จุฬาลงกรณ์มหาวิทยาลัย
คณะวิศวกรรมศาสตร์
ภาควิชาวิศวกรรมคอมพิวเตอร์
2110-263 DIGITAL COMPUTER LOGIC LAB I

ชื่อ _____
เลขประจำตัว _____
หมายเลขเครื่อง _____
วันที่ _____

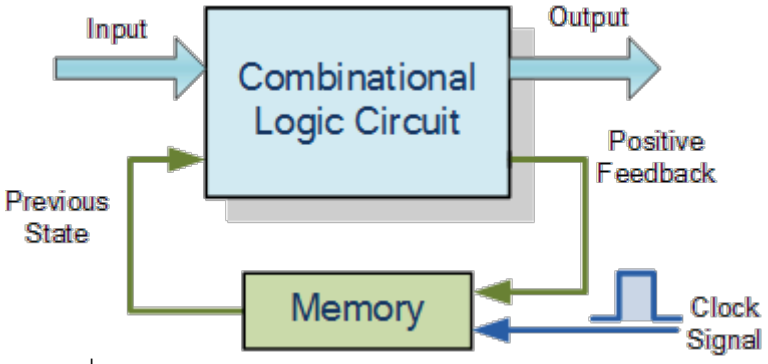
7. Latch และ Flip-flop

วัตถุประสงค์

1. เพื่อให้นิสิตเข้าใจโครงสร้าง Latch และ Flip-flop
2. เพื่อให้นิสิตเข้าใจโครงสร้าง Flip-flop ที่เป็นแบบ Edge Triggered และ Master/Slave
3. เพื่อให้นิสิตเข้าใจโครงสร้าง Flip-flop ชนิด S-R ,J-K , D และ T
4. เพื่อให้นิสิตได้ฝึกใช้ part สำเร็จรูปต่าง ๆ

บทนำ

Sequential Logic เป็นวงจร Logic ที่ผลลัพธ์ของวงจรขึ้นอยู่กับ input และ Memory ของตัววงจร ซึ่งแตกต่างจาก Combinational Logic ที่ผลลัพธ์ของวงจรนั้นจะขึ้นอยู่กับ input อย่างเดียว โดยวงจร Sequential Logic สามารถนำไปสร้าง Finite-state machines ซึ่งเป็นโครงสร้างพื้นฐานสำหรับวงจร Digital ทั้งหมดได้



รูปที่ 1 : โครงสร้างการทำงานของวงจร Sequential Logic อย่างง่าย

นอกจากนี้วงจร Sequential Logic สามารถแบ่งออกมาได้เป็น 2 ประเภทคือ

1. Synchronous เป็นวงจรที่ผลลัพธ์จะขึ้นอยู่กับ Memory ของวงจรอย่างเดียว
2. Asynchronous เป็นวงจรที่ผลลัพธ์จะเปลี่ยนแปลงทันทีตาม input และ Memory ของวงจร

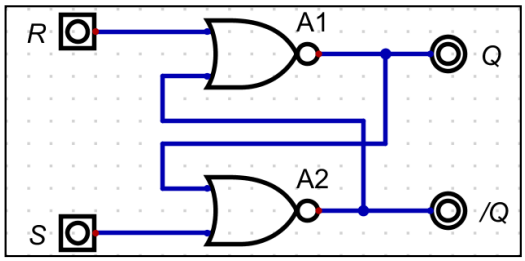
สำหรับปฏิบัติการครั้งนี้เราจะทำการทดลองเรื่อง Memory ของวงจร Sequential Logic โดยในการสร้าง Memory นั้นจะมี output 2 ตัวคือ Q และ /Q ซึ่งทั้งคู่จะมีค่าความจริงที่ตรงกันข้ามกัน และ Memory มีรูปแบบการสร้างอยู่หลากหลายวิธี อาทิเช่น

1. R-S Latch เป็น Memory ขั้นพื้นฐาน โดยสามารถแบ่งออกได้เป็น 2 ประเภทคือ

    1. NOR R-S Latch โดยสร้างมาจาก NOR Gate 2 อัน

Input		Output		Description
R	S	Q	/Q	
0	0	Q	/Q	Hold
0	1	1	0	Set
1	0	0	1	Reset
1	1	0	0	Forbidden

ตารางค่าความจริงของ NOR R-S Latch



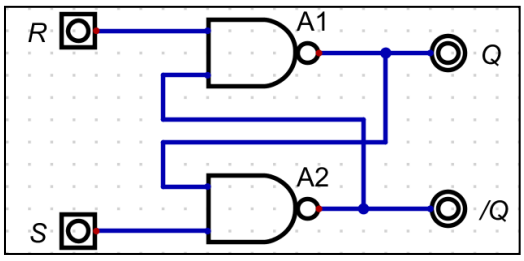
วงจร NOR R-S Latch

- เมื่อทั้ง R และ S เป็น 0 จะทำให้ที่ A1 output จะขึ้นอยู่กับ /(0 + /Q) = Q และที่ A2 output จะขึ้นอยู่กับ /(Q + 0) = /Q ซึ่งทำให้ทั้ง Q และ /Q ยังมีค่าความจริงเหมือนเดิม
- เมื่อ R เป็น 0 และ S เป็น 1 จะทำให้ A2 มี output เป็น 0 ส่งผลให้ /Q มีค่าเป็น 0 ซึ่งทำให้ A1 มี input เป็น 0 ทั้งคู่ส่งผลให้ output ของ A1 และ Q มีค่าเป็น 1
- เมื่อ R เป็น 1 และ S เป็น 0 จะทำให้ A1 มี output เป็น 0 ส่งผลให้ Q มีค่าเป็น 0 ซึ่งทำให้ A2 มี input เป็น 0 ทั้งคู่ส่งผลให้ output ของ A2 และ /Q มีค่าเป็น 1
- เมื่อทั้ง R และ S เป็น 1 จะทำให้ที่ A1 และ A2 มี output เป็น 0 ทั้งคู่และส่งผลให้ทั้ง Q และ /Q มีค่าเป็น 0 ทั้งคู่เช่นกัน

