



ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

ภาคการศึกษาที่.....1.....ปีการศึกษา.....2564

รหัสวิชา 010113026 ชื่อวิชา Digital Laboratory

ตอนเรียน .....7..... หมายเลขโต๊ะ.....-

รหัสนักศึกษา.....62010113026.....

ชื่อ-นามสกุล.....นายโสมภพ สุขสมบูรณ์.....

อาจารย์ผู้สอน.....CSP.....

เวลาทำการทดลอง THU: 13.00-16.00 วันที่ 2/ก.ย./64

## การทดลองที่ 6

### Latch and Flip-Flops

#### วัตถุประสงค์

1. เพื่อให้สามารถใช้โปรแกรมคอมพิวเตอร์จำลองการทำงานของวงจรลอจิกเกตได้
2. เพื่อให้เข้าใจพื้นฐานของอุปกรณ์ที่มีความจำ และ อุปกรณ์ฟลิป-ฟลอป
3. เพื่อให้เข้าใจพื้นฐานของ D Flip-Flop , T Flip-Flop , JK Flip-Flop

#### อุปกรณ์

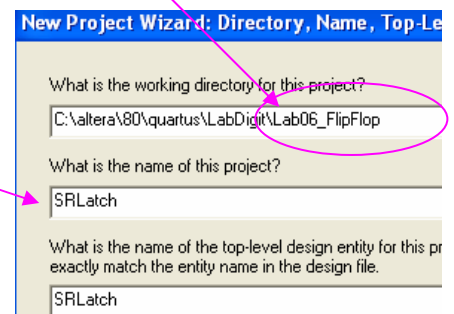
1. ระบบคอมพิวเตอร์ 1 เครื่อง พร้อมติดตั้งโปรแกรม Quartus II เวอร์ชัน 8.0 (Student Edition) ขึ้นไป

#### การทดลองตอนที่ 1

##### วงจร Latch

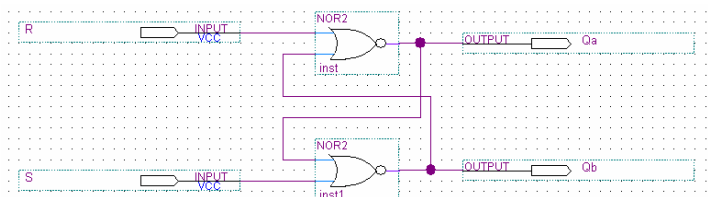
#### คำสั่งการทดลอง

1. ให้ น.ศ. สร้างโปรเจกต์สำหรับเก็บงานชิ้นใหม่เพื่อเก็บงานในการทดลองนี้ชื่อ “Lab06\_FlipFlop” จากนั้นให้สร้างโปรเจกต์ชื่อ “SRLatch” ขึ้นดังรูปที่ 1 และใช้ชิพเบอร์ EP3C10E144C8



รูปที่ 1

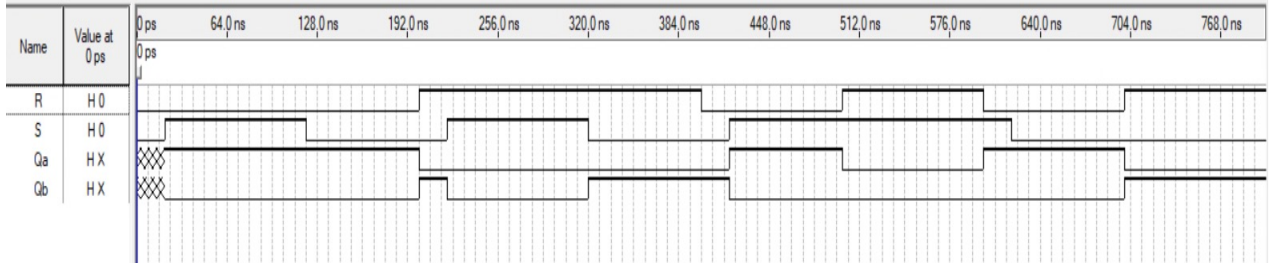
เขียนวงจรทดลองดังรูปที่ 2 ด้วย Graphic Editor Tool ของโปรแกรม Quartus II และทำการคอมไพล์ให้เรียบร้อย จากนั้นให้ทำการสร้าง symbol ของวงจรขึ้นมาเตรียมไว้ใช้ในขั้นถัดไป



รูปที่ 2

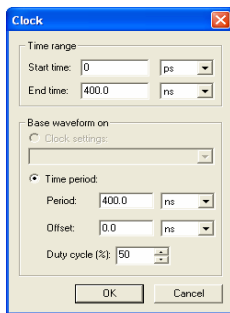


2. สร้างไฟล์แสดงแผนภาพทางเวลา (Timing diagram) ในรูปที่ 3 (ดูวิธีสร้างจากรูปที่ 4) โดยตั้งค่าสำหรับการแสดงผลจำลองการทำงานมีค่า End Time = 800 ns Grid Size = 1 ns แล้วจำลองการทำงานโหมด Functional mode บันทึกผลที่ได้ลงในรูปที่ 3

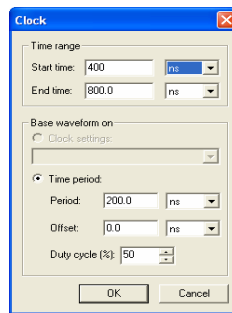


## วิธีสร้างรูปคลื่นของ R และ S

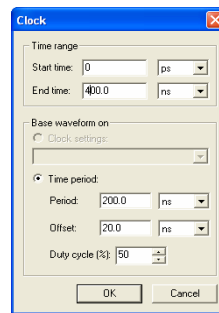
- ก) กำหนดอินพุต R ให้เป็นแบบสัญญาณนาฬิกา ในช่วง 0 – 400 ns ดังรูปที่ 4a  
ในช่วง 400 – 800 ns ดังรูปที่ 4b
- ข) กำหนดอินพุต S ให้เป็นแบบสัญญาณนาฬิกา ในช่วง 400 – 800 ns ดังรูปที่ 4C  
ในช่วง 400 – 800 ns ดังรูปที่ 4d และกลับเฟส กราฟทั้งรูป (ใช้เครื่องมือ inv)



