



ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

ภาคการศึกษาที่.....ปีการศึกษา.....

รหัสวิชา 010113026 ชื่อวิชา Digital Laboratory

ตอนเรียน หมายเลขโต๊ะ.....

รหัสนักศึกษา..... ชื่อ-นามสกุล.....

อาจารย์ผู้สอน.....เวลาที่ทำการทดลอง วันที่.....

การทดลองที่ 4

Combinational-Circuit Building Block

วัตถุประสงค์

1. เพื่อให้สามารถใช้โปรแกรมคอมพิวเตอร์จำลองการทำงานของวงจรลอจิกเกตได้
2. เพื่อให้เข้าใจวิธีการสร้างอุปกรณ์ที่ซับซ้อน โดยอาศัยการต่อวงจรด้วยเกตพื้นฐาน
3. เพื่อให้เข้าใจคุณลักษณะพื้นฐานของ มัลติเพล็กซ์เซอร์ ดีมัลติเพล็กซ์เซอร์
4. เพื่อให้เข้าใจคุณลักษณะพื้นฐานของ เอนโคเดอร์ ดีโคเดอร์
5. เพื่อให้เข้าใจหลักการทำงานของตัวเปรียบเทียบ 4-bit Comparator

เครื่องมือและอุปกรณ์

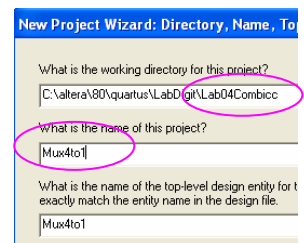
1. ระบบคอมพิวเตอร์ 1 เครื่อง พร้อมติดตั้งโปรแกรม Quartus II เวอร์ชัน 8.0 (Student Edition) ขึ้นไป

การทดลองตอนที่ 1 หลักการทำงานของมัลติเพล็กซ์เซอร์ 4-to-1 Multiplexer

คำแนะนำ: ในการทดลองนี้ จะเป็นมีรูปแบบ เป็นการสร้างโปรเจกต์ใหม่ขึ้นมาเพื่อหาคำตอบโปรแกรมเก่า ไว้เป็นขั้นๆ เช่นในข้อ 1-3 เราจะสร้างอุปกรณ์ชื่อ MUX4to1 ขึ้นมา

คำสั่งการทดลอง

1. ให้ น.ศ. สร้างโฟลเดอร์สำหรับเก็บงานขึ้นใหม่เพื่อเก็บงานที่จะทดลองในการทดลองนี้ชื่อ “Lab04Combicc” จากนั้นให้สร้างโปรเจกต์ชื่อ “Mux4to1” ขึ้นมา ดังรูปที่ 1 ให้ใช้ชิพ EP3C10E144C8



รูปที่ 1

2. เขียนวงจรมัลติเพล็กซ์เซอร์ขนาด 4-to-1

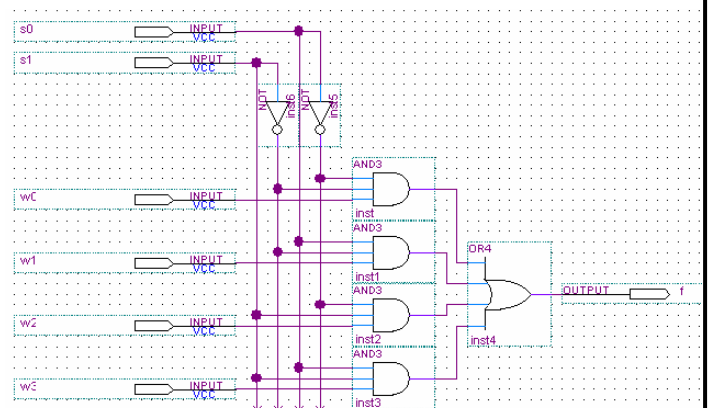
ดังรูปที่ 2 ด้วย Graphic Editor Tool

- ทำการคอมไพล์วงจรให้เรียบร้อย

- สร้าง symbol ของวงจร โดยไปที่เมนู

File >> create/update

>> Create symbol file for current file



รูปที่ 2

3. สร้างไฟล์แสดงแผนภาพทางเวลา (Timing diagram)

ตั้งค่า End Time = 1.0 us และ Grid Size = 1 ns (ดูในเมนู Edit >> End time)



4. กำหนดสัญญาณอินพุต w0, w1, w2 และ w3 ให้มีรูปคลื่นเป็นแบบสัญญาณนาฬิกา โดยสัญญาณและเลือกที่แถบเครื่องมือสร้างรูปคลื่น ดังรูปที่ 3a จากนั้นกำหนดค่าต่างๆตามรูปที่ 3b, 3c, 3d, และ 3e ตามลำดับ