    - NAND R-S Latch โดยสร้างมาจาก NAND Gate 2 อัน

Input		Output		Description
R	S	Q	/Q	
0	0	1	1	Forbidden
0	1	1	0	Set
1	0	0	1	Reset
1	1	Q	/Q	Hold

ตารางค่าความจริงของ NAND R-S Latch



วงจร NOR R-S Latch

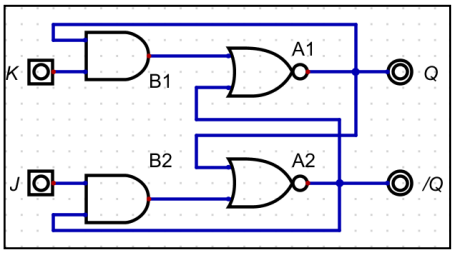
- เมื่อทั้ง R และ S เป็น 0 จะทำให้ที่ A1 และ A2 มี output เป็น 1 ทั้งคู่และส่งผลให้ทั้ง Q และ /Q มีค่าเป็น 1 ทั้งคู่เช่นกัน
- เมื่อ R เป็น 0 และ S เป็น 1 จะทำให้ A1 มี output เป็น 1 ส่งผลให้ Q มีค่าเป็น 1 ซึ่งทำให้ A2 มี input เป็น 1 ทั้งคู่ส่งผลให้ output ของ A2 และ /Q มีค่าเป็น 0
- เมื่อ R เป็น 1 และ S เป็น 0 จะทำให้ A2 มี output เป็น 1 ส่งผลให้ /Q มีค่าเป็น 1 ซึ่งทำให้ A1 มี input เป็น 1 ทั้งคู่ส่งผลให้ output ของ A1และ Q มีค่าเป็น 0
- เมื่อทั้ง R และ S เป็น 1 จะทำให้ที่ A1 output จะขึ้นอยู่กับ /(/Q && 0) = Q และที่ A2 output จะขึ้นอยู่กับ /(Q && 0) = /Q ซึ่งทำให้ทั้ง Q และ /Q ยังมีค่าความจริงเหมือนเดิม

จากตารางค่าความจริงจะสังเกตได้ว่า R-S Latch จะมี Forbidden Input ที่จะทำให้ทั้ง Q และ /Q มีค่าความจริงที่เหมือนกัน ซึ่งอาจจะก่อให้เกิดปัญหาตามมาได้

2. J-K Flip-Flop เป็น Memory ที่พัฒนาต่อจาก R-S Latch เพื่อแก้ไขปัญหา Forbidden Input

Input		Output		Description
K	J	Q	/Q	
0	0	Q	/Q	Hold
0	1	1	0	Set
1	0	0	1	Reset
1	1	/Q	Q	Toggle

ตารางค่าความจริงของ J-K Flip-Flop

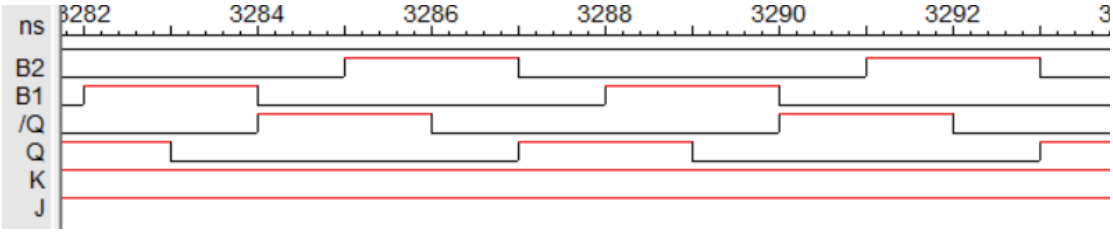


วงจร J-K Flip-Flop

วงจร J-K Flip-Flop มีการเพิ่ม AND Gate 2 อันเพื่อป้องกันในกรณี Forbidden Input โดยที่เมื่อ J และ K เป็น 1 ทั้งคู่นั้น output จาก B1 และ B2 จะไม่เป็น 1 ทั้งคู่เนื่องจาก Q และ /Q มีค่าความจริงที่ตรงกันข้ามกัน และจะทำให้เกิดการเปลี่ยนแปลงดังนี้ (กำหนดให้ทุก Gate มี Delay เท่ากัน)

1. B1 จะมี output เป็น 1 && Q = Q และ B2 จะมี output เป็น 1 && /Q = /Q
2. A1 จะมี output เป็น / (Q + /Q) = 0 และ A2 จะมี output เป็น / (Q + /Q) = 0
3. B1 จะมี output เป็น 1 && 0 = 0, B2 จะมี output เป็น 1 && 0 = 0, A1 จะมี output เป็น / (Q + 0) = /Q และ A2 จะมี output เป็น / (/Q + 0) = Q
4. B1 จะมี output เป็น 1 && /Q = /Q, B2 จะมี output เป็น 1 && Q = Q, A1 จะมี output เป็น / (Q + 0) = /Q และ A2 จะมี output เป็น / (/Q + 0) = Q
5. B1 จะมี output เป็น 1 && /Q = /Q, B2 จะมี output เป็น 1 && Q = Q, A1 จะมี output เป็น / (/Q + Q) = 0 และ A2 จะมี output เป็น / (Q + /Q) = 0