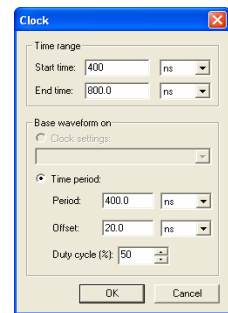
a



b



c



d

รูปที่ 4

## ผลการทดลอง

ให้สังเกตทุกจุดของอินพุต Qa Qb ที่มีการเปลี่ยนแปลง

- ก) เมื่อ R = '0' S = '1' จุดที่ 1 ในช่วงเวลา t = 20 ns ได้ค่า Qa = 1 Qb = 0  
จุดที่ 2 ในช่วงเวลา t = 420 ns ได้ค่า Qa = 1 Qb = 0  
จุดที่ 3 ในช่วงเวลา t = 600 ns ได้ค่า Qa = 1 Qb = 0
- ข) เมื่อ R = '0' S = '0' จุดที่ 1 ในช่วงเวลา t = 0 ns ได้ค่า Qa = X Qb = X  
จุดที่ 2 ในช่วงเวลา t = 120 ns ได้ค่า Qa = 1 Qb = 0  
จุดที่ 3 ในช่วงเวลา t = 400 ns ได้ค่า Qa = 0 Qb = 1  
จุดที่ 4 ในช่วงเวลา t = 620 ns ได้ค่า Qa = 1 Qb = 0
- ค) เมื่อ R = '1' S = '0' จุดที่ 1 ในช่วงเวลา t = 200 ns ได้ค่า Qa = 0 Qb = 1  
จุดที่ 2 ในช่วงเวลา t = 320 ns ได้ค่า Qa = 0 Qb = 1
- ง) เมื่อ R = '1' S = '1' จุดที่ 1 ในช่วงเวลา t = 220 ns ได้ค่า Qa = 0 Qb = 0  
จุดที่ 2 ในช่วงเวลา t = 500 ns ได้ค่า Qa = 0 Qb = 0



# การทดลองที่ 6 Latch and Flip-Flops

หน้า  
3 / 9

จากความสัมพันธ์กันระหว่าง S, R, Qa และ Qb ให้อธิบายสรุปหลักการทำงาน หรือคุณสมบัติของวงจร SR Latch

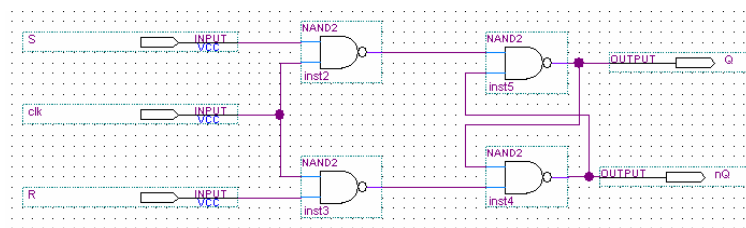
เมื่อ R และ S สลับกันเท่ากับ 0 ค่า Output จะคงที่ Logic ถัดหน้าไว้, เมื่อ R = '1' และ S = '1' ค่า Output จะเปลี่ยน 0 เป็น 1, R = '1' และ S = '0' ค่า Output Reset ค่า Qa = 0, Qb = 1 และเมื่อ R = '0' และ S = '1' จะเกิดการ Set ค่า Qa = 1 และ Qb = 0

S	R	Qa	Qb
0	0	0 / 1	0 / 1
0	1	0	1
1	0	1	0
1	1	0	0

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

3. ให้ปิดโปรเจกต์ที่สร้างมาในขั้นตอนที่ 1 – 2 ก่อนที่จะทำการทดลองต่อไป
4. สร้างโปรเจกต์ชื่อ “GatedSRLatch” ขึ้นมาและให้เก็บไว้ในโฟลเดอร์เดิมจากนั้นสร้างไฟล์ชื่อเดียวกันกับโปรเจกต์สำหรับเก็บงานวงจรในรูปที่ 5 ใช้ชิพ EP3C10E144C8 แล้วทำการคอมไพล์ให้เรียบร้อย

รูปที่ 5



5. สร้างไฟล์แสดงแผนภาพทางเวลา (Timing diagram) ให้ตั้งค่าแสดงผลจำลองการทำงานดังนี้

End Time = 0.5 us      Grid Size = 1 ns

สัญญาณอินพุต      clk ให้เป็นแบบนาฬิกา period 100 ns, offset 0

R ให้เป็นแบบนาฬิกา period 300 ns, offset 15 ns

S ให้เป็นแบบนาฬิกา period 200 ns, offset 15 ns

จำลองการทำงานโหมด “Functional mode” บันทึกผลที่ได้ลงในรูปที่ 6



รูปที่ 6

หมายเหตุ 1. สัญลักษณ์ ในกราฟของ Quartus II หมายถึงการไม่ทราบค่า (unknown)  
2. การอ่านค่ากราฟในจุดที่มีการเปลี่ยนค่าจะอ่านแบบกราฟมีค่าต่อเนื่องทางขวา

## ผลการทดลอง

ให้สังเกตทุกจุดของเอาต์พุต Q และ nQ (not Q,  $\bar{Q}$ ) ที่มีการเปลี่ยนแปลง

ก) ที่ t = 15 ns	เมื่อ clk = '0'	R = '0'	S = '0'	ค่าเอาต์พุต Q =	X	nQ =	X
ที่ t = 50 ns	เมื่อ clk = '1'	R = '0'	S = '0'	ค่าเอาต์พุต Q =	X	nQ =	X
ที่ t = 100 ns	เมื่อ clk = '0'	R = '0'	S = '0'	ค่าเอาต์พุต Q =	X	nQ =	X
ที่ t = 115 ns	เมื่อ clk = '0'	R = '0'	S = '1'	ค่าเอาต์พุต Q =	X	nQ =	X
ที่ t = 150 ns	เมื่อ clk = '1'	R = '1'	S = '1'	ค่าเอาต์พุต Q =	1	nQ =	0
ที่ t = 165 ns	เมื่อ clk = '1'	R = '1'	S = '1'	ค่าเอาต์พุต Q =	1	nQ =	1