ค่าพารามิเตอร์ของ w0

Time range
Start time: 0 ps
End time: 0.25 us

Base waveform on
Clock settings

Time period:
Period: 25.0 ns
Offset: 0.0 ns
Duty cycle (%): 30

รูปที่ 3b

ค่าพารามิเตอร์ของ w1

Time range
Start time: 0.25 us
End time: 0.50 us

Base waveform on
Clock settings

Time period:
Period: 50.0 ns
Offset: 0.0 ns
Duty cycle (%): 80

รูปที่ 3c

ค่าพารามิเตอร์ของ w2

Time range
Start time: 500 ns
End time: 0.750 us

Base waveform on
Clock settings

Time period:
Period: 75.0 ns
Offset: 0.0 ns
Duty cycle (%): 50

รูปที่ 3d

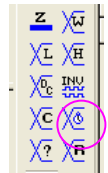
ค่าพารามิเตอร์ของ w3

Time range
Start time: 25 ns
End time: 1.0 us

Base waveform on
Clock settings

Time period:
Period: 50.0 ns
Offset: 0.0 ns
Duty cycle (%): 80

รูปที่ 3e



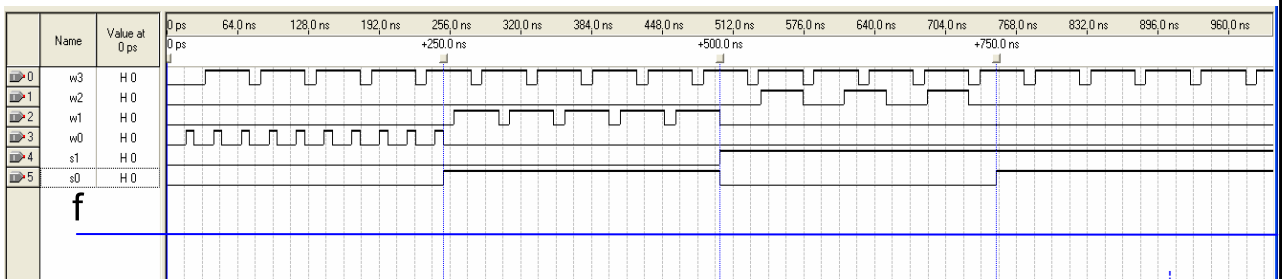
รูปที่ 3a

5. กำหนดของ s0 และ s1 ให้มีรูปคลื่นแบบนาฬิกาดังรูปที่ 4

s0 : Period = 500 ns s1 : Period 1000 ns

6. จำลองการทำงานในโหมด Functional บันทึกผลในกราฟรูปที่ 4

บันทึกผลการทดลอง



รูปที่ 4

- เมื่อ s1= '0' s0= '0' สัญญาณที่ปรากฏที่เอาต์พุต f มีรูปร่างเหมือนสัญญาณ (อินพุต)
- เมื่อ s1= '0' s0= '1' สัญญาณที่ปรากฏที่เอาต์พุต f มีรูปร่างเหมือนสัญญาณ (อินพุต)
- เมื่อ s1= '1' s0= '0' สัญญาณที่ปรากฏที่เอาต์พุต f มีรูปร่างเหมือนสัญญาณ (อินพุต)
- เมื่อ s1= '1' s0= '1' สัญญาณที่ปรากฏที่เอาต์พุต f มีรูปร่างเหมือนสัญญาณ (อินพุต)

สมการของ f =

หาสัญญาณ s1 s0 ของอุปกรณ์ที่ชื่อ Mux4to1 มีหน้าที่อะไร

ให้อธิบายหลักการทำงานของอุปกรณ์ที่ชื่อ Mux4to1

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

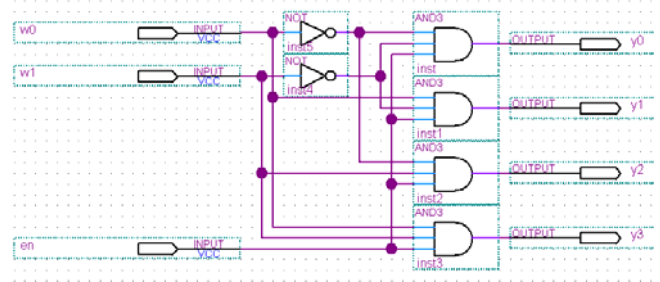


การทดลองตอนที่ 2 หลักการทำงานของ 2-to-4 Decoder

7. ให้ปิดโปรเจกต์เดิม
8. สร้างโปรเจกต์ใหม่ชื่อ “2to4Decoder” โดยเก็บไว้ในโฟลเดอร์เดิม(ข้อที่ 1 – 6) ใช้ชิพ EP3C10E144C8
9. ใช้ Graphic Editor Tool เขียนวงจรดีโคเดอร์ขนาด 2-to-4 ดังรูปที่ 5 เก็บไว้ในไฟล์ชื่อเดียวกันกับโปรเจกต์แล้วทำการคอมไพล์ให้เรียบร้อย
10. สร้าง symbol file ของวงจร

