



ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

ภาคการศึกษาที่ 1..... ปีการศึกษา 2564

รหัสวิชา 010113026 ชื่อวิชา Digital Laboratory

ตอนเรียน 7..... หมายเลขโต๊ะ.....

รหัสนักศึกษา 6201011631188

ชื่อ-นามสกุล ชาญวิเศษณ์ สุขสมบุรณ์

อาจารย์ผู้สอน CSP

เวลาที่ทำการทดลอง 13.00-16.00 วันที่ 22 / กค / 64

## การทดลองที่ 1

### คุณลักษณะทางเวลาและการทำงานของอุปกรณ์ลอจิกเกต

#### วัตถุประสงค์

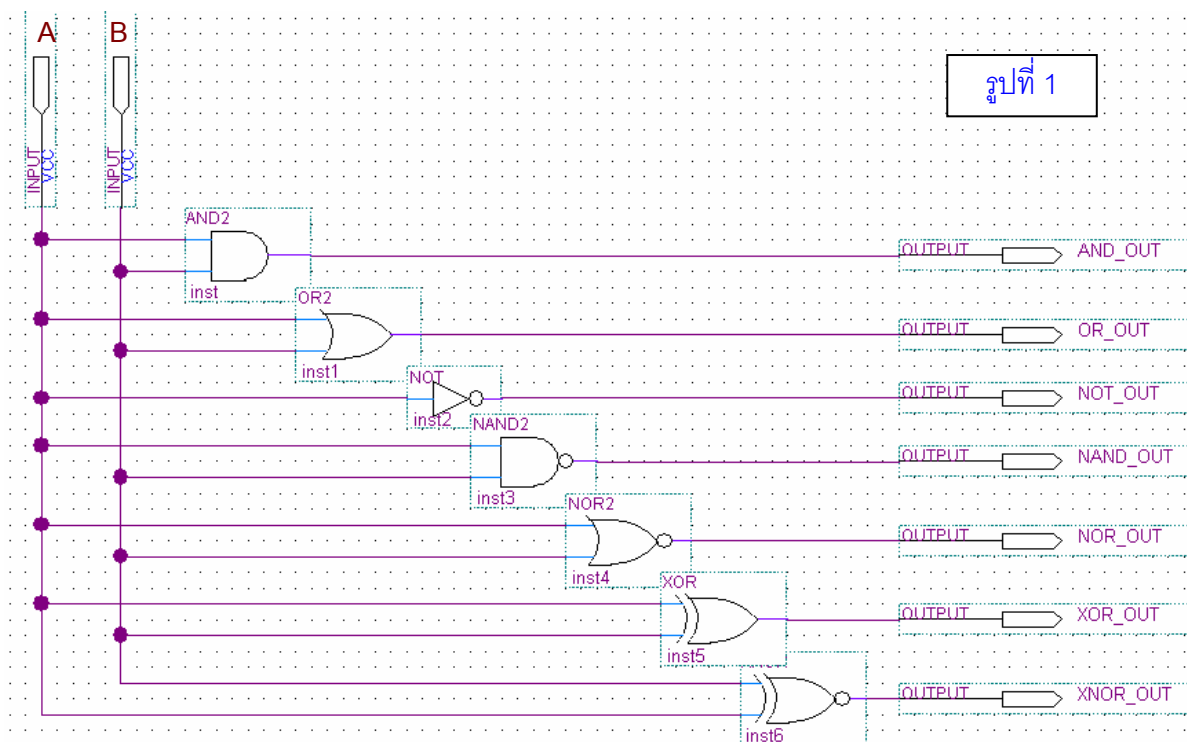
1. เพื่อให้สามารถใช้โปรแกรมคอมพิวเตอร์จำลองการทำงานของวงจรลอจิกเกต (Logic gates) ได้
2. เพื่อให้เข้าใจคุณลักษณะพื้นฐานทางเวลา ของอุปกรณ์ดิจิทัล ลอจิกเกต
3. เพื่อให้เข้าใจขั้นตอนการเขียนภาษา VHDL เพื่อแสดงคุณสมบัติของอุปกรณ์ลอจิกเกตได้

#### อุปกรณ์

1. ระบบคอมพิวเตอร์ 1 เครื่อง พร้อมติดตั้งโปรแกรม Quartus II เวอร์ชัน 8.x (Student Edition) ขึ้นไป

#### การทดลอง

1. ให้ น.ศ. สร้างโฟลเดอร์สำหรับเก็บงานโปรเจกต์ไฟล์ขึ้นใหม่เพื่อเก็บงานที่จะทดลองในการทดลองนี้
2. เขียนวงจรในรูปที่ 1 ด้วย Graphic Editor Tool ของ Quartus II ใช้ชิพ FPGA เบอร์ EP3C10E144C8 และทำการคอมไพล์ให้เรียบร้อย





3. สร้างไฟล์แสดงแผนภาพทางเวลา (Timing diagram) ด้วย Vector Waveform Editor Tool โดยให้ค่าของสัญญาณอินพุต A และ B มีค่าเป็น 00, 01, 10, และ 11 ดังรูปที่ 2 สำหรับการสร้างกราฟให้ทำดังนี้

ก) ตั้งค่าของพารามิเตอร์สำหรับแสดงผลจำลองการทำงาน (เมนู **Edit**)

End Time = 300 ns (เมนู **Edit >> End Time --> Time : 300 ns**)

Grid Size = 1 ns (เมนู **Edit >> Grid size --> Period: 1.0 ns**)