ที่ t = 200 ns	เมื่อ clk = '0'	R = '1'	S = '1'	ค่าเข้าที่พืท Q =	1	nQ =	0
ที่ t = 215 ns	เมื่อ clk = '0'	R = '1'	S = '0'	ค่าเข้าที่พืท Q =	1	nQ =	0
ที่ t = 250 ns	เมื่อ clk = '1'	R = '1'	S = '0'	ค่าเข้าที่พืท Q =	0	nQ =	1
ที่ t = 300 ns	เมื่อ clk = '0'	R = '1'	S = '0'	ค่าเข้าที่พืท Q =	0	nQ =	1
ที่ t = 315 ns	เมื่อ clk = '0'	R = '0'	S = '1'	ค่าเข้าที่พืท Q =	0	nQ =	1
ที่ t = 350 ns	เมื่อ clk = '1'	R = '0'	S = '1'	ค่าเข้าที่พืท Q =	1	nQ =	0
ที่ t = 400 ns	เมื่อ clk = '0'	R = '0'	S = '1'	ค่าเข้าที่พืท Q =	1	nQ =	0
ที่ t = 415 ns	เมื่อ clk = '0'	R = '0'	S = '0'	ค่าเข้าที่พืท Q =	1	nQ =	0
ที่ t = 450 ns	เมื่อ clk = '1'	R = '0'	S = '0'	ค่าเข้าที่พืท Q =	1	nQ =	0
ที่ t = 465 ns	เมื่อ clk = '1'	R = '0'	S = '1'	ค่าเข้าที่พืท Q =	0	nQ =	1

ข) จากผลการทดลองข้างต้นสามารถสรุปเป็นหลักการทํางาน หรือคุณ

สมบัติของวงจร Gated SR Latch ได้คือ เมื่อ clk = '0' ไม่ว่า R & S จะมีค่า

เท่าใด Output จะยังคงเดิมจากค่าก่อนหน้า และเมื่อ clk = '1'

และเมื่อ R = '0' & S = '0' ค่า Output จะยังคงเดิม , R = '1' & S = '0'

ค่า Output Q = '0' (Reset) , R = '0' , S = '1' ค่า Output Q = '1' (Set)

และเมื่อ R & S มีค่า '1' แล้ว Q & nQ = '1'

Clk	S	R	Q(t+1)
0	x	x	Q(t)
1	0	0	Q(t)
1	0	1	0
1	1	0	1
1	1	1	x

ค) สถานะ Q(t+1) ในตารางความจริง มีความหมายว่าอย่างไร

สถานะถัดไปของ Output

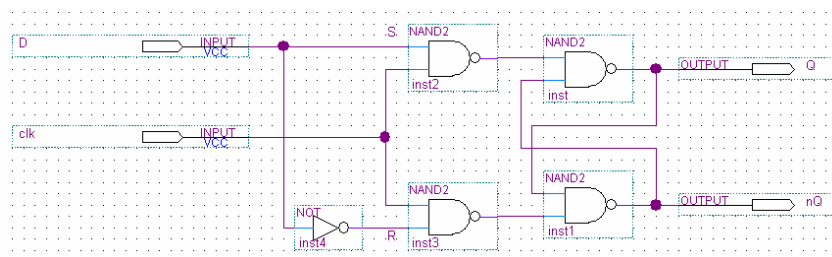
ลายเซ็นอาจารย์ผู้ควบคุม..... /...../.....

6. ให้ปิดโปรเจกต์ที่สร้างในขั้นตอน 4-5 ก่อนที่จะทำการทดลองต่อไป

7. สร้างโปรเจกต์ชื่อ “GatedDLatch” ขึ้นมาและให้เก็บไว้ในโฟลเดอร์เดียวกันกับโปรเจกต์ที่สร้างตอนก่อนหน้า

จากนั้นสร้างวงจรทดลองดังรูปที่ 7 ให้ใช้ชิพ EP3C10E144C8

8. ทำการคอมไพล์ให้เรียบร้อยและสร้าง Symbol file เพื่อเตรียมใช้งานในขั้นตอนถัดไป



รูปที่ 7

9. สร้างไฟล์แสดงแผนภาพทางเวลา ดังรูปที่ 8 กำหนดค่าสำหรับแสดงผลจำลองการทำงานโดยให้

End Time = 0.5 us Grid Size = 1 ns

สัญญาณอินพุต clk ให้เป็นแบบนาฬิกา period 100 ns, offset 0

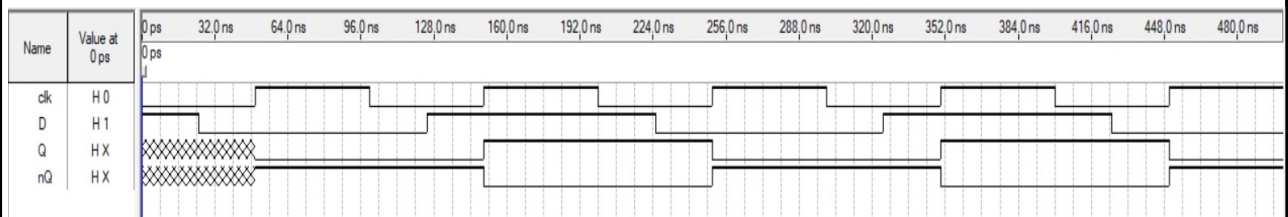
D ให้เป็นแบบนาฬิกา period 200 ns, offset 25,

จำลองการทำงานโหมด Functional mode บันทึกผลที่ได้ลงในรูปที่ 8



# การทดลองที่ 6 Latch and Flip-Flops

หน้า  
5 / 9



รูปที่ 8

ผลการทดลอง ให้สังเกตทุกจุดของเอาต์พุต Q และ nQ ( not Q ,  $\bar{Q}$  ) ที่มีการเปลี่ยนแปลง