File >> create/update

>> Create symbol file for current file

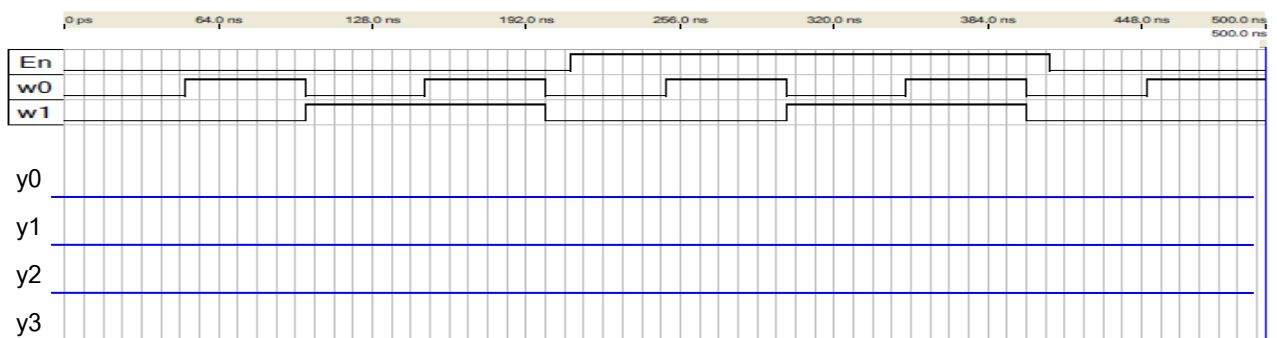


รูปที่ 5

11. สร้างไฟล์แสดงแผนภาพทางเวลา กำหนดค่าของพารามิเตอร์แสดงผลจำลองการทำงานโดยให้มีค่าดังนี้
 - End Time = 500 ns Grid Size = 1 ns (ดูที่เมนู Edit >> Grid size)
 - กำหนดสัญญาณอินพุต En, w0 และ w1 ให้มีรูปคลื่นดังในรูปที่ 6 โดย
 - En กำหนดให้ Start time = 10 ns , Period = 400 ns Duty Cycle[%] = 50
 - w1 กำหนดให้ Start time = 0 ps , Period = 200 ns Duty Cycle[%] = 50
 - w0 กำหนดให้ Start time = 0 ps , Period = 100 ns Duty Cycle[%] = 50

12. ให้ทำการจำลองการทำงานโหมด Functional mode บันทึกผลลงในกราฟรูปที่ 6

บันทึกผลการทดลอง



รูปที่ 6

เมื่อ w1 = '0' w0 = '0' สัญญาณ เป็นเอาต์พุตที่ถูกเลือก (ค่าเป็น '1' เพียงตัวเดียวจากทั้งหมด4ตัว)
 เมื่อ w1 = '0' w0 = '1' สัญญาณ เป็นเอาต์พุตที่ถูกเลือก (ค่าเป็น '1' เพียงตัวเดียวจากทั้งหมด4ตัว)
 เมื่อ w1 = '1' w0 = '0' สัญญาณ เป็นเอาต์พุตที่ถูกเลือก (ค่าเป็น '1' เพียงตัวเดียวจากทั้งหมด4ตัว)
 เมื่อ w1 = '1' w0 = '1' สัญญาณ เป็นเอาต์พุตที่ถูกเลือก (ค่าเป็น '1' เพียงตัวเดียวจากทั้งหมด4ตัว)

สมการของเอาต์พุต =



ขาสัญญาณ w1 w0 ของอุปกรณ์ที่ชื่อ 2to4Decode มีหน้าที่อะไร

ให้อธิบายหลักการทำงานของอุปกรณ์ที่ชื่อ 2to4Decode

จากผลการทดลองในรูปที่ 6 ให้เขียนตารางความจริง

En	w1	w0	y0	y1	y2	y3
1	0	0				
1	0	1				
1	1	0				
1	1	1				
0	x	x				

ลายเซ็นอาจารย์ผู้ควบคุม /...../.....

การทดลองตอนที่ 3 หลักการทำงานของ 1-to-4 Demultiplexer

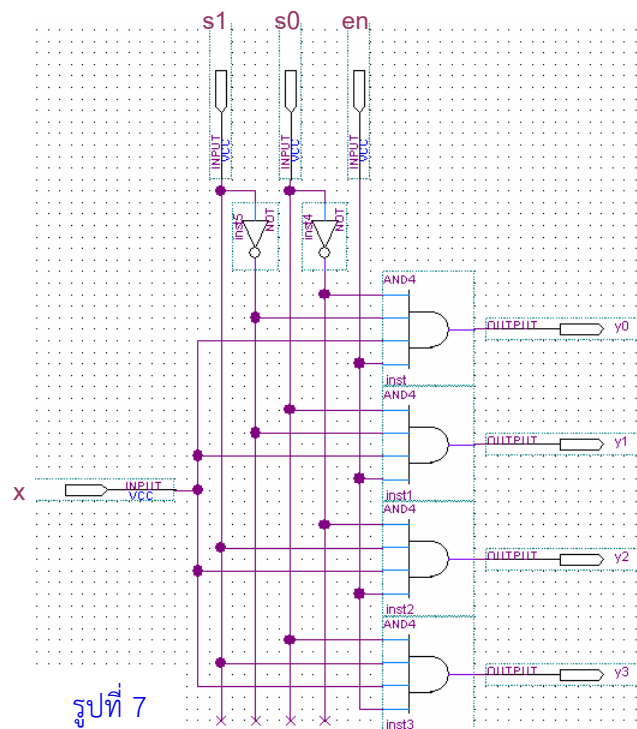
13. ให้ทำการปิดโปรเจกต์ที่สร้างมาในขั้นตอนที่ 7 – 12 ก่อนที่จะทำการทดลองต่อไป