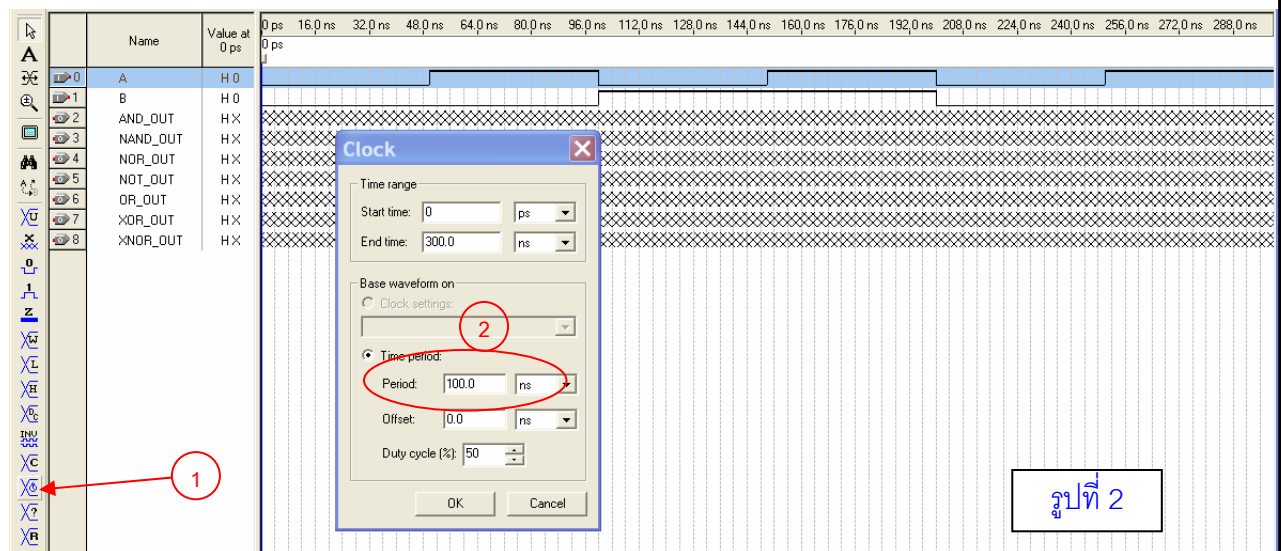
ข) สร้างสัญญาณลอจิกในรูปที่ 2 โดยใช้เมาส์กดเลือกสัญญาณ A จะปรากฏเครื่องมือช่วยสร้างรูปคลื่น

- สัญญาณลอจิก A ใช้ตัวสร้างรูปคลื่นแบบ Overwrite clock **(1)**

โดยมี Period (คาบ) 100 ns **(2)** Offset 0.0 ns Duty cycle 50%

- สัญญาณลอจิก B ใช้ตัวสร้างรูปคลื่นแบบ Overwrite clock เช่นเดียวกันกับ A

โดยมี Period (คาบ) 200 ns Offset 0.0 ns Duty cycle 50%

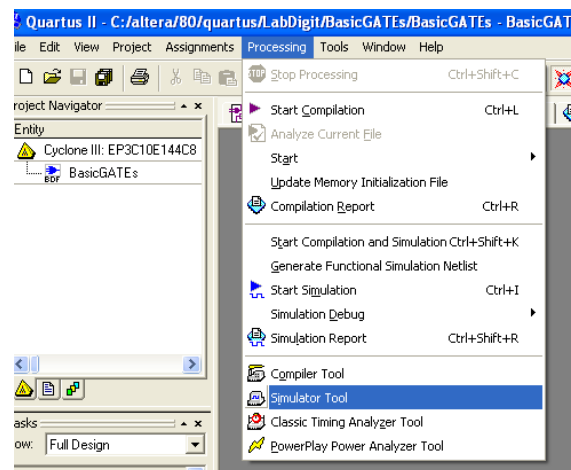


3. ตั้งค่าพารามิเตอร์สำหรับจำลองการทำงานสภาวะลอจิกให้เป็นแบบในอุดมคติ (**Functional**) โดย (การทำงานแบบอุดมคติหรือแบบไม่ใช่เวลาจริง, **Functional Simulation**)