ก)	ที่ t = 25 ns	เมื่อ clk = '0' D = '0'	ค่าเอาต์พุต Q = X	nQ = X
	ที่ t = 50 ns	เมื่อ clk = '1' D = '0'	ค่าเอาต์พุต Q = 0	nQ = 1
	ที่ t = 100 ns	เมื่อ clk = '0' D = '0'	ค่าเอาต์พุต Q = 0	nQ = 1
	ที่ t = 125 ns	เมื่อ clk = '0' D = '1'	ค่าเอาต์พุต Q = 0	nQ = 1
	ที่ t = 150 ns	เมื่อ clk = '1' D = '1'	ค่าเอาต์พุต Q = 1	nQ = 0
	ที่ t = 200 ns	เมื่อ clk = '0' D = '1'	ค่าเอาต์พุต Q = 1	nQ = 0
	ที่ t = 225 ns	เมื่อ clk = '0' D = '0'	ค่าเอาต์พุต Q = 1	nQ = 0
	ที่ t = 250 ns	เมื่อ clk = '1' D = '0'	ค่าเอาต์พุต Q = 0	nQ = 1
	ที่ t = 300 ns	เมื่อ clk = '0' D = '0'	ค่าเอาต์พุต Q = 0	nQ = 1
	ที่ t = 325 ns	เมื่อ clk = '0' D = '1'	ค่าเอาต์พุต Q = 0	nQ = 1
	ที่ t = 350 ns	เมื่อ clk = '1' D = '1'	ค่าเอาต์พุต Q = 1	nQ = 0
	ที่ t = 400 ns	เมื่อ clk = '0' D = '1'	ค่าเอาต์พุต Q = 1	nQ = 0

ข) เอาต์พุต Q จะเปลี่ยนค่าตามอินพุต D ก็ต่อเมื่อ clk มีค่าเท่ากับ 1

จากผลการทดลองข้างต้นสามารถสรุปเป็นหลักการทำงานหรือคุณสมบัติของวงจร Gated D Latch ได้ดังนี้

เมื่อ clk = '1' output Q จะสลับค่าเท่ากับ Input D  
และเมื่อ clk = '0' output Q: ถ้าสมมุติเดิมจากสมมุติก่อนหน้า

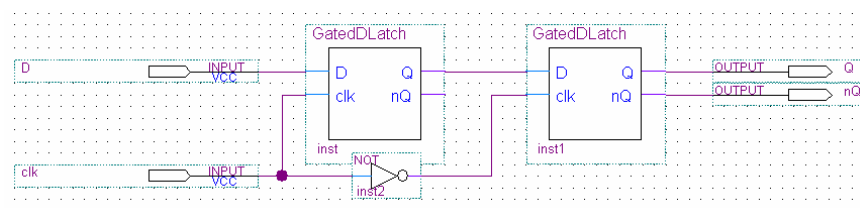
clk	D	Q(t+1)
0	x	Q(t)
1	0	0
1	1	1

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

## การทดลองตอนที่ 2 วงจร Flip-Flop

10. ให้ปิดโปรเจกต์ที่สร้างมาในขั้นตอนที่ 7-9 ก่อนที่จะทำการทดลองต่อไป

11. สร้างโปรเจกต์ชื่อ “MasterSlaveDFF” ขึ้นมาและให้เก็บไว้ในโฟลเดอร์เดิม ทำการสร้างวงจรในรูปที่ 9 โดยใช้อุปกรณ์ GatedDLatch จาก symbol file ที่ได้จากการทดลองในข้อ 7-9 ใช้ชิพ EP3C10E144C8 จากนั้นทำการคอมไพล์ให้เรียบร้อย



รูปที่ 9





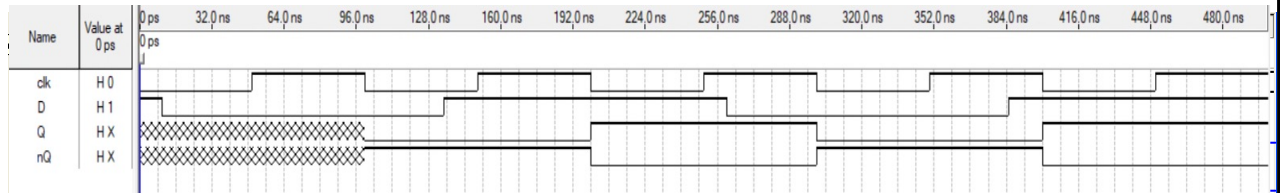
12. สร้างไฟล์แสดงแผนภาพทางเวลา กำหนดค่าสำหรับแสดงผลจำลองการทำงานให้มีค่า


End Time = 0.5 us      Grid Size = 1 ns

สัญญาณอินพุต      clk ให้เป็นแบบนาฬิกา period 100 ns, offset 0

D ให้เป็นแบบนาฬิกา period 250 ns, offset 10 ns,


จำลองการทำงานโหมด Functional mode บันทึกผลที่ได้ลงในรูปที่ 10




หมายเหตุ : 1. สัญลักษณ์ '↑' หมายถึง มีการเปลี่ยนค่า จาก 0→1 ส่วน '↓' เปลี่ยนจาก 1→0  
2. สัญลักษณ์  ในกราฟของ Quartus II หมายถึงการไม่ทราบค่า (unknown)

รูปที่ 10

สังเกตผลการทดลอง

ก) ที่ t = 10 ns	เมื่อ clk = '0' D = '0'	ค่าเอาต์พุต Q = 	nQ = 
ที่ t = 50 ns	เมื่อ clk = '↑' D = '0'	ค่าเอาต์พุต Q = 	nQ = 
ที่ t = 100 ns	เมื่อ clk = '↓' D = '0'	ค่าเอาต์พุต Q = 0	nQ = 1
ที่ t = 135 ns	เมื่อ clk = '0' D = '1'	ค่าเอาต์พุต Q = 0	nQ = 1
ที่ t = 150 ns	เมื่อ clk = '↑' D = '1'	ค่าเอาต์พุต Q = 0	nQ = 1
ที่ t = 200 ns	เมื่อ clk = '↓' D = '1'	ค่าเอาต์พุต Q = 1	nQ = 0
ที่ t = 250 ns	เมื่อ clk = '↑' D = '1'	ค่าเอาต์พุต Q = 1	nQ = 0
ที่ t = 260 ns	เมื่อ clk = '1' D = '0'	ค่าเอาต์พุต Q = 1	nQ = 0
ที่ t = 300 ns	เมื่อ clk = '↓' D = '0'	ค่าเอาต์พุต Q = 0	nQ = 1