14. สร้างโปรเจกต์ชื่อ “1to4Demux” ขึ้นมาใหม่โดยให้เก็บไว้ในโฟลเดอร์เดิม

ให้ใช้ชิพ EP3C10E144C8 จากนั้นเขียน

วงจรดีมัลติเพล็กซ์ 1-to-4 ในรูปที่ 7

เก็บไว้ในไฟล์ชื่อเดียวกันกับชื่อโปรเจกต์



รูปที่ 7

15. ทำการคอมไพล์ พร้อมสร้าง symbol file

File >> create/update

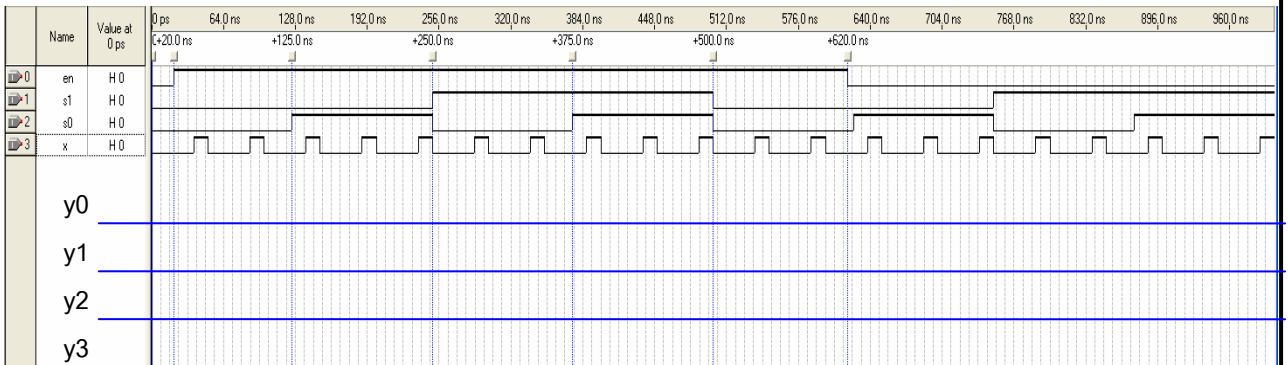
>> Create symbol file for current file

16. สร้างไฟล์แสดงแผนภาพทางเวลาของของสัญญาณ โดยให้ s1 s0 และ en มีรูปคลื่นดังรูปที่ 8

จำลองการทำงานในโหมด Functional mode บันทึกผลการทำงานลงในกราฟรูปที่ 8



บันทึกผลการทดลอง



หมายเหตุ : x เป็นสัญญาณแบบรายคาบ Period=50 ns, duty cycle=25%

รูปที่ 8

ถ้าสัญญาณ en = '1' และ

- s1 = '0' s0 = '0' สัญญาณจากอินพุตถูกเลือกให้ไปที่ขาเอาต์พุตชื่อ..... (มีเพียงขาเดียวจากที่มี4ขา)
- s1 = '0' s0 = '1' สัญญาณจากอินพุตถูกเลือกให้ไปที่ขาเอาต์พุตชื่อ..... (มีเพียงขาเดียวจากที่มี4ขา)
- s1 = '1' s0 = '0' สัญญาณจากอินพุตถูกเลือกให้ไปที่ขาเอาต์พุตชื่อ..... (มีเพียงขาเดียวจากที่มี4ขา)
- s1 = '1' s0 = '1' สัญญาณจากอินพุตถูกเลือกให้ไปที่ขาเอาต์พุตชื่อ..... (มีเพียงขาเดียวจากที่มี4ขา)

ถ้าสัญญาณ en = '0' มีสัญญาณอินพุตไปปรากฏที่เอาต์พุตหรือไม่ เพราะเหตุใด

ขาสัญญาณ s1 s0 อุปกรณ์ที่ชื่อ 1to4Demux มีหน้าที่อะไร.....

ขาสัญญาณ en มีหน้าที่อะไร.....

