

จุฬาลงกรณ์มหาวิทยาลัย

คณะวิศวกรรมศาสตร์

ภาควิชาวิศวกรรมคอมพิวเตอร์

2110-263 DIGITAL COMPUTER LOGIC LAB I

ชื่อ _____

เลขประจำตัว _____

หมายเลขเครื่อง _____

วันที่ _____

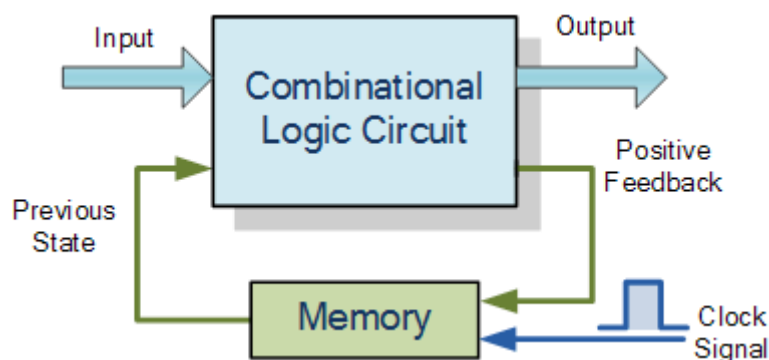
7. Latch และ Flip-flop

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 ที่ผลลัพธ์ของวงจรมันจะขึ้นอยู่กับ input อย่างเดียว โดยวงจร Sequential Logic สามารถนำไปสร้าง Finite-state machines ซึ่งเป็นโครงสร้างพื้นฐานสำหรับวงจร Digital ทั้งหมดได้



รูปที่ 1 : โครงสร้างการทำงานของวงจร Sequential Logic อย่างง่าย

นอกจากนี้วงจร Sequential Logic สามารถแบ่งออกมาได้เป็น 2 ประเภทคือ

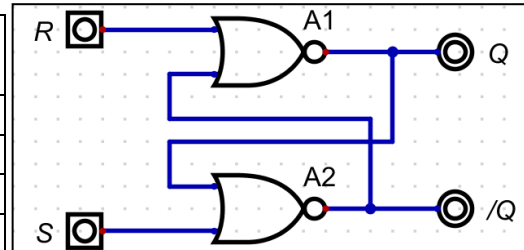
1. Synchronous เป็นวงจรที่ผลลัพธ์จะขึ้นอยู่กับ Memory ของวงจรอย่างเดียว
2. Asynchronous เป็นวงจรที่ผลลัพธ์จะเปลี่ยนแปลงทันทีตาม input และ Memory ของวงจร

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



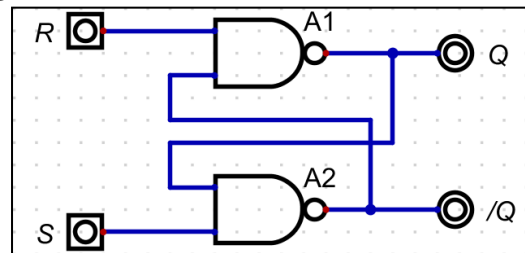
ตารางค่าความจริงของ NOR R-S Latch

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

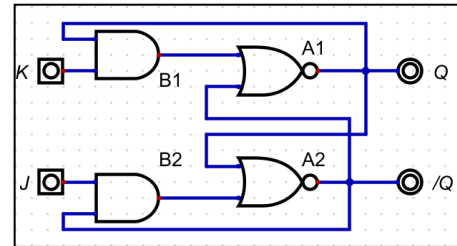
วงจร NOR R-S Latch

- 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 A1 และ 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 มีความจริงที่เหมือนกัน ซึ่งอาจจะก่อให้เกิดปัญหาตามมาได้

2. J-K Flip-Flop เป็น Memory ที่พัฒนาต่อจาก R-S Latch เพื่อแก้ไขปัญห 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



