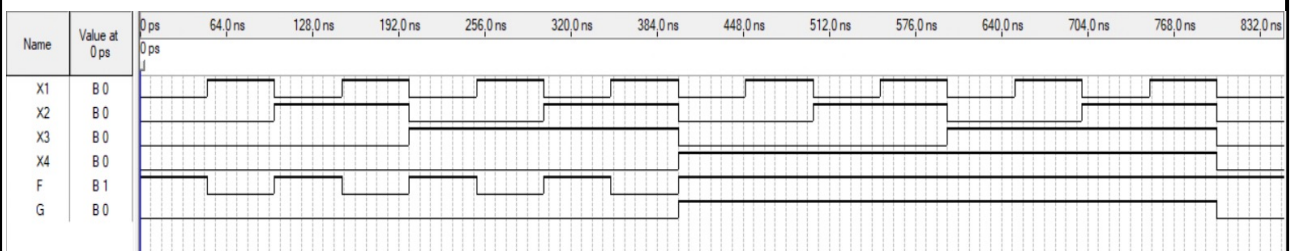
3. สร้างไฟล์แสดงแผนภาพทางเวลา (Timing diagram) ด้วย Vector Waveform Editor Tool ดังรูปที่ 2

- กำหนดสัญญาณอินพุต  $X_4 X_3 X_2 X_1$  ให้มีค่าเริ่มจาก 0000, 0001, ... , 1110, และ 1111 ตามลำดับ
- กำหนดค่าของพารามิเตอร์สำหรับแสดงผลจำลองการทำงานโดยให้มิต่างดังนี้

End Time = 850 ns (ดูที่เมนู Edit >> End Time)

Grid Size = 1 ns (ดูที่เมนู Edit >> Grid size)

3.1 กำหนดโหมดการจำลองเป็นแบบ Functional mode บันทึกผลกราฟของ F และ G ลงในรูปที่ 2



คำแนะนำ ให้กำหนด X1 มีคาบ (Period) = 100 ns

รูปที่ 2



นำค่าสถานะลอจิกของเอาต์พุต F และ G ที่ได้จากกราฟรูปที่ 2 (Functional mode) มาเขียนลงในตารางความจริงและใช้ k-map ลดรูปสมการของลอจิกฟังก์ชัน บันทึกผลการคำนวณที่ได้

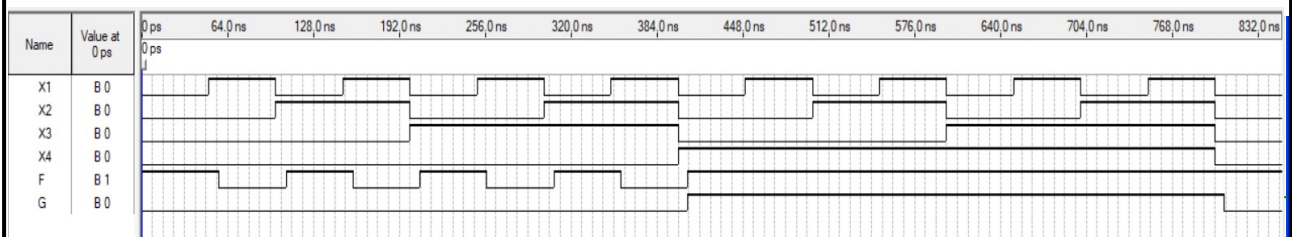
Time	T	Input				Output f
		X <sub>4</sub>	X <sub>3</sub>	X <sub>2</sub>	X <sub>1</sub>	
at 0 ns	T <sub>0</sub>	0	0	0	0	1
at 50 ns	T <sub>1</sub>	0	0	0	1	0
at 100 ns	T <sub>2</sub>	0	0	1	0	1
at 150 ns	T <sub>3</sub>	0	0	1	1	0
at 200 ns	T <sub>4</sub>	0	1	0	0	1
at 250 ns	T <sub>5</sub>	0	1	0	1	0
at 300 ns	T <sub>6</sub>	0	1	1	0	1
at 350 ns	T <sub>7</sub>	0	1	1	1	0
at 400 ns	T <sub>8</sub>	1	0	0	0	1
at 450 ns	T <sub>9</sub>	1	0	0	1	1
at 500 ns	T <sub>10</sub>	1	0	1	0	1
at 550 ns	T <sub>11</sub>	1	0	1	1	1
at 600 ns	T <sub>12</sub>	1	1	0	0	1
at 650 ns	T <sub>13</sub>	1	1	0	1	1
at 700 ns	T <sub>14</sub>	1	1	1	0	1
at 750 ns	T <sub>15</sub>	1	1	1	1	1