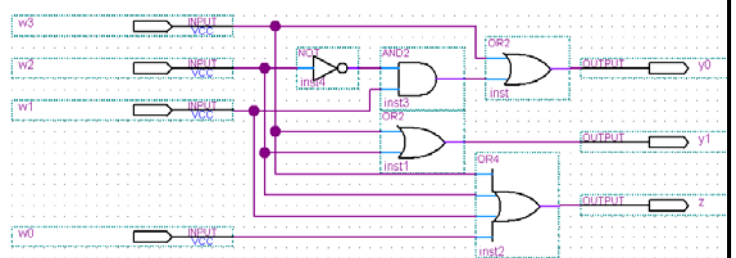
อุปกรณ์ที่ชื่อ 1to4Demux มีสมการเอาต์พุตคือ

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

การทดลองตอนที่ 4

หลักการทำงานของ 4-to-2 Priority Encoder

- ให้ทำการ**ปิดโปรเจกต์**ที่สร้างมาในขั้นตอนที่ 13 – 16 ก่อนที่จะทำการทดลองต่อไป
- สร้างโปรเจกต์ชื่อ “4to2PrioEncode” ขึ้นมาใหม่โดย**ให้เก็บไว้ในโฟลเดอร์เดิม** เขียนวงจรในรูปที่ 9 บันทึกในไฟล์ชื่อเดียวกันกับโปรเจกต์ ทำการคอมไพล์ให้เรียบร้อย



รูปที่ 9

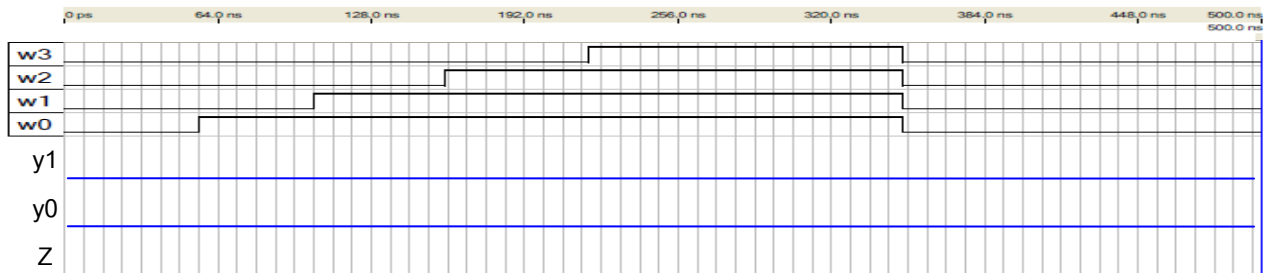


19. สร้าง symbol file เตรียมไว้ใช้งานขั้นตอนต่อไป

File >> create/update >> Create symbol file for current file

20. สร้างไฟล์แสดงแผนภาพทางเวลาของของสัญญาณ โดยให้ w3 w2 w1 w0 มีรูปคลื่นดังรูปที่ 10
จำลองการทำงานในโหมด Functional mode บันทึกผลลงในกราฟรูปที่ 10

บันทึกผลการทดลอง



รูปที่ 10

เมื่อ w3='0',w2='0',w1='0',w0='0' ได้เอาท์พุท z = y1= y0=.... ค่าเลขฐานสิบของ y_1y_0 =

เมื่อ w3='0',w2='0',w1='0',w0='1' ได้เอาท์พุท z = y1= y0=.... ค่าเลขฐานสิบของ y_1y_0 =

เมื่อ w3='0',w2='0',w1='1',w0='1' ได้เอาท์พุท z = y1= y0=.... ค่าเลขฐานสิบของ y_1y_0 =

เมื่อ w3='0',w2='1',w1='1',w0='1' ได้เอาท์พุท z = y1= y0=.... ค่าเลขฐานสิบของ y_1y_0 =

เมื่อ w3='1',w2='1',w1='1',w0='1' ได้เอาท์พุท z = y1= y0=.... ค่าเลขฐานสิบของ y_1y_0 =

ค่าของ z จะเป็น '1' เมื่อ

เป็น '0' เมื่อ

ให้เปรียบเทียบทั้ง 4 กรณีข้างต้นที่มีค่าของ z = '1'

y_1y_0 (มีค่าเลขฐานสิบ) = 0 เมื่อมีอินพุตเข้าที่ขา

y_1y_0 (มีค่าเลขฐานสิบ) = 1 เมื่อมีอินพุตเข้าที่ขา

y_1y_0 (มีค่าเลขฐานสิบ) = 2 เมื่อมีอินพุตเข้าที่ขา

y_1y_0 (มีค่าเลขฐานสิบ) = 3 เมื่อมีอินพุตเข้าที่ขา

ระหว่างอินพุตขา w3, w2, w1, w0 ขาอินพุตใดที่มีความสำคัญมากที่สุด

ขาอินพุตใดที่มีความสำคัญน้อยที่สุด

ให้อธิบายหลักการทำงานของอุปกรณ์ Priority Encoder

.....

.....