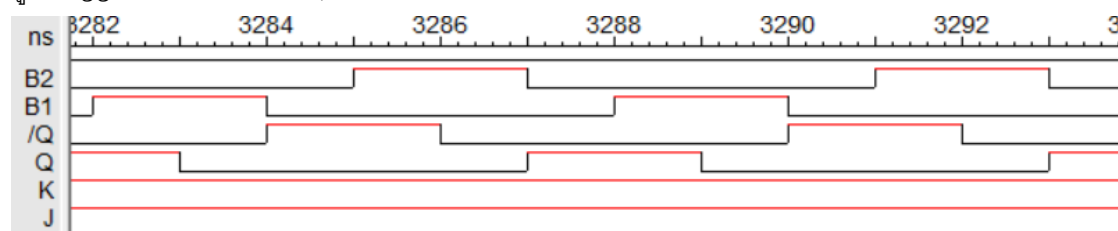
ตารางค่าความจริงของ J-K Flip-Flop

วงจร J-K Flip-Flop

วงจร J-K Flip-Flop มีการเพิ่ม AND Gate 2 อันเพื่อป้องกันในกรณี Forbidden Input โดยที่เมื่อ J และ K เป็น 1 ทั้งคู่ output จาก B1 และ B2 จะไม่เป็น 1 ทั้งคู่เนื่องจาก Q และ /Q มีความจริงที่ตรงกันข้ามกัน 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 ๆ ในขณะที่ J และ K ยังคงเป็น 1 อยู่ ซึ่งจะทำให้ค่า Q สลับค่าไปมาระหว่าง 0 กับ 1 เหตุการณ์นี้เรียกว่า Race Condition (เพราะในความเป็นจริงเราต้องการให้ค่า Q ถูก Toggle แค่ 1 รอบเท่านั้น)