ซึ่งการเปลี่ยนแปลงนี้จะถูกทำซ้ำไปเรื่อย ๆในขณะที่ J และ K ยังคงเป็น 1 อยู่ ซึ่งจะทำให้ค่า Q สลับค่าไปมาระหว่าง 0 กับ 1 เหตุการณ์นี้เรียกว่า Race Condition (เพราะในความเป็นจริงเราต้องการให้ค่า Q ถูก Toggle แค่ 1 รอบเท่านั้น)

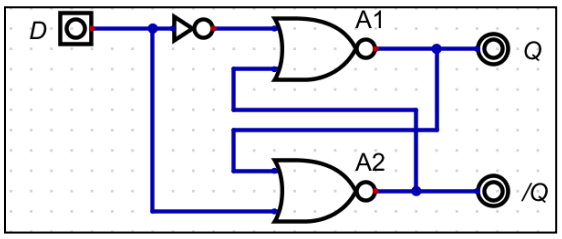


Timing Diagram **ขณะเกิด** Race Condition

3. D Flip-Flop เป็น Memory ที่พัฒนาต่อจาก R-S Latch โดยที่ D Flip-Flop จะทำได้แค่ Set หรือ Reset เท่านั้น

input	Output		Description
D	Q	/Q	
0	0	1	Reset
1	1	0	Set

ตารางค่าความจริงของ D Flip-Flop

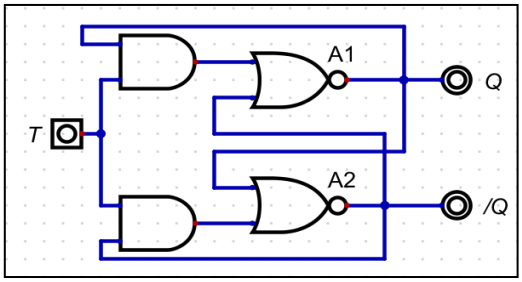


วงจร D Flip-Flop

4. T Flip-Flop เป็น Memory ที่พัฒนาต่อจาก J-K Flip-Flop โดยที่ T Flip-Flop จะทำได้แค่ Hold หรือ Toggle เท่านั้น

input	Output		Description
T	Q	/Q	
0	Q	/Q	Hold
1	/Q	Q	Toggle

ตารางค่าความจริงของ T Flip-Flop



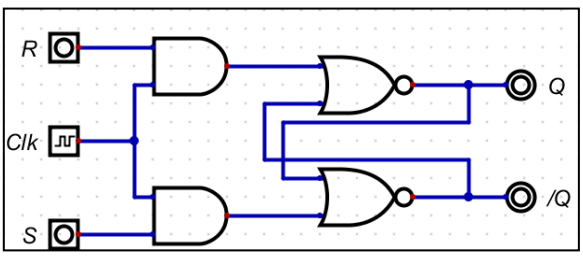
วงจร T Flip-Flop

แต่ในขณะเดียวกัน T Flip-Flop ก็ยังคงเกิด Race Condition เช่นเดียวกับ J-K Flip-Flop

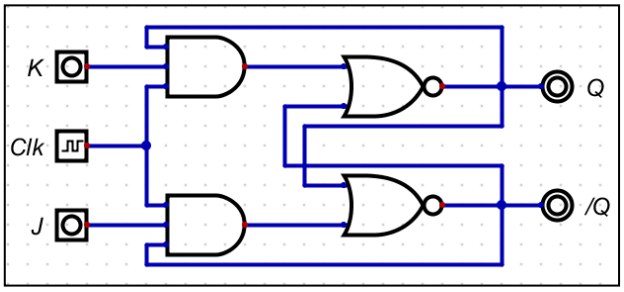
แต่การสร้าง Memory ข้างต้นนั้นยังไม่เพียงพอที่จะนำไปใช้งานจริง เนื่องจาก Memory ข้างต้นนั้นจะมีการเปลี่ยนแปลงค่าที่เก็บไว้ตาม input ที่รับเข้ามาในทันที แต่ว่าในวงจรส่วน Combinational Logic นั้นมี Propagational Delay อยู่ ทำให้ต้องใช้ระยะเวลาหนึ่งก่อนที่ผลลัพธ์ที่ถูกต้องของ Combinational Logic จะออกมา แต่ Memory ข้างต้นมีการเปลี่ยนแปลงตาม Input ในทันที ทำให้อาจเกิดความผิดพลาดขึ้นมาได้

จากย่อหน้าก่อนหน้า เราจึงมีการนำ Clock เข้ามาเพื่อควบคุม Memory ให้เปลี่ยนแปลง output หลังจาก Propagational Delay ของวงจร Combinational Logic โดยสามารถแบ่งประเภทได้ดังนี้