ก) ไปที่เมนู

**Processing → Simulator Tool**

ดังรูปที่ 3



รูปที่ 3



ข) ที่หน้าต่าง Simulator Tool รูปที่ 4 ให้ทำตามลำดับขั้นตอนดังนี้

๑. ให้เลือก

Simulation mode: **Functional**

๒. จากนั้นกดปุ่ม

**Generate Functional Simulation Netlist**

เพื่อกำหนดรูปแบบประมวลผล

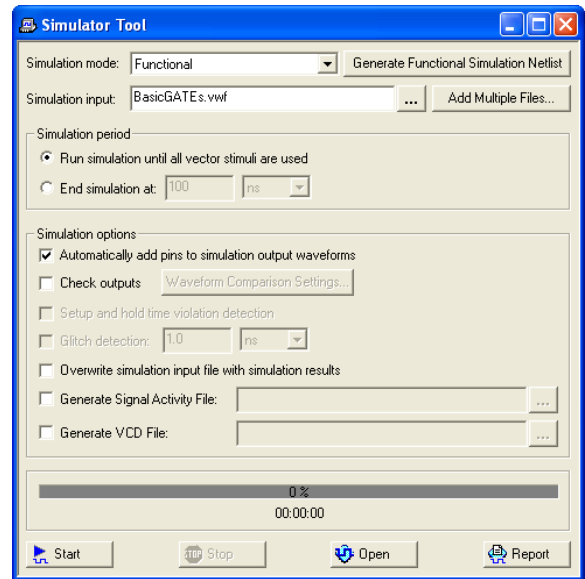
๓. กดปุ่ม

**Start**

เพื่อเริ่มต้นการจำลองการทำงาน

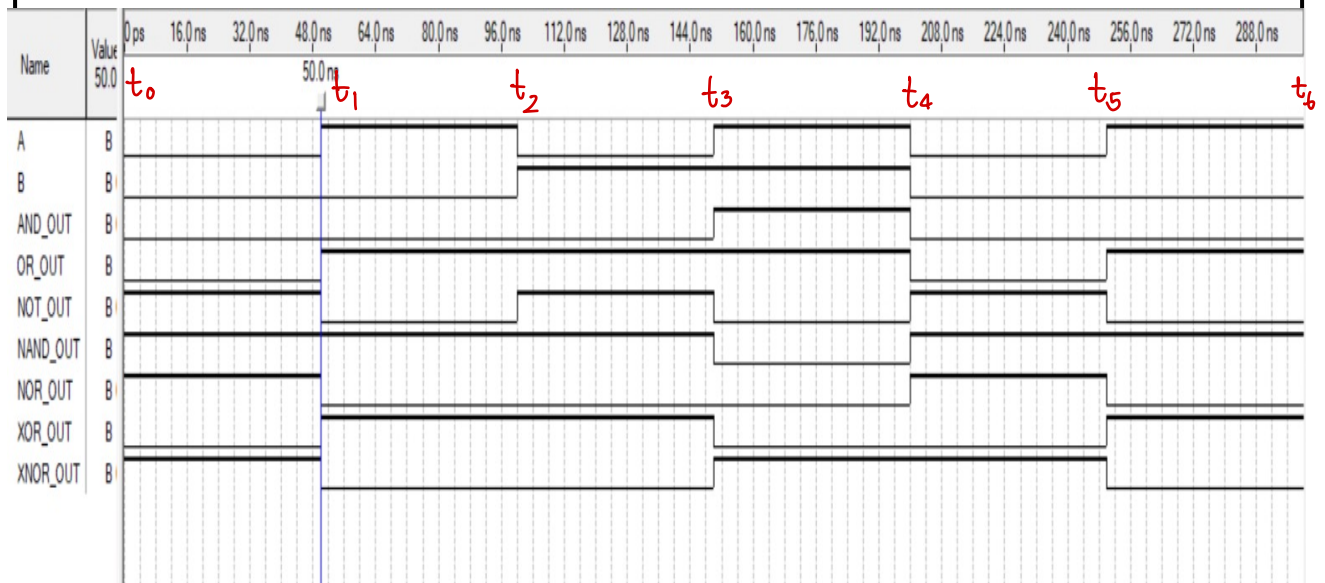
๔. กดปุ่ม

**Report** เพื่อดูผลการจำลองการทำงาน



รูปที่ 4

4. ให้บันทึกผลการจำลองการทำงานอย่างละเอียดลงในกราฟรูปที่ 5 (ใช้ไม้บรรทัดช่วยเขียนส่วนที่เป็นเส้นตรง)



รูปที่ 5



นำค่าสถานะลอจิกที่ได้จากกราฟแผนภาพทางเวลาในรูปที่ 5 มาเขียนเป็นตารางความจริง (Truth table) ที่เวลาในจุดต่างๆ ดังตาราง

**หมายเหตุ** การอ่านค่าลอจิกจากกราฟ ให้อ่านแบบ “กราฟมีความต่อเนื่องทางขวา”

โดยดูจากขอบ (ขอบขาขึ้น ↑ หรือขอบขาลง ↓) ของกราฟแล้วให้อ่านค่าลอจิกด้านขวาของขอบนั้นๆ

	A	B	AND	OR	NOT	NAND	NOR	XOR	XNOR
t <sub>0</sub>	0	0	0	0	1	1	1	0	1
t <sub>1</sub>	1	0	0	1	0	1	0	1	0
t <sub>2</sub>	0	1	0	1	1	1	0	1	0
t <sub>3</sub>	1	1	1	1	0	0	0	0	1

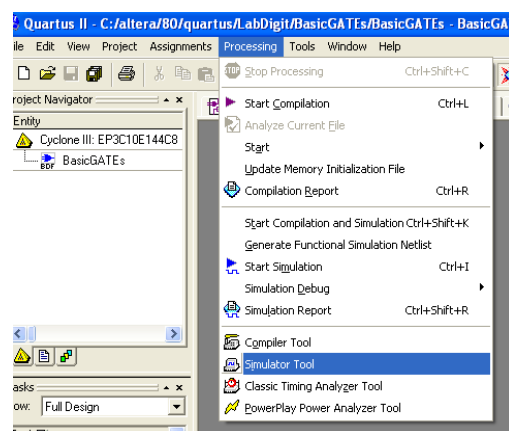
เพราะเหตุใดในตารางความจริงจึงไม่จำเป็นต้องมีการเขียนค่าลอจิกเอาต์พุตของเกทที่เวลา t<sub>4</sub>, t<sub>5</sub>, t<sub>6</sub> เหมือนในแผนภาพทางเวลารูปที่ 5 คำ Logic ที่ได้ออก Output ที่เวลา t<sub>4</sub>-t<sub>6</sub> เป็นกรณีที่เกิดขึ้นในค่า Output เหมือน t<sub>1</sub>-t<sub>3</sub>