Time	T	Input				Output g
		X <sub>4</sub>	X <sub>3</sub>	X <sub>2</sub>	X <sub>1</sub>	
at 0 ns	T <sub>0</sub>	0	0	0	0	0
at 50 ns	T <sub>1</sub>	0	0	0	1	0
at 100 ns	T <sub>2</sub>	0	0	1	0	0
at 150 ns	T <sub>3</sub>	0	0	1	1	0
at 200 ns	T <sub>4</sub>	0	1	0	0	0
at 250 ns	T <sub>5</sub>	0	1	0	1	0
at 300 ns	T <sub>6</sub>	0	1	1	0	0
at 350 ns	T <sub>7</sub>	0	1	1	1	0
at 400 ns	T <sub>8</sub>	1	0	0	0	1
at 450 ns	T <sub>9</sub>	1	0	0	1	1
at 500 ns	T <sub>10</sub>	1	0	1	0	1
at 550 ns	T <sub>11</sub>	1	0	1	1	1
at 600 ns	T <sub>12</sub>	1	1	0	0	1
at 650 ns	T <sub>13</sub>	1	1	0	1	1
at 700 ns	T <sub>14</sub>	1	1	1	0	1
at 750 ns	T <sub>15</sub>	1	1	1	1	1

$$F = \overline{X_1} + X_4$$

$$G = X_4$$

## 3.2 เปลี่ยนโหมดการจำลองเป็นแบบ Timing mode บันทึกผลกราฟของ F และ G ลงในรูปที่ 3



รูปที่ 3



จากกราฟแผนภาพทางเวลาในรูปที่ 3

ก) มีปรากฏการณ์ glitch Hazard ขึ้นที่เวลาใดบ้าง ให้ระบุ

จากรูปที่ 3 ไม่เกิดปรากฏการณ์ glitch Hazard

ข) ค่าของเวลาหน่วงของ F มีค่ามากที่สุดที่เวลา  $t = 4.667$  ns น้อยที่สุดที่เวลา  $t = 4.19$  ns

เมื่อค่าของอินพุตเปลี่ยนสถานะจาก  $X_4X_3X_2X_1 = 0101$  ไปเป็น  $X_4X_3X_2X_1 = 0110$

ค) ค่าของเวลาหน่วงของ G มีค่ามากที่สุดที่เวลา  $t = 6.846$  ns น้อยที่สุดที่เวลา  $t = 6.654$  ns

เมื่อค่าของอินพุตเปลี่ยนสถานะจาก  $X_4X_3X_2X_1 = 1111$  ไปเป็น  $X_4X_3X_2X_1 = 0000$

ง) การเปลี่ยนสถานะของ  $X_4X_3X_2X_1$  ส่งผลอย่างไรต่อค่าเวลาหน่วง (delay time, propagation delay)

ของวงจร หากเป็นกรณีเปลี่ยนสถานะ logic จาก 1  $\rightarrow$  0 จะทำให้ค่า Delay time มากกว่ากรณีเปลี่ยนจาก 0  $\rightarrow$  1 เนื่องจากข้อผิดพลาดของกระบวนการในลักษณะ

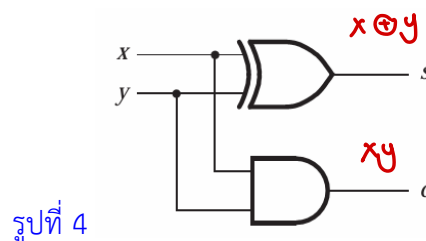
จ) การเกิดค่าเวลาหน่วง (delay time, propagation delay) ของอุปกรณ์แต่ละตัวในวงจรจะส่งผลอย่างไร

ต่อการเกิดปรากฏการณ์ glitch Hazard กรณีเกิด glitch Hazard เป็นผลมาจากค่า Delay time ของ Input แต่ละตัวมีค่าไม่เท่ากัน ส่งผลให้ค่าของ Output ที่เกิดจาก Input นั้นเมื่อเปลี่ยนแปลง จึงเกิด glitch ใน Timing diagram