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

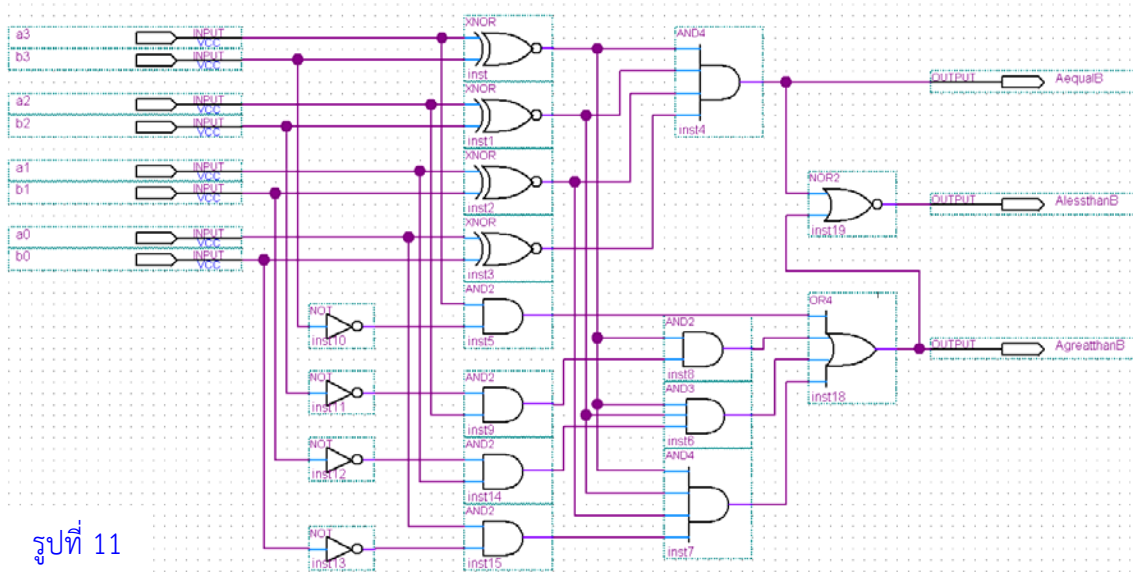
การทดลองตอนที่ 5 หลักการทำงานของ 4-bit Comparator Circuit

21. ให้ทำการ**ปิดโปรเจกต์**ที่สร้างมาในขั้นตอนที่ 17 – 20 ก่อนที่จะทำการทดลองต่อไป

22. สร้างโปรเจกต์ชื่อ “4BitComparator” ขึ้นมาใหม่โดย**ให้เก็บไว้ในโฟลเดอร์เดิม**



23. สร้างไฟล์ใหม่ขึ้นมาให้มีชื่อเดียวกันกับโปรเจกเพื่อเขียนวงจรในรูปที่ 11
24. คอมไพล์และสร้าง symbol file ของวงจรก่อนทำการทดลองขั้นต่อไป



รูปที่ 11

25. สร้างไฟล์แสดงแผนภาพทางเวลาดังรูปที่ 12 โดยให้ดำเนินการตามขั้นตอนต่อไปนี้
- กำหนดค่าของพารามิเตอร์สำหรับแสดงผลจำลองการทำงานโดยให้มีค่าดังนี้
End Time = 1.0 us Grid Size = 1 ns
 - จัดกลุ่มให้แสดงผลเป็นแบบ Hexadecimal โดย
a3, a2, a1, a0 จัดเข้าเป็นกลุ่ม A ให้มีค่าเป็นแบบเลขนับ (Count value, **ดูวิธีทำในหน้าถัดไป**)
เริ่มนับจาก 0 เพิ่มขึ้นครั้งละ 4 ทุกๆ 100 ns
b3, b2, b1, b0 จัดเข้าเป็นกลุ่ม B ให้มีค่าเป็นแบบเลขนับ (Count value)
เริ่มนับจาก 0 เพิ่มขึ้นครั้งละ 1 ทุกๆ 100 ns
26. จำลองการทำงานในโหมด Functional mode และบันทึกผลที่ได้ลงในรูปที่ 12

บันทึกผลการทดลอง

	0 ps	128,0 ns	256,0 ns	384,0 ns	512,0 ns	640,0 ns	768,0 ns	896,0 ns	1.0 us	1.0 us										
A	0	4	8	C	0	4	8	C	0	4	8	C	0	4	8	C				
B	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1	2	3
AeqB																				
AltB																				
AgtB																				

รูปที่ 12

AeqB จะมีสถานะลอจิกเป็น “1” เมื่อ.....
จะมีสถานะลอจิกเป็น “0” เมื่อ.....
AltB จะมีสถานะลอจิกเป็น “1” เมื่อ.....
จะมีสถานะลอจิกเป็น “0” เมื่อ.....



AgtB จะมีสถานะลอจิกเป็น “1” เมื่อ.....
จะมีสถานะลอจิกเป็น “0” เมื่อ.....
ลายเซ็นอาจารย์ผู้ควบคุม..... /...../.....

