
Chapter 5

Synchronous Sequential Logic

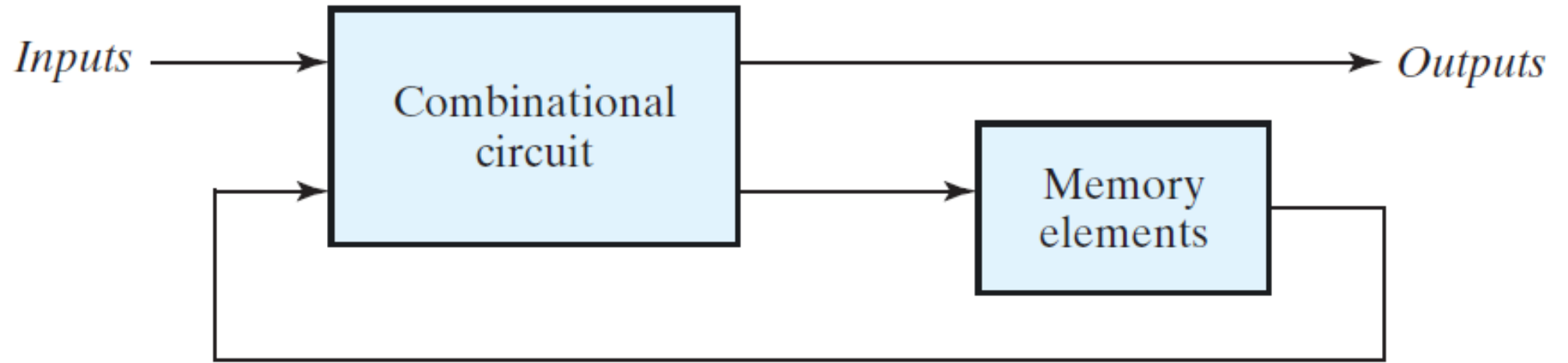


FIGURE 5.1
Block diagram of sequential circuit

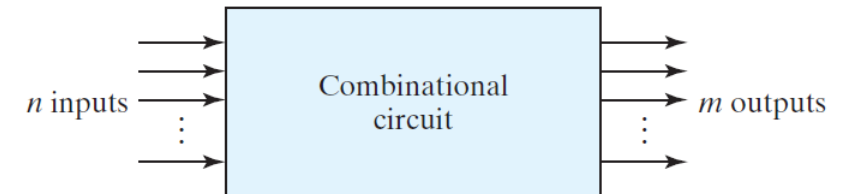
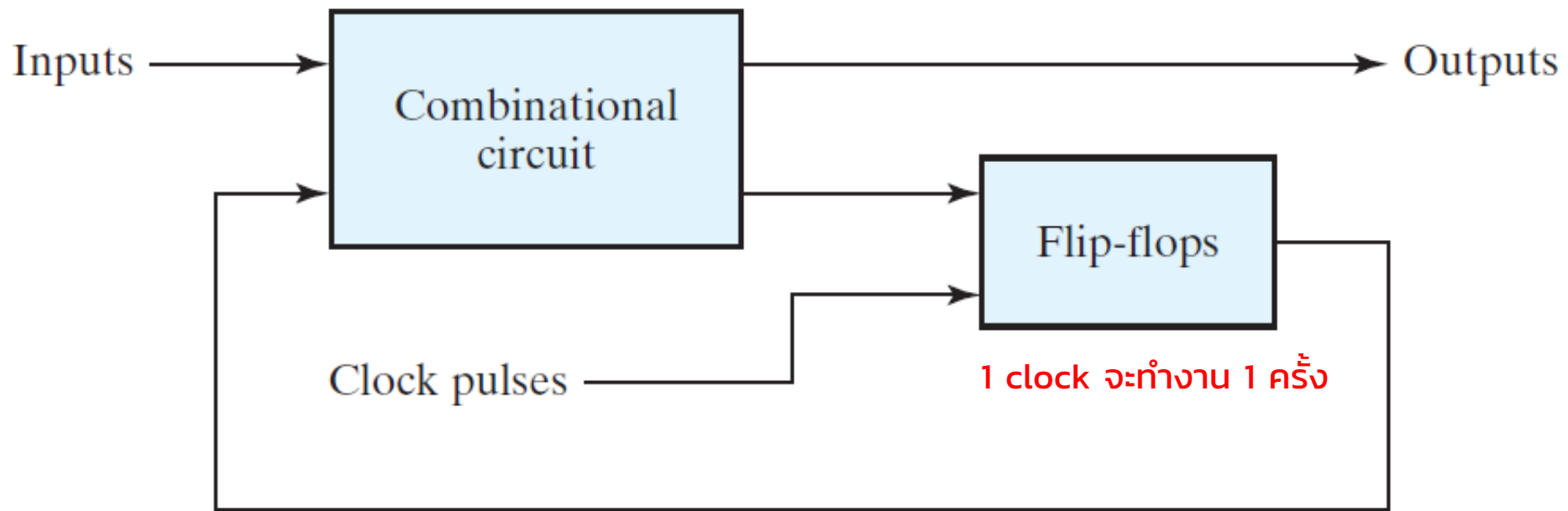


FIGURE 4.1
Block diagram of combinational circuit



(a) Block diagram



1 คาบ (period)

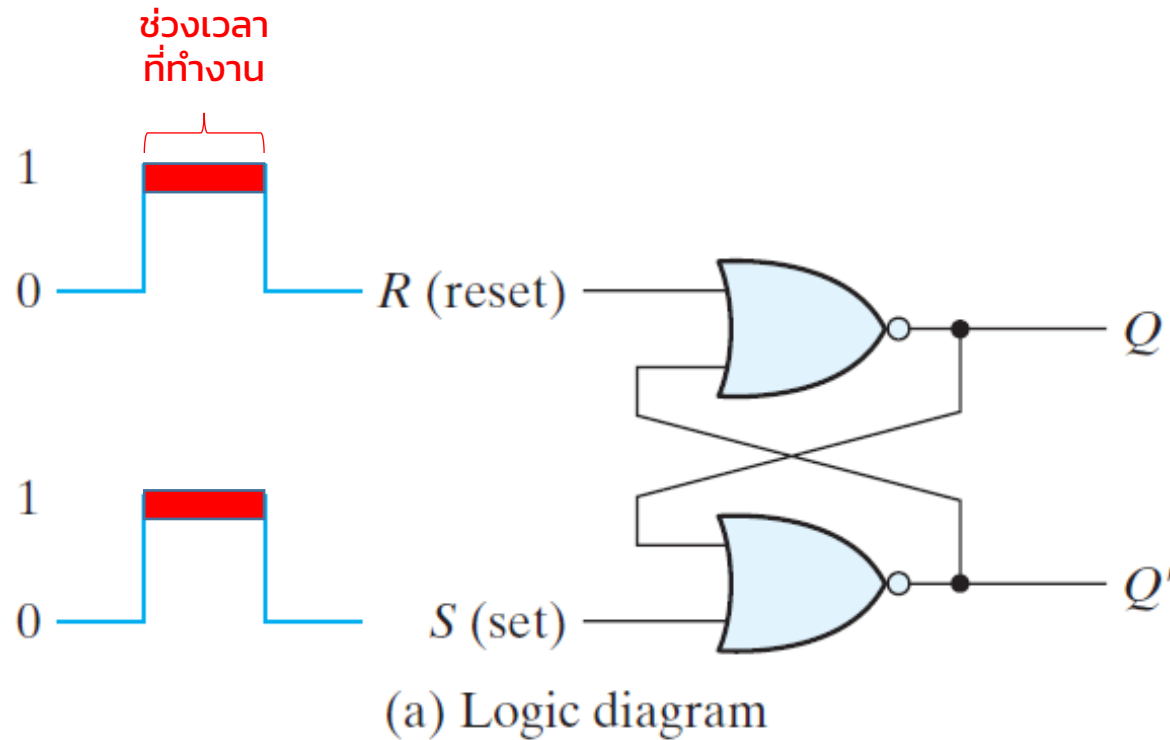
(b) Timing diagram of clock pulses

ความถี่ 1 Hz คือ 1 คาบ ใน 1 วินาที

ความถี่ 100 Hz คือ 100 คาบ ใน 1 วินาที

FIGURE 5.2
Synchronous clocked sequential circuit

SR Latch (หน่วยความจำขนาด 1 บิต)



S	R	Q	Q'
1	0	1	0
0	0	1	0 (after $S = 1, R = 0$)
0	1	0	1
0	0	0	1 (after $S = 0, R = 1$)
1	1	0	0 (forbidden)