## การทดลองตอนที่ 2

วงจร Adder: วงจร, ลอจิกฟังก์ชัน, ผลกระทบของเวลาหน่วง

- ให้เขียนวงจรเพื่อจำลองการทำงาน Half Adder ในรูปที่ 4 ด้วยโปรแกรม Quartus II ให้ใช้ชิพเบอร์ EP3C10E144C8 (แนะนำให้สร้างไฟล์เดสคอปใหม่ ตั้งชื่อโปรเจกต์และชื่อของไฟล์ชิ้นงานขึ้นใหม่ทั้งหมด)



- สร้างไฟล์แสดงแผนภาพทางเวลา (Timing diagram) ด้วย Vector Waveform Editor Tool ดังรูปที่ 5 โดย X มีค่าของ period = 100 ns

6.1 กำหนดค่าของพารามิเตอร์สำหรับแสดงผลจำลองการทำงานโดยให้มีค่าดังนี้

End Time = 250 ns (เมนู Edit >> End Time)

Grid Size = 1 ns (เมนู Edit >> Grid size)



6.2 กำหนดโหมดการจำลองเป็นแบบ Functional mode บันทึกผลการจำลองที่ได้ลงในรูปที่ 5

6.3 เปลี่ยนโหมดการจำลองเป็นแบบ Timing mode บันทึกผลการจำลองที่ได้ลงในรูปที่ 6

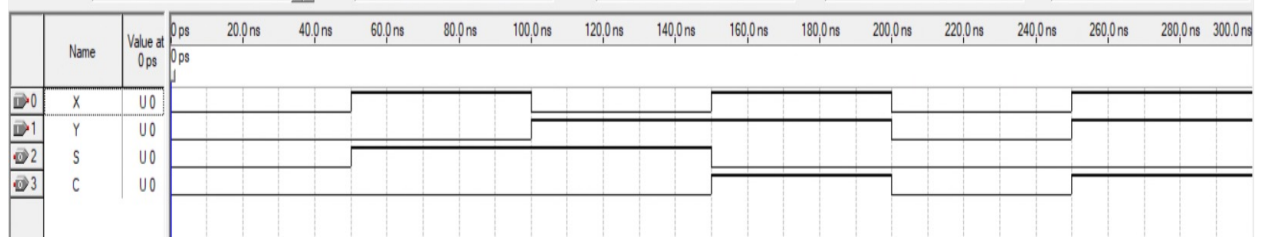
## ผลการทดลอง

จากรูปที่ 5 ให้เขียนสมการลอจิกฟังก์ชันของเอาต์พุต s และ c

$$s(x,y) = \text{---} x \oplus y \text{---}$$

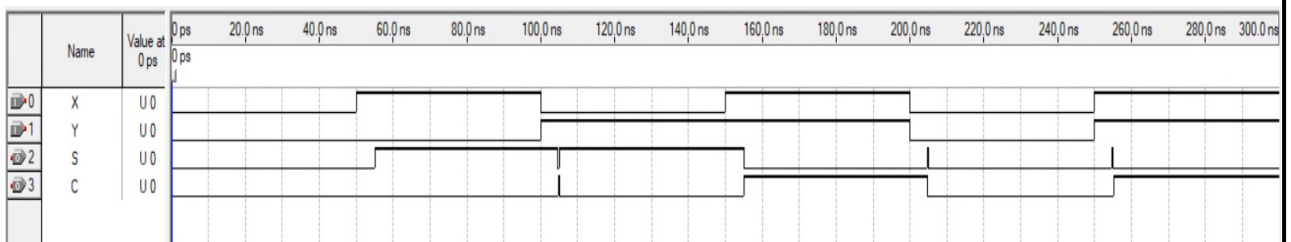
$$c(x,y) = \text{---} xy \text{---}$$

a) ผลกราฟ S และ C จากการจำลองแบบ Functional mode



รูปที่ 5

b) ผลกราฟ S และ C จากการจำลองแบบ Timing mode



รูปที่ 6

- c) จากรูปที่ 6 สังเกตผลลัพธ์ ทั้ง S และ C [หมายเหตุ:  $x = 0 \rightarrow 1$  อ่านว่า x เปลี่ยนค่าจาก '0' ไปเป็น '1']
- ที่เวลา  $t = 50 \text{ ns}$  ค่าของ  $x = 0 \rightarrow 1$  และ  $y = 0 \rightarrow 0$  ผลบวกที่ถูกต้องจะเกิดขึ้นช้ากว่าอินพุต = 5.094 ns
  - ที่เวลา  $t = 100 \text{ ns}$  ค่าของ  $x = 1 \rightarrow 0$  และ  $y = 0 \rightarrow 1$  ผลบวกที่ถูกต้องจะเกิดขึ้นช้ากว่าอินพุต = ..... ns