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

5. จากข้อ 3 เปลี่ยนพารามิเตอร์สำหรับการประมวลผลเพื่อจำลองการทำงานสถานะลอจิกให้เป็นแบบตามเวลาจริง (หรือเรียกว่าจำลองการทำงานแบบ **Timing Simulation**) โดย

ก) ไปที่เมนู

**Processing → Simulator Tool**



รูปที่ 6

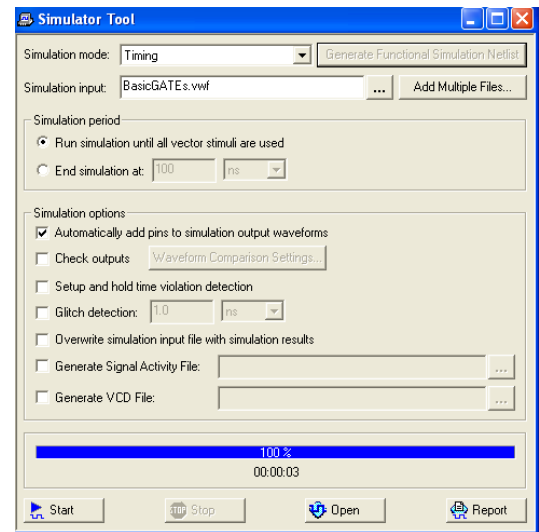


ข) ที่หน้าต่าง Simulator Tool ให้ดำเนินการ

๑. กำหนด Simulation mode: **Timing**

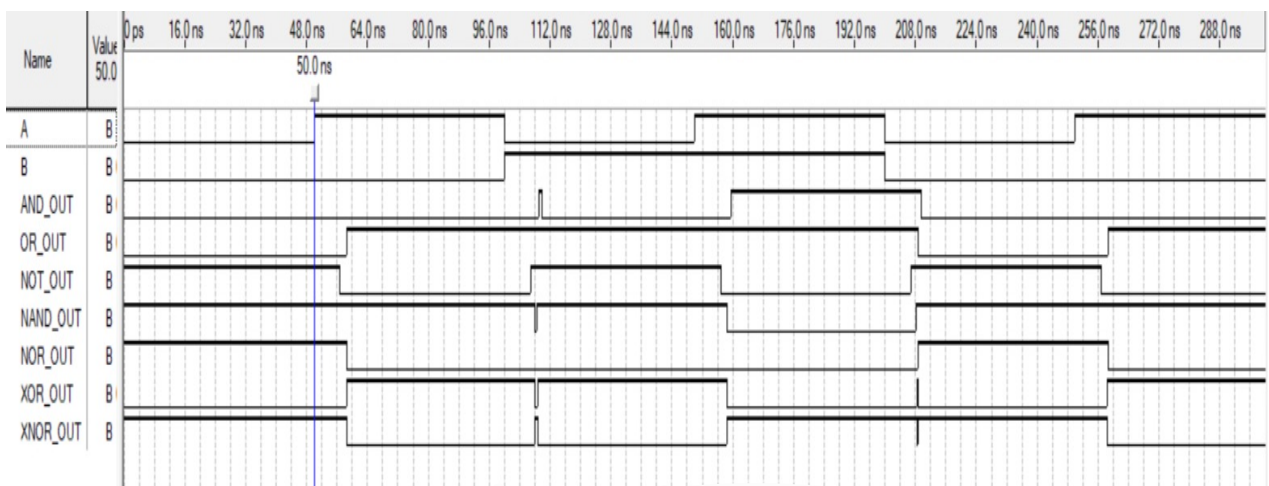
๒. กดปุ่ม **Start**  
เพื่อเริ่มต้นจำลองการทำงาน

๓. กดปุ่ม **Report**  
เพื่อดูผลจำลองการทำงาน



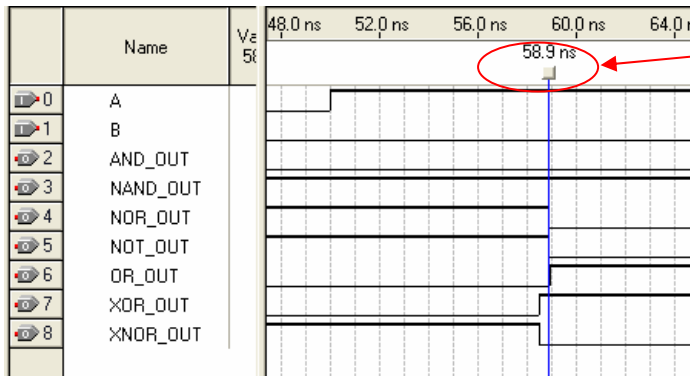
รูปที่ 7

6.บันทึกผลการทำงานอย่างละเอียดลงในกราฟรูปที่ 8 (ให้วาดกราฟแกนเวลาอย่างละเอียดและถูกต้อง)