1. Gated Memory มีแนวคิดคือจะให้ output ของ Memory เปลี่ยนแปลงตอนที่ Clock มีค่าเป็น 1 หรือ 0 เท่านั้น



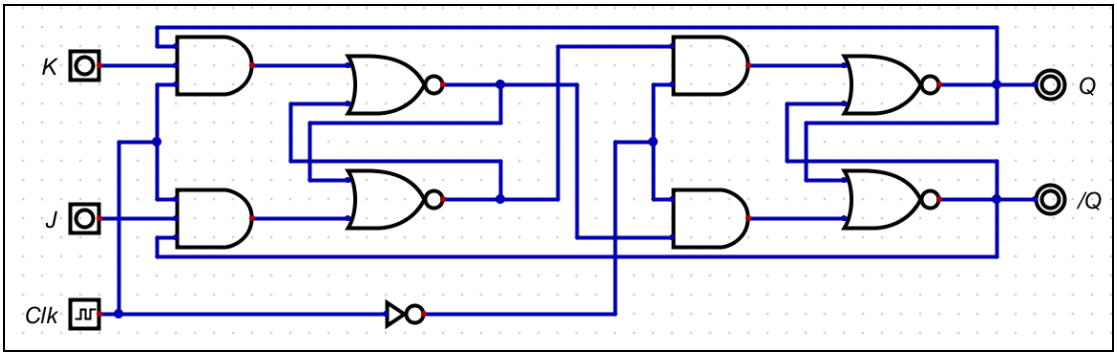
วงจร R-S Latch



วงจร J-K Flip-Flop

จากวงจรตัวอย่างข้างต้น จะสังเกตุได้ว่าเมื่อ Clock เป็น 0 จะทำให้ output ที่ออกจาก And Gate เป็น 0 ทั้งคู่ ซึ่งจะทำให้ output ของ Memory นั้นไม่เกิดการเปลี่ยนแปลง และเมื่อ Clock เป็น 1 นั้น output ที่เกิดขึ้นก็จะขึ้นอยู่กับ Input ที่เราตั้งค่าไว้ แต่สำหรับวงจร J-K Flip-Flop และ T Flip-Flop ก็ยังเกิดปัญหาเรื่อง Race Condition อยู่เหมือนเดิม

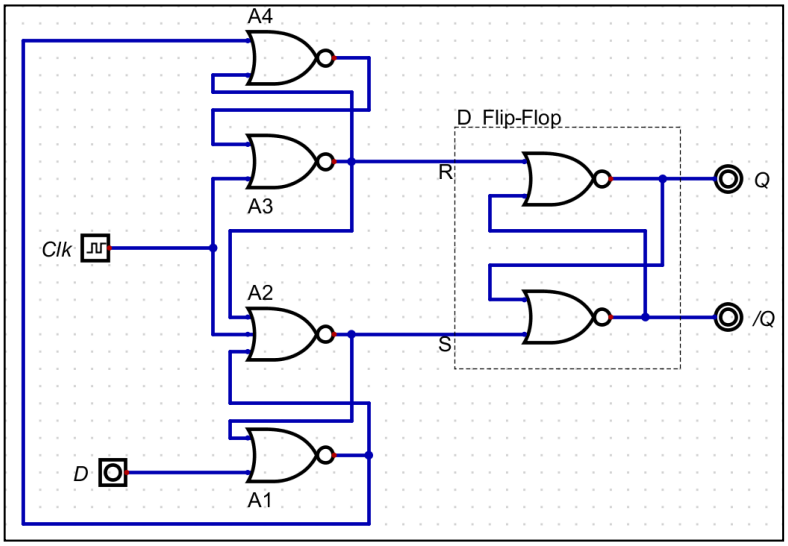
2. **Master/Slave J-K Flip-Flop** มีแนวคิดคือการนำ Memory มาต่อพ่วงกันโดยให้แต่ละ Memory เปลี่ยน output ณ Clock ที่ต่างกัน คล้ายๆกับ Air lock



วงจร Master/Slave J-K Flip-Flop

จากวงจรตัวอย่างจะเห็นได้ว่าวงจรจะประกอบไปด้วย J-K Flip-Flop 2 ตัว โดยที่ตัวด้านซ้ายจะถูกเรียกว่า Master และอีกตัวหนึ่งจะถูกเรียกว่า Slave โดยที่ Master จะเปลี่ยน Output ในขณะที่ Clock เป็น 1 และ Slave จะเปลี่ยน Output ในขณะที่ Clock เป็น 0 ทำให้ โดยที่ Slave จะเปลี่ยน Output ไปตามค่าที่ Output ออกมาจาก Master ซึ่งจะทำให้สามารถแก้ไขปัญหา Race Condition ได้ แต่วงจร Master/Slave ก็มีข้อเสียอยู่ตรงที่ถ้าค่าที่ตัว Master ผิดพลาด ก็จะทำให้ค่าที่ออกมาจาก Slave ผิดด้วยเช่นกัน