  - ที่เวลา  $t = 150 \text{ ns}$  ค่าของ  $x = 0 \rightarrow 1$  และ  $y = 1 \rightarrow 1$  ผลบวกที่ถูกต้องจะเกิดขึ้นช้ากว่าอินพุต = 5.108 ns
  - ที่เวลา  $t = 200 \text{ ns}$  ค่าของ  $x = 1 \rightarrow 0$  และ  $y = 1 \rightarrow 0$  ผลบวกที่ถูกต้องจะเกิดขึ้นช้ากว่าอินพุต = 4.844 ns
  - ที่เวลา  $t = 250 \text{ ns}$  ค่าของ  $x = 0 \rightarrow 1$  และ  $y = 0 \rightarrow 1$  ผลบวกที่ถูกต้องจะเกิดขึ้นช้ากว่าอินพุต = 5.108 ns

- d) ให้พิจารณาค่าของเอาต์พุตเมื่อ กรณีที่อินพุต (X,Y) เปลี่ยนแปลงครั้งละหนึ่งสัญญาณ กับกรณีที่เปลี่ยนแปลงพร้อมกันทั้งสองสัญญาณ ทั้งสองกรณิดังกล่าวส่งผลต่อเวลาการเกิดผลลัพธ์ที่ถูกต้องของเอาต์พุตอย่างไร
- กรณีที่ Input เปลี่ยนครั้งละหนึ่งสัญญาณ จะไม่ทำให้ Output เกิด glitch Hazard
- แต่ในกรณีที่ Input เปลี่ยนแปลงพร้อมกันทั้งคู่ จะเกิด glitch hazard ขึ้นที่ตัว Output ทั้งสองตัว

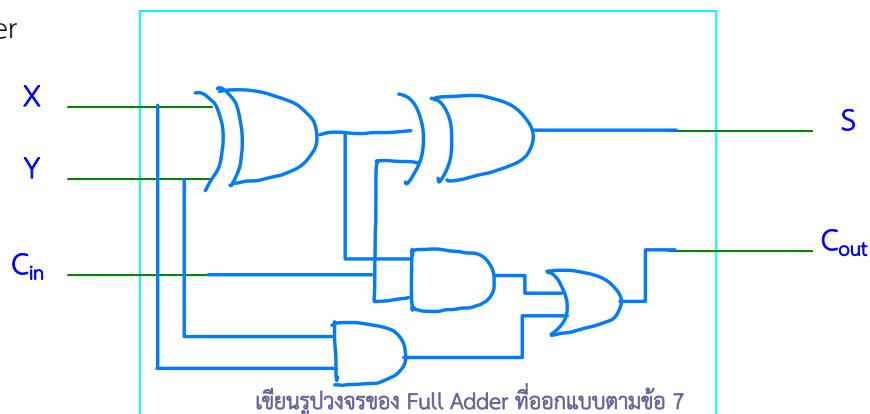
ลายเซ็นอาจารย์ผู้ควบคุม..... /...../.....



7. ให้สร้าง (ออกแบบ) วงจร full Adder จากวงจร Half Adder และเขียนวงจรที่สมบูรณ์ลงในรูปที่ 7  
(แนะนำให้สร้างไฟล์เดสก์ท็อปใหม่ ตั้งชื่อโปรเจกต์และชื่อของไฟล์ชิ้นงานขึ้นใหม่ทั้งหมด)
8. ให้จำลองการทำงาน Full Adder ด้วย Quartus II และให้ใช้ชิพเบอร์ EP3C10E144C8 โดยที่กำหนดค่าของ  $C_{in}$ , X, Y ดังรูปที่ 8 และค่าของ End Time = 250 ns, Grid Size = 1 ns
  - 8.1 ทำการจำลองการทำงานในโหมด Functional Mode บันทึกผลการจำลองที่ได้ลงในรูปที่ 8
  - 8.2 ทำการจำลองการทำงานในโหมด Timing Mode บันทึกผลการจำลองที่ได้ลงในรูปที่ 9