ข) เอาต์พุต Q เปลี่ยนเป็นค่าเดียวกันกับอินพุต D เมื่อ clk =  หรือ ขอบาลงของ Clk

ค) จากผลการทดลอง ให้สรุปหลักการทำงานหรือคุณสมบัติ ของวงจร Master Slave D Flip-Flop ได้ดังนี้

เมื่อ clk = '↑' Output จะคงค่าเดิมไว้ , clk = '0' & '1' ค่า Output จะไม่เกิดการเปลี่ยนแปลงใดๆ และเมื่อ clk = '↓' Output Q จะรับค่าเท่ากับค่าของ Input D

clk	D	Q(t+1)
↑	X	Q(t)
0	0	0
↓	1	1

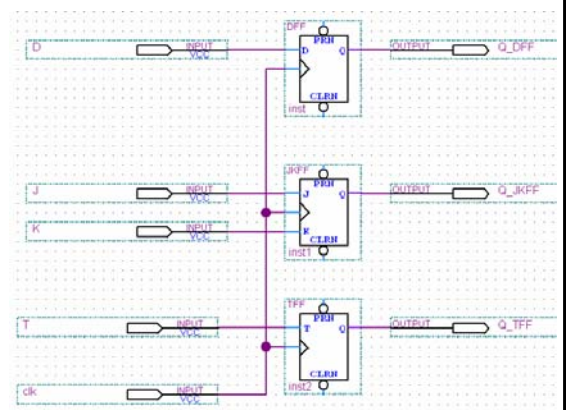
ลายเซ็นอาจารย์ผู้ควบคุม..... /...../.....

13. ให้ปิดโปรเจกต์ที่สร้างมาในขั้นตอนที่ 10-12

14. สร้างโปรเจกต์ชื่อ "ALL\_FLIP\_FLOP" เก็บไว้ในโฟลเดอร์  
เดิมสร้างไฟล์ชื่อเดียวกันกับโปรเจกต์ขึ้นมาเก็บวงจรทดลอง  
ในรูปที่ 11 คอมไพล์ให้เรียบร้อย

ในไลบรารีของ Quartus II  
อุปกรณ์ D Flip-Flop ใช้ชื่อเป็น dff  
อุปกรณ์ JK Flip-Flop ใช้ชื่อเป็น jkff  
อุปกรณ์ T Flip-Flop ใช้ชื่อเป็น tff

รูปที่ 11





## การทดลองที่ 6 Latch and Flip-Flops

หน้า  
7 / 9

15. สร้างไฟล์แสดงแผนภาพทางเวลา กำหนดค่าสำหรับแสดงผลจำลองการทำงานโดยให้มีค่าดังกราฟในรูปที่ 12

End Time = 1.0 us      Grid Size = 1 ns

สัญญาณอินพุต clk ให้เป็นแบบนาฬิกา period 100 ns, offset 0 ns

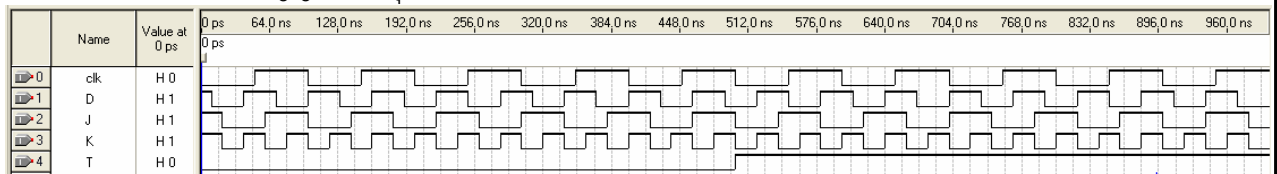
D ให้เป็นแบบนาฬิกา period 60 ns, offset 10 ns

J ให้เป็นแบบนาฬิกา period 80 ns, offset 20 ns

K ให้เป็นแบบนาฬิกา period 40 ns, offset 20 ns

T ให้เป็นแบบนาฬิกา period 1000 ns, offset 0 ns

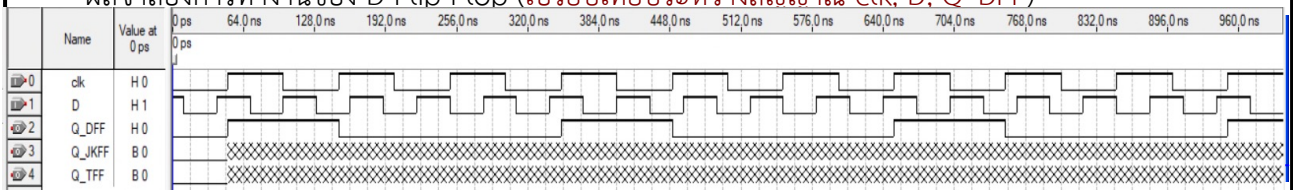
กราฟแสดงค่าสัญญาณอินพุต clk, D, J, K, T



รูปที่ 12

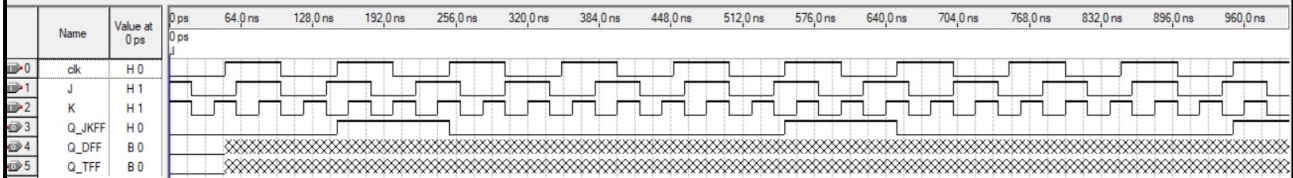
16 จำลองการทำงานโหมด Functional mode บันทึกผลที่ได้ลงในรูปที่ 13 -15