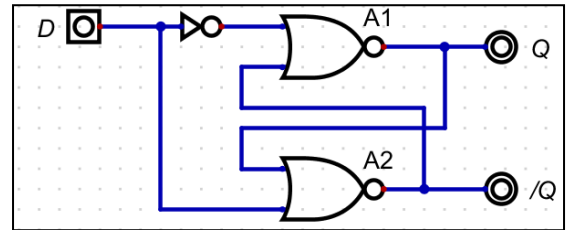


Timing Diagram ขณะเกิด Race Condition

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

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

ตารางค่าความจริงของ D Flip-Flop

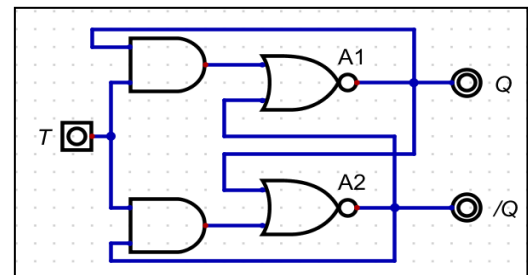


วงจร D Flip-Flop

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

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

ตารางค่าความจริงของ T Flip-Flop



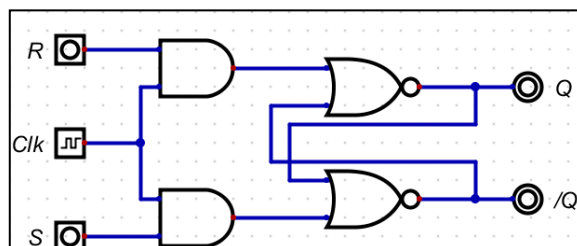
วงจร T Flip-Flop

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

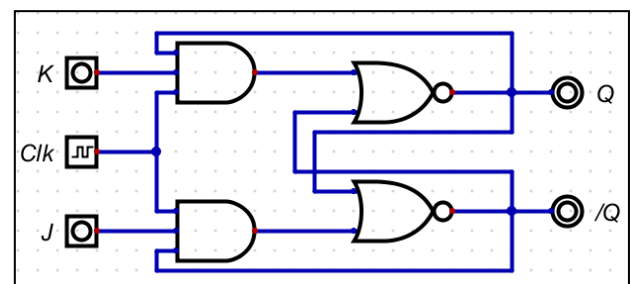
แต่การสร้าง Memory ข้างต้นนั้นยังไม่เพียงพอที่จะนำไปใช้งานจริง เนื่องจาก Memory ข้างต้นนั้นจะมีการเปลี่ยนแปลงค่าที่เก็บไว้ตาม input ที่รับเข้ามาในทันที แต่ว่าในวงจรส่วน Combinational Logic นั้นมี Propagational Delay อยู่ ทำให้ต้องใช้ระยะเวลาหนึ่งก่อนที่จะผลลัพธ์ที่ถูกต้องของ Combinational Logic จะออกมา 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



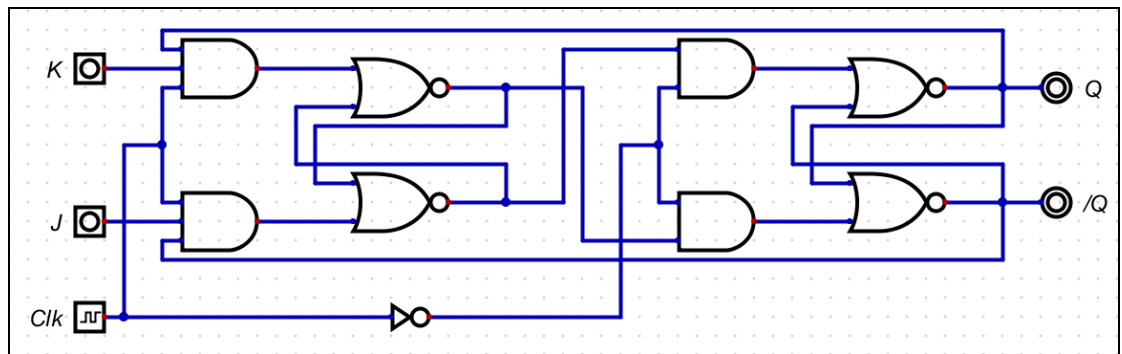
วงจร R-S Latch



วงจร J-K Flip-Flop

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

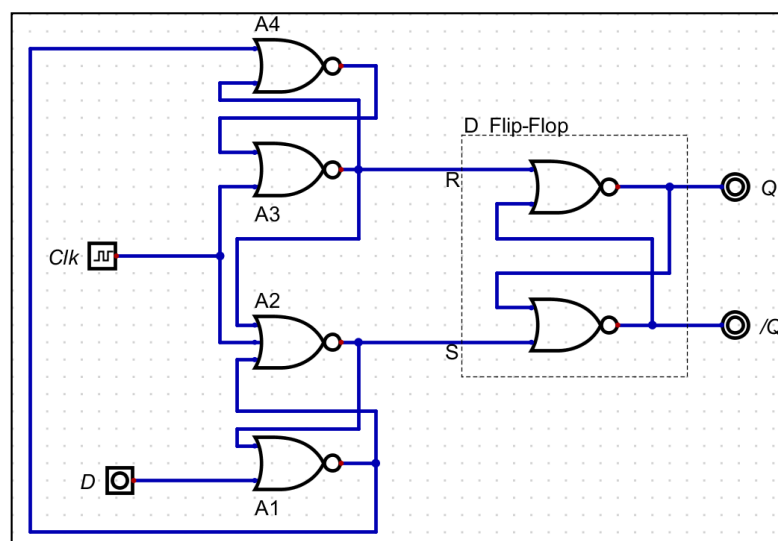
2. Master/Slave J-K Flip-Flop มีแนวคิดคือการนำ Memory มาต่อพ่วงกันโดยให้แต่ละ Memory เปลี่ยน output ณ Clock ที่ต่างกัน คล้ายๆกับ Air lock