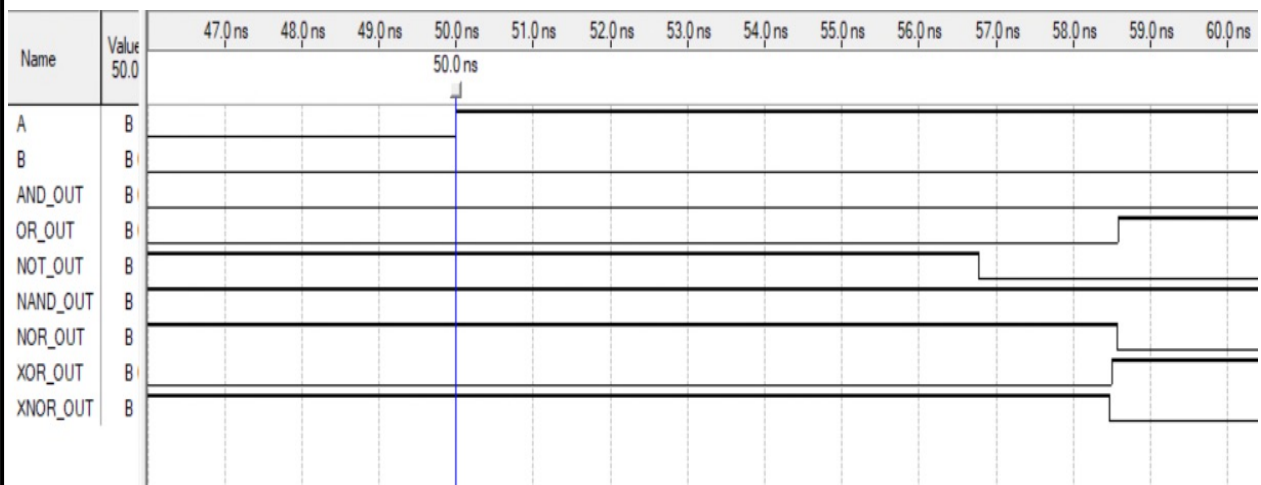
รูปที่ 8

ให้ทำการขยายกราฟ timing diagram (กดปุ่ม **Ctrl** ค้างไว้และหมุน **wheel** ที่เมาส์) จนได้ขนาดที่ชัดเจนพอเหมาะ จากนั้นใช้เมาส์กดที่ปุ่ม **Cursor** (รูปที่ 9) ค้างไว้แล้วลากเลื่อนดูค่าของเวลาในจุดที่ต้องการได้ทำการขยายดูกราฟในช่วงเวลา 49 - 60 ns และบันทึกกราฟอย่างละเอียดลงในรูปที่ 10  
ขยายดูกราฟในช่วงเวลา 99 - 110 ns และทำการบันทึกกราฟอย่างละเอียดลงในรูปที่ 11  
ขยายดูกราฟในช่วงเวลา 149 - 160 ns และทำการบันทึกกราฟอย่างละเอียดลงในรูปที่ 12

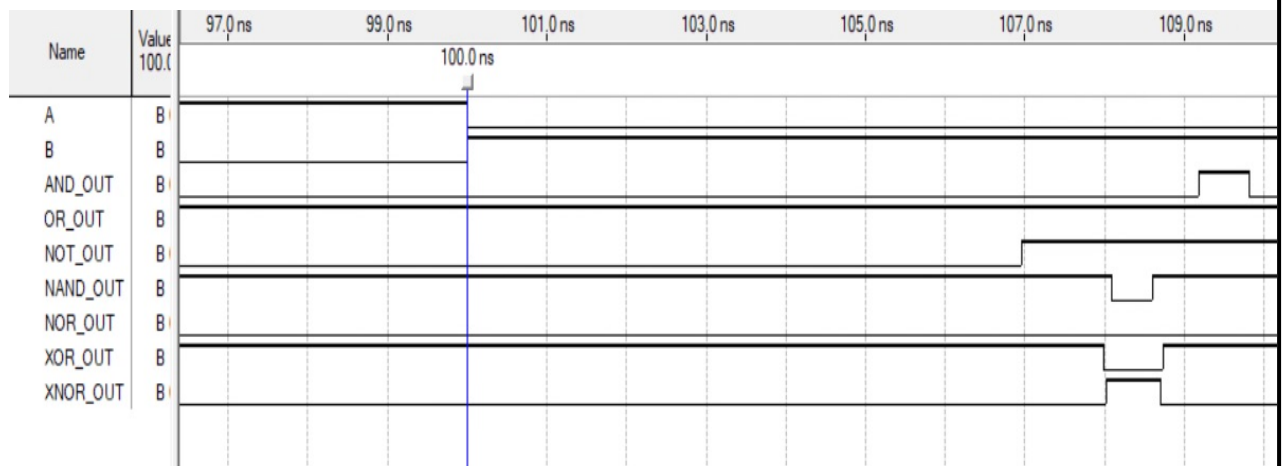


เพื่อจะได้ดูค่าของเวลาบน timing diagram ได้อย่างละเอียด ให้ใช้ปุ่มเคอร์เซอร์โดยใช้เมาส์กดที่ตรงจุดสีเหลี่ยมแล้วลากไปตามแกนนอน หรือใช้คีย์ลูกศรบนแป้นพิมพ์เลื่อนแทนเมาส์ก็ได้เช่นกัน