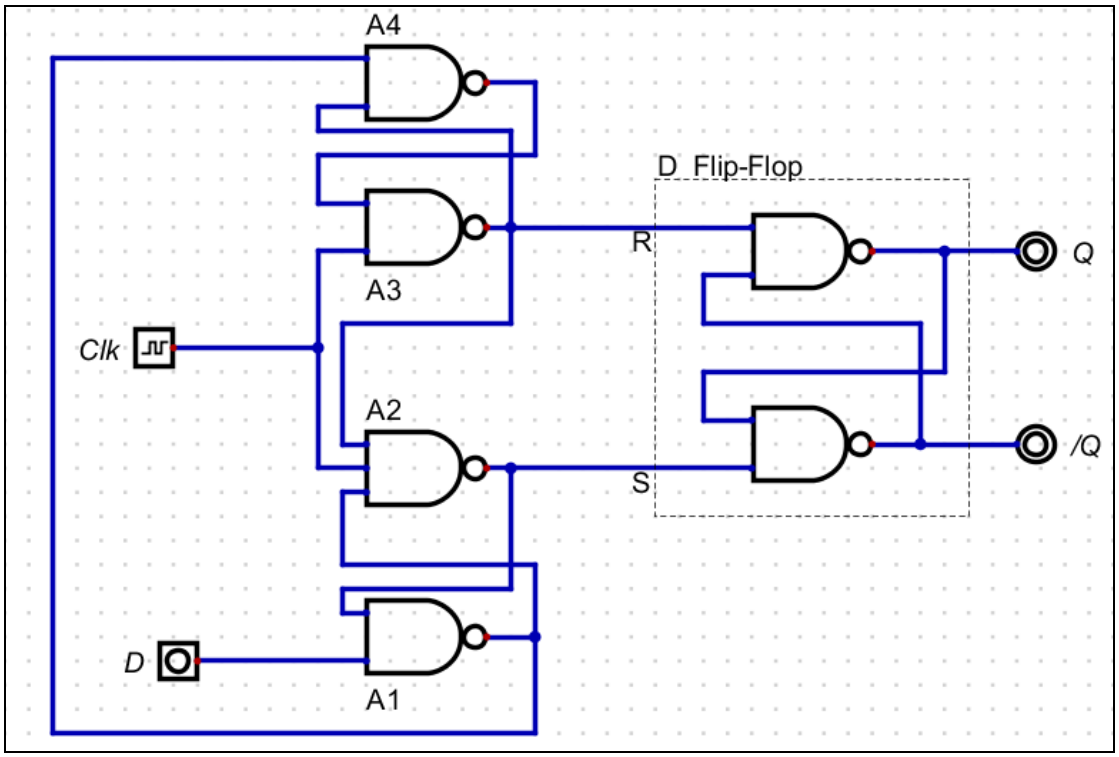
3. **Edge Trigger Memory** มีแนวคิดคือจะเปลี่ยนค่าตอนที่ Clock มีการเปลี่ยนแปลงจาก 1 ไปเป็น 0 (Negative Edge Trigger) หรือ เปลี่ยนค่าตอนที่ Clock มีการเปลี่ยนแปลงจาก 0 ไปเป็น 1 (Positive Edge Trigger)



Negative Edge Trigger D Flip-Flop NOR

Negative Edge Trigger จะมีหลักการทำงานดังนี้

1. เมื่อ clock เป็น 1, A2 และ A3 จะมี output เป็น 0 ทำให้เกิดการ Hold, A1 จะมีค่า output คือ /(0 + D) = /D และ A4 จะมีค่า output คือ /(0 + /D) = D
2. เมื่อ clock ถูกเปลี่ยนค่าไปเป็น 0 จะเกิดเหตุการณ์ดังนี้
    - A3 จะมี output เป็น / (0 + D) = /D และ A2 จะมี output เป็น / (0 + 0 + /D) = D
    - A1 จะมี output เป็น / (D + D) = /D, A4 จะมี output เป็น / (/D + /D) = D และ A2 จะมี output เป็น / (0 + /D + /D) = D
3. เมื่อ clock มีค่าเป็น 0 แล้วค่า input D มีการเปลี่ยนแปลงไปเป็น /D จะเกิดเหตุการณ์ดังนี้
    - A1 จะมี output คือ / (D + /D) = 0
    - A4 จะมี output คือ / (0 + /D) = D และ A2 จะมี output คือ / (/D + 0 + 0) = D