วิธีการสร้างสัญญาณแบบระบบเลขนับ (Count value) ตามการทดลองข้อ 25

- 1) ใช้เมาส์เลือกสัญญาณที่ต้องการ เช่นสัญญาณ A จะปรากฏแถบสีน้ำเงินขึ้นที่รูปภาพของสัญญาณนั้นๆ
- 2) จากนั้นใช้เมาส์คลิกที่แถบเครื่องมือแบบ Count ดังในรูปที่ 13
- 3) จะปรากฏเมนูให้ตั้งค่า สำหรับสัญญาณในรูปที่ 12 ให้กำหนดค่าต่างๆ ดังนี้

สัญญาณอินพุต A

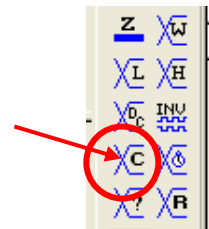
ที่แถบ **Counting** ให้ Start value: 0 , Increment by: 4

ที่แถบ **Timing** ให้ Period: 50 ns

สัญญาณอินพุต B

ที่แถบ **Counting** ให้ Start value: 0 , Increment by: 1

ที่แถบ **Timing** ให้ Period : 50 ns



รูปที่ 13

การทดลองตอนที่ 6 ภาษา VHDL ของอุปกรณ์ที่มีฟังก์ชันการทำงานซับซ้อน

27. ให้เขียนภาษา VHDL ดังในรูปที่ 14 -18

a) ให้สร้างโปรเจกต์ขึ้นใหม่ สำหรับเก็บงาน Designed ของแต่ละรูป และเก็บไฟล์ไว้ในโฟลเดอร์เดิม

ไฟล์เก็บชิ้นงาน (Designed file) ให้ใช้ชื่อ *****.VHD (***** ดูคำอธิบายของในแต่ละรูป)

b) คอมไพล์และจำลองการทำงานในโหมด **Functional** เปรียบเทียบกับผลที่ได้จากการออกแบบโดยใช้เกทพื้นฐาน

```
1  --VHDL code for a 4-to-1 multiplexer.
2
3  LIBRARY ieee ;
4  USE ieee.std_logic_1164.all ;
5
6  ENTITY Multiplex4t1 IS
7  =   PORT ( w0, w1, w2, w3 : IN STD_LOGIC ;
8         s      : IN STD_LOGIC_VECTOR(1 DOWNTO 0) ;
9         f      : OUT STD_LOGIC ) ;
10 END Multiplex4t1 ;
11
12 ARCHITECTURE Behavior OF Multiplex4t1 IS
13 =   BEGIN
14     WITH s SELECT
15         f <=      w0 WHEN "00" ,
16                 w1 WHEN "01" ,
17                 w2 WHEN "10" ,
18                 w3 WHEN OTHERS ;
19 END Behavior ;
```

ชื่ออุปกรณ์ : 4-to-1 Multiplexer

ชื่อโปรเจกต์ : Multiplex4t1

ชื่อไฟล์ : Multiplex4t1.vhd

ให้เปรียบเทียบผลการจำลองกับรูปที่ 4

รูปที่ 14



```
1  --VHDL code for a 2-to-4 binary decoder
2
3  LIBRARY ieee ;
4  USE ieee.std_logic_1164.all ;
5
6  ENTITY decoder2t4 IS
7  =   PORT ( w : IN STD_LOGIC_VECTOR(1 DOWNTO 0) ;
8          En : IN STD_LOGIC ;
9          y : OUT STD_LOGIC_VECTOR(0 TO 3) ) ;
10     END decoder2t4 ;
11
12  ARCHITECTURE Behavior OF decoder2t4 IS
13  SIGNAL Enw : STD_LOGIC_VECTOR(2 DOWNTO 0) ;
14  =   BEGIN
15      Enw <= En & w ;
16      WITH Enw SELECT
17          y <=      "1000" WHEN "100" ,
18                  "0100" WHEN "101" ,
19                  "0010" WHEN "110" ,
20                  "0001" WHEN "111" ,
21                  "0000" WHEN OTHERS ;
22  END Behavior ;
```

ชื่ออุปกรณ์ : 2-to-4 Binary Decoder

ชื่อโปรเจกต์ : decoder2t4

ชื่อไฟล์ : decoder2t4.vhd

ให้เปรียบเทียบผลการจำลองกับรูปที่ 6

รูปที่ 15

```
1  --VHDL code for a Demultiplexer 1-to-4
2
3  LIBRARY ieee ;
4  USE ieee.std_logic_1164.all ;
5
6  ENTITY demultiplex1t4 IS
7  =   PORT ( x : IN STD_LOGIC ;
8          s : IN STD_LOGIC_VECTOR(1 DOWNTO 0) ;
9          y0, y1 : OUT STD_LOGIC ;
10         y2, y3 : OUT STD_LOGIC ) ;
11     END demultiplex1t4 ;
12
13  ARCHITECTURE Behavior OF demultiplex1t4 IS
14  =   BEGIN
15      y0 <= x WHEN s="00" ELSE '0';
16      y1 <= x WHEN s="01" ELSE '0';
17      y2 <= x WHEN s="10" ELSE '0';
18      y3 <= x WHEN s="11" ELSE '0';
19  END Behavior ;
```