รูปที่ 9

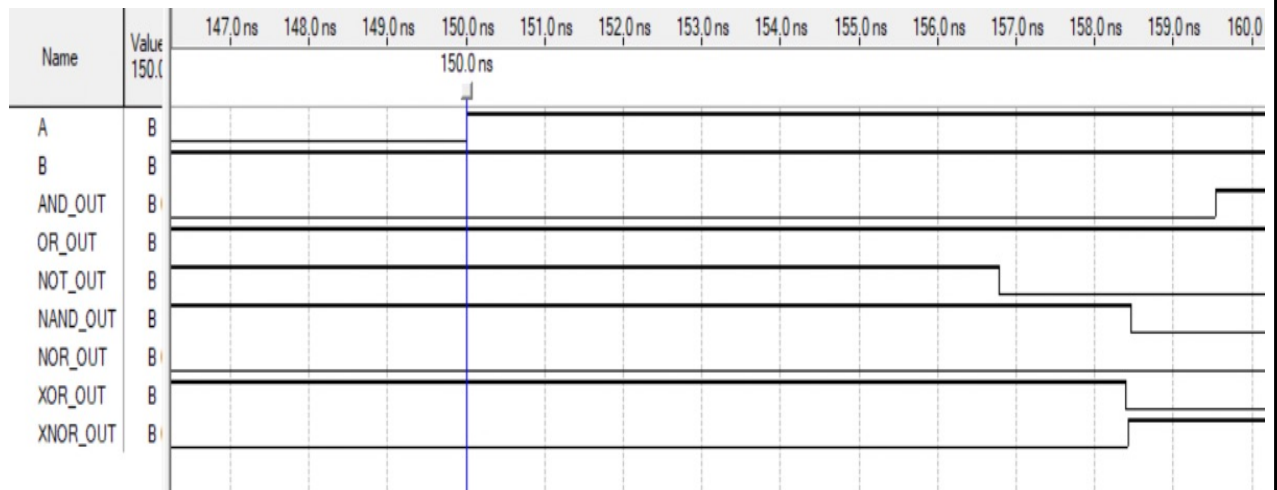


รูปที่ 10



รูปที่ 11





รูปที่ 12

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

จากกราฟรูปที่ 10 - 12 นำค่าของเวลาหน่วง (delay time) ของสัญญาณเอาท์พุตที่เกิดขึ้นจากการเปลี่ยนค่าของอินพุต AB มาเขียนบันทึกในตาราง

ตารางบันทึกผลของเวลาหน่วง (delay time)

ที่เวลา t=50 ns (เหตุการณ์ : ค่าของ A เปลี่ยนจาก 0 → 1 ส่วนค่าของ B คงที่หรือเปลี่ยนจาก 0 → 0)

เกท	ค่าลอจิกเอาท์พุต	เวลาที่เอาท์พุตเปลี่ยนแปลง (ns)	ค่าdelay (เอาท์พุตช้ากว่าอินพุต, ns)
AND	0	-	-
OR	1	58.578	8.578
NOT	0	56.776	6.776
NAND	1	-	-
NOR	0	58.559	8.559
XOR	1	58.449	8.449
XNOR	0	58.468	8.468

ที่เวลา t=100 ns (เหตุการณ์ : ค่าของ A เปลี่ยนจาก 1 → 0 ส่วนค่าของ B เปลี่ยนจาก 0 → 1)

เกท	ค่าลอจิกเอาท์พุต	เวลาที่เอาท์พุตเปลี่ยนแปลง (ns)	ค่าdelay (เอาท์พุตช้ากว่าอินพุต, ns)
AND	0	-	-
OR	1	-	-
NOT	1	106.969	6.969
NAND	1	-	-
NOR	0	-	-
XOR	1	-	-
XNOR	0	-	-



ที่เวลา  $t=150$  ns (เหตุการณ์ : ค่าของ A เปลี่ยนจาก 0  $\rightarrow$  1 ส่วนค่าของ B คงที่หรือเปลี่ยนจาก 1  $\rightarrow$  1)

เกท	ค่าลอจิกเข้าที่พุท	เวลาที่เข้าที่พุทเปลี่ยนแปลง (ns)	ค่าdelay (เข้าที่พุทช้ากว่าอินพุท, ns)
AND	1	159.546	9.546
OR	1	—	—
NOT	0	156.776	6.776
NAND	0	158.455	8.455
NOR	0	—	—
XOR	0	158.396	8.396
XNOR	1	158.427	8.427

พิจารณาอุปกรณ์ที่มีเพียงอินพุตเดียว ( NOT gate ) เปรียบเทียบช่วงเวลาต่าง ๆ กัน

ที่  $t_1$  ค่าอินพุตเปลี่ยนจาก 0  $\rightarrow$  1 แต่ต้องใช้เวลา 6.776 ns กว่าค่าของเข้าที่พุทจะเปลี่ยนจาก 1  $\rightarrow$  0

ที่  $t_2$  ค่าอินพุตเปลี่ยนจาก 1  $\rightarrow$  0 แต่ต้องใช้เวลา 6.969 ns กว่าค่าของเข้าที่พุทจะเปลี่ยนจาก 0  $\rightarrow$  1

น.ศ. คิดว่าสาเหตุใดที่ทำให้ค่าเวลาหน่วงของเข้าที่พุททั้งสองกรณีไม่เท่ากัน

ถ้าไม่นับมองเป็นอุปกรณ์ชนิดหนึ่ง เสมือน Capacitor ที่ 0 ( Logic 0 )

ถ้าต่อวงจรที่มีประจุ ก็ต่อวงจร ( Logic 1 ) เมื่อเทียบกับ ตัวโอ้เวลาจากทฤษฎี

พิจารณาอุปกรณ์ที่มีมากกว่าหนึ่งอินพุต ( AND gate ) ในช่วงเวลา  $t_2$  ผลลัพธ์: สัญญาณออก