Positive Edge Trigger D Flip-Flop NAND

Positive Edge Trigger จะมีหลักการทำงานดังนี้

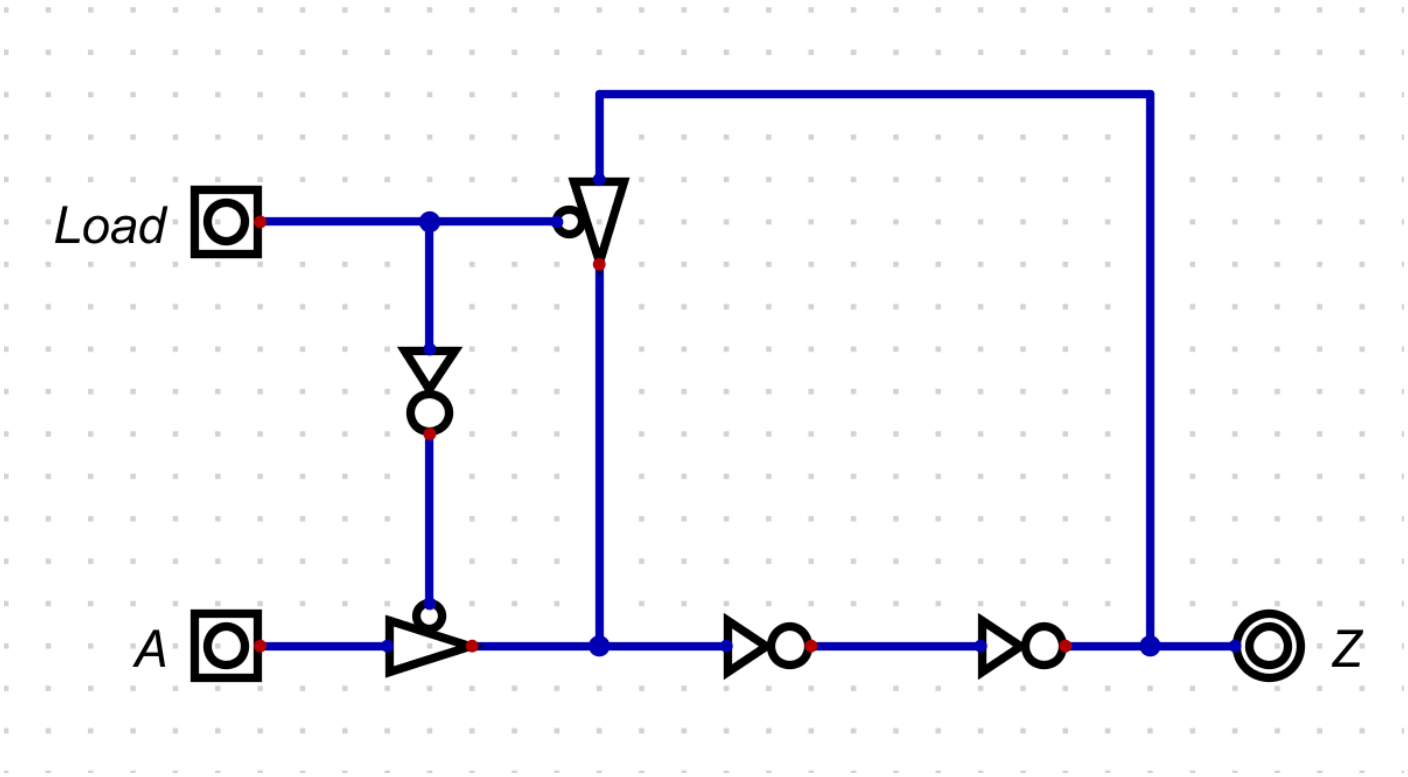
1. เมื่อ clock เป็น 0, A2 และ A3 จะมี output เป็น 1 ทำให้เกิดการ Hold, A1 จะมีค่า output คือ /(1 && D) = /D และ A4 จะมีค่า output คือ /(1 && /D) = D
2. เมื่อ clock ถูกเปลี่ยนค่าไปเป็น 1 จะเกิดเหตุการณ์ดังนี้
    - A3 จะมี output เป็น / (1 && D) = /D และ A2 จะมี output เป็น / (1 && 1 && /D) = D

- A1 จะมี output เป็น / (D && D) = /D, A4 จะมี output เป็น / (/D && /D) = D และ A2 จะมี output เป็น / (1 && /D && /D) = D

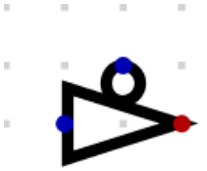
3. เมื่อ clock มีค่าเป็น 1 แล้วค่า input D มีการเปลี่ยนแปลงไปเป็น /D จะเกิดเหตุการณ์ดังนี้
   - A1 จะมี output คือ / (D && /D) = 1
   - A4 จะมี output คือ / (1 && /D) = D และ A2 จะมี output คือ / (/D && 1 && 1) = D

Simple Memory

ให้นิสิตสร้างวงจร Simple Memory ที่มี Input คือ A, Load ขนาด 1 Bit และ Output คือ Z ขนาด 1 บิท ตามรูป



รูปที่ 1 วงจร Simple Memory



รูปที่ 2 อุปกรณ์ Driver, Inverted Selector (อยู่ใน Folder Wires)

ข้อมูลนำเข้า

- A ขนาด 1 Bit
- Load ขนาด 1 Bit

ข้อมูลส่งออก

- Z ขนาด 1 Bit

ชุดข้อมูลทดสอบ

- 100% โปรแกรมทำงานถูกต้องตาม Input ทุกรูปแบบ

RS Latch (NOR)

ให้นิสิตสร้างวงจร RS Latch โดยใช้ Nor Gate ที่มี Input คือ R, S ขนาด 1 Bit และ Output คือ Q, \Q ขนาด 1 บิท โดยวงจรจะมีการทำงานดังต่อไปนี้

S(t)	R(t)	Q(t)	Q(t+Δ)	
0	0	0	0	HOLD
0	0	1	1	
0	1	0	0	RESET
0	1	1	0	
1	0	0	1	SET
1	0	1	1	
1	1	0	X	Not Allowed
1	1	1	X	