ชื่ออุปกรณ์ : 1-to-4 Demultiplexer

ชื่อโปรเจกต์ : demultiplex1t4

ชื่อไฟล์ : demultiplex1t4.vhd

ให้เปรียบเทียบผลการจำลองกับรูปที่ 8

รูปที่ 16

```
1  --VHDL code for a priority encoder
2
3  LIBRARY ieee ;
4  USE ieee.std_logic_1164.all ;
5
6  ENTITY priority IS
7  =   PORT ( w : IN STD_LOGIC_VECTOR(3 DOWNTO 0) ;
8          y : OUT STD_LOGIC_VECTOR(1 DOWNTO 0) ;
9          z : OUT STD_LOGIC ) ;
10     END priority ;
11
12  ARCHITECTURE Behavior OF priority IS
13  =   BEGIN
14      y <=      "11" WHEN w(3) = '1' ELSE
15              "10" WHEN w(2) = '1' ELSE
16              "01" WHEN w(1) = '1' ELSE
17              "00" ;
18
19      z <= '0' WHEN w = "0000" ELSE '1' ;
20  END Behavior ;
```

ชื่ออุปกรณ์ : 4-to-2 Priority Encoder

ชื่อโปรเจกต์ : priority

ชื่อไฟล์ : priority.vhd

ให้เปรียบเทียบผลการจำลองกับรูปที่ 10

รูปที่ 17



```

1  -- VHDL code for a four-bit comparator (unsigned)
2
3  LIBRARY ieee ;
4  USE ieee.std_logic_1164.all ;
5  USE ieee.std_logic_unsigned.all ;
6
7  ENTITY compare IS
8  =   PORT ( A, B : IN   STD_LOGIC_VECTOR(3 DOWNTO 0) ;
9        AeqB : OUT  STD_LOGIC ;
10       AgtB : OUT  STD_LOGIC ;
11       AltB : OUT  STD_LOGIC ) ;
12      END compare ;
13
14  ARCHITECTURE Behavior OF compare IS
15  =   BEGIN
16       AeqB <= '1' WHEN A = B ELSE '0' ;
17       AgtB <= '1' WHEN A > B ELSE '0' ;
18       AltB <= '1' WHEN A < B ELSE '0' ;
19  END Behavior ;

```

ชื่ออุปกรณ์ : 4-bit Comparator
ชื่อโปรเจกต์ : compare
ชื่อไฟล์ : compare.vhd
ให้เปรียบเทียบผลการจำลองกับรูปที่ 12

รูปที่ 18

บันทึกของอาจารย์ผู้ควบคุม

โปรแกรม VHDL	ชื่ออุปกรณ์ที่ออกแบบ	ผลการทดลอง (ผ่าน/ไม่ผ่าน)	ลายเซ็นอาจารย์ผู้ควบคุม
รูปที่ 14 เทียบกับรูปที่ 4			
รูปที่ 15 เทียบกับรูปที่ 6			
รูปที่ 16 เทียบกับรูปที่ 8			
รูปที่ 17 เทียบกับรูปที่ 10			
รูปที่ 18 เทียบกับรูปที่ 12			

งานมอบหมายท้ายการทดลอง

(ให้เขียนลงบนกระดาษ A4 ที่มีเส้นบรรทัดแนบท้ายเอกสารการทดลองส่งคร่าวถัดไป)

- ให้ทำรายงานสรุป เปรียบเทียบการออกแบบสร้างอุปกรณ์ดิจิทัลที่ใช้ในการทดลอง ระหว่างวิธีสร้างด้วยอุปกรณ์ลอจิกเกต และวิธีสร้างด้วยภาษา VHDL ดังนี้
 - ชุดคำสั่ง/โครงสร้างภาษา VHDL ที่ทำพฤติกรรมเป็นอุปกรณ์ประเภท Multiplexer
 - ชุดคำสั่ง/โครงสร้างภาษา VHDL ที่ทำพฤติกรรมเป็นอุปกรณ์ประเภท Decoder
 - ชุดคำสั่ง/โครงสร้างภาษา VHDL ที่ทำพฤติกรรมเป็นอุปกรณ์ประเภท Demultiplexer
 - ชุดคำสั่ง/โครงสร้างภาษา VHDL ที่ทำพฤติกรรมเป็นอุปกรณ์ประเภท Encoder
 - ชุดคำสั่ง/โครงสร้างภาษา VHDL ที่ทำพฤติกรรมเป็นอุปกรณ์ประเภท Priority Encoder
 - ชุดคำสั่ง/โครงสร้างภาษา VHDL ที่ทำพฤติกรรมเป็นอุปกรณ์ประเภท Comparator