



การออกแบบวงจรตรวจนับลำดับ แบบมัวร์ที่ใช้ T Flip-Flop (Moore Machine Design of a Sequence Detector Circuits T Flip-Flop¹)

รศ.ดร.สุรัชย์ สุขสกุลชัย

(¹Roth, C.H., Fundamentals of Logic Design, 5th Ed.)

Out Line

- การออกแบบ Moore State Graph สำหรับตรวจนับลำดับตัวเลขไบนารี
- การสร้าง Next State Table จาก Moore State Graph
- การสร้าง Transition Table จาก Next State Table
- การสร้าง T Flip-Flop Table จาก Transition Table
- การแปลง T Flip-Flop Table เป็นสมการบูลีนโดยใช้ Karnaugh Map
- บทสรุป

2

วัตถุประสงค์

- สามารถออกแบบ Moore State Graph สำหรับตรวจนับลำดับตัวเลขไบนารี
- สามารถสร้างตรวจนับลำดับตัวเลขไบนารีโดยใช้ JK Flip-Flop ได้

3

บทนำ

- State Graph คือผังแสดงการทำงานของวงจรลำดับ (sequential circuit) แบ่งออกเป็น 2 แบบคือ
 - แบบมัวร์ (Moore Machine)
 - แบบเมลลี (Mealy Machine)
- วงจรนับก็เป็นวงจรลำดับชนิดหนึ่ง
 - แต่ไม่จัดอยู่ในทั้งสองแบบ
 - แต่มีหลักการพื้นฐานคล้ายๆกัน
- ขั้นตอนการแปลง State Graph เป็นวงจรลอจิกจะมีขั้นตอนคล้ายๆกัน

4

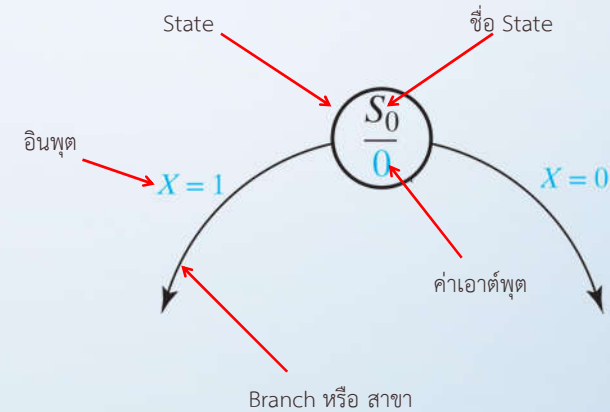
บทนำ

- Moore Machine เป็นมีหลักการทำงาน จะเป็น State Graph ที่เป็น

5

บทนำ

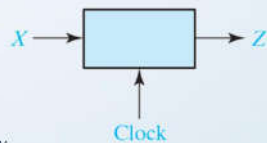
Moore Machine



6

ตัวอย่างการออกแบบ Moore State Graph

- เพื่อให้เข้าใจวิธีการสร้าง State Graph
- ตัวอย่างการออกแบบวงจรตรวจจับลำดับตัวเลขไบนารี “101” จะถูกใช้ในการอธิบาย



- ค่า X คืออินพุตที่ป้อนเข้า
- ค่า Z คือค่าเอาต์พุตเมื่อลำดับของเลขที่ต้องการถูกตรวจจับได้
 - Z=0 ถ้าลำดับไม่ถูกต้อง
 - Z=1 ถ้าลำดับถูกต้อง
 - สัญญาณ Clock มีไว้สำหรับให้วงจรทำงานพร้อมกัน

7

ขั้นตอน 1: การสร้าง Moore State Graph

- ลำดับที่ต้องการตรวจจับคือ 101

X =	0	0	1	1	0	1	1	0	0	1	0	1	0	1	0	0
Z =	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0
(time:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15)

8

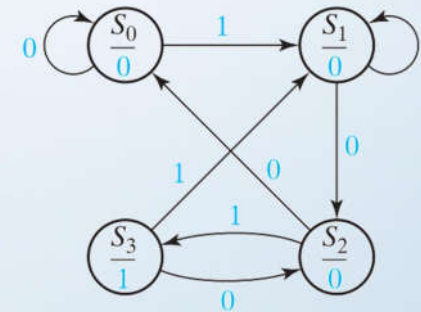
ขั้นตอน 1: การสร้าง Moore State Graph

- ลำดับที่ต้องการตรวจจับคือ 101

9

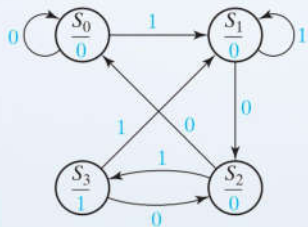
ขั้นตอน 1: การสร้าง Moore State Graph

- State Graph สำหรับตรวจจับลำดับ 101 ที่ได้



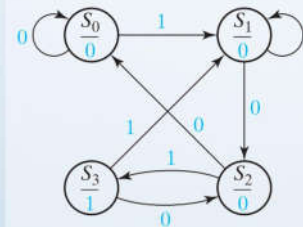
10

ขั้นตอน 2: การสร้าง Next State Table จาก State Graph



11

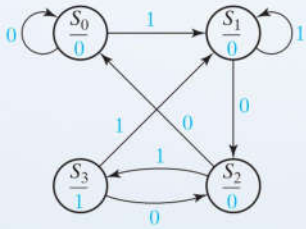
ขั้นตอน 2: การสร้าง Next State Table จาก State Graph