### ผลการทดลอง

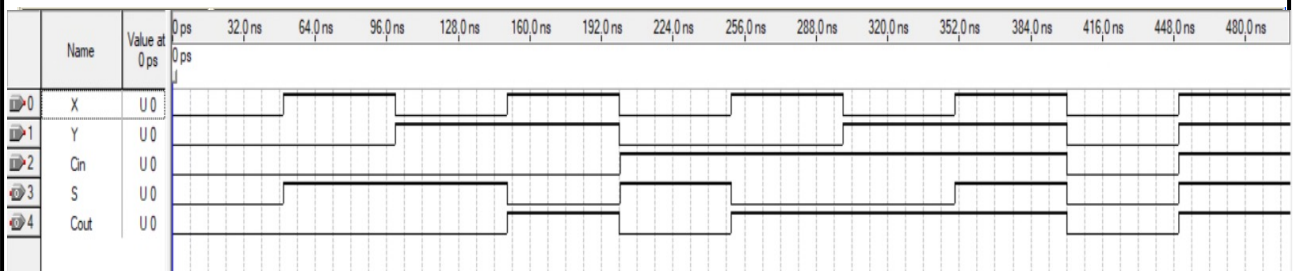
#### a) วงจร Full Adder



เขียนรูปวงจรของ Full Adder ที่ออกแบบตามข้อ 7

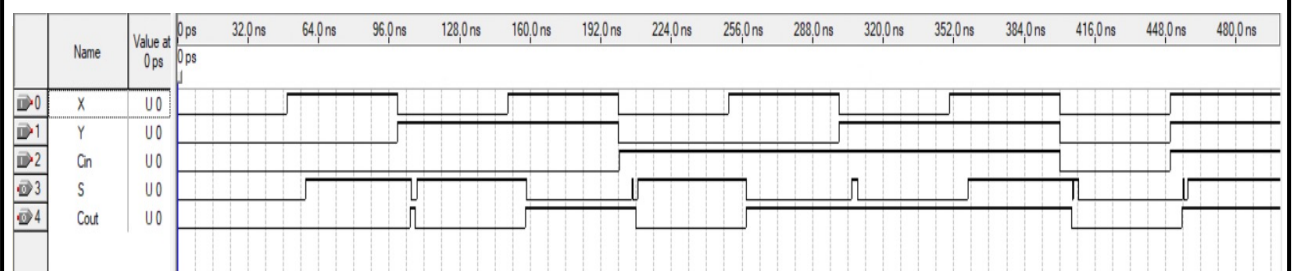
รูปที่ 7

#### b) ผลกราฟ S และ Cout จากการจำลองการทำงานในโหมด Functional Mode



รูปที่ 8

#### c) ผลกราฟ S และ Cout จากการจำลองการทำงานในโหมด Timing Mode



รูปที่ 9





d) จากรูปที่ 9 สังเกตผลลัพธ์ที่เอาต์พุตทั้ง S และ Cout เปรียบเทียบกับการเปลี่ยนสถานะของอินพุต

ที่ t = 50 ns ค่า x = 0 → 1 , y = 0 → 0 , Cin = 0 → 0	ผลบวกที่ต้องเกิดขึ้นซ้ำ = 8.341 ns
ที่ t = 100 ns ค่า x = 1 → 0 , y = 0 → 1 , Cin = 0 → 0	ผลบวกที่ต้องเกิดขึ้นซ้ำ = ..... ns
ที่ t = 150 ns ค่า x = 0 → 1 , y = 1 → 1 , Cin = 0 → 0	ผลบวกที่ต้องเกิดขึ้นซ้ำ = 8.141 ns
ที่ t = 200 ns ค่า x = 1 → 0 , y = 1 → 0 , Cin = 0 → 1	ผลบวกที่ต้องเกิดขึ้นซ้ำ = 8.6 ns
ที่ t = 250 ns ค่า x = 0 → 1 , y = 0 → 0 , Cin = 1 → 1	ผลบวกที่ต้องเกิดขึ้นซ้ำ = 8.141 ns
ที่ t = 300 ns ค่า x = 1 → 0 , y = 0 → 1 , Cin = 1 → 1	ผลบวกที่ต้องเกิดขึ้นซ้ำ = ..... ns
ที่ t = 350 ns ค่า x = 0 → 1 , y = 1 → 1 , Cin = 1 → 1	ผลบวกที่ต้องเกิดขึ้นซ้ำ = 8.341 ns
ที่ t = 400 ns ค่า x = 1 → 0 , y = 1 → 0 , Cin = 1 → 0	ผลบวกที่ต้องเกิดขึ้นซ้ำ = 8.4 ns
ที่ t = 450 ns ค่า x = 0 → 1 , y = 0 → 1 , Cin = 0 → 1	ผลบวกที่ต้องเกิดขึ้นซ้ำ = 8.341 ns