ที่  $t_2$  ค่าของอินพุตเปลี่ยนแปลงพร้อมกันทั้งคู่ ( A เปลี่ยนจาก 1  $\rightarrow$  0 ส่วน B เปลี่ยนจาก 0  $\rightarrow$  1 ) ที่เข้าที่พุทเกิดปรากฏการณ์ glitch

น.ศ. คิดว่าสาเหตุที่ทำให้เข้าที่พุทของ AND gate เกิดปรากฏการณ์เช่นนี้คือ Input มีการเปลี่ยนแปลง และ มีค่า Delay ไม่เท่ากัน ส่งผลให้เกิด glitch

มีอุปกรณ์ชนิดใดบ้างที่เข้าที่พุทเกิดปรากฏการณ์เช่นเดียวกันกับ AND gate และจะเกิดเมื่อค่าของอินพุตมีสถานะเช่นใด NAND, XOR, XNOR เกิดเมื่อ Output มีสถานะ: คงเดิม ไม่มีการ

เปลี่ยนแปลง แม้ Input เปลี่ยน

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

7. ให้ น.ศ. สร้างไฟล์เตอร์สำหรับเก็บงานโปรเจกต์ไฟล์ขึ้นใหม่เพื่อเก็บงานที่จะทดลองในการทดลองนี้ จากนั้นให้เขียนภาษา VHDL เพื่อจำลองการทำงานของโปรแกรมในรูปที่ 13

ชื่อโปรเจกต์ LAB01VHD

ชื่อไฟล์ LAB01VHD.vhd (ชื่อโปรเจกต์และชื่อไฟล์ต้องตรงกันจึงจะคอมไพล์ได้)





```
library ieee;
use ieee.std_logic_1164.all;

entity LAB01VHD is
  port
    ( A, B                : in  std_logic;
      OR_OUT, AND_OUT     : out std_logic;
      NOT_OUT, NAND_OUT   : out std_logic;
      NOR_OUT, XOR_OUT    : out std_logic;
      XNOR_OUT            : out std_logic );
end entity;

architecture rtl of LAB01VHD is
begin
  OR_OUT      <= A OR B ;
  AND_OUT     <= A AND B ;
  NOT_OUT     <= NOT A ;
  NAND_OUT    <= A NAND B ;
  NOR_OUT     <= A NOR B ;
  XOR_OUT     <= A XOR B ;
  XNOR_OUT    <= A XNOR B ;
end rtl;
```

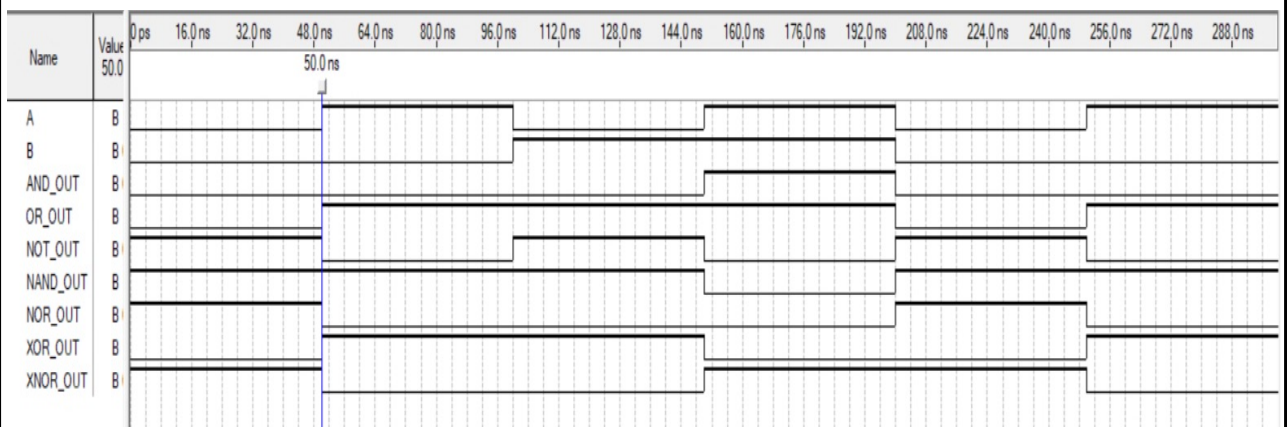
รูปที่ 13

หมายเหตุ น.ศ. ต้องเปิดโปรเจกใหม่ และตั้งชื่อโปรเจกเป็นชื่อเดียวกันกับชื่อของไฟล์และชื่อของ **entity**

8. ทำการคอมไพล์และจำลองการทำงานโดยกำหนดให้

- Simulation mode: Functional (ตั้งค่าเช่นเดียวกันกับในข้อ 3)
- สัญญาณอินพุต A และ B ให้มีค่าเช่นเดียวกันกับการทดลองในรูปที่ 14