Present State	Next State		Present Output(Z)
	X = 0	X = 1	
S_0	S_0	S_1	0
S_1	S_2	S_1	0
S_2	S_0	S_3	0
S_3	S_2	S_1	1

12

ขั้นตอน 2: การสร้าง Next State Table จาก State Graph



Present State	Next State		Present Output(Z)
	X = 0	X = 1	
00	00	01	0
01	01	01	0
11	11	10	0
10	10	01	1

ขั้นตอน 2: การสร้าง Next State Table จาก State Graph

Present State	Next State		Present Output(Z)
	X = 0	X = 1	
00	00	01	0
01	01	01	0
11	11	10	0
10	10	01	1

ขั้นตอน 3: การสร้าง Transition Table (T Flip-Flop Table)

Present State AB	Next State		Present Output(Z)
	X = 0 A^+B^+	X = 1 A^+B^+	
00	00	01	0
01	11	01	0
11	00	10	0
10	11	01	1

ขั้นตอน 3: การสร้าง Transition Table (T Flip-Flop Table)

Present State AB	Next State		Present Output(Z)	T_A		T_B	
	X = 0 A^+B^+	X = 1 A^+B^+		X = 0	X = 1	X = 0	X = 1
00	00	01	0				
01	11	01	0				
11	00	10	0				
10	11	01	1				

ขั้นตอน 3: การสร้าง Transition Table (T Flip-Flop Table)

Present State AB	Next State		Present Output(Z)	T_A		T_B	
	X = 0 $A+B^+$	X = 1 A^+B^+		X = 0	X = 1	X = 0	X = 1
00	00	01	0	0	0		
01	11	01	0	1	0		
11	00	10	0	1	0		
10	11	01	1	0	1		

ขั้นตอน 3: การสร้าง Transition Table (T Flip-Flop Table)

Present State AB	Next State		Present Output(Z)	T_A		T_B	
	X = 0 $A+B^+$	X = 1 A^+B^+		X = 0	X = 1	X = 0	X = 1
00	00	01	0	0	0		
01	11	01	0	1	0		
11	00	10	0	1	0		
10	11	01	1	0	1		

ขั้นตอน 3: การสร้าง Transition Table (T Flip-Flop Table)

Present State AB	Next State		Present Output(Z)	T_A		T_B	
	X = 0 $A+B^+$	X = 1 A^+B^+		X = 0	X = 1	X = 0	X = 1
00	00	01	0	0	0	0	1
01	11	01	0	1	0	0	0
11	00	10	0	1	0	1	1
10	11	01	1	0	1	1	1

ขั้นตอน 4: การแปลง T Flip-Flop Table เป็นสมการบูลีน

Present State AB	Next State		Present Output(Z)	T_A		T_B	
	X = 0 $A+B^+$	X = 1 A^+B^+		X = 0	X = 1	X = 0	X = 1
00	00	01	0	0	0	0	1
01	11	01	0	1	0	0	0
11	00	10	0	1	0	1	1
10	11	01	1	0	1	1	1

ขั้นตอน 4: การแปลง T Flip-Flop Table เป็นสมการบูลีน

Present State AB
00
01
11
10

	T_A		T_B	
X	0	1	0	1
0	0	0	0	1
1	1	0	0	0
1	1	0	1	1
0	0	1	1	1

ขั้นตอน 4: การแปลง T Flip-Flop Table เป็นสมการบูลีน

Present State AB	T_A		T_B	
X	0	1	0	1
00	0	0	0	1
01	1	0	0	0
11	1	0	1	1
10	0	1	1	1

X	0	1
AB	00	01
01	11	10
10		

T_A

X	0	1
AB	00	01
01	11	10
10		

T_B

21

22

ขั้นตอน 4: การแปลง T Flip-Flop Table เป็นสมการบูลีน

Present State AB	T_A		T_B	
X	0	1	0	1
00	0	0	0	1
01	1	0	0	0
11	1	0	1	1
10	0	1	1	1

X	0	1
AB	00	01
01	11	10
10		

T_A

X	0	1
AB	00	01
01	11	10
10		

T_B

$$T_A = XB + XAB'$$

$$T_B = A + XB'$$

ขั้นตอน 4: การแปลง T Flip-Flop Table เป็นสมการบูลีน

Present State AB	T_A		T_B	
X	0	1	0	1
00	0	0	0	1
01	1	0	0	0
11	1	0	1	1
10	0	1	1	1

X	0	1
AB	00	01
01	11	10
10		

T_A

X	0	1
AB	00	01
01	11	10
10		

T_B

$$T_A = X'B + XAB'$$

$$T_B = A + XB'$$

23

24

ขั้นตอน 4: การแปลง ZT Flip-Flop Table เป็นสมการบูลีน

Present State AB	T_A		T_B	
	X = 0	X = 1	X = 0	X = 1
00	0	0	0	1
01	1	0	0	0
11	1	0	1	1
10	0	1	1	1