ให้พิจารณาค่าของเอาต์พุต ในกรณีที่อินพุต (X,Y) เปลี่ยนแปลงครั้งละหนึ่งสัญญาณ เปลี่ยนแปลงพร้อมกันสองสัญญาณ เปลี่ยนแปลงพร้อมกันสามสัญญาณ ทั้งสามกรณีดังกล่าวส่งผลต่อเวลาการเกิดผลลัพธ์ที่ต้องของเอาต์พุตอย่างไร

เปลี่ยนแปลง 1 ครั้ง ไม่เกิด glitch และ Delay time

เปลี่ยนแปลง 2 ครั้ง เกิด glitch

เปลี่ยนแปลง 3 ครั้ง เกิด glitch และ Delay time

ผลการทดลองจากข้อ d การบวกเลขในกรณีที่  $X + Y + Cin$  มีรูปแบบการบวกในกรณีใด ที่ทำให้ได้คำตอบที่ถูกต้องช้าที่สุด

กรณีที่  $x = 1 \rightarrow 0$  ,  $y = 1 \rightarrow 0$  ,  $Cin = 0 \rightarrow 1$

ที่ t = 200 ns

ลายเซ็นอาจารย์ผู้ควบคุม..... /...../.....

## การทดลองตอนที่ 3 วงจร Adder: VHDL

- ให้เขียนภาษา VHDL ดังต่อไปนี้ดังต่อไปนี้เพื่อจำลองการทำงานของโปรแกรมบน Quartus II
  - ให้สร้างโปรเจกต์ขึ้นใหม่ ชื่อ FullAdder (เก็บงาน Designed ของเราไว้ในโฟลเดอร์เดิมก็ได้)
  - ให้สร้างไฟล์ชิ้นงาน (Designed file) ชื่อ FullAdder.VHD ขึ้นมาและเขียนภาษา VHDL ดังรูปที่ 10

```
LIBRARY ieee;
USE      ieee.std_logic_1164.all ;

ENTITY FullAdder IS
PORT    ( Cin , X , Y      : IN  std_logic ;
          Cout , S         : OUT std_logic );
END      FullAdder ;

ARCHITECTURE rtl OF FullAdder IS
BEGIN
    S    <= X XOR Y XOR Cin ;
    Cout <= (X AND Y) OR (Cin AND X) OR (Cin AND Y) ;
END rtl ;
```

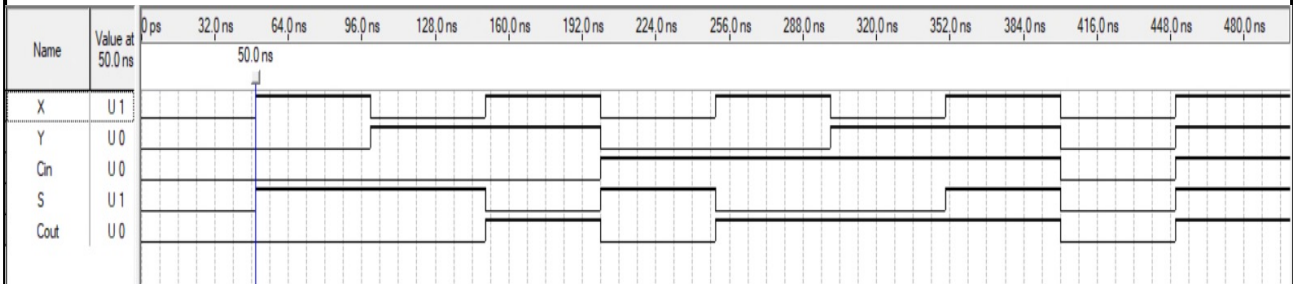
รูปที่ 10

ทำการคอมไพล์ และจำลองการทำงานในโหมด Functional Mode กำหนดสัญญาณ Cin , X และ Y ดังในรูปที่ 11 บันทึกผลการจำลองที่ได้ จากนั้นเปลี่ยนโหมดจำลองเป็น Timing Mode บันทึกผลลงในรูปที่ 12