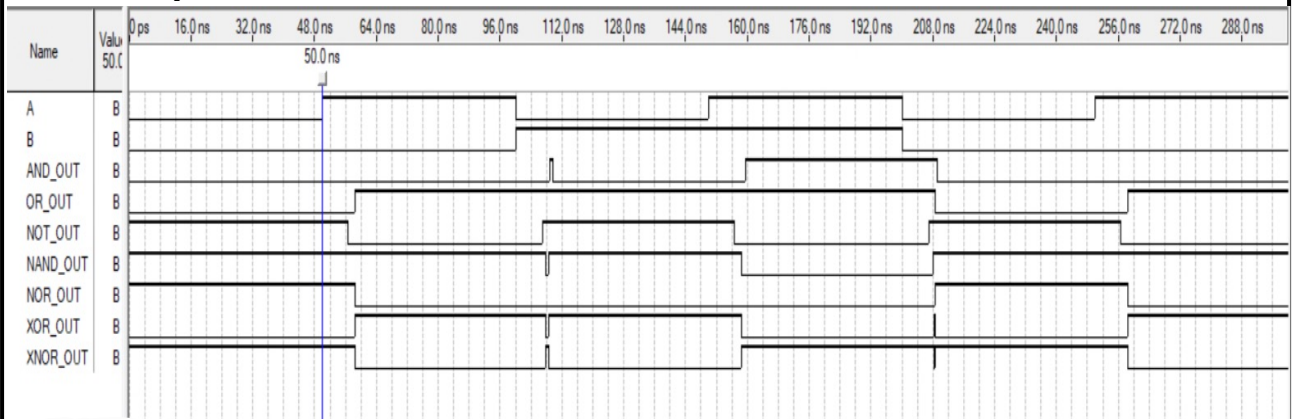
บันทึกผลการจำลองการทำงานอย่างละเอียดลงในกราฟรูปที่ 14



รูปที่ 14



9. เปลี่ยนโหมดการจำลองการทำงานให้เป็นแบบ **Timing** (ตั้งค่าเช่นเดียวกันกับในข้อ 5) และบันทึกผลการจำลองลงในรูปที่ 15



รูปที่ 15

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

ทำการซูมขยายดูกราฟในช่วงเวลา 49-60 ns ช่วง 99-110 ns และ 149-160 ns ตรวจสอบว่าเกิดปรากฏการณ์เช่นเดียวกันกับในรูปที่ 10 - 12 หรือไม่ ถ้ามี เกิดขึ้นที่เวลาใดบ้าง ค่าอินพุตมีการเปลี่ยนแปลงในสถานะใดให้อธิบาย เหมือนกับรูป 10-12 ที่เวลา 49-60 ns มีทร Delat เกิดขึ้น 99-110 อุปกรณ์ที่มี Output คงเดิม; เกิด glitch แต่ค่า Output ที่มีทรเปลี่ยนแปลงจะเกิดทร Delay เช่นได้ขงกณ 149-160

เปรียบเทียบและวิจารณ์ผลของเวลาหน่วง (delay time) ของ NOT gate ที่ได้จากการทดลองด้วยวงจรลอจิก (schematic) กับการทดลองด้วยภาษา VHDL

NOT gate ที่ได้จาก schematic มี Delay คือ 6.776, 6.996 และ 6.776 ns  
NOT gate ที่ได้จาก VHDL มี Delay คือ 6.689, 6.949 และ 6.689 ns  
จะพบว่า ค่า Delay ของ VHDL มีค่าน้อยกว่า schematic

**สรุปผลการทดลอง** จากการทดลอง เราพบว่า ในทรทำาหนรงของวงจรนั้น จะเกิดทร Delay ของอุปกรณ์ ในโปรแกรมเราสามารถใช้ค่าได้ที่ต่อทรค่าจริง หรือค่าสมททฤษฎี

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

งานมอบหมาย (ให้เขียนลงบนกระดาษ A4 ที่มีเส้นบรรทัดแนบท้ายเอกสารการทดลองส่งครวถัดไป)

1. ให้ทำสรุปความรู้ที่ได้จากการทดลองนี้
2. ให้ทำรายงานสรุปเรื่องคุณสมบัติทางเวลา (Timing Characteristics) ของเกท

งานมอบหมาย (ให้เขียนลงบนกระดาษ A4 ที่มีเส้นบรรทัดแนบท้ายเอกสารการทดลองส่งคร่าวๆไป)

1. ให้ทำสรุปความรู้ที่ได้จากการทดลองนี้

2. ให้ทำรายงานสรุปเรื่องคุณสมบัติทางเวลา (Timing Characteristics) ของเกท

1. ให้นำสรุปความรู้ที่ได้จากการทดลองนี้

จากการทดลองนี้ ทำให้นักทราบว่า ค่า Output หรือ Input เมื่อเกิดการเปลี่ยนค่า Logic

จ. เกิดการ Delay จาก Device ซึ่งเป็นความรู้ใหม่ แตกต่างจากทฤษฎีที่เรียนในวิชา

Digital ที่ timing diagram มีความ Smooth, Output มีการเปลี่ยน  
พลวัตที่ Input เปลี่ยน

2. ให้นำรายงานสรุปเรื่องคุณสมบัติของเกท

① AND gate จะให้ค่า Logic เป็น 1 เมื่อ Input ทั้งหมดเป็น 1

OR gate จะให้ค่า Logic เป็น 0 เมื่อ Input ทั้งหมดเป็น 0

NOT gate จะให้ค่า Logic ตรงกันข้ามกับ Input เสมอ.

NAND gate จะให้ค่า Logic เป็น 0 เมื่อ Input ทั้งหมดเป็น 1

NOR gate จะให้ค่า Logic เป็น 1 เมื่อ Input ทั้งหมดเป็น 0

XOR gate จะให้ค่า Logic เป็น 0 เมื่อ Input มีค่าเดียวกัน

XNOR gate จะให้ค่า Logic เป็น 1 เมื่อ Input มีค่าเดียวกัน