ผลจำลองการทำงานของ D Flip-Flop (เปรียบเทียบระหว่างสัญญาณ clk, D, Q DFF)



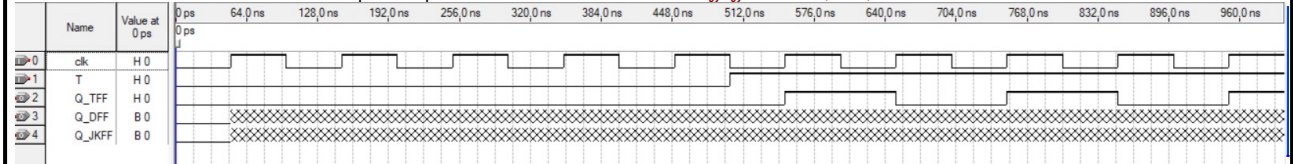
รูปที่ 13

ผลจำลองการทำงานของ JK Flip-Flop (เปรียบเทียบระหว่างสัญญาณ clk, J, K, Q JKFF)



รูปที่ 14

ผลจำลองการทำงานของ T Flip-Flop (เปรียบเทียบระหว่างสัญญาณ clk, T, Q TFF)



รูปที่ 15

ผลการทดลอง      หมายเหตุ : ให้ใช้สัญลักษณ์ '↑' แทนการ มีการเปลี่ยนค่า จาก 0→1 ส่วน '↓' เปลี่ยนจาก 1→0 ในการเขียนตารางความจริง

ก) จากรูปที่ 13 ค่าของอินพุต D ไปปรากฏที่เอาต์พุต Q\_DFF เมื่อ clk มีค่า..... ↑ (ของขาขึ้น)

ข) จากผลการทดลอง สรุปหลักการทำงาน (คุณสมบัติ) หรือสมการ หรือ ตารางความจริงของวง

จร Positive Edge Trig D Flip-Flop ได้ดังนี้      เมื่อ clk = '↑' Output Q จ. ค่า

เดือขาขึ้น Input D      และเมื่อ clk = '↓' Output Q จ. ค่าตกข:      ได้ม ไร่

clk	D	Q(t+1)
↓	x	Qch
↑	0	0
↑	1	1



## การทดลองที่ 6 Latch and Flip-Flops

หน้า  
8 / 9

ค) จากรูปที่ 14 ค่าของเอาต์พุต Q\_JKFF จะเกิดการเปลี่ยนแปลงเมื่อ clk มีค่า  $\uparrow$  (ของทรีน)

ง) ให้เขียนความสัมพันธ์กันระหว่างสัญญาณ J K และ Q\_JKFF โดยเขียนเป็น  
สมการหรือ ตารางความจริง เมื่อ  $clk = '1'$  และ  $J = '0'$  &  $K = '0'$  Output  
จะคงที่เดิม,  $J = '0'$  &  $K = '1'$  Output Q จะมีค่า '1',  $J = '1'$ ,  $K = '0'$   
Output Q จะมีค่า '1', J & K = '1' ค่า Output จะกลับค่าเดิม Logic don't care  
และเมื่อ  $clk = '0'$  Output จะคงที่เดิมไว้

clk	J	K	Q(t+1)
↓	x	x	Q(t)
↑	0	0	Q(t)
↑	0	1	0
↑	1	0	1
↑	1	1	Q(t)

จ) จากรูปที่ 15 ค่าของเอาต์พุต Q\_TFF จะเกิดการเปลี่ยนแปลงเมื่อ clk มีค่า  $\uparrow$  (ของทรีน)

ถ้าสัญญาณ T = '0' ค่าของเอาต์พุต Q\_TFF จะมีค่า  $\uparrow$  คงที่เดิม

ถ้าสัญญาณ T = '1' ค่าของเอาต์พุต Q\_TFF จะมีการเปลี่ยนแปลงอย่างไร ถ้า  $clk = '1'$  Output  
จะมีค่าตรงข้ามกับสถานะเดิม

ถ้าสัญญาณ T = '1' ค่าของเอาต์พุต Q\_TFF จะมีค่าของเวลาใน 1 คาบ  
คิดเป็นจำนวน 2 เท่าเมื่อเทียบกับ clk

ความถี่ของ Q\_TFF คิดเป็นจำนวน 0.5 เท่าเมื่อเทียบกับ clk

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



### การทดลองตอนที่ 3 สร้างวงจร Flip-Flop ด้วยภาษา VHDL

17. ให้ปิดโปรเจกต์ที่สร้างมาในขั้นตอนที่ 10-16 ก่อนที่จะทำการทดลองต่อไป

18. เขียนภาษา VHDL ดังในรูปที่ 16 และ 18

a) ให้สร้างโปรเจกต์ขึ้นใหม่ สำหรับเก็บงาน Designed ของแต่ละรูป และเก็บไฟล์ไว้ในโฟลเดอร์เดิม  
ไฟล์เก็บชิ้นงาน (Designed file) ให้ใช้ชื่อ \*\*\*\*\*.VHD (ดูคำอธิบายในแต่ละรูป)

b) ทำการคอมไพล์ และจำลองการทำงานในโหมด  
Functional Mode ดังรูปที่ 17 และรูปที่ 19

c) เปรียบเทียบกับผลที่ได้จากการออกแบบด้วยวิธีการ  
เขียนด้วยวงจรลอจิก (การทดลองข้อที่ 1-16)