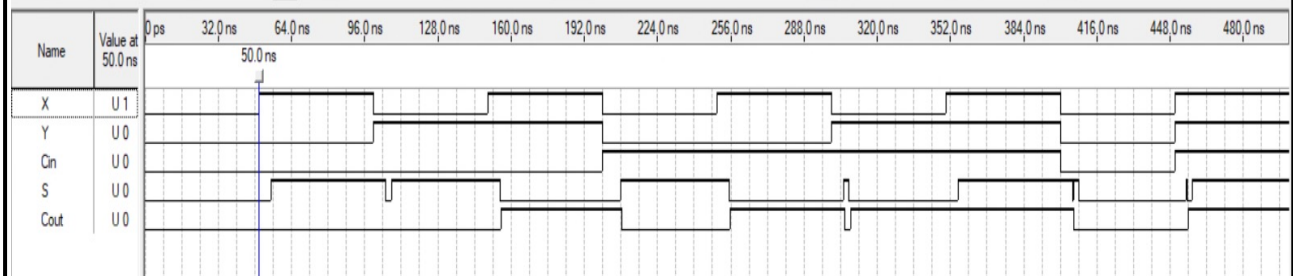
## ผลการทดลอง

a) ผลกราฟ S และ Cout จากการจำลองการทำงานในโหมด Functional Mode



รูปที่ 11

b) ผลกราฟ S และ Cout จากการจำลองการทำงานในโหมด Timing Mode



รูปที่ 12

c) จากรูปที่ 12 สังเกตผลลัพธ์ ทั้ง S และ Cout

ที่ t = 50 ns ค่าของ x = 0 → 1 , y = 0 → 0 , Cin = 0 → 0

ที่ t = 100 ns ค่าของ x = 1 → 0 , y = 0 → 1 , Cin = 0 → 0

ที่ t = 150 ns ค่าของ x = 0 → 1 , y = 1 → 1 , Cin = 0 → 0

ที่ t = 200 ns ค่าของ x = 1 → 0 , y = 1 → 0 , Cin = 0 → 1

ที่ t = 250 ns ค่าของ x = 0 → 1 , y = 0 → 0 , Cin = 1 → 1

ที่ t = 300 ns ค่าของ x = 1 → 0 , y = 0 → 1 , Cin = 1 → 1

ที่ t = 350 ns ค่าของ x = 0 → 1 , y = 1 → 1 , Cin = 1 → 1

ที่ t = 400 ns ค่าของ x = 1 → 0 , y = 1 → 0 , Cin = 1 → 0

ที่ t = 450 ns ค่าของ x = 0 → 1 , y = 0 → 1 , Cin = 0 → 1

ผลบวกที่ถูกต้องเกิดขึ้นช้า = 5.208 ns

ผลบวกที่ถูกต้องเกิดขึ้นช้า = - ns

ผลบวกที่ถูกต้องเกิดขึ้นช้า = 5.766 ns

ผลบวกที่ถูกต้องเกิดขึ้นช้า = 8.4 ns

ผลบวกที่ถูกต้องเกิดขึ้นช้า = 5.766 ns

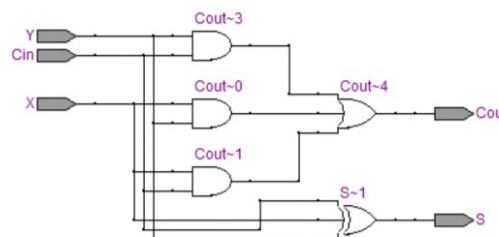
ผลบวกที่ถูกต้องเกิดขึ้นช้า = - ns

ผลบวกที่ถูกต้องเกิดขึ้นช้า = 5.208 ns

ผลบวกที่ถูกต้องเกิดขึ้นช้า = 7.965 ns

ผลบวกที่ถูกต้องเกิดขึ้นช้า = 7.788 ns

d) ให้เขียนวงจรลอจิกที่ได้จากความหมายของภาษา VHDL ในรูปที่ 10 (ดูในเมนู Tools > Netlist Viewer > RTL Viewer)