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 ซึ่งจะทำให้สามารถแก้ไข ปัญหา Race Condition ได้ แต่วงจร Master/Slave ก็มีข้อเสียอยู่ตรงที่ถ้าค่าที่ตัว Master ผิดพลาด ก็จะทำให้ค่าที่ออกมาจาก Slave ผิดด้วยเช่นกัน

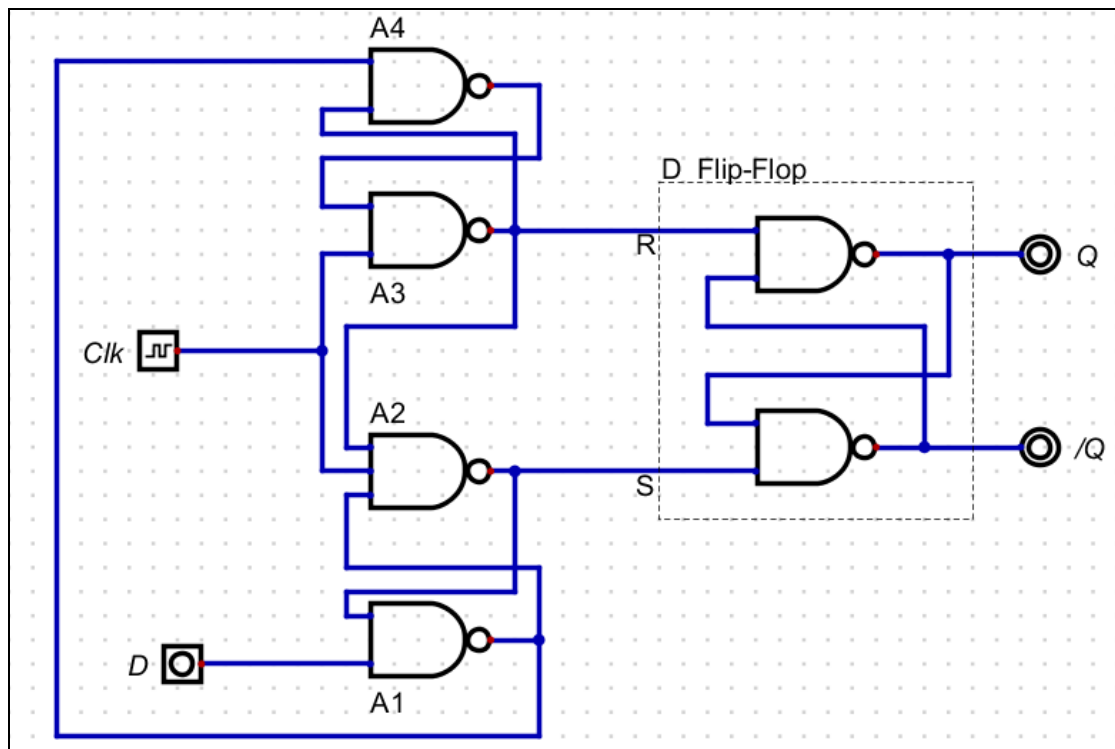
3. Edge Trigger Memory มีแนวคิดคือจะเปลี่ยนค่าตอนที่ Clock มีการเปลี่ยนแปลงจาก 1 ไปเป็น 0 (Negative Edge Trigger) หรือ เปลี่ยนค่าตอนที่ Clock มีการเปลี่ยนแปลงจาก 0 ไปเป็น 1 (Positive Edge Trigger)



Negative Edge Trigger D Flip-Flop

Negative Edge Trigger จะมีหลักการทำงานดังนี้

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



Positive Edge Trigger D Flip-Flop

Positive Edge Trigger จะมีหลักการทำงานดังนี้

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

- 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 $(/D \&\& 1 \&\& 1) = D$