d) แจ้งอาจารย์ผู้สอนเพื่อตรวจผลการทดลอง  
ให้ครบทุกอุปกรณ์

ชื่ออุปกรณ์ : D Flip-Flop  
ชื่อโปรเจกต์ : DFF\_VHD  
ชื่อไฟล์ : DFF\_VHD.vhd

```

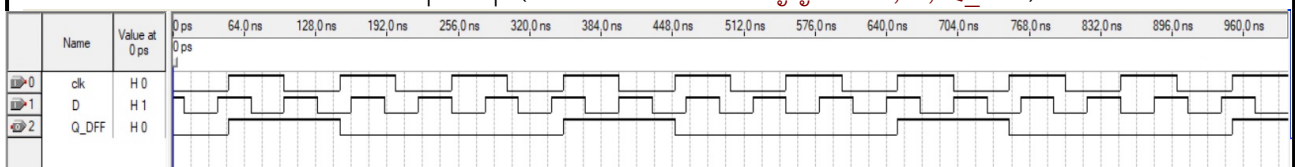
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY DFF_VHD IS
    PORT ( D, clk : IN      STD_LOGIC ;
          Q_DFF : OUT     STD_LOGIC);
END DFF_VHD ;

ARCHITECTURE Behavior OF DFF_VHD IS
    BEGIN
        PROCESS ( clk )
        BEGIN
            IF clk' EVENT AND clk = '1' THEN
                Q_DFF <= D ;
            END IF ;
        END PROCESS ;
    END Behavior ;
    
```

รูปที่ 16

ผลจำลองการทำงานของ D Flip-Flop (เปรียบเทียบระหว่างสัญญาณ clk, D, Q\_DFF)







ชื่ออุปกรณ์ : D Flip-Flop with asynchronous clear

ชื่อโปรเจกต์ : DFFasync\_VHD

ชื่อไฟล์ : DFFasync\_VHD.vhd

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

ENTITY DFFasync_VHD IS
PORT ( D, Resetn, Clock : IN STD_LOGIC ;
      Q : OUT STD_LOGIC ) ;
END DFFasync_VHD ;

ARCHITECTURE Behavior OF DFFasync_VHD IS
BEGIN
PROCESS ( Resetn, Clock )
BEGIN
IF Resetn = '0' THEN
Q <= '0' ;
ELSIF Clock'EVENT AND Clock = '1' THEN
Q <= D ;
END IF ;
END PROCESS ;
END Behavior ;
```

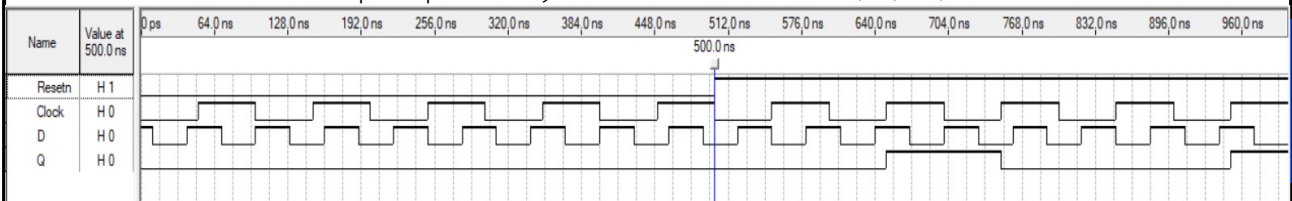
รูปที่ 18

## บันทึกผลการทดลอง

ก) เปรียบเทียบผลการทดลองในรูปที่ 13 และรูปที่ 17

จากผลการ Simulate ในโปรแกรมที่เขียนไว้  
นั่นคือ เมื่อ  $clk = '0'$  output จะลatchเอา  
 $clk = '1'$  output จะเปลี่ยนไปตามค่า Input D

ผลจำลองการทำงานของ D Flip-Flop with asynchronous clear ( clock, D, Q, Resetn)



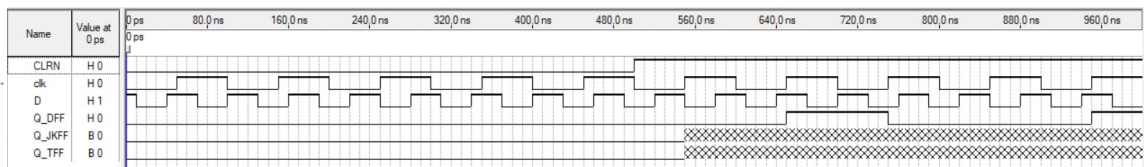
รูปที่ 19 (ให้กำหนดค่าของกราฟ D, clock เหมือนในรูปที่ 13)

ข) อุปกรณ์ D Flip-Flop ที่มีการเพิ่มขาควบคุมชื่อ Resetn (ขาควบคุมนี้ทำงานเป็น asynchronous clear) ดังรูปที่ 18 มีการทำงานที่แตกต่างจาก D Flip-Flop ในรูปที่ 16 อย่างไร ... เป็นขาที่ควบคุมที่ reset ค่า Output เป็น '0' โดยไม่สนใจ Clock จากรูปที่ 19 จะเห็นว่าในช่วง Resetn = '0' ไร้ clock ค่า output Q จะถูกรีเซ็ตเป็น '0' ตลอดเวลา

ค) ขาควบคุมที่ทำงานแบบ asynchronous มีการทำงานอย่างไร ... ควบคุมการแสดงผลของ Output ถ้ามีค่าของขา clock โดยไม่สนใจ clock หรือ Input อื่นๆ

ง) ขาควบคุมขาใดในอุปกรณ์ D Flip-Flop รูปที่ 11 มีหน้าที่เป็น asynchronous clear เช่นเดียวกันกับขา Resetn ของอุปกรณ์ในรูปที่ 18 ... CLRN

น.ศ. รู้ได้อย่างไร ให้อธิบายวิธีการพิสูจน์ว่าขาควบคุมที่ถูกกล่าวอ้างมานั้นทำหน้าที่เป็น asynchronous clear จริงๆ



## งานมอบหมายท้ายการทดลอง

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

(ให้เขียนลงบนกระดาษ A4 ที่มีเส้นบรรทัดและรวมใส่ท้ายเอกสารการทดลอง ส่งอาจารย์ผู้สอนในคราวถัดไป)

- 1) ให้สร้างอุปกรณ์ JK Flip-Flop โดยใช้ภาษา VHDL พร้อมทั้งพิสูจน์การทำงานด้วย timing diagram
- 2) ให้สร้างอุปกรณ์ T Flip-Flop โดยใช้ภาษา VHDL พร้อมทั้งพิสูจน์การทำงานด้วย timing diagram

# J-K Flip-flop (VHDL)

វិទ្យាស្ថានបច្ចេកទេស  
បច្ចេកទេស 620101631188 Lab. 7 CSP  
(Digital lab)

