



# Algorithmic State Machine



## Algorithmic State Machine (ASM)

**Algorithm**= วิธีการแก้ปัญหา อธิบายโดยภาษาระดับสูงถึงระดับต่ำ หรือ Flowchart

**State Machine**= คือวงจร Synchronous Sequential นั่นเอง มี State เป็นตัวกำหนดการเปลี่ยน State และ สร้าง O/P ขึ้นอยู่กับ State และ I/P

**Algorithmic State Machine**= State Machine ทำงานเป็นขั้นตอนและเงื่อนไขอธิบายโดยใช้ Algorithm (Flowchart)

# ระบบชื่อสัญญาณของ ASM

ชื่อสัญญาณ I/P,O/P ของ ASM จะใช้ตัวอักษร 3-4 ตัวอักษร ซึ่งอาจสื่อความหมายได้ เช่น

RST สำหรับการ Reset เครื่อง

CLR สำหรับการ Clear เครื่อง

และอาจมีการบอก Active Logic (โดยใช้ H, Y, 1 และ L, N, 0) นำหน้าชื่อสัญญาณ เช่น

H.RST หมายถึง ทำการ Reset เมื่อ RST = 1

L.RST หมายถึง ทำการ Reset เมื่อ RST = 0

N.RED หมายถึง สัญญาณ RED Active ที่ลอจิก 0

Y.RED หมายถึง สัญญาณ RED Active ที่ลอจิก 1



## ระบบข้อสัญญาของ ASM

O/P line อาจมี I นำหน้า หมายถึง **Immediate Output** ก็จะมีการเปลี่ยนแปลงได้ใน **ช่วงเวลาของ State** ตามเงื่อนไขของ I/P หากมีการเปลี่ยน I/P อาจทำให้ค่าของ **Immediate O/P** นี้เปลี่ยนแปลงได้

ถ้า O/P ใดไม่มี I นำหน้า หมายถึง **O/P นั้นขึ้นอยู่กับ State** เพียงอย่างเดียว การเปลี่ยนค่า O/P นั้นเกิดขึ้นเมื่อมีการเปลี่ยน State เท่านั้น



## การเขียน ASM Chart

ASM chart จะประกอบไปด้วยหลายๆ ASM block

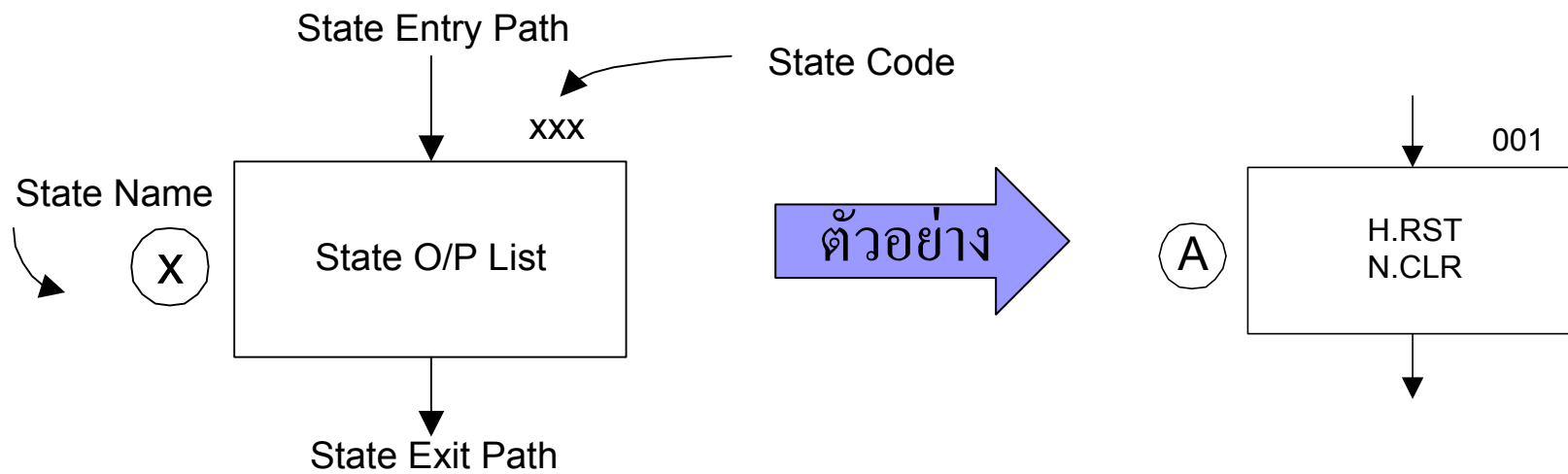
1 ASM Block จะแสดงการทำงานใน 1 State

ทุกอย่างใน Block นั้นถือว่าทำงานพร้อมกันหมดในแต่ละ ASM Block  
ประกอบไปด้วย

- State Box
- Decision Box
- Condition O/P Box

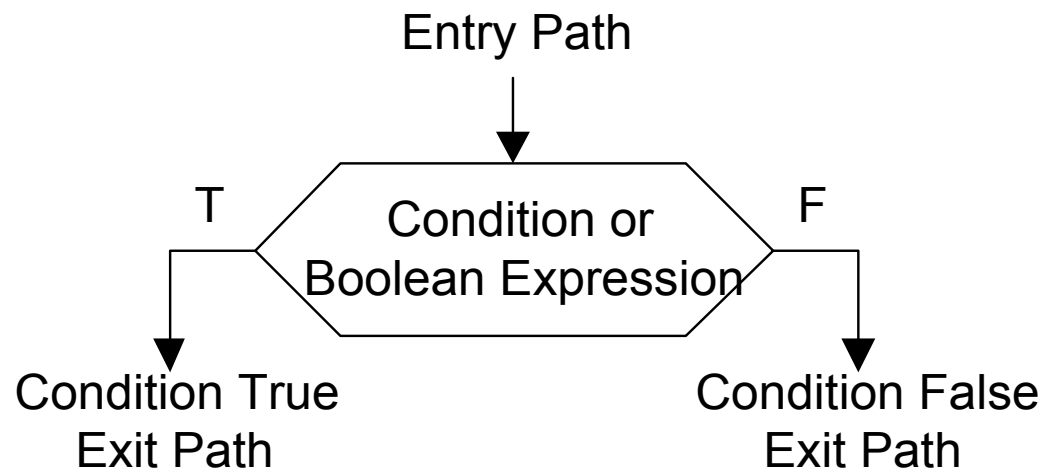
# การเขียน ASM Chart

## ■ State Box



## การเขียน ASM Chart

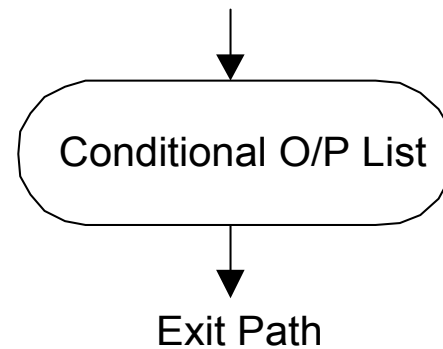
- Decision Box      T, Y, 1 คือ จริง และ F, N, 0 คือ เท็จ



# การเขียน ASM Chart

## ■ Condition O/P Box

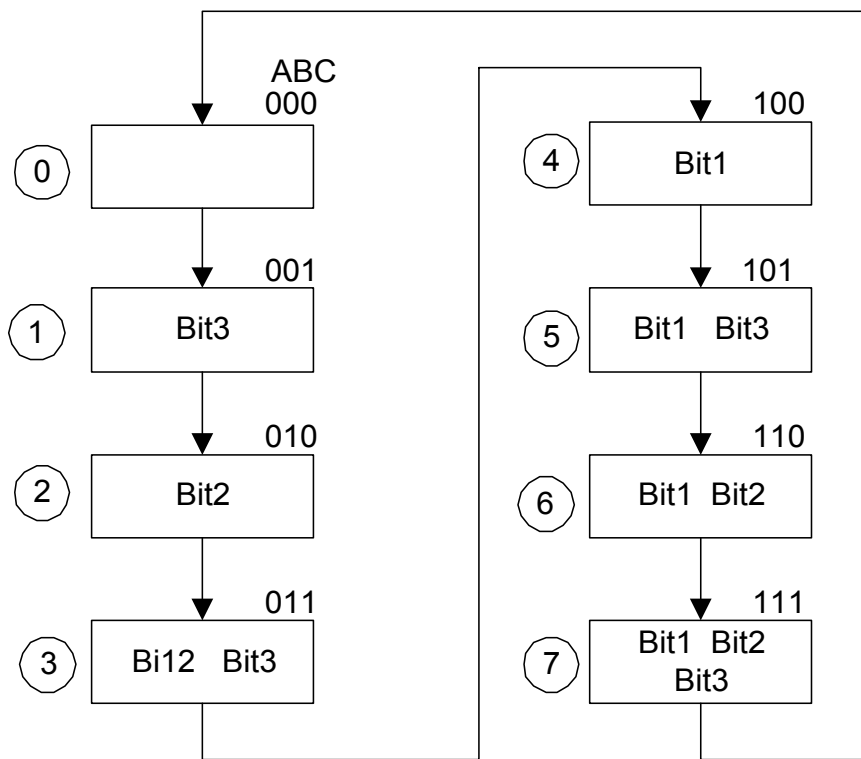
Entry Path From Decision Box



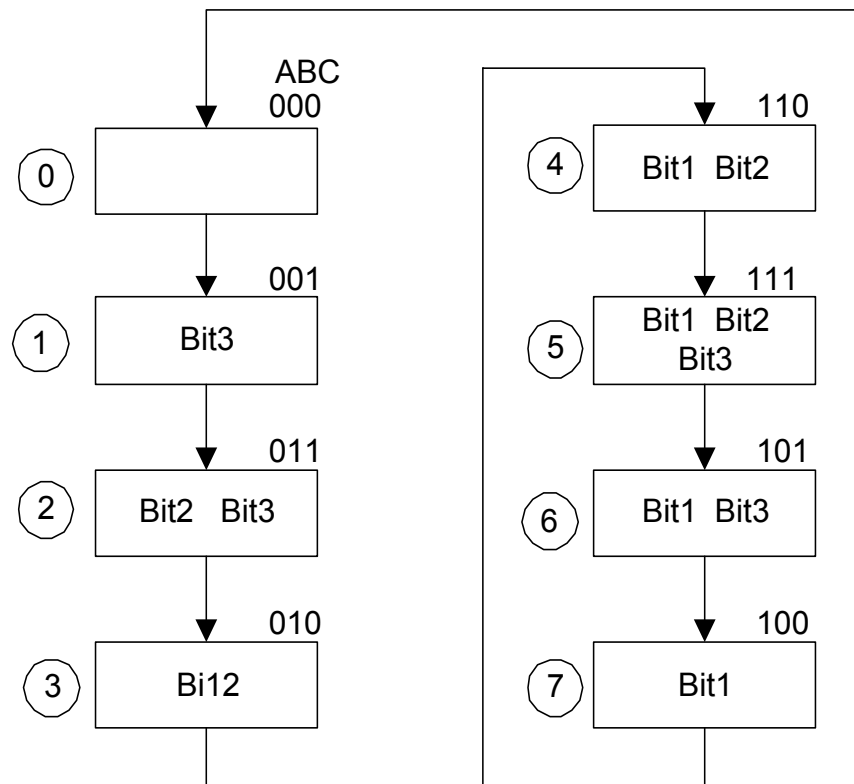
Conditional O/P Box จะมีทางเข้าซึ่งออกมาจาก Decision Box เท่านั้น ภายใน Box จะเป็นรายชื่อ O/P ที่ Active เนื่องจาก ทางเข้าของ Conditional O/P Box มาจาก Decision Box นั่นคือ รายชื่อ O/P ใน Conditional O/P box นี้จะ Active เมื่อได้รับผลอย่างใดอย่างหนึ่ง (จริง หรือ เท็จ) จาก Decision Box นั้น นั่นก็หมายถึง O/P เหล่านี้จะขึ้นอยู่กับทั้ง State และ I/P ที่ใช้ทดสอบเงื่อนไขใน Decision Box นั้น



# ตัวอย่างการใช้ ASM Chart กับวงจรนับ



ASM Chart ของวงจร 3 Bit Binary Counter

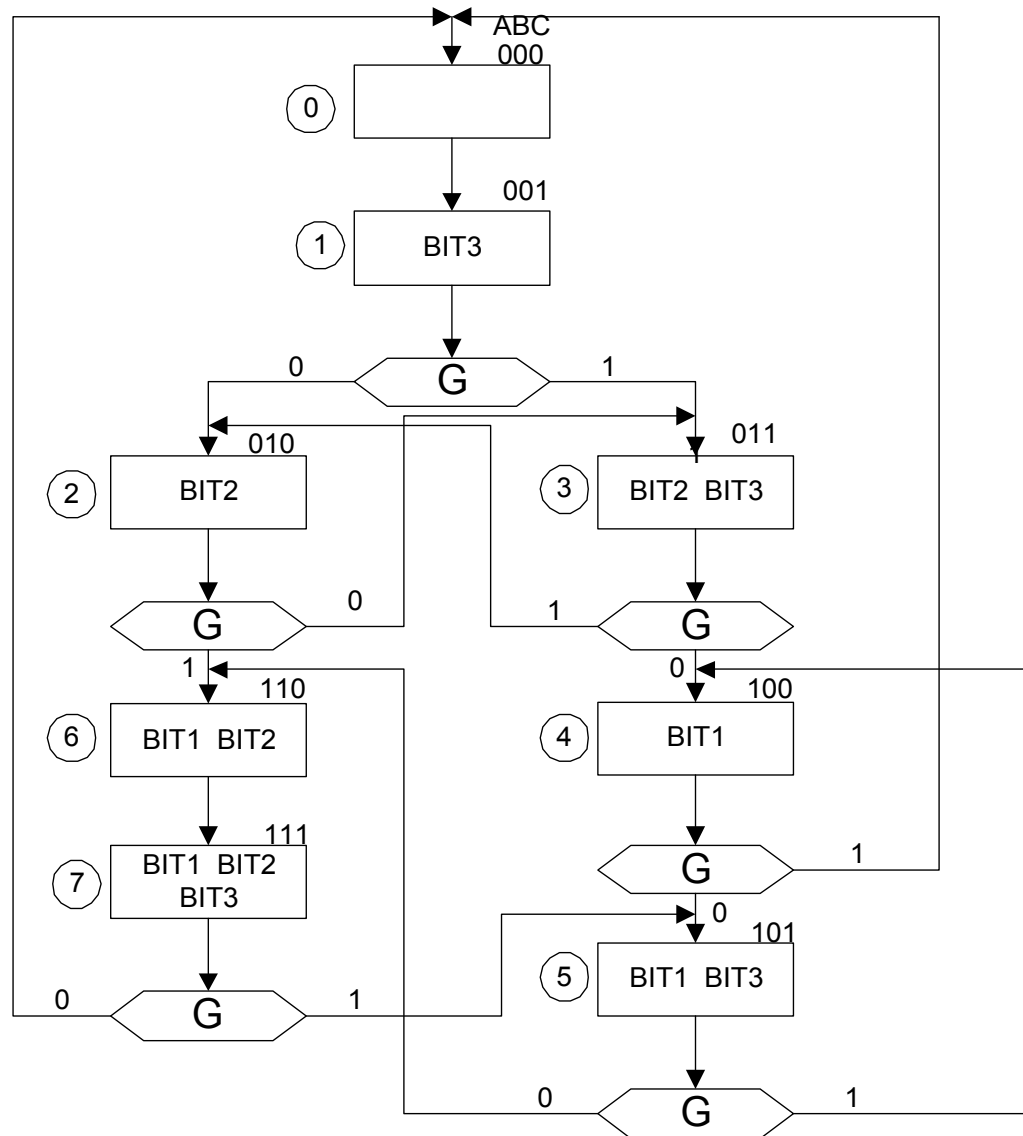


ASM Chart ของวงจร 3 Bit Gray-Code Counter

# 3 bit Binary/Gray code Counter

G=0 นับ Binary

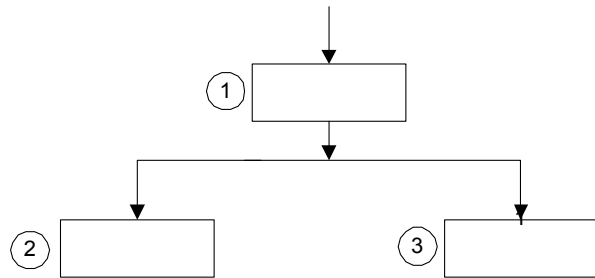
G=1 นับ Gray Code



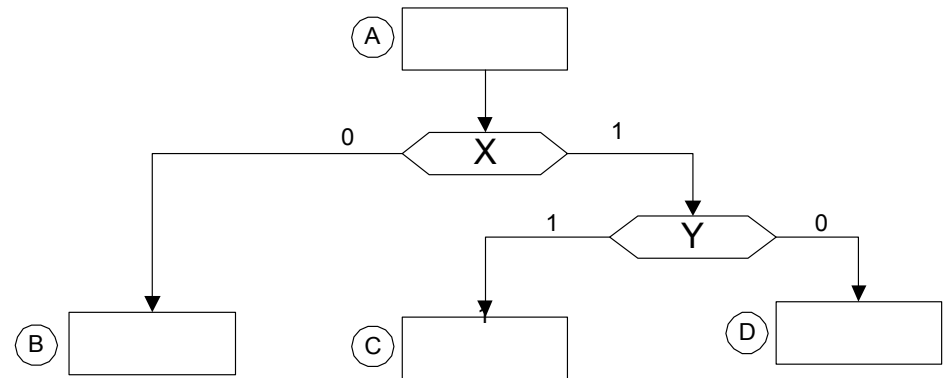
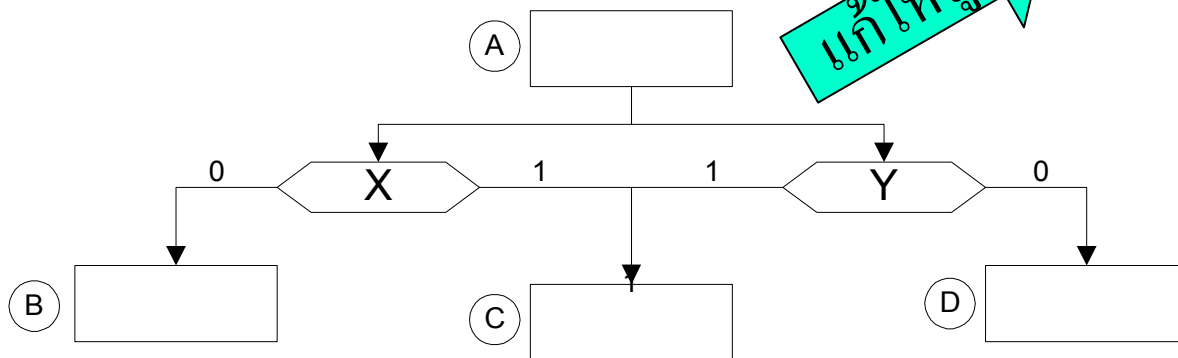
# ตัวอย่าง ASM Chart ที่ผิด

ผิดที่มีการวิ่งไป 2 Next State

ตัวอย่างที่ 1



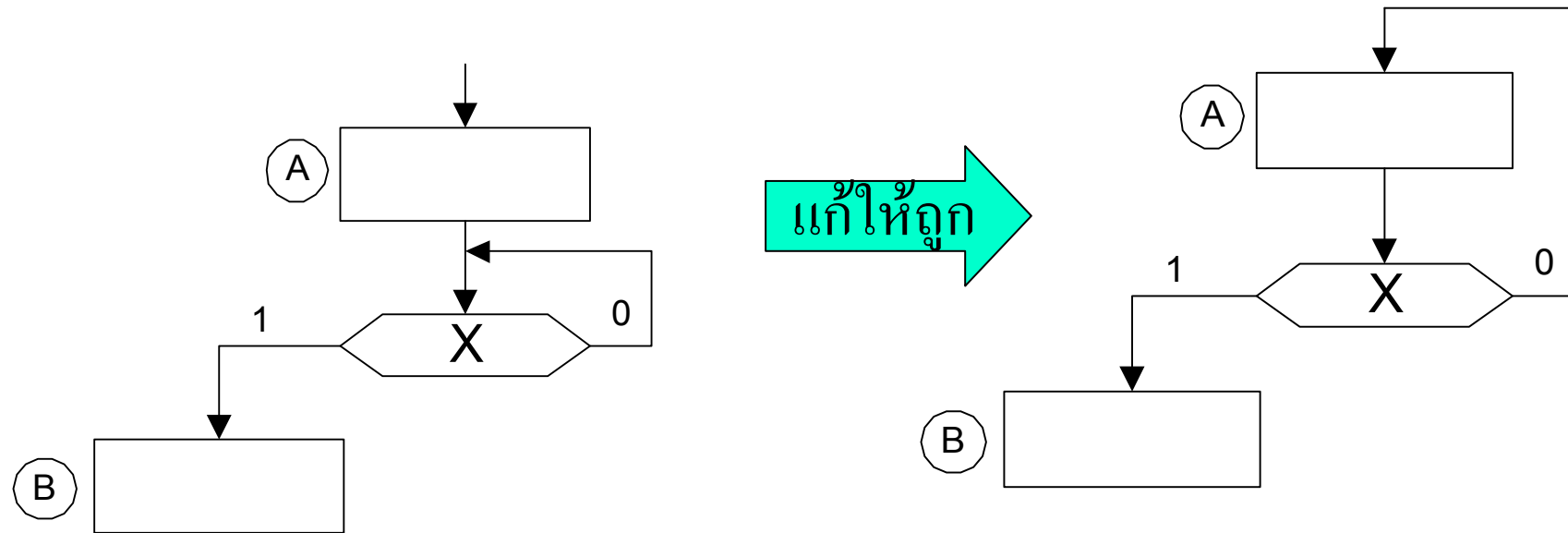
ตัวอย่างที่ 2



จะเห็นว่ามีการเช็คเงื่อนไขโดยใช้ X และ Y  
หาก  $X=0$  และ  $Y=0$  ในเวลาเดียวกัน  
จะมีการวิ่งไป 2 State คือ B และ D  
หรือ  $X=0, Y=1$   
ก็วิ่งไป 2 State คือ B และ C

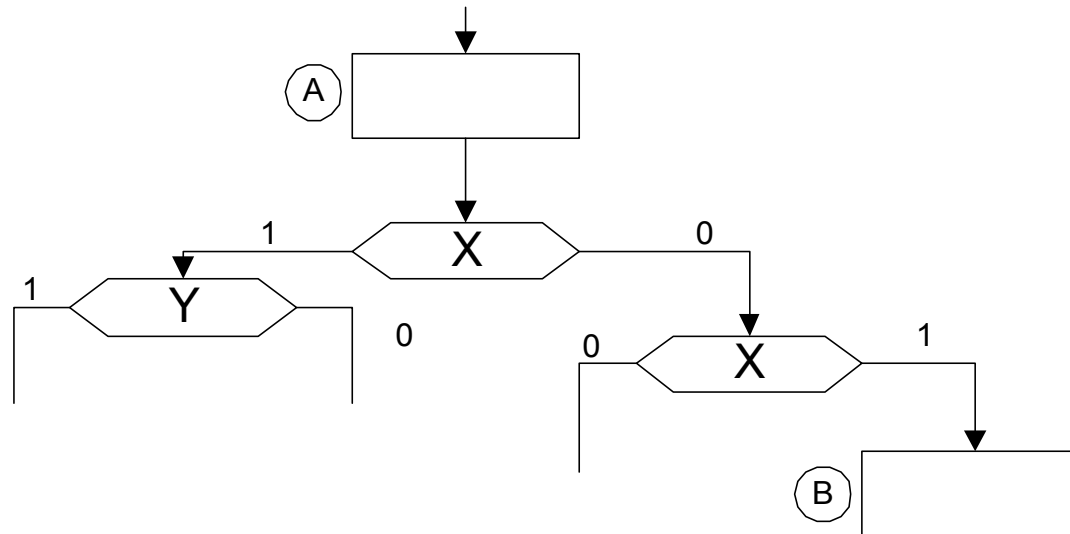
# ตัวอย่าง ASM Chart ที่ผิด

## ตัวอย่างที่ 3 ผิดเพราะมีวงรอบใน ASM Block



# ตัวอย่าง ASM Chart ที่ผิด

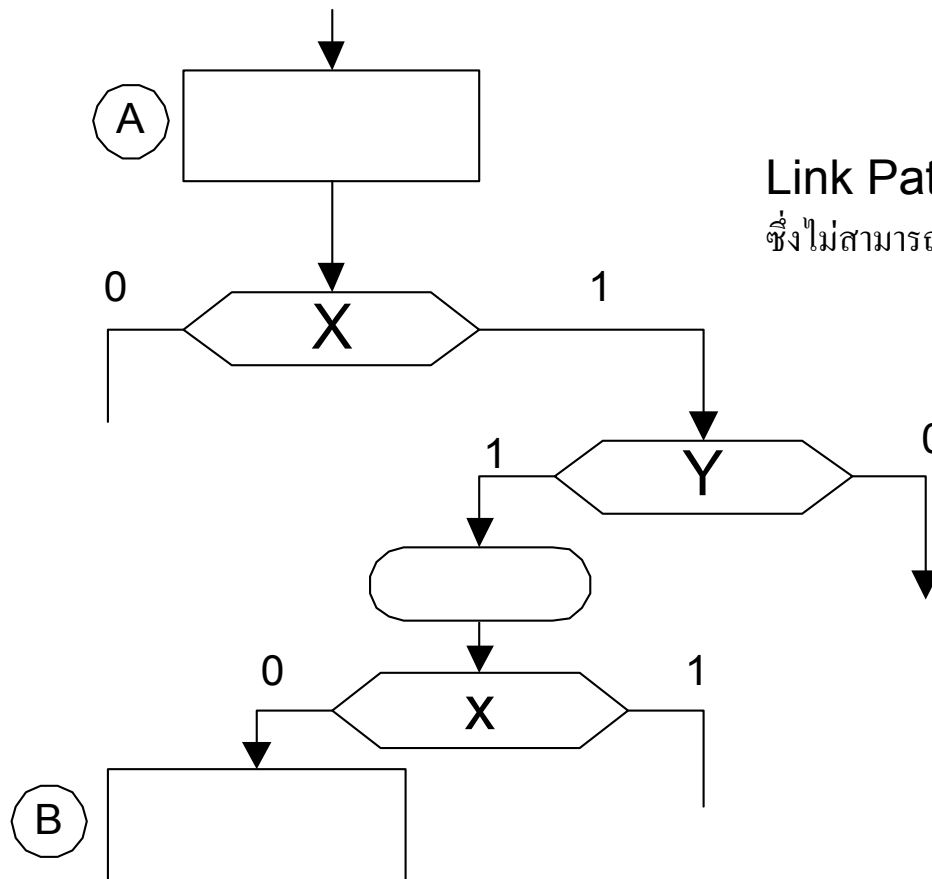
ตัวอย่างที่ 4 ผิดเพราะมีเส้นทางที่ไม่สามารถวิ่งไปได้



จะเห็นว่า Link Path จาก State A ไป B ผ่าน Condition แรกคือ **X=0** แต่ Condition ที่ 2 คือ **X=1** ซึ่งเป็นไปไม่ได้ที่ X จะเป็น 0, 1 ในเวลาเดียวกัน

# ตัวอย่าง ASM Chart ที่ผิด

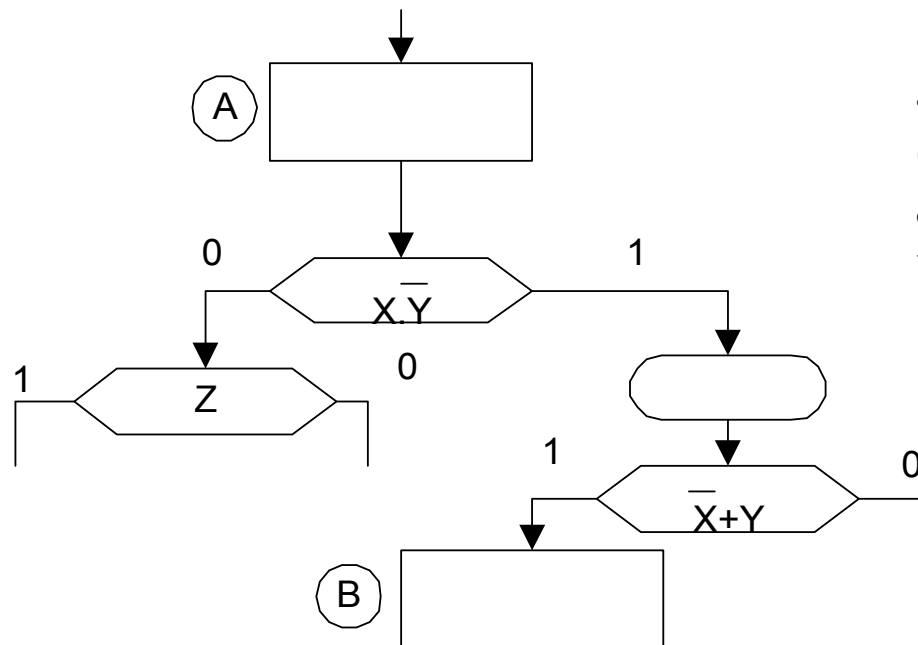
ตัวอย่างที่ 5 มีเส้นทางที่ไม่สามารถวิ่งไปได้



Link Path จาก A ไป B เงื่อนไขคือ  $x=1, y=1, x=0$   
ซึ่งไม่สามารถเป็นไปได้

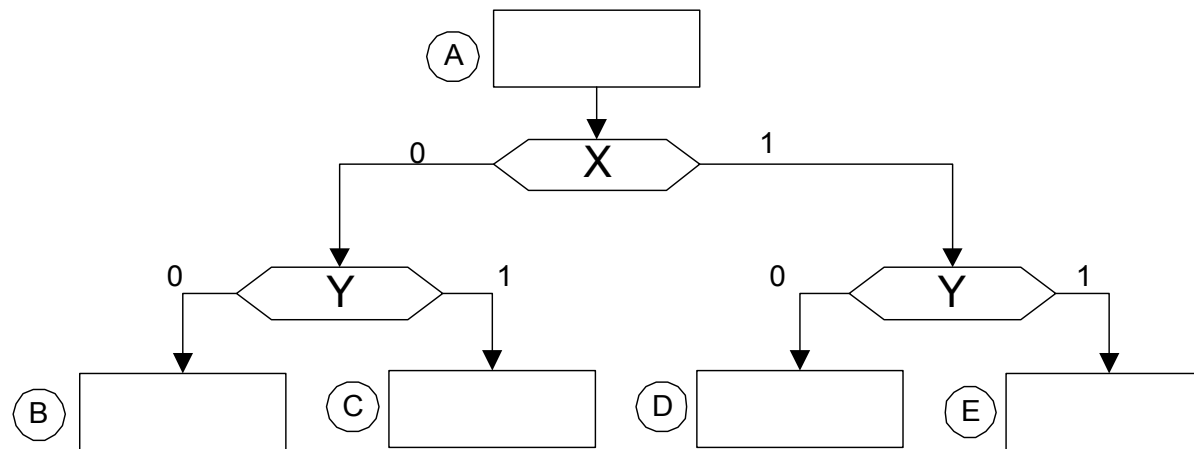
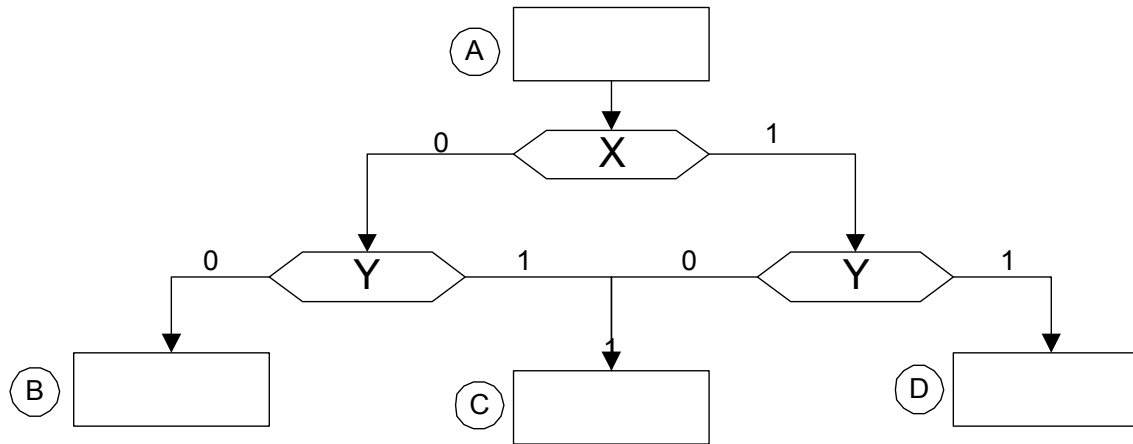
# ตัวอย่าง ASM Chart ที่ผิด

## ตัวอย่างที่ 6 มีเส้นทางที่ไม่สามารถวิ่งไปได้



Link Path จาก A ไป B ผ่าน Condition ที่ 1  
คือ  $X \cdot Y = 1$  คือ  $X=1, Y=0$  และ  
Condition ที่ 2 คือ  $X + Y = 1$  ซึ่ง  $X=0, Y=1$   
ซึ่งเป็นไปไม่ได้ เพราะ Condition ทั้ง 2  
ไม่สามารถเป็นจริงในเวลาเดียวกัน

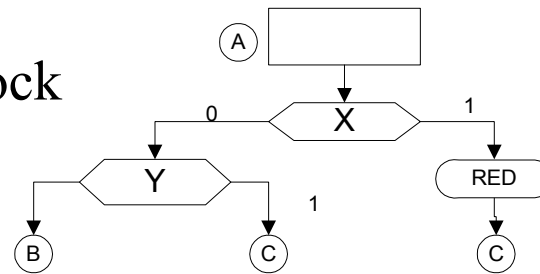
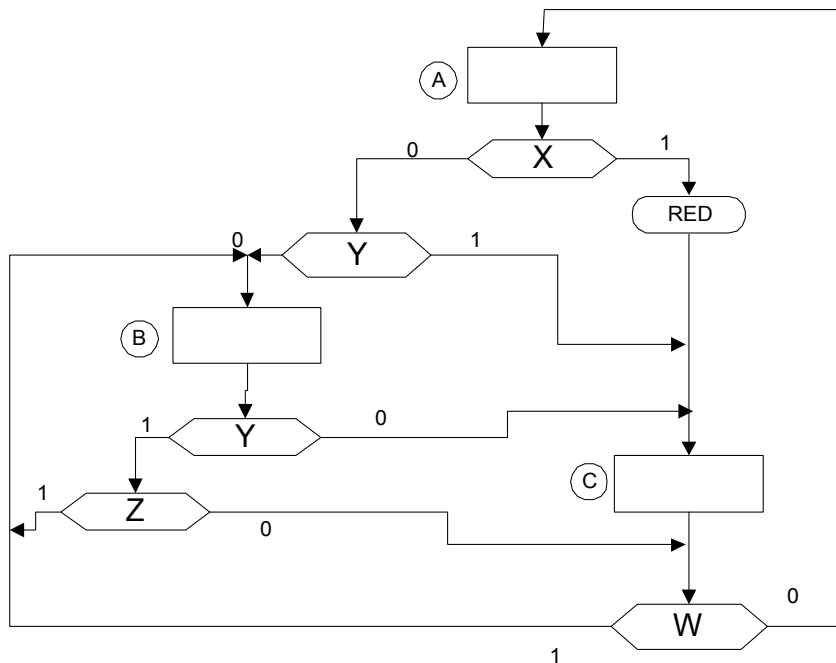
# ตัวอย่างการใช้ตัวแปรหลายตัวในโครงข่ายการตัดสินใจ อย่างถูกต้อง





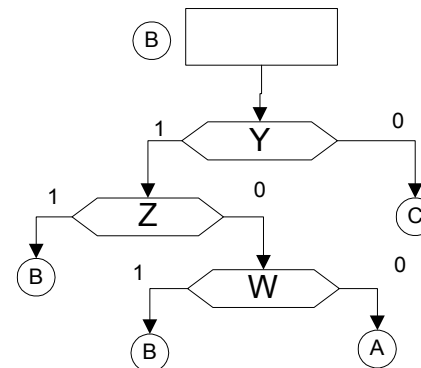
# การใช้ Decision Box ร่วมกันของหลาย ASM Block

บางครั้งเพื่อความกระชับรัดขอ ASM Chart อาจมีการใช้โครงข่ายการตัดสินใจร่วมกันของหลาย ASM Block



$$\textcircled{A} \rightarrow \textcircled{B} = \bar{X}\bar{Y}$$

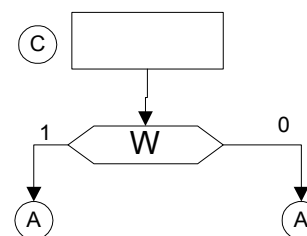
$$\textcircled{A} \rightarrow \textcircled{C} = X + \bar{X}Y$$



$$\textcircled{B} \rightarrow \textcircled{B} = YZ + YZW$$

$$\textcircled{B} \rightarrow \textcircled{A} = Y\bar{Z}\bar{W}$$

$$\textcircled{B} \rightarrow \textcircled{C} = \bar{Y}$$



$$\textcircled{C} \rightarrow \textcircled{B} = W$$

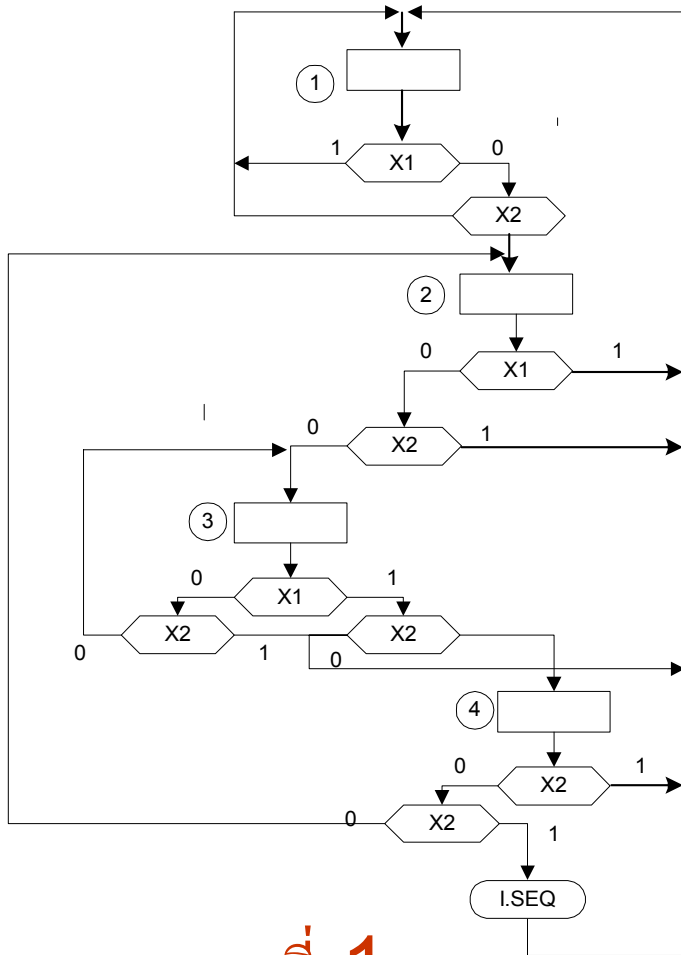
$$\textcircled{C} \rightarrow \textcircled{A} = \bar{W}$$

## ตัวอย่างการออกแบบ ASM Chart

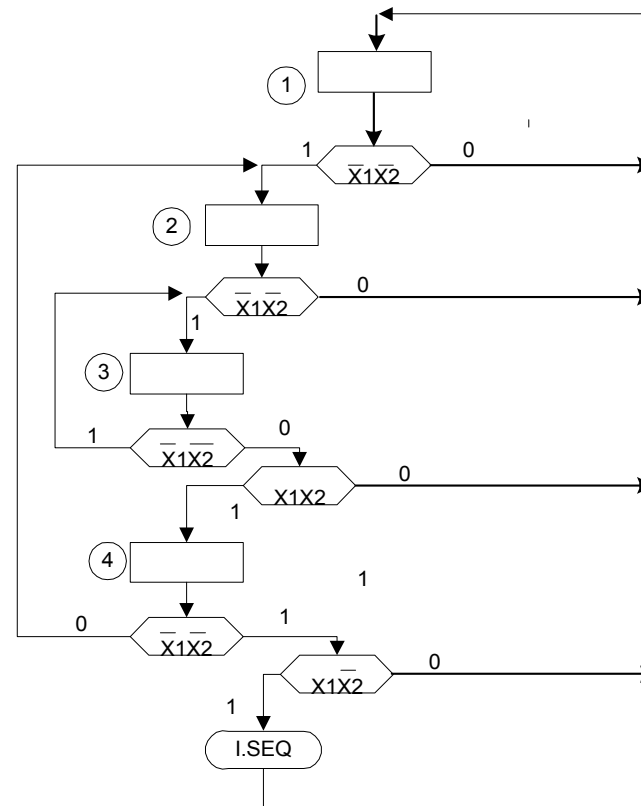
จงออกแบบวงจร Sequence Detector ของข้อมูล 2 บิต วงจรมี 2 I/P คือ  
ข้อมูล 2 บิตนั้น และ 1 O/P ซึ่งจะเป็น 1 เมื่อ I/P มีลำดับเป็น 00, 00,  
11,10 นอกจากนี้ O/P จะเป็น 0 เช่น

I/P Sequence	x1	=	0 0 0 0 1 1 0 0 1 0 0 1 1
	x2	=	0 1 0 0 1 0 0 0 0 0 0 1 0
O/P Sequence	SEQ	=	0 0 0 0 0 1 0 0 0 0 0 0 0

# ตัวอย่างการออกแบบ ASM Chart



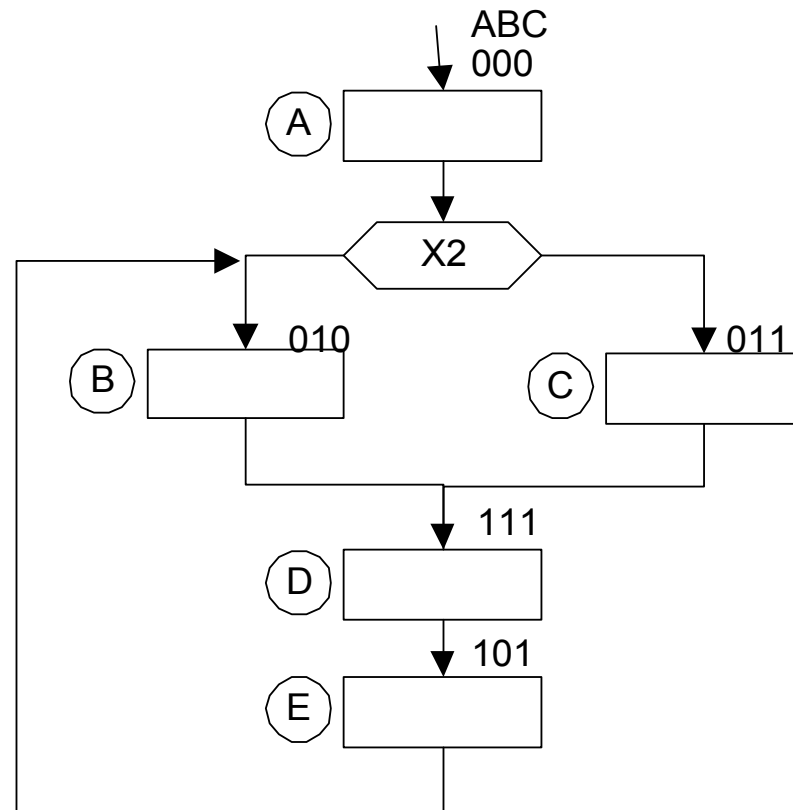
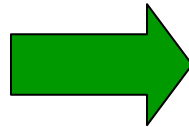
แบบที่ 1



แบบที่ 1

# State Assignment

State Name	State Code		
	A	B	C
A	0	0	0
B	0	1	0
C	0	1	1
D	1	1	1
E	1	0	1





## วิธี Minimum State Locus

Minimum State Locus คือ วิธีการกำหนดรหัสของ State ให้ได้ฟังก์ชันที่มี  
ความซับซ้อนน้อย

วิธีทำก็คือ เขียนลำดับของ Link Path ทั้งหมดออกมา แล้วทดลองแทนชื่อ  
State ด้วย State Code แบบต่างๆ แล้วหาจำนวนบิตที่เปลี่ยนค่าในแต่ละ  
Link Path แล้วนำมาบวกกัน เพื่อหา State Locus แล้วเลือกเอาแบบที่มี  
State Locus น้อยที่สุด



## วิธี Minimum State Locus

A → B	=	000 → 010	=	1 บิต
A → C	=	000 → 011	=	2 บิต
B → D	=	010 → 111	=	2 บิต
C → D	=	011 → 111	=	1 บิต
D → E	=	111 → 101	=	1 บิต
E → B	=	101 → 010	=	3 บิต
<b>State Locus =</b>				<b>10 บิต</b>

## วิธี Minimum State Locus

A → B	=	000 → 010	=	1 บิต
A → C	=	000 → 100	=	1 บิต
B → D	=	010 → 110	=	1 บิต
C → D	=	100 → 110	=	1 บิต
D → E	=	110 → 111	=	1 บิต
E → B	=	111 → 010	=	2 บิต
		<b>State Locus</b>	=	<b>7 บิต</b>

ถือว่าเป็น Minimum State Locus



# การสร้างตาราง ASM และการหา NS Function และ O/P Function

ASM Table นั้น จะมี 3 Table คือ

- Next State Table หรือ State Transition Table
- State O/P Table
- Conditional O/P Table



# การสร้างตาราง ASM และการหา NS Function และ O/P

## Function

วงจรมี 3 I/P คือ

Q1, Q2, Q3 ซึ่ง Active High ทั้งหมด

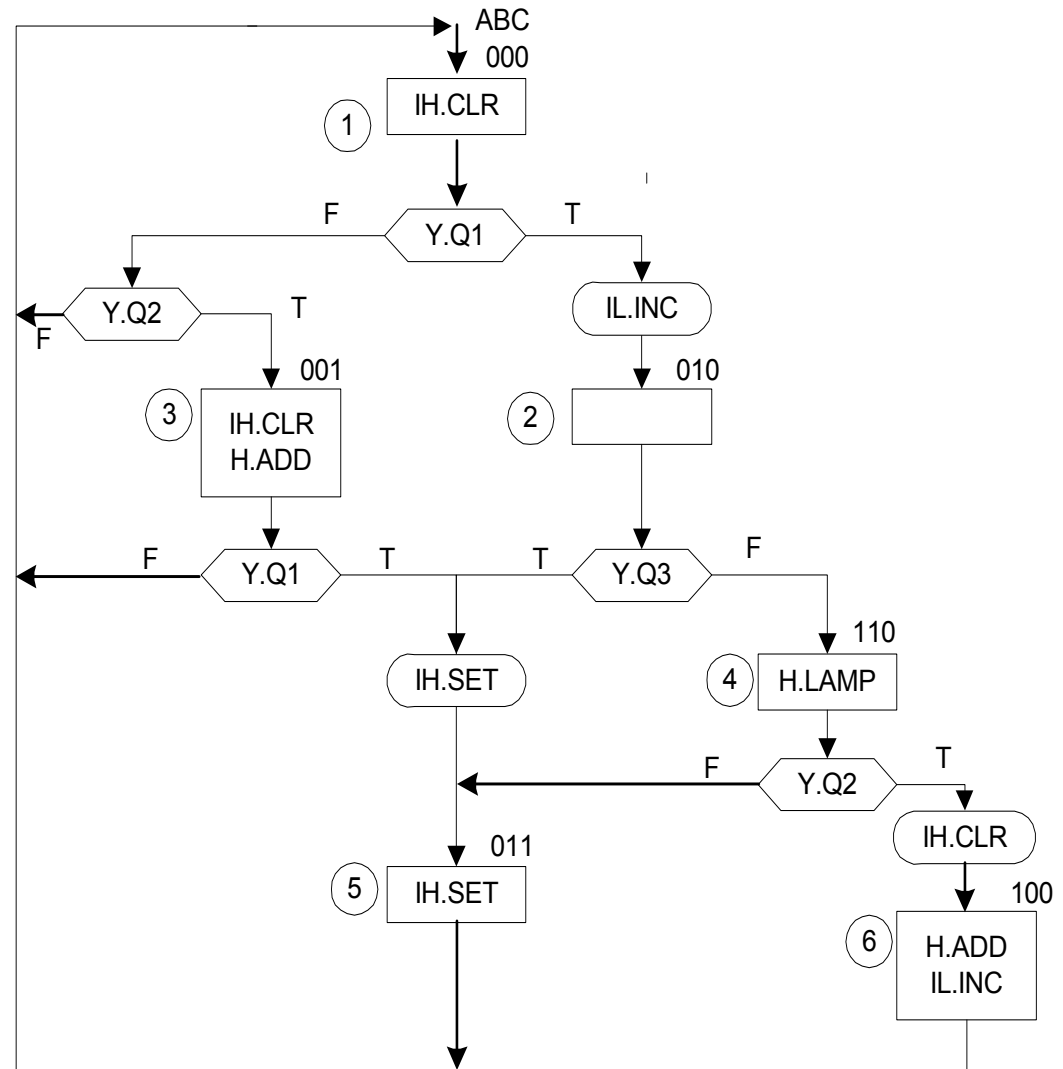
5 O/P คือ

CLR, SET เป็น immediate และ Active High

INC เป็น immediate และ Active Low

ADD, LAMP เป็น State O/P และ Active High

วงจรมีใช้ A, B, C เป็น State Variable



# การสร้างตาราง ASM และการหา NS Function และ O/P

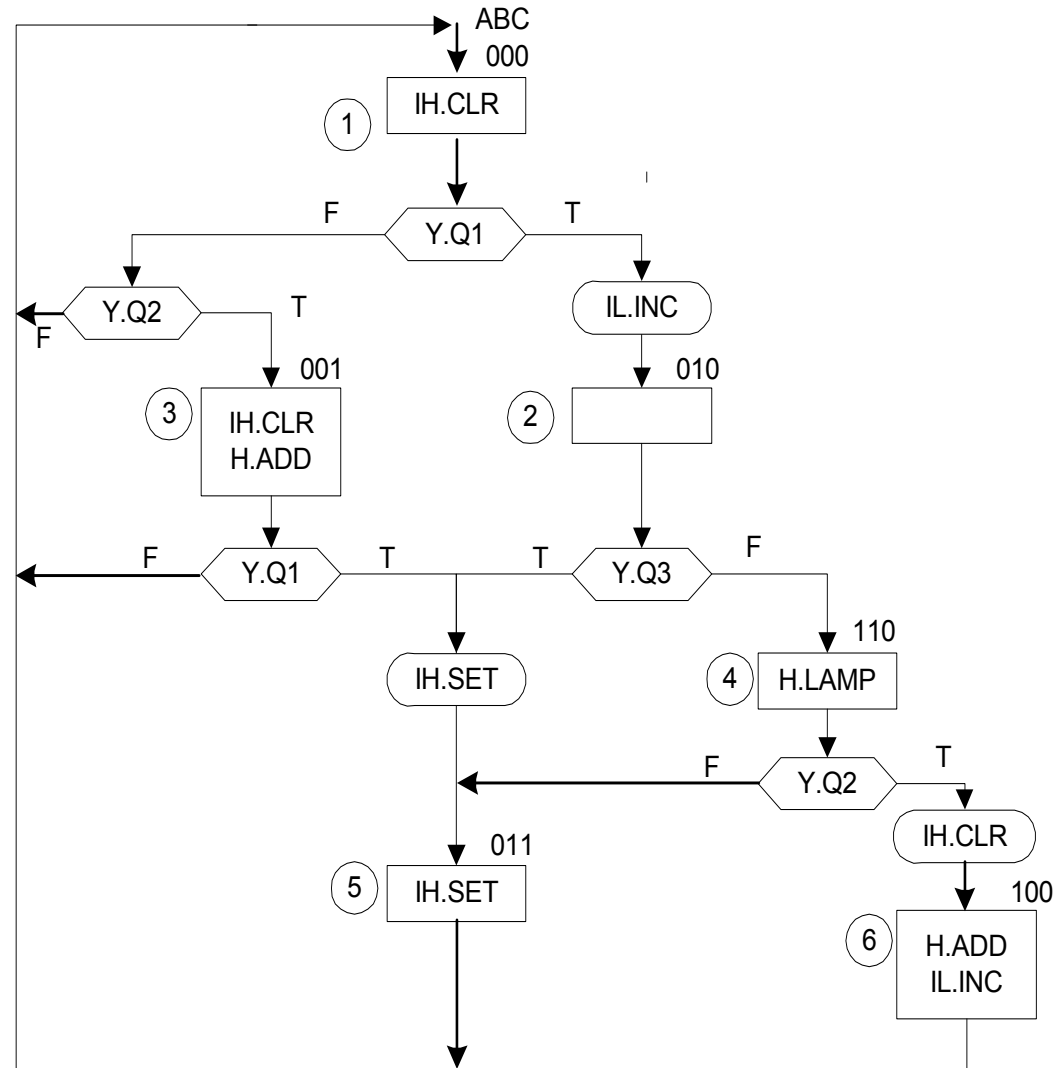
## Function

ก่อนอื่นเราต้องหา Link Path ทั้งหมดใน ASM Chart

ให้ B = ASM Block และ L = Link Path

เราจะเห็นว่าวงจรนี้มี 6 ASM Block และ 11 Link Path

B1L1	:	1	→	1	โดย	Q1Q2
B1L2	:	1	→	2	โดย	Q1
B1L3	:	1	→	3	โดย	Q1Q2
B2L4	:	2	→	4	โดย	Q3
B2L5	:	2	→	5	โดย	Q3
B3L6	:	3	→	1	โดย	Q1
B3L7	:	3	→	5	โดย	Q1
B4L8	:	4	→	5	โดย	Q2
B4L9	:	4	→	6	โดย	Q2
B5L10	:	5	→	1	โดย	-
B6L11	:	6	→	1	โดย	-





# 1. Next State Table

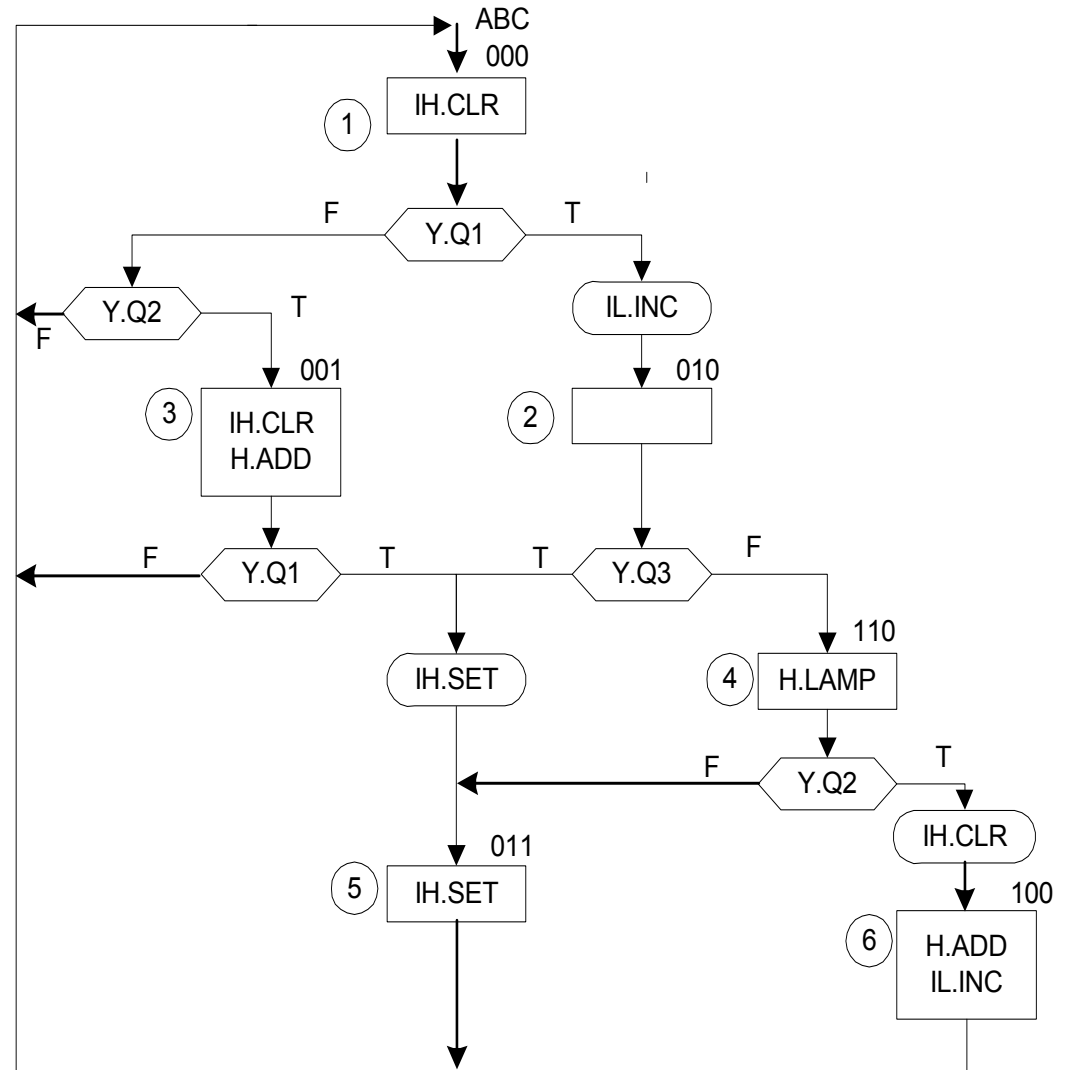
มี 3 ส่วน คือ

- ส่วนที่ 1 เป็น I/P ของวงจร
- ส่วนที่ 2 เป็น Present State
- ส่วนที่ 3 เป็น Next State

# Next State Table

(a) Symbolic Form

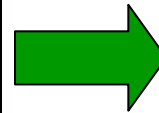
I/P Y.Q1 Y.Q2 Y.Q3			PS	NS	Comment
0	0	-	1	1	B1L1
1	-	-	1	2	B1L2
0	1	-	1	3	B1L3
-	-	0	2	4	B2L4
-	-	1	2	5	B2L5
0	-	-	3	1	B3L6
1	-	-	3	5	B3L7
-	0	-	4	5	B4L8
-	1	-	4	6	B4L9
-	-	-	5	1	B5L10
-	-	-	6	1	B6L11



## Next State Table

(a) Symbolic Form

I/P			PS	NS	Comment
Y.Q1	Y.Q2	Y.Q3			
0	0	-	1	1	B1L1
1	-	-	1	2	B1L2
0	1	-	1	3	B1L3
-	-	0	2	4	B2L4
-	-	1	2	5	B2L5
0	-	-	3	1	B3L6
1	-	-	3	5	B3L7
-	0	-	4	5	B4L8
-	1	-	4	6	B4L9
-	-	-	5	1	B5L10
-	-	-	6	1	B6L11



(b) Assigned Form

I/P			PS			NS		
Q1	Q2	Q3	A	B	C	NA	NB	NC
0	0	-	0	0	0	0	0	0
1	-	-	0	0	0	0	1	0
0	1	-	0	0	0	0	0	1
-	-	0	0	1	0	1	1	0
-	-	1	0	1	0	0	1	1
0	-	-	0	0	1	0	0	0
1	-	-	0	0	1	0	1	1
-	0	-	1	1	0	0	1	1
-	1	-	1	1	0	1	0	0
-	-	-	0	1	1	0	0	0
-	-	-	1	0	0	0	0	0

# ถ้าใช้ D flip-flop เขียน Excitation Table

Excitation Table

I/P			P/S			N/S			Memory I/P		
Q1	Q2	Q3	A	B	C	NA	NB	NC	DA	DB	DC
0	0	-	0	0	0	0	0	0	0	0	0
1	-	-	0	0	0	0	1	0	0	1	0
0	1	-	0	0	0	0	0	1	0	0	1
-	-	0	0	1	0	1	1	0	1	1	0
-	-	1	0	1	0	0	1	1	0	1	1
0	-	-	0	0	1	0	0	0	0	0	0
1	-	-	0	0	1	0	1	1	0	1	1
-	0	-	1	1	0	0	1	1	0	1	1
-	1	-	1	1	0	1	0	0	1	0	0
-	-	-	0	1	1	0	0	0	0	0	0
-	-	-	1	0	0	0	0	0	0	0	0

		BC			
		00	01	11	10
DA	A	0	0	0	$\overline{Q3}$
	1	0	-	-	$Q2$

$$DA = ABQ2 + \overline{A}B\overline{C}Q3$$

		BC			
		00	01	11	10
DB	A	$\overline{Q1}$	$\overline{Q1}$	0	1
	1	0	-	-	$\overline{Q2}$

$$DB = \overline{A}\overline{B}Q1 + \overline{A}B\overline{C} + ABQ3$$

		BC			
		00	01	11	10
DC	A	$\overline{Q1}Q2$	$Q1$	0	$Q3$
	1	0	-	-	$\overline{Q2}$

$$DC = \overline{A}\overline{B}\overline{C}Q1Q2 + \overline{B}CQ1 + \overline{A}B\overline{C}Q3 + AB\overline{Q2}$$

# ทำให้ JK Flip-Flop เขียน Excitation Table

I/P			PS			NS			Memory I/P					
Q1	Q2	Q3	A	B	C	NA	NB	NC	JA	KA	JB	KB	JC	KC
0	0	-	0	0	0	0	0	0	0	-	0	-	0	-
1	-	-	0	0	0	0	1	0	0	-	1	-	0	-
0	1	-	0	0	0	0	0	1	0	-	0	-	1	-
-	-	0	0	1	0	1	1	0	1	-	-	0	0	-
-	-	1	0	1	0	0	1	1	0	-	-	0	1	-
0	-	-	0	0	1	0	0	0	0	-	0	-	-	1
1	-	-	0	0	1	0	1	1	0	-	1	-	-	0
-	0	-	1	1	0	0	1	1	-	1	-	0	1	-
-	1	-	1	1	0	1	0	0	-	0	-	1	0	-
-	-	-	0	1	1	0	0	0	0	-	-	1	-	1
-	-	-	1	0	0	0	0	0	-	1	0	-	-	0

JA

BC	00	01	11	10
A=0	0	0	0	$\overline{Q3}$
A=1	-	-	-	-

KA

BC	00	01	11	10
A=0	-	-	-	-
A=1	-	-	-	$\overline{Q2}$

JB

BC	00	01	11	10
A=0	$\overline{Q1}$	$\overline{Q1}$	-	-
A=1	0	-	-	-

KB

BC	00	01	11	10
A=0	-	-	1	0
A=1	-	-	-	$\overline{Q2}$

JC

BC	00	01	11	10
A=0	$\overline{Q1}Q2$	-	-	$\overline{Q2}$
A=1	-	-	-	$\overline{Q2}$

KC

BC	00	01	11	10
A=0	-	$\overline{Q1}$	1	-
A=1	0	-	-	-

$$JA = B\overline{C}\overline{Q3}$$

$$JB = \overline{A}Q1$$

$$JC = \overline{A}\overline{Q1}Q2 + \overline{A}BQ2 + A\overline{Q2}$$

$$KA = \overline{B} + \overline{Q2}$$

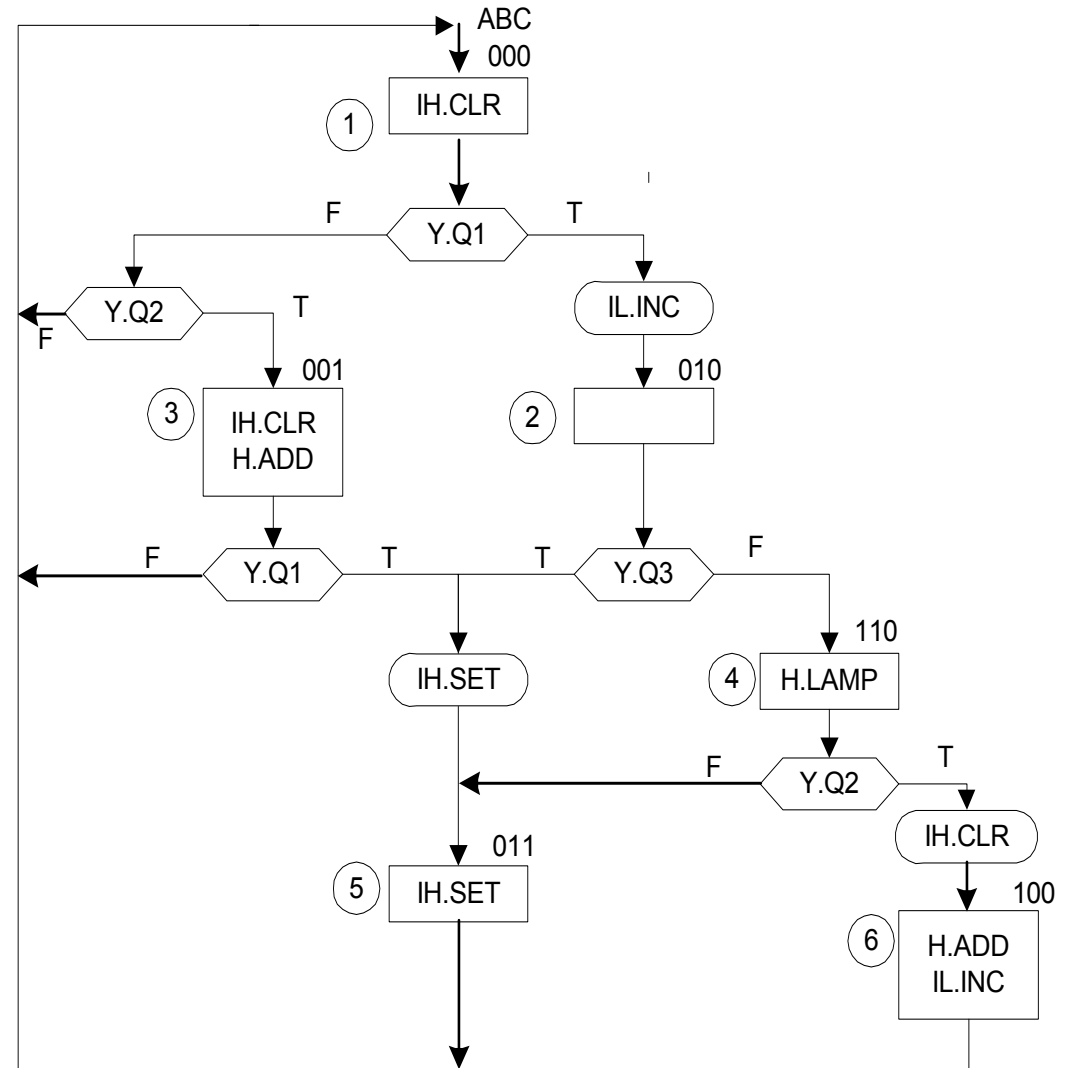
$$KB = C + A\overline{Q2}$$

$$KC = B + \overline{A}\overline{Q1}$$

## 2. State Output Table

(a) Symbolic Form

FS	O/P	
	H.ADD	H.LAMP
1		
2		
3	A	
4		A
5		A
6	A	





## 2. State Output Table

(a) Symbolic Form

PS	O/P	
	H.ADD	H.LAMP
1		
2		
3	A	
4		A
5		A
6	A	

(b) Assigned Form

PS			O/P	
A	B	C	H.ADD	H.LAMP
0	0	0	0	0
0	1	0	0	0
0	0	1	1	0
1	1	0	0	1
0	1	1	0	1
1	0	0	1	0

H.ADD =  $\bar{B}C + A\bar{B}$

H.ADD		BC			
A		00	01	11	10
0		0	1	0	0
1		1	-	-	0

H.LAMP

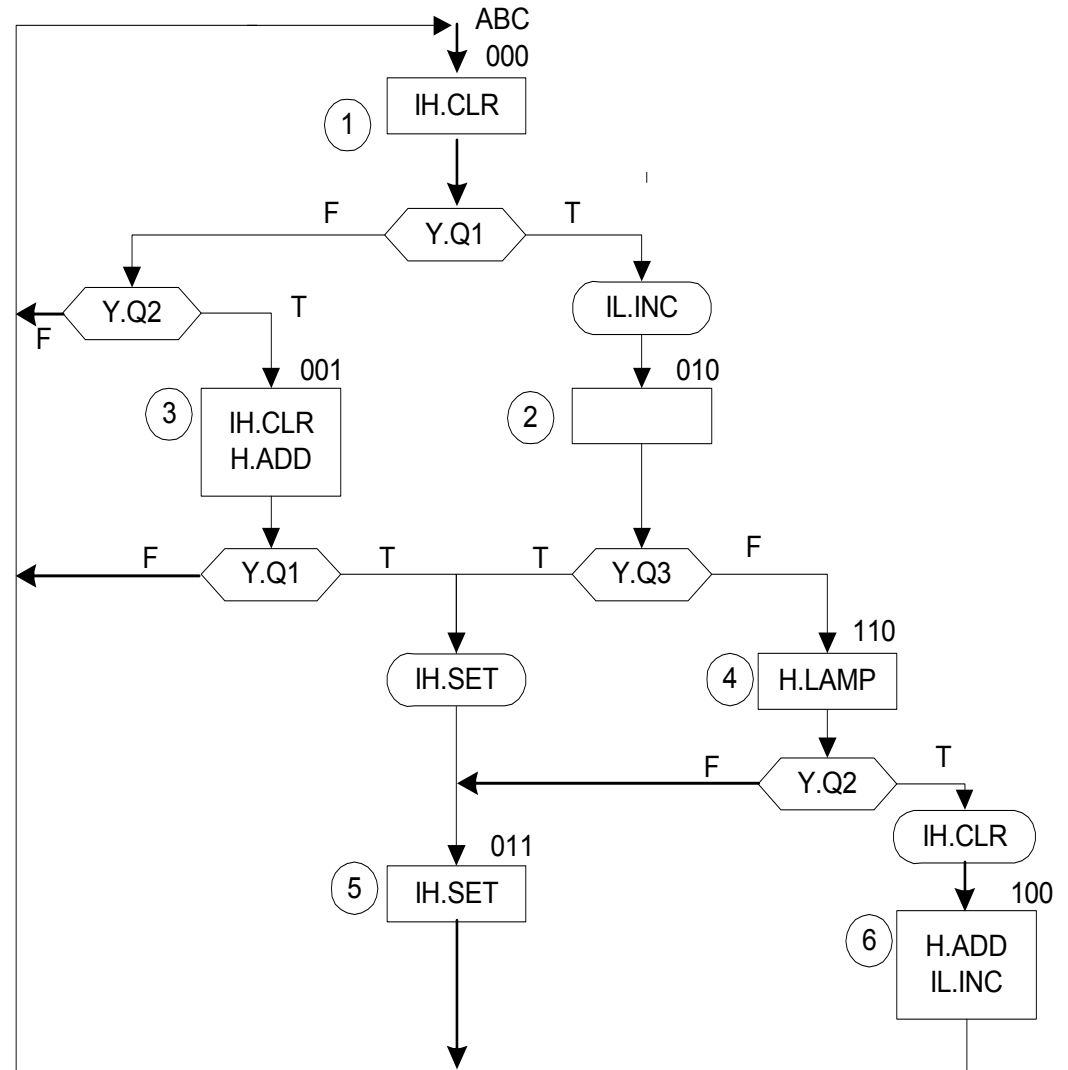
H.LAMP		BC			
A		00	01	11	10
0		0	0	1	0
1		0	-	-	1

H.LAMP =  $BC + AB$

# 3. Conditional Output Table

(a) Symbolic Form

I/P			PS	Conditional O/P		
Y.Q <sub>1</sub>	Y.Q <sub>2</sub>	Y.Q <sub>3</sub>		IL.INC	IH.SET	IH.CLR
0	0	-	1			A
1	-	-	1	A		A
0	1	-	1			A
-	-	0	2			
-	-	1	2		A	
0	-	-	3			A
1	-	-	3		A	A
-	0	-	4			
-	1	-	4			A
-	-	-	5		A	
-	-	-	6	A		



### 3. Conditional Output Table

(a) Symbolic Form

I/P	PS	Conditional O/P
Y.Q <sub>1</sub> Y.Q <sub>2</sub> Y.Q <sub>3</sub>		IL.INC IH.SET IH.CLR
0 0 -	1	A
1 - -	1	A A
0 1 -	1	A
- - 0	2	
- - 1	2	A
0 - -	3	A
1 - -	3	A A
- 0 -	4	
- 1 -	4	A
- - -	5	A
- - -	6	A

(b) Assigned Form

I/P	PS	O/P
Q <sub>1</sub> Q <sub>2</sub> Q <sub>3</sub>	A B C	INC SET CLR
0 0 -	0 0 0	1 0 1
1 - -	0 0 0	0 0 1
0 1 -	0 0 0	1 0 1
- - 0	0 1 0	1 0 0
- - 1	0 1 0	1 1 0
0 - -	0 0 1	1 0 1
1 - -	0 0 1	1 1 1
- 0 -	1 1 0	1 0 0
- 1 -	1 1 0	1 0 1
- - -	0 1 1	1 1 0
- - -	1 0 0	0 0 0

### 3. Conditional Output Table

(b) Assigned Form

I/P			PS			O/P		
Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>	A	B	C	INC	SET	CLR
0	0	-	0	0	0	1	0	1
1	-	-	0	0	0	0	0	1
0	1	-	0	0	0	1	0	1
-	-	0	0	1	0	1	0	0
-	-	1	0	1	0	1	1	0
0	-	-	0	0	1	1	0	1
1	-	-	0	0	1	1	1	1
-	0	-	1	1	0	1	0	0
-	1	-	1	1	0	1	0	1
-	-	-	0	1	1	1	1	0
-	-	-	1	0	0	0	0	0

IL.INC

A	BC			
	00	01	11	10
0	$\overline{Q_1}$	1	1	1
1	0	-	-	1

$$IL.INC = \overline{A}Q_2 + B + C$$

IH.SET

A	BC			
	00	01	11	10
0	0	$Q_1$	1	$Q_3$
1	0	-	-	0

$$IH.SET = ABQ_1 + ABC + ABQ_3$$

IH.CLR

A	BC			
	00	01	11	10
0	1	1	0	0
1	0	-	-	$Q_2$

$$IH.CLR = \overline{A}\overline{B} + ABQ_2$$