รูปที่ 1 ตารางการทำงานของวงจร RS Latch แบบ Nor Gate

ข้อมูลนำเข้า

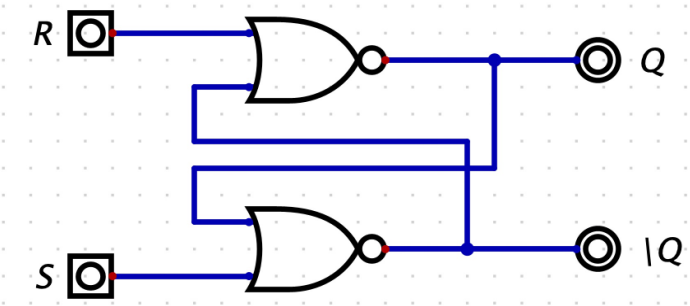
- R ขนาด 1 Bit
- S ขนาด 1 Bit

ข้อมูลส่งออก

- Q ขนาด 1 Bit
- \Q ขนาด 1 Bit

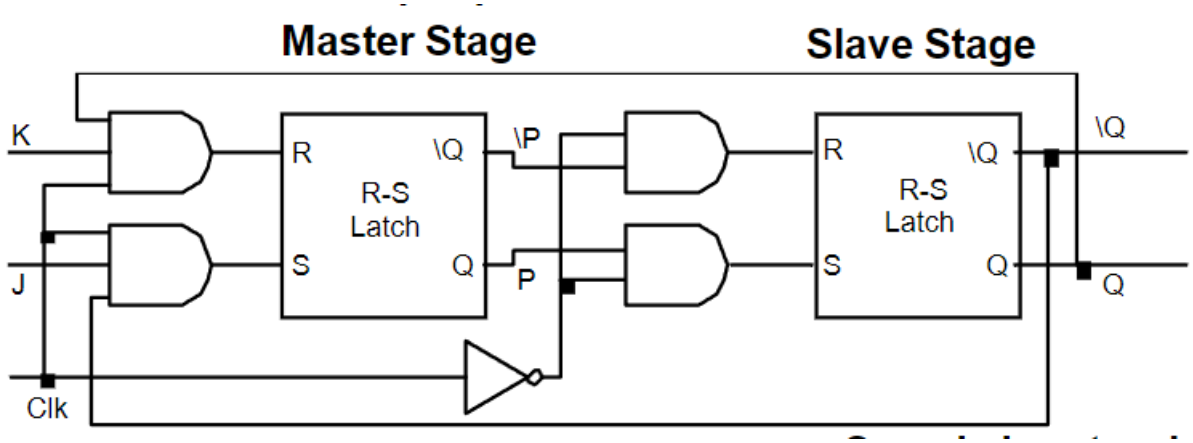
ชุดข้อมูลทดสอบ

- 100% โปรแกรมทำงานถูกต้องตาม Input ทุกรูปแบบ



Master-Slave JK Flip-flop

ให้นิสิตสร้างวงจร Master-Slave JK Flip-Flop ที่มี Input คือ J, K, Clock, \Clear ขนาด 1 Bit และ Output คือ Q, \Q ขนาด 1 บิท โดยให้สร้างโดยใช้วงจร RS Latch แบบ Nor Gate, input Clock คือสัญญาณนาฬิกาของระบบ และ \Clear เป็น asynchronous input และเมื่อ \Clear มีค่าเป็น 1 ให้เปลี่ยนค่า Q เป็น 0 และ /Q เป็น 1 โดยทันทีโดยไม่ต้องรอสัญญาณ Clock.



รูปที่ 1 รูปวงจร Master-Slave แบบไม่มี \Clear

ข้อมูลนำเข้า

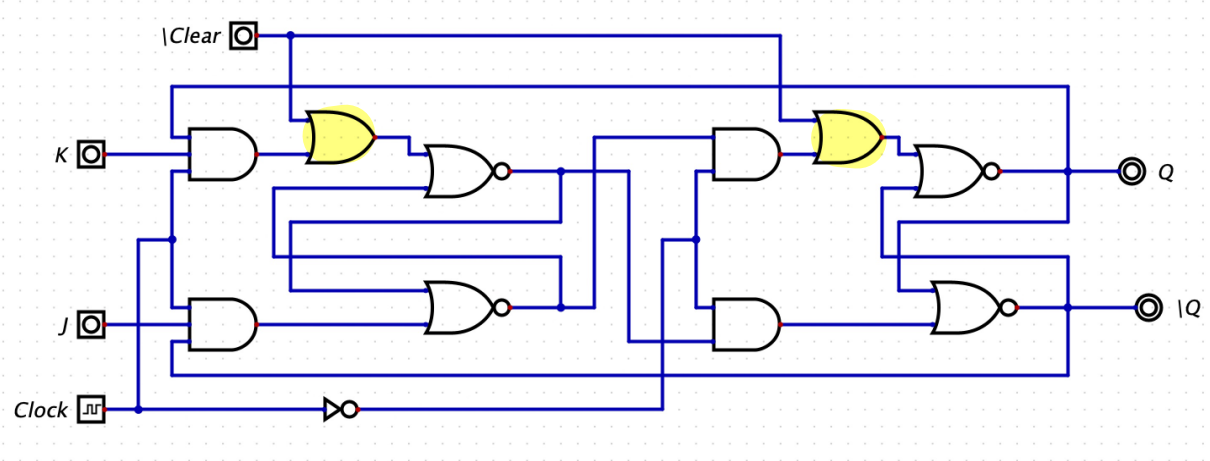
- K ขนาด 1 Bit
- J ขนาด 1 Bit
- Clock ขนาด 1 Bit
- \Clear ขนาด 1 Bit

ข้อมูลส่งออก

- Q ขนาด 1 Bit
- \Q ขนาด 1 Bit

ชุดข้อมูลทดสอบ

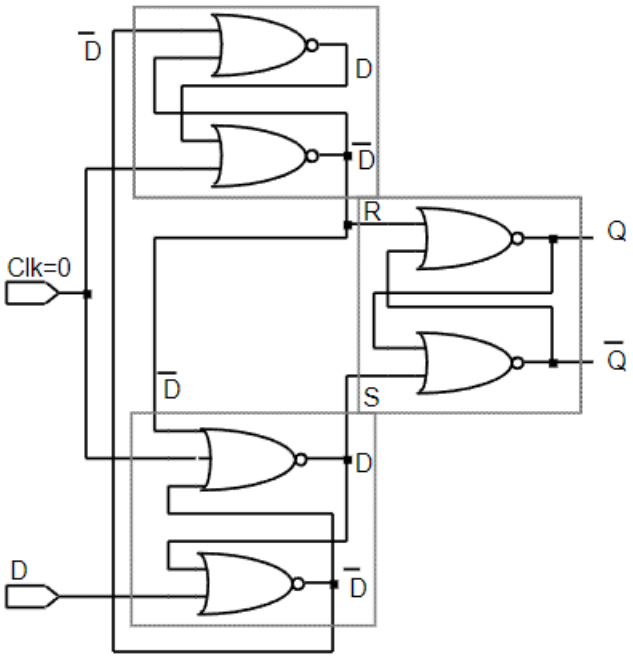
- 100% โปรแกรมทำงานถูกต้องตาม Input ทุกรูปแบบ