```

1  LIBRARY ieee;
2  USE ieee.std_logic_1164.all ;

3  ENTITY JKFF_VHD IS
4  PORT (J,K,CLK : IN STD_LOGIC ;
5        Q_JKFF : OUT STD_LOGIC) ;
6  END JKFF_VHD ;

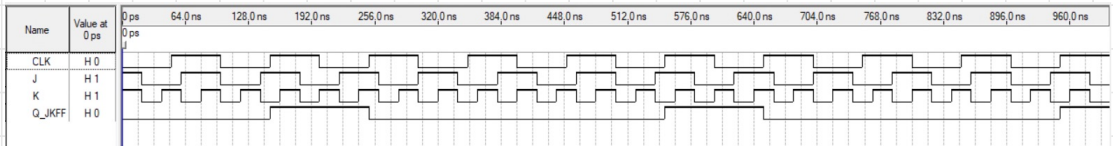
7  ARCHITECTURE Behavior OF JKFF_VHD IS
8  SIGNAL X : STD_LOGIC ;
9
10 BEGIN
11     PROCESS (CLK)
12     BEGIN
13         IF CLK'EVENT AND CLK='1' THEN
14             IF (J='0' AND K='0') THEN
15                 X <= X ;
16             ELSIF (J='0' AND K='1') THEN
17                 X <= '0' ;
18             ELSIF (J='1' AND K='0') THEN
19                 X <= '1' ;
20             ELSIF (J='1' AND K='1') THEN
21                 X <= NOT X ;
22             END IF ;
23         END IF ;
24     END PROCESS ;
25     Q_JKFF <= X ;
26 END Behavior ;

```

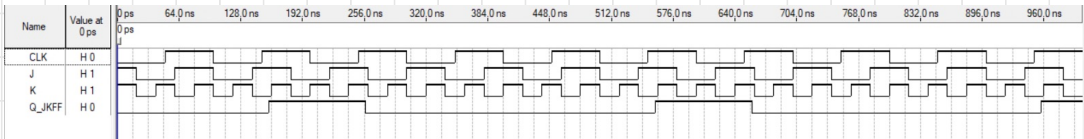
Flow Status	Successful - Thu Sep 02 02:47:53 2021
Quartus II Version	8.1 Build 163 10/28/2008 SJ Web Edition
Revision Name	JKFF_VHD
Top-level Entity Name	JKFF_VHD
Family	Cyclone III
Device	EP3C10E144C8
Timing Models	Final
Met timing requirements	N/A
Total logic elements	1 / 10,320 (< 1 %)
Total combinational functions	1 / 10,320 (< 1 %)
Dedicated logic registers	1 / 10,320 (< 1 %)
Total registers	1
Total pins	4 / 95 (4 %)
Total virtual pins	0
Total memory bits	0 / 423,936 (0 %)
Embedded Multiplier 9-bit elements	0 / 46 (0 %)
Total PLLs	0 / 2 (0 %)

(Compilation)

(VHDL code)



(functional mode)



(timing mode)

# T Flip-flop (VHDL)

```

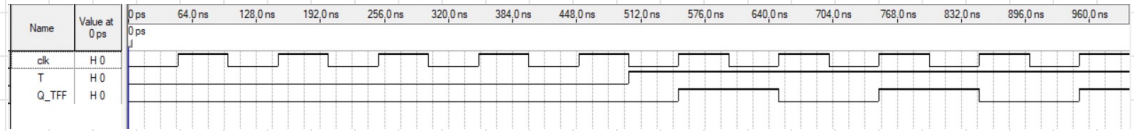
1  -- VHDL for T flip-flop --
2  LIBRARY ieee ;
3  USE ieee.std_logic_1164.all ;
4
5  ENTITY TFF_VHD IS
6  PORT ( T, clk : IN STD_LOGIC ;
7        Q_TFF : OUT STD_LOGIC) ;
8  END TFF_VHD ;
9
10 ARCHITECTURE Behavior OF TFF_VHD IS
11 SIGNAL X : STD_LOGIC ;
12
13 BEGIN
14   PROCESS (clk)
15   BEGIN
16     IF clk 'event and clk='1' THEN
17       IF T = '0' THEN
18         X <= X ;
19       ELSIF T = '1' THEN
20         X <= NOT X ;
21       END IF ;
22     END IF ;
23   END PROCESS ;
24   Q_TFF <= X ;
25 END Behavior ;
26

```

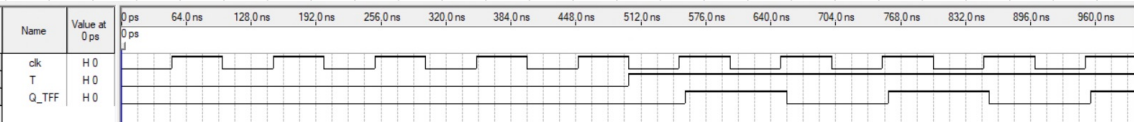
Flow Status	Successful - Thu Sep 02 02:49:50 2021
Quartus II Version	8.1 Build 163 10/28/2008 SJ Web Edition
Revision Name	TFF_VHD
Top-level Entity Name	TFF_VHD
Family	Cyclone III
Device	EP3C10E144C8
Timing Models	Final
Met timing requirements	N/A
Total logic elements	1 / 10,320 (< 1 %)
Total combinational functions	1 / 10,320 (< 1 %)
Dedicated logic registers	1 / 10,320 (< 1 %)
Total registers	1
Total pins	3 / 95 (3 %)
Total virtual pins	0
Total memory bits	0 / 423,936 (0 %)
Embedded Multiplier 9-bit elements	0 / 46 (0 %)
Total PLLs	0 / 2 (0 %)

(Compilation)

(VHDL code)



(functional mode)



(timing mode)