AB \ X	0	1
00	0	0
01	1	0
11	1	0
10	0	1

T_A

$$T_A = X'B + XAB'$$

AB \ X	0	1
00	0	1
01	0	0
11	1	1
10	1	1

T_B

$$T_B = A + XB'$$

ขั้นตอน 4: การสร้างวงจร Logic จากสมการบูลีน

Present State AB	T_A		T_B	
	X = 0	X = 1	X = 0	X = 1
00	0	0	0	1
01	1	0	0	0
11	1	0	1	1
10	0	1	1	1

$$T_A = X'B + XAB'$$

$$T_B = A + XB'$$

ขั้นตอน 5: การหาสมการบูลีนของเอาต์พุต

Present State AB	Next State		Present Output(Z)	T_A		T_B	
	X = 0 $A+B^+$	X = 1 $A+B^+$		X = 0	X = 1	X = 0	X = 1
00	00	01	0	0	0	0	1
01	11	01	0	1	0	0	0
11	00	10	0	1	0	1	1
10	11	01	1	0	1	1	1

ขั้นตอน 5: การหาสมการบูลีนของเอาต์พุต

Present State AB
00
01
11
10

Present Output(Z)
0
0
0
1

ขั้นตอน 5: การหาสมการบูลีนของเอาต์พุต

Present State AB	Present Output(Z)
00	0
01	0
11	0
10	1

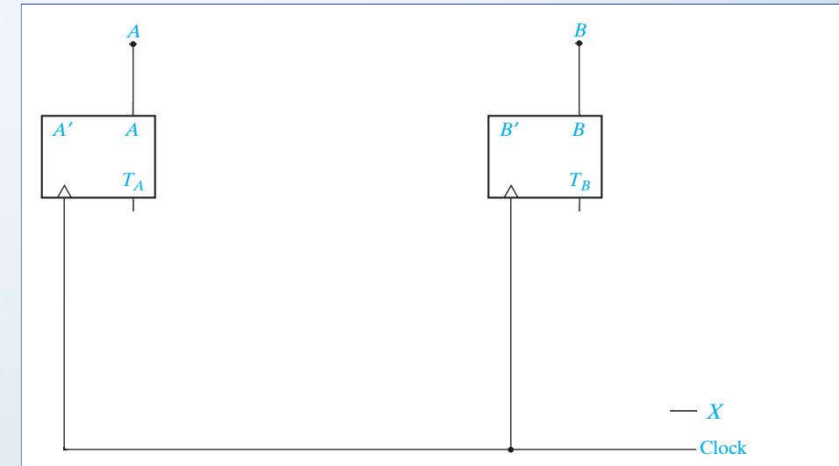
$$Z = AB'$$

ขั้นตอน 6: การสร้างวงจร Logic จากสมการบูลีน

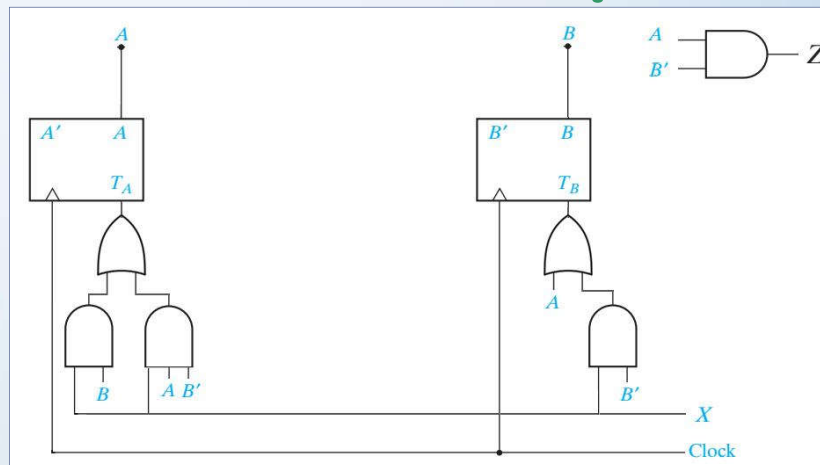
$$T_A = X'B + XAB'$$

$$T_A = A + XB'$$

$$Z = AB'$$



ขั้นตอน 6: การสร้างวงจร Logic จากสมการบูลีน



$$T_A = X'B + XAB'$$

$$T_A = A + XB'$$

$$Z = AB'$$

บทสรุป