asynchronous

Positive Edge Trigger D Flip-flop

ให้นิสิตสร้างวงจร Positive Edge Trigger D Flip-Flop ที่มี Input คือ D, Clock, \Clear ขนาด 1 Bit และ Output คือ Q, \Q ขนาด 1 บิท โดยให้สร้างวงจรจาก RS Latch แบบ Nor Gate ซึ่งมี Clock คือสัญญาณนาฬิกาของระบบ, \Clear เป็น synchronous input เมื่อ \Clear มีค่าเป็น 1 ให้เปลี่ยนค่า Q เป็น 0 และ \Q เป็น 1



รูปที่ 1 Negative Edge Trigger D Filp-flop

ข้อมูลนำเข้า

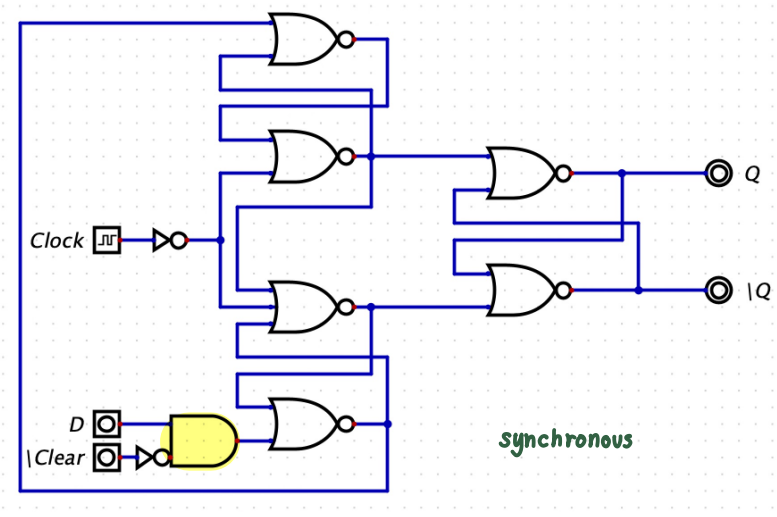
- D ขนาด 1 Bit
- Clock ขนาด 1 Bit
- \Clear ขนาด 1 Bit

ข้อมูลส่งออก

- Q ขนาด 1 Bit
- \Q ขนาด 1 Bit

ชุดข้อมูลทดสอบ

- 100% โปรแกรมทำงานถูกต้องตาม Input ทุกรูปแบบ



Positive Edge Trigger T Flip flop

ให้นิสิตสร้างวงจร Positive Edge Trigger T Flip flop ที่มี Input คือ T, Clock, \Clear ขนาด 1 Bit และ Output คือ Q, \Q ขนาด 1 บิท โดยให้สร้างวงจรจาก RS Latch แบบ Nor Gate ซึ่งมี Clock คือสัญญาณนาฬิกาของระบบ, \Clear เป็น synchronous input เมื่อ \Clear มีค่าเป็น 1 ให้เปลี่ยนค่า Q เป็น 0 และ \Q เป็น 1

Input	Outputs	
	Present State	Next State
T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

รูปที่ 1 ตารางค่าความจริงของวงจร T Flip Flop

ข้อมูลนำเข้า

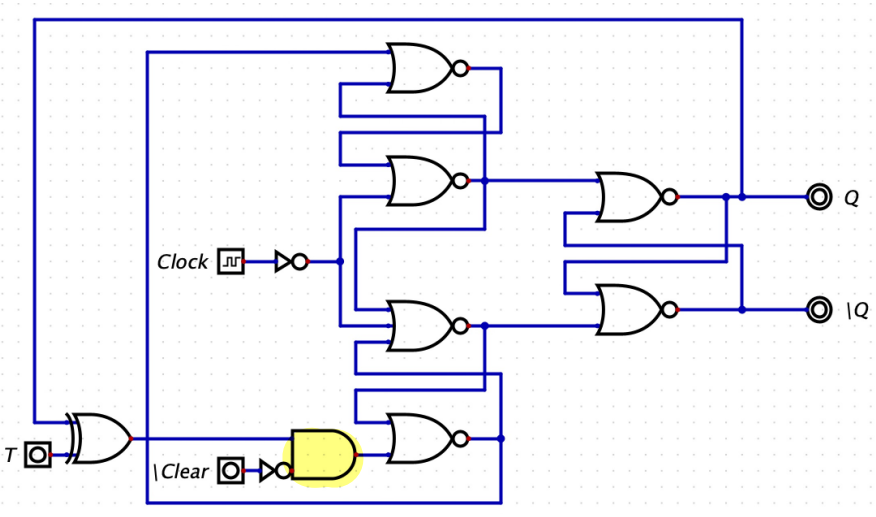
- T ขนาด 1 Bit
- Clock ขนาด 1 Bit
- \Clear ขนาด 1 Bit

ข้อมูลส่งออก

- Q ขนาด 1 Bit
- \Q ขนาด 1 Bit

ชุดข้อมูลทดสอบ

- 100% โปรแกรมทำงานถูกต้องตาม Input ทุกรูปแบบ



synchronous