### คำถาม

1. ให้สังเกตผลการทดลองในรูปที่ 12 และข้อ d) แล้วเปรียบเทียบกับรูปที่ 9 ว่ามีความเหมือนหรือแตกต่างกันอย่างไรในด้านเวลาหน่วง Delay time จาก Pic.12 มีค่าน้อยกว่า Delay time จาก Pic.9 อย่างมาก เช่นที่เวลาหน่วงมากที่สุดของแปลรูป คือ 8.4 กับ 8.6 ms ต่างกันอย่างมาก
2. น.ศ.คิดว่า การออกแบบวงจรลอจิกด้วยวิธีที่เขียนวงจร logic diagram เมื่อเปรียบเทียบกับวิธีที่เขียนด้วยภาษา VHDL มีข้อเด่นข้อด้อยต่างกันอย่างไร การเขียนด้วย logic diagram ทำง่ายมองเห็นภาพรวมของวงจรง่ายกว่า VHDL เมื่อ User เป็นมือใหม่ แต่ข้อเสียคือ เมื่อทำการ Simulate จะเกิด Delay time มากกว่า VHDL

ลายเซ็นอาจารย์ผู้ควบคุม..... /...../.....

### งานมอบหมายท้ายการทดลอง

(ให้เขียนลงบนกระดาษ A4 ที่มีเส้นบรรทัดและรวมใส่ท้ายเอกสารการทดลอง ส่งอาจารย์ผู้สอนในคราวถัดไป)

1. ให้อธิบายคำสั่งต่างๆ ของภาษา VHDL ที่แสดงในรูปที่ 10 อย่างละเอียด
2. ให้อธิบายและยกตัวอย่างประกอบ เรื่องความสำคัญของ **เวลาที่ใช้ในการบวกเลขของวงจร ADDER** ถ้าวางจรบวกเลขในรูปที่ 7 ถูกนำไปสร้างเป็นวงจรบวกเลขขนาด 4 บิตหรือมากกว่า

## งานมอบหมายท้ายการทดลอง

(ให้เขียนลงบนกระดาษ A4 ที่มีเส้นบรรทัดและรวมใส่ท้ายเอกสารการทดลอง ส่งอาจารย์ผู้สอนในคราวถัดไป)

- ให้อธิบายคำสั่งต่างๆ ของภาษา VHDL ที่แสดงในรูปที่ 10 อย่างละเอียด
- ให้อธิบายและยกตัวอย่างประกอบ เรื่องความสำคัญของเวลาที่ใช้ในการบวกเลขของวงจร ADDER ถ้าวางจรวกเลขในรูปที่ 7 ถูกนำไปสร้างเป็นวงจรบวกเลขขนาด 4 บิตหรือมากกว่า

①

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY FullAdder IS
    PORT ( Cin, X, Y : IN std_logic;
          Cout, S : OUT std_logic );
END FullAdder;

ARCHITECTURE rtl OF FullAdder IS
    BEGIN
        S <= X XOR Y XOR Cin;
        Cout <= (X AND Y) OR (Cin AND X) OR (Cin AND Y);
    END rtl;
```

คำสั่ง VHDL ทุกรูปที่ 10

คำสั่ง library เป็นคำสั่งที่ให้เราเรียกใช้ library ที่ต้องการใช้

คำสั่ง use เป็นคำสั่งที่เราเรียกใช้ package ที่เราเรียกจาก library

คำสั่ง entity เป็นคำสั่งที่ใช้กำหนดสถานะตัวแปรที่เราสร้างขึ้น

คำสั่ง end จบการทำงาน และส่งค่าไปยัง file อื่นๆ

คำสั่ง architecture เป็นคำสั่งที่คอยกำหนดการทำงานของวงจรที่เราออกแบบด้วย logic gate

② ให้อธิบายและยกตัวอย่างประกอบ เรื่องความสำคัญของเวลาที่ใช้ในการบวกเลขของวงจร Adder ถ้าวางจรวกเลขในรูปที่ 7 ถูกนำไปสร้างเป็นวงจรบวกเลขขนาด 4 bit หรือมากกว่า

ตอบ สำหรับฝึกออกแบบ จำเป็นให้ความสำคัญกับการแสดงผลของ Output ทุกรูป 7 เมื่อทำการ Simulate จะพบว่าค่า Delay time นั้นค่อนข้างมาก แต่เมื่อเราไปสร้างวงจรที่มีหลายอินพุตขึ้น ค่า Delay time จะมีการสะสมมากขึ้น ทำให้เวลาแสดงผลของวงจรช้าลง นอกเสียจากที่เราต้องลด Delay