(b) Function table

FIGURE 5.3
SR latch with NOR gates

SR Latch คือ memory ขนาด 1 บิต

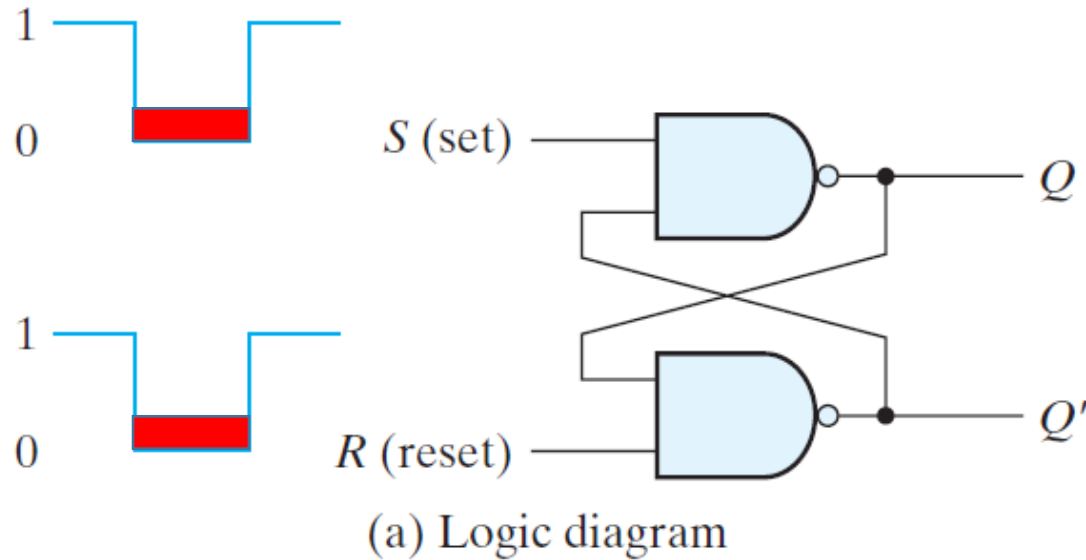
SR = 00 ไม่เปลี่ยนค่า

SR = 01 reset ($Q = 0$)

SR = 10 set ($Q = 1$)

SR = 11 ห้าม $S = R = 1$ พร้อมกัน

SR Latch



SR = 01 set ($Q = 1$) สลับ set/reset
SR = 10 reset ($Q = 0$)

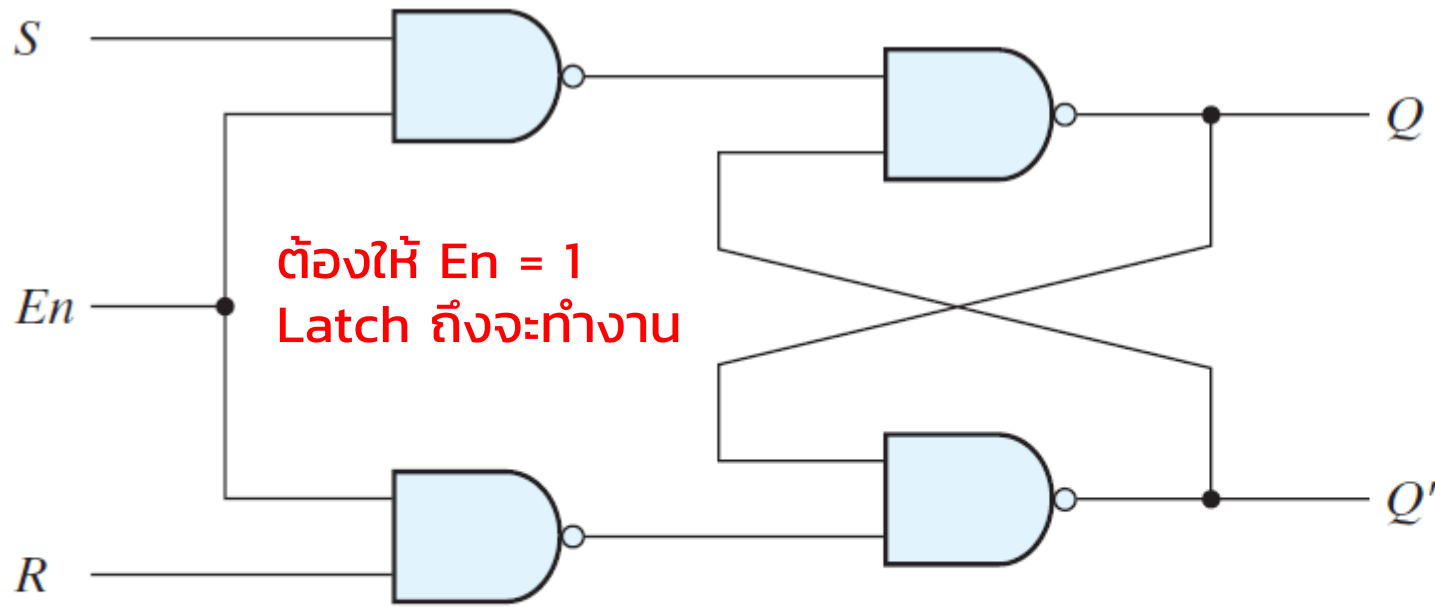
S	R	Q	Q'
1	0	0	1
1	1	0	1 (after $S = 1, R = 0$)
0	1	1	0
1	1	1	0 (after $S = 0, R = 1$)
0	0	1	1 (forbidden)

(b) Function table

FIGURE 5.4

SR latch with NAND gates

ถ้าสร้างจาก NAND ต้องใช้ negative clock เพื่อ set/reset



(a) Logic diagram

$SR = 01$ reset ($Q = 0$)

$SR = 10$ set ($Q = 1$)

สลับ set/reset อีกครั้ง
กลับไปเหมือนเดิม

En	S	R	Next state of Q
0	X	X	No change
1	0	0	No change
1	0	1	$Q = 0$; reset state
1	1	0	$Q = 1$; set state
1	1	1	Indeterminate

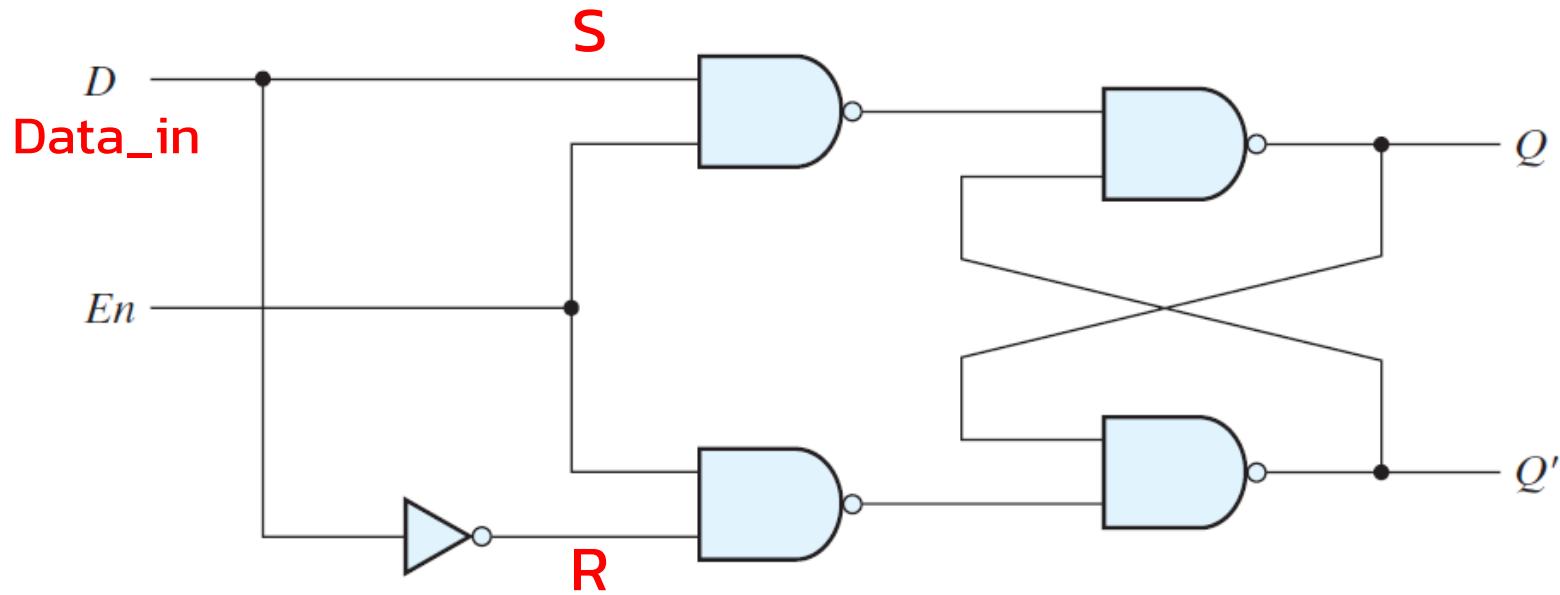
(b) Function table

FIGURE 5.5

SR latch with control input

ประโยชน์ของ En คือ เมื่อปิดการทำงานไว้ (En) ค่าใน memory จะไม่เปลี่ยนแปลง (ไม่ต้องสนใจว่า S, R มีค่าอะไร)

D Latch (Transparent Latch)



(a) Logic diagram

<i>En</i>	<i>D</i>	Next state of <i>Q</i>
0	X	No change
1	0	$Q = 0$; reset state
1	1	$Q = 1$; set state

(b) Function table

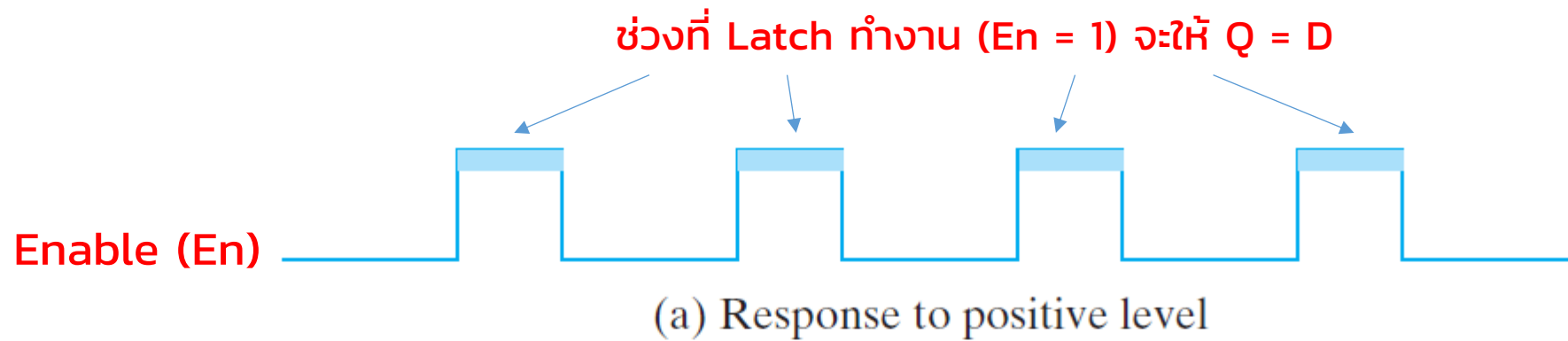
FIGURE 5.6
D latch

ໃຊ້ D ແລະ SR

D = 0 SR = 01 (reset)

D = 1 SR = 10 (set)

Latch



Flip-flop

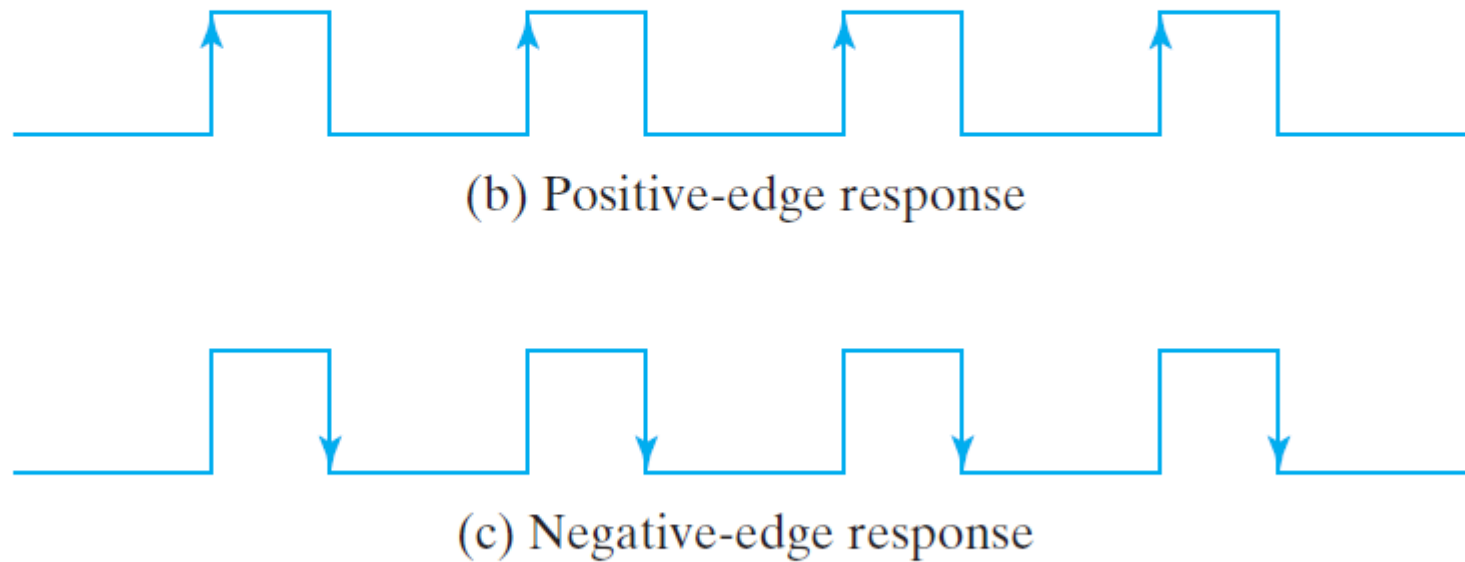


FIGURE 5.8

Clock response in latch and flip-flop

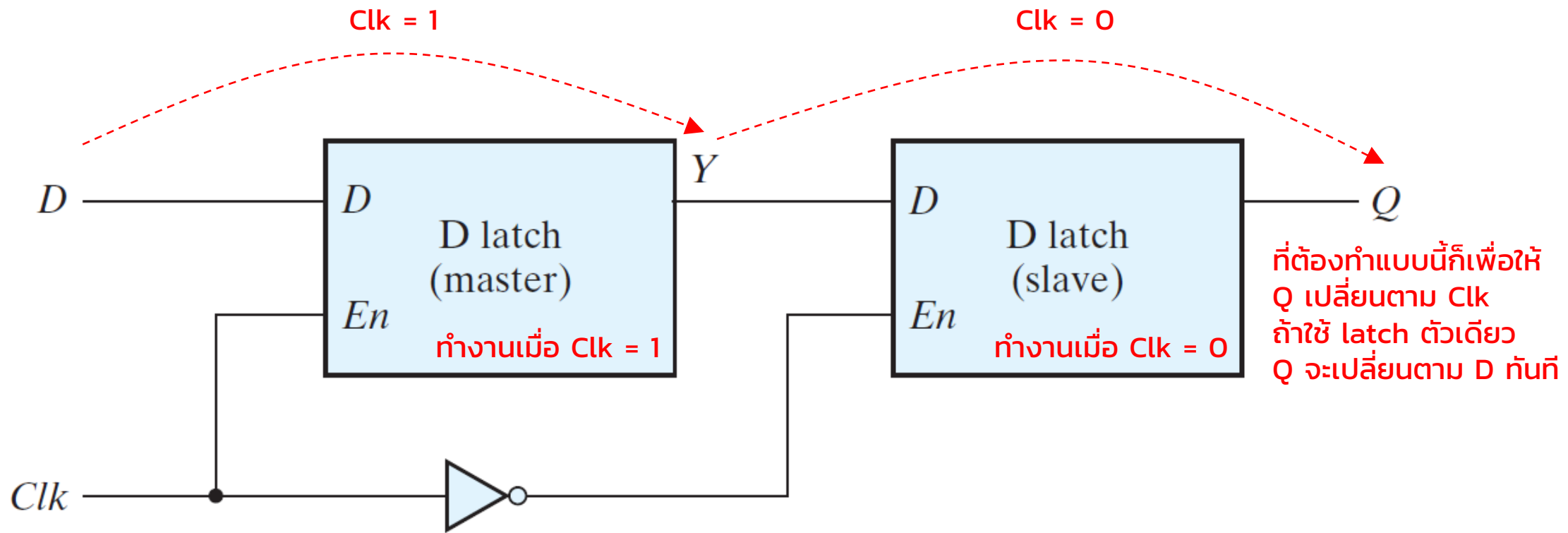
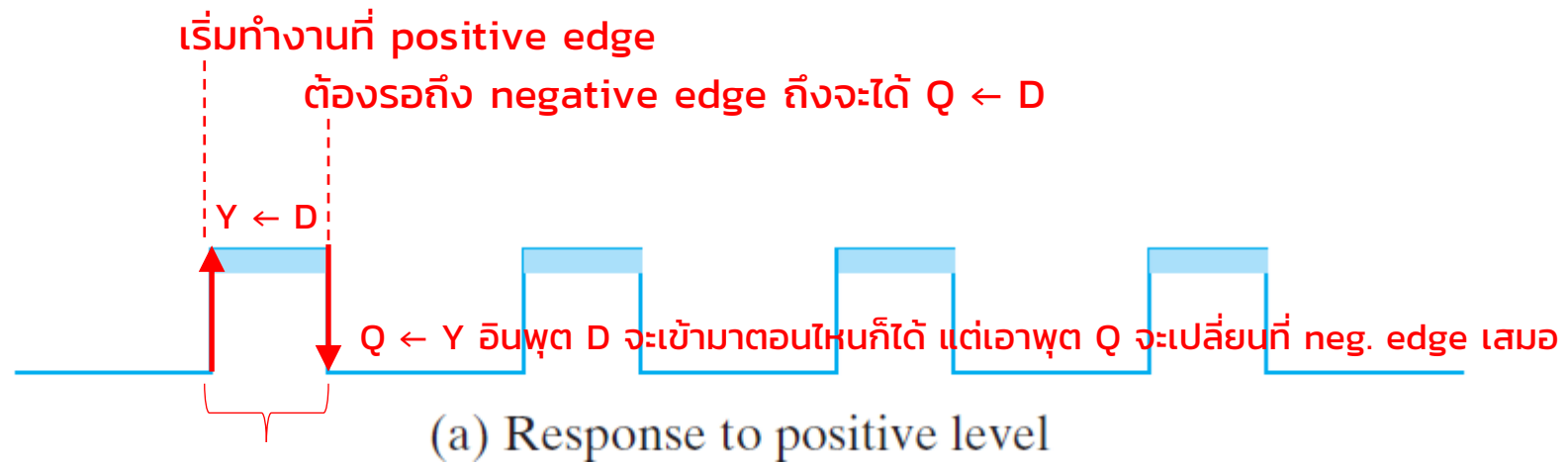


FIGURE 5.9
Master-slave D flip-flop



D ต้อง stable ในช่วงนี้ (กว้างมาก)

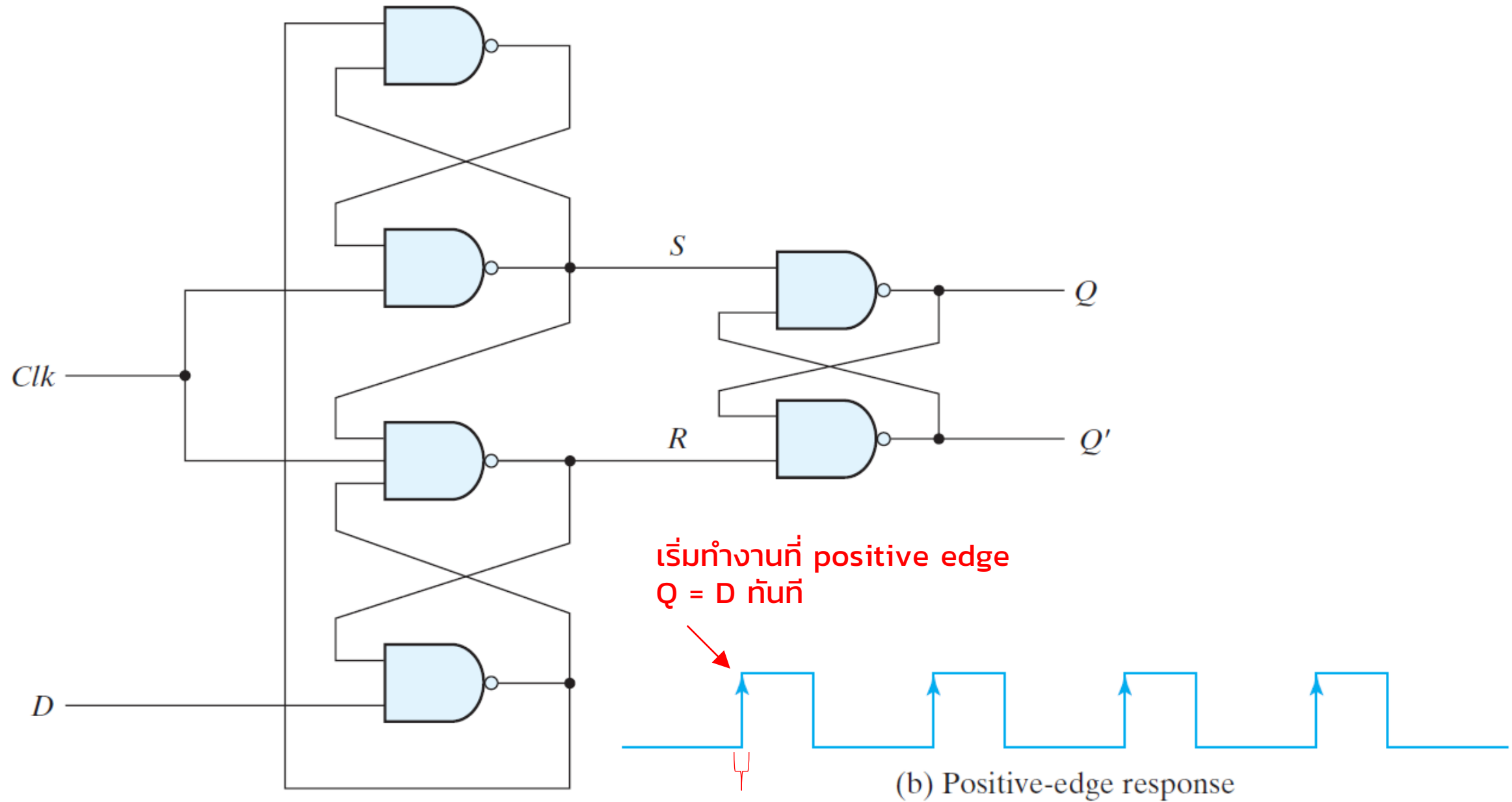


FIGURE 5.10
D-type positive-edge-triggered flip-flop

D ต้อง stable ในช่วงนี้
แคบกว่า master-slave flip-flop มาก

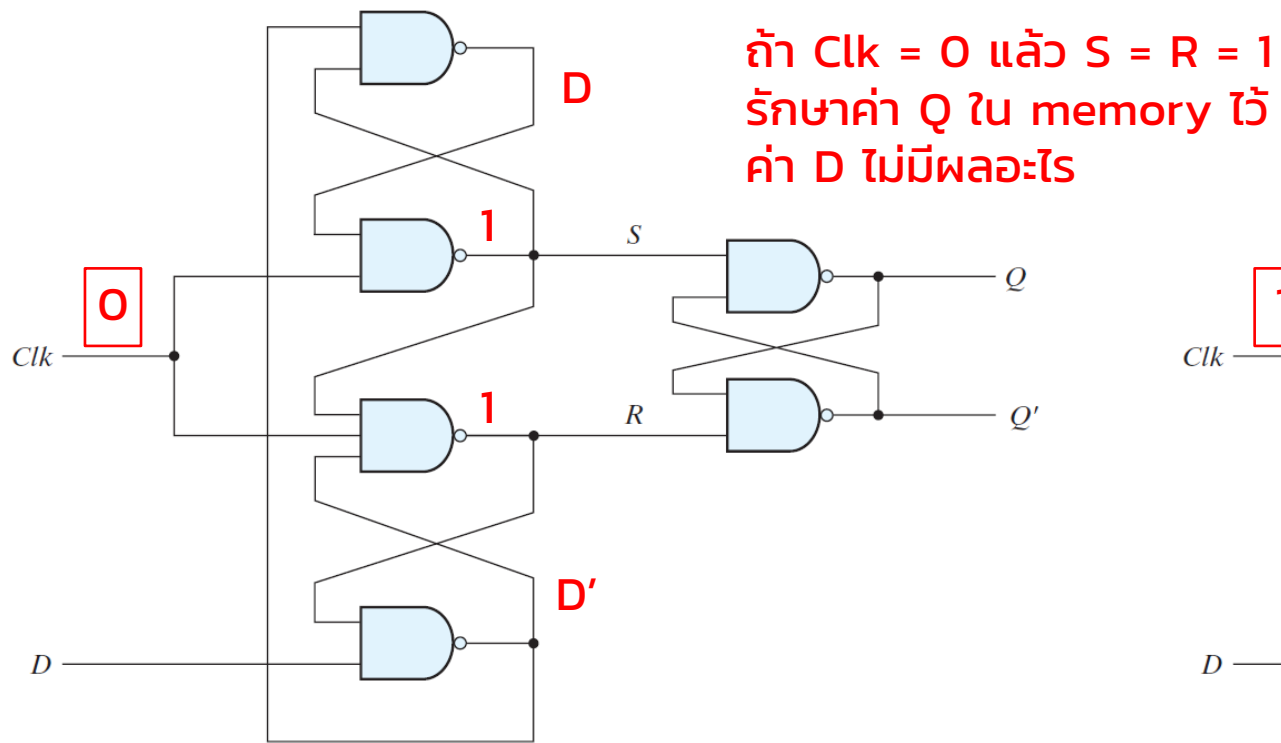


FIGURE 5.10

D-type positive-edge-triggered flip-flop

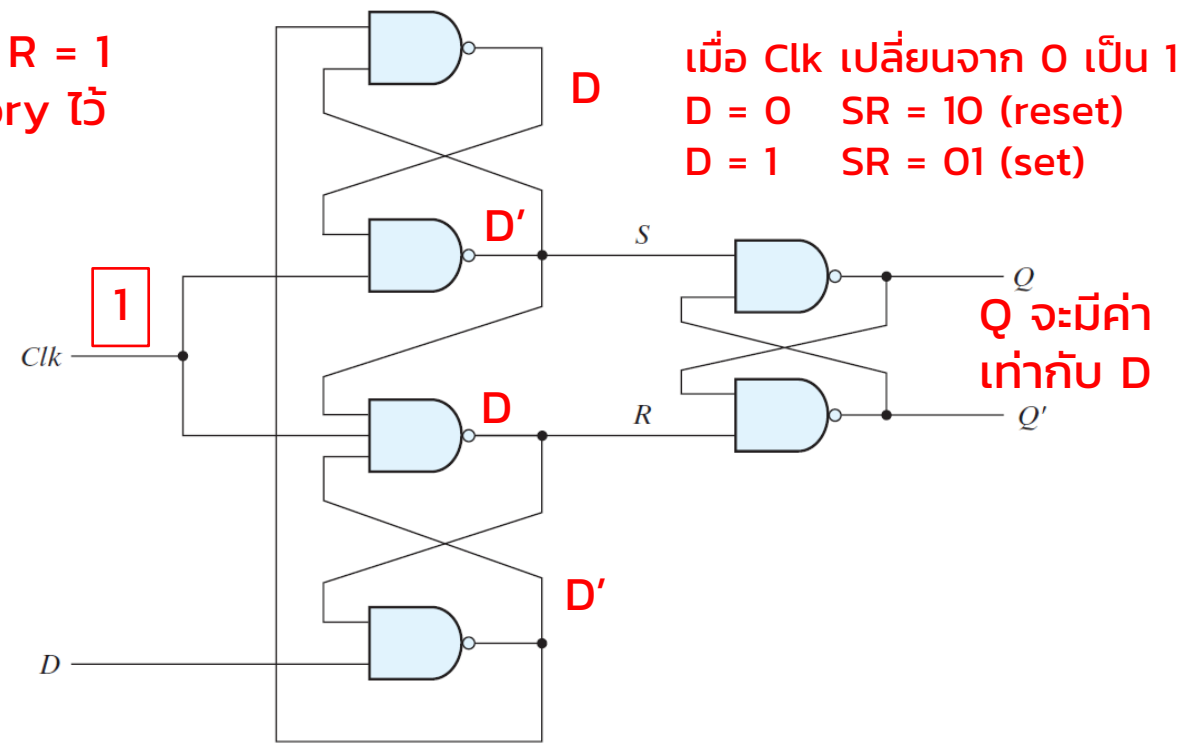


FIGURE 5.10

D-type positive-edge-triggered flip-flop

ในรูปถัดไปจะแจกทั้ง 2 กรณีคือ
 กรณีที่ 1: $D = 0$
 กรณีที่ 2: $D = 1$

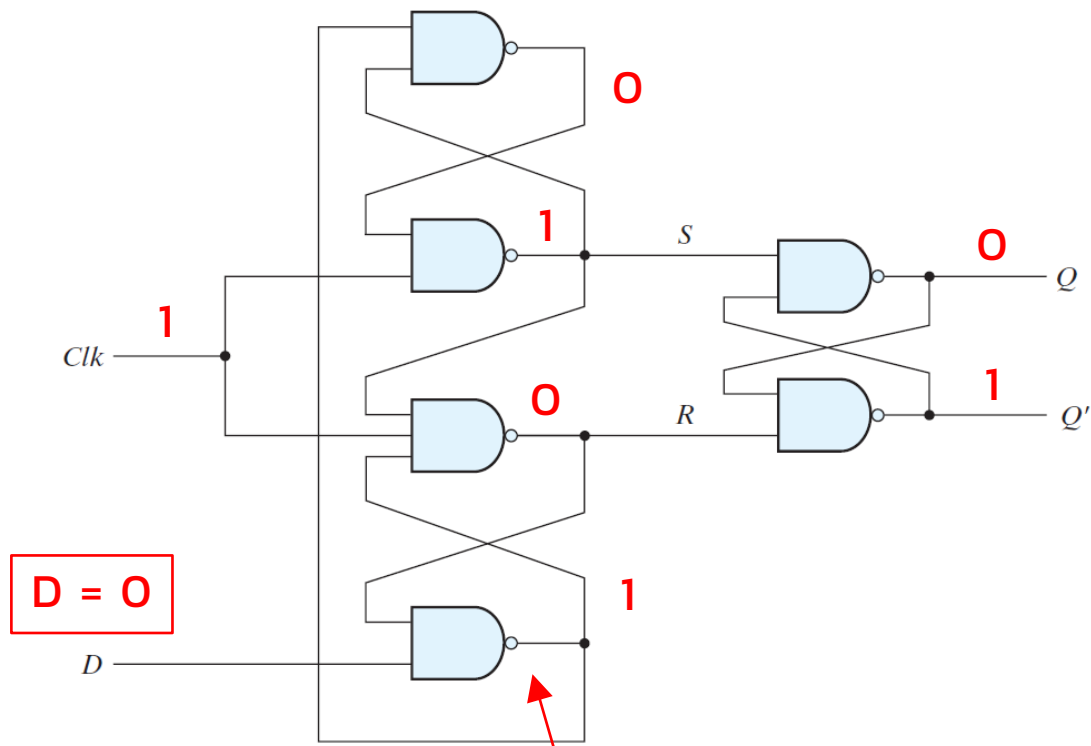


FIGURE 5.10

D-type positive-edge-triggered flip-flop

ถ้า D stable ก็เป็นตามรูปนี้
ถ้า D ไม่ stable เปลี่ยนจาก 0 เป็น 1
output ตรงนี้จะไม่เปลี่ยน
ดังนั้น Q จะไม่เปลี่ยน

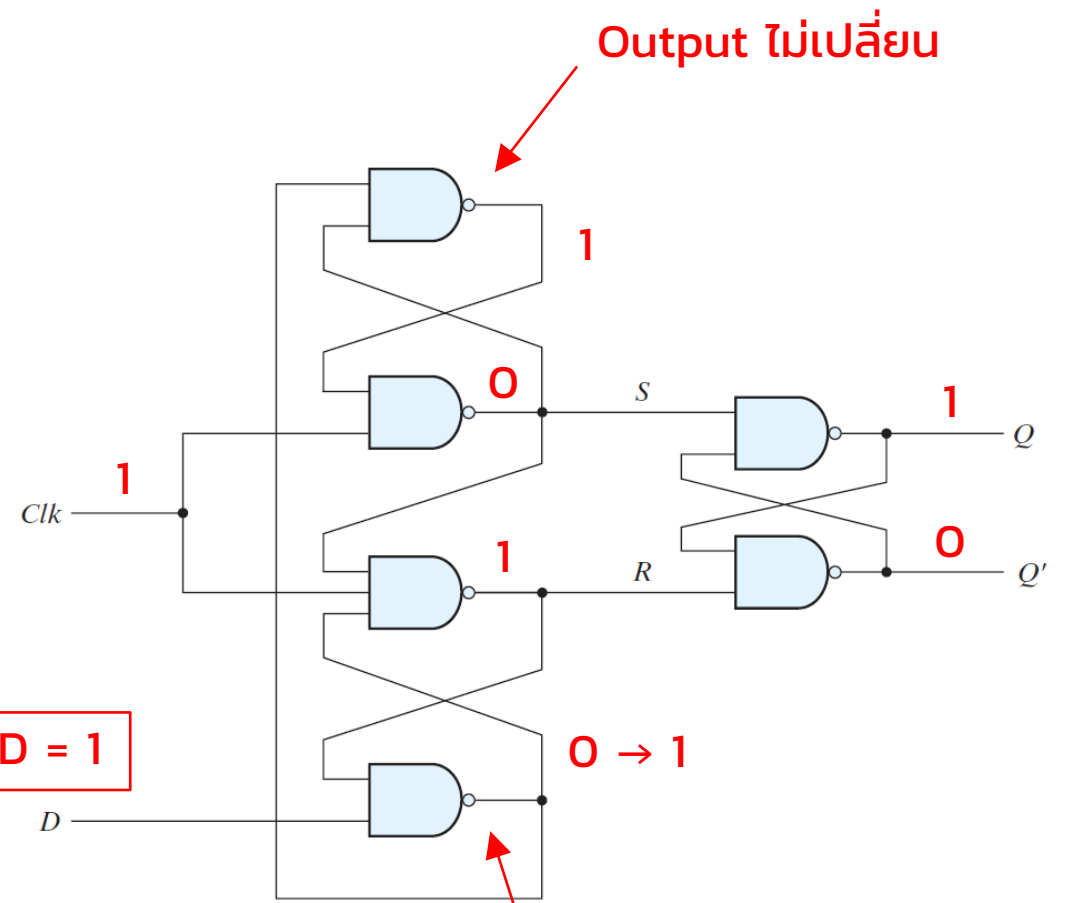
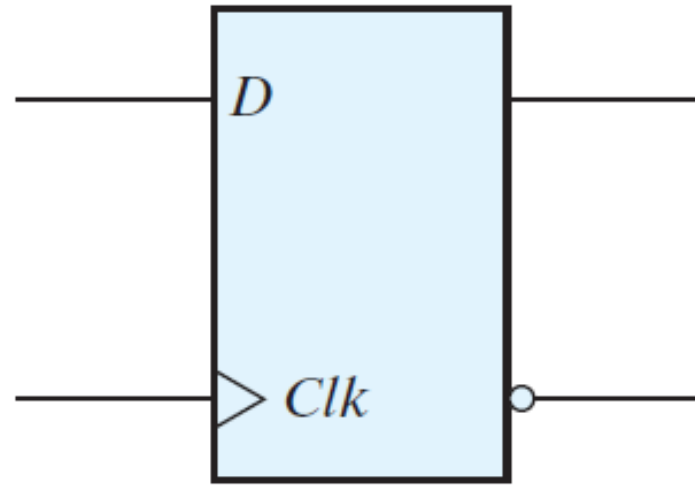


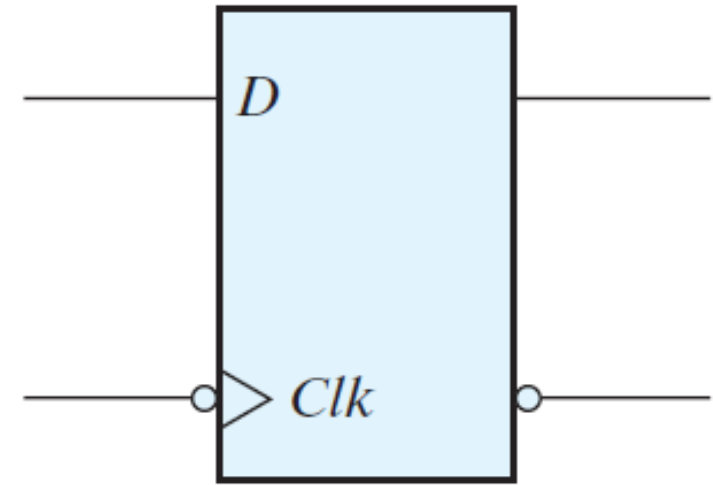
FIGURE 5.10

D-type positive-edge-triggered flip-flop

ถ้า D stable ก็เป็นตามรูปนี้
ถ้า D ไม่ stable เปลี่ยนจาก 1 เป็น 0
output ตรงนี้จะเปลี่ยนเป็น 1
แต่ไม่เปลี่ยน output ของ NAND ข้างบน
ดังนั้น Q จะไม่เปลี่ยน



(a) Positive-edge

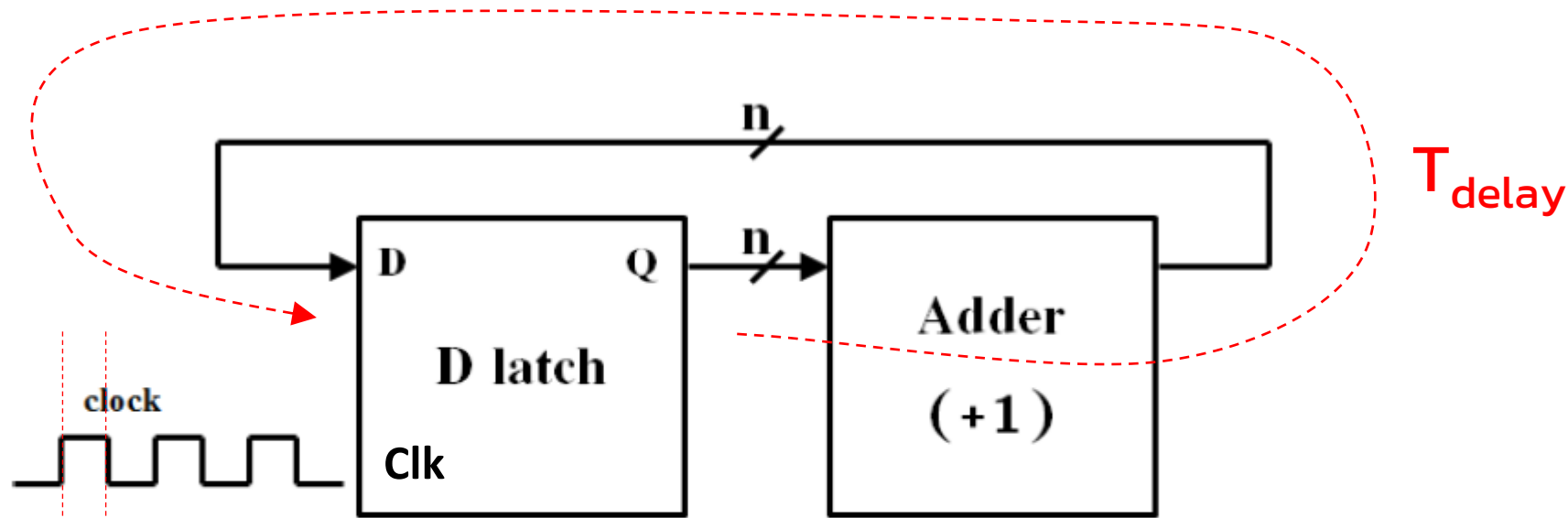


(a) Negative-edge

FIGURE 5.11

Graphic symbol for edge-triggered D flip-flop

ตัวอย่างว่าทำไมใช้ Latch ไม่ได้ ต้องใช้ Flip-flop



รูปที่ 5.4 วงจรนับเลขแบบใช้ D latch

วงจรนี้จะทำงานถูกต้อง คือนับ +1 ทุกๆ clock cycle

ถ้า $T_{\text{clock}} < T_{\text{delay}}$ (เพื่อปิด D latch ให้ทัน ก่อนที่ Q จะวนกลับมา)

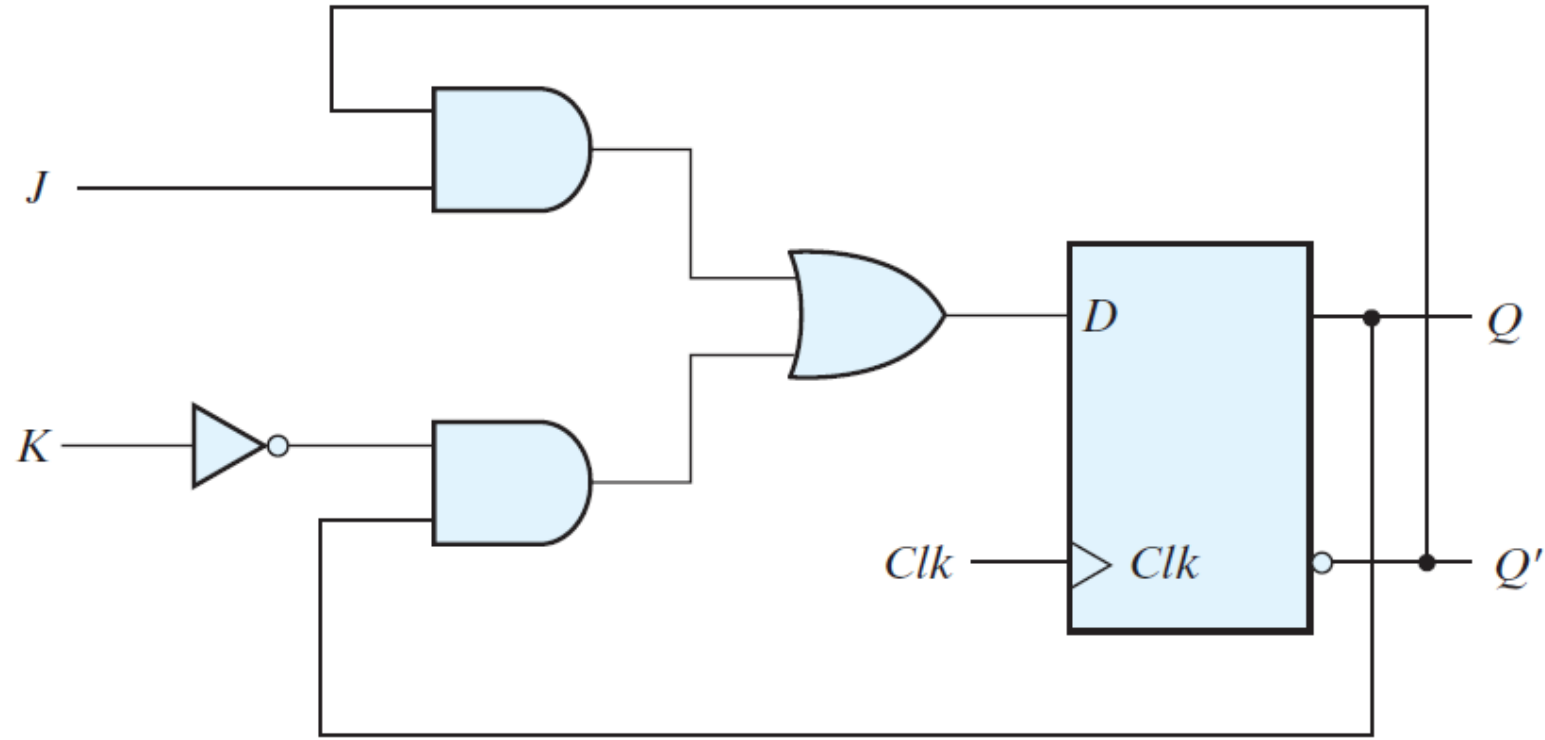
ต้องใช้ clock ความถี่สูงๆ เพื่อให้ T_{clock} น้อยๆ

แต่ถ้าใช้ Flip-flop จะใช้ความถี่ต่ำเท่าใดก็ได้ ไม่จำเป็นต้องใช้ความถี่สูงมาก

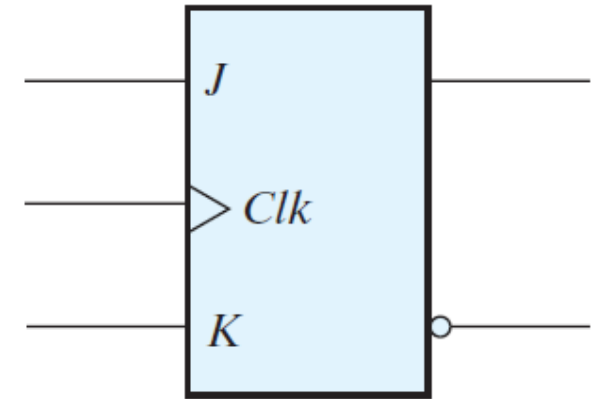
จะใช้ Master-slave D flip-flop หรือ

D-type positive-edge-triggered flip-flop ก็ได้

JK flip-flop



(a) Circuit diagram



(b) Graphic symbol

FIGURE 5.12
JK flip-flop

T flip-flop

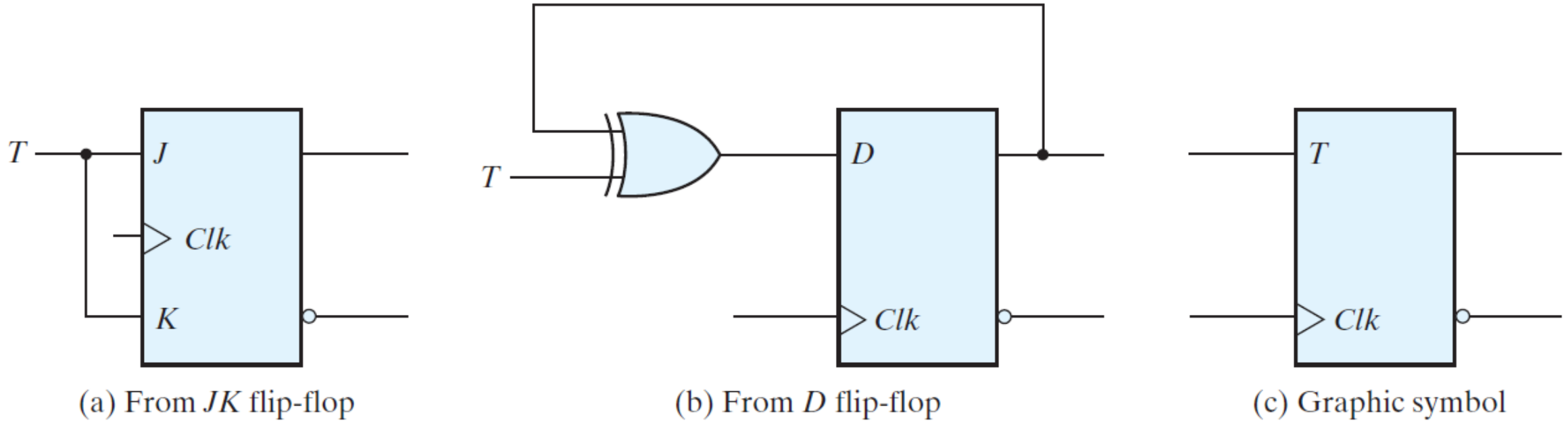


FIGURE 5.13
T flip-flop

***D* Flip-Flop**

<i>D</i>	$Q(t + 1)$
0	0 Reset
1	1 Set

***T* Flip-Flop**

<i>T</i>	$Q(t + 1)$
0	$Q(t)$ No change
1	$Q'(t)$ Complement

***JK* Flip-Flop**

<i>J</i>	<i>K</i>	$Q(t + 1)$
0	0	$Q(t)$ No change
0	1	0 Reset
1	0	1 Set
1	1	$Q'(t)$ Complement

Characteristic Equations

The logical properties of a flip-flop, as described in the characteristic table, can be expressed algebraically with a characteristic equation. For the D flip-flop, we have the characteristic equation

$$Q(t + 1) = D$$

D	$Q(t + 1)$
0	0 Reset
1	1 Set

which states that the next state of the output will be equal to the value of input D in the present state. The characteristic equation for the JK flip-flop can be derived from the characteristic table or from the circuit of Fig. 5.12. We obtain

$$Q(t + 1) = JQ' + K'Q$$

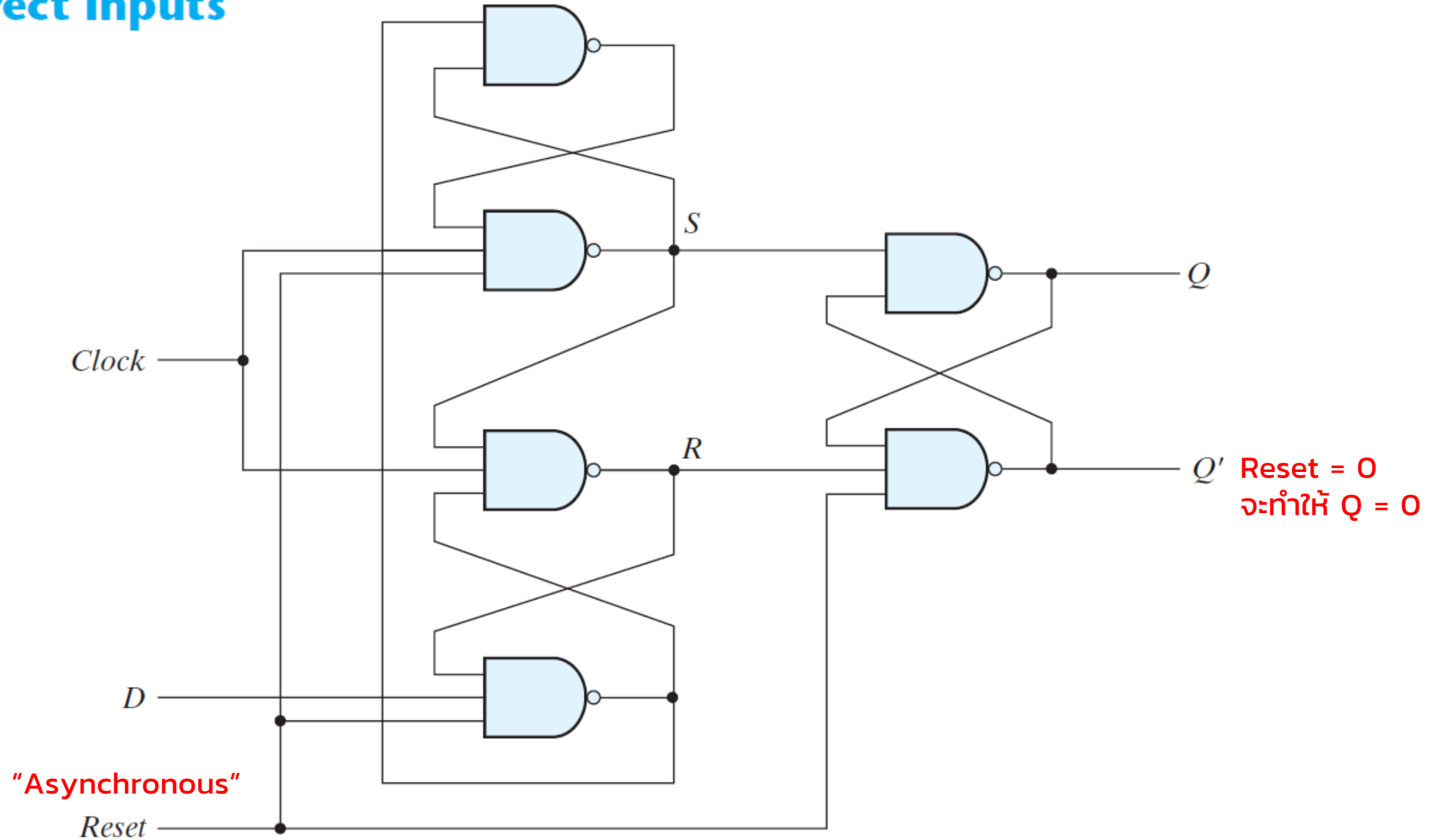
J	K	$Q(t + 1)$
0	0	$Q(t)$ No change
0	1	0 Reset
1	0	1 Set
1	1	$Q'(t)$ Complement

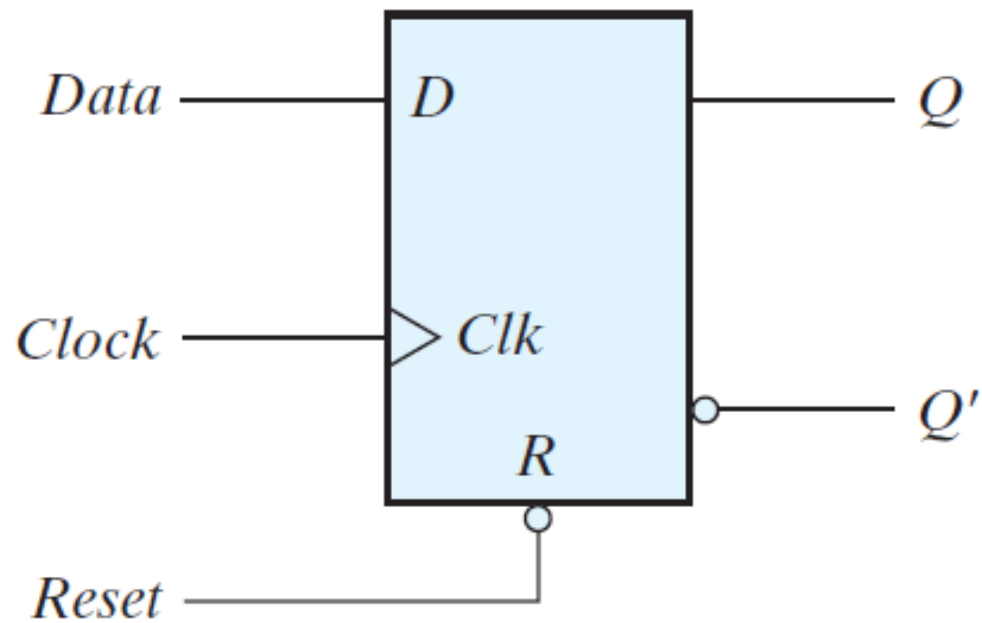
where Q is the value of the flip-flop output prior to the application of a clock edge. The characteristic equation for the T flip-flop is obtained from the circuit of Fig. 5.13:

$$Q(t + 1) = T \oplus Q = TQ' + T'Q$$

T	$Q(t + 1)$
0	$Q(t)$ No change
1	$Q'(t)$ Complement

Direct Inputs





(b) Graphic symbol

		Reset	Clock	Data		
R	Clk	D			Q	Q'
0	X	X			0	1
1	↑	0			0	1
1	↑	1			1	0

(b) Function table

FIGURE 5.14

D flip-flop with asynchronous reset

ANALYSIS OF CLOCKED SEQUENTIAL CIRCUITS

$$A(t + 1) = A(t)x(t) + B(t)x(t)$$

$$B(t + 1) = A'(t)x(t)$$

$$A(t + 1) = Ax + Bx$$

$$B(t + 1) = A'x$$

$$y(t) = [A(t) + B(t)]x'(t)$$

$$y = (A + B)x'$$

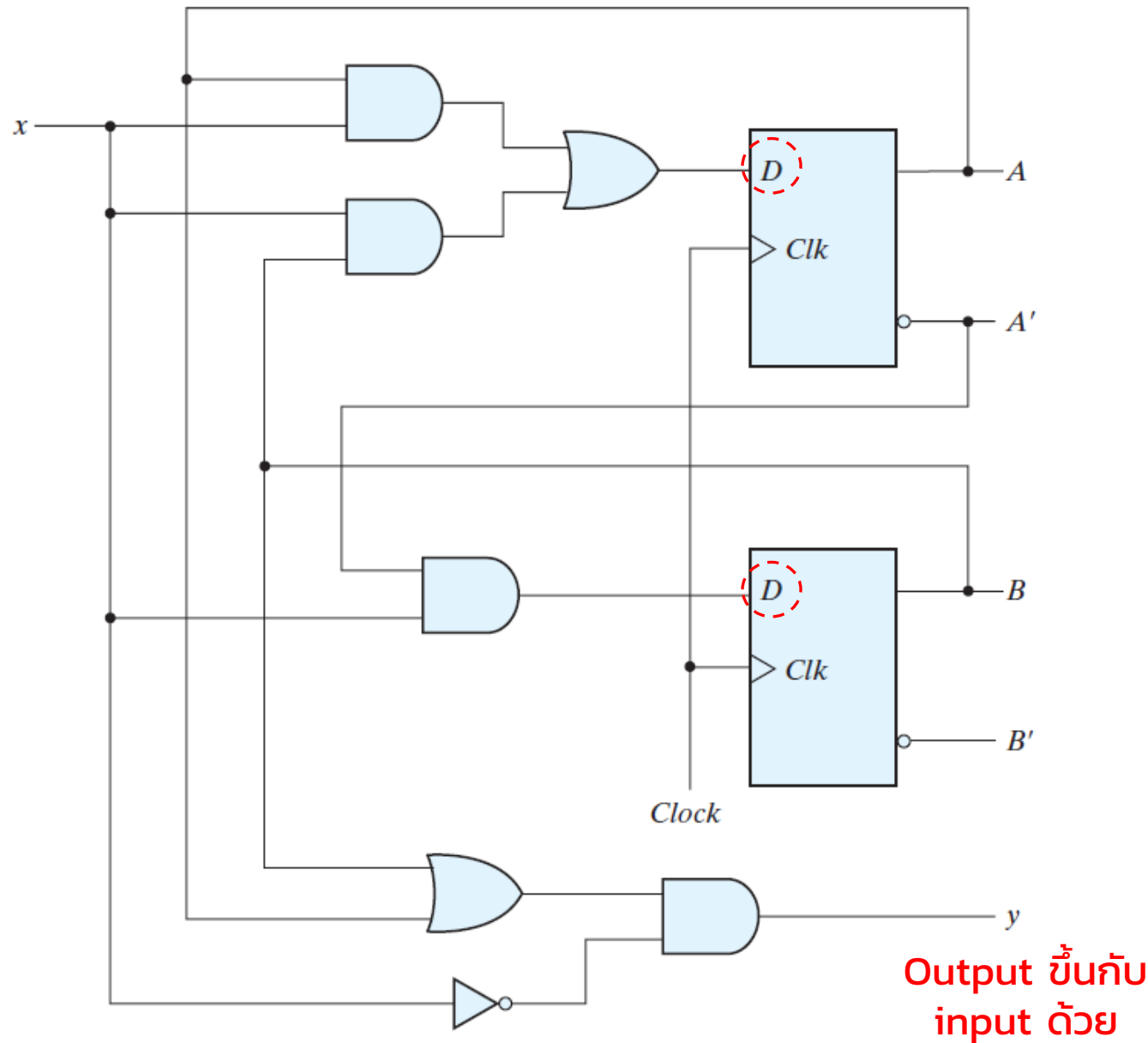


FIGURE 5.15
Example of sequential circuit

Table 5.2
State Table for the Circuit of Fig. 5.15

Present State		Input	Next State		Output
A	B		A	B	
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

เปลี่ยนที่ positive edge
เปลี่ยนตาม input กันที ไม่รอ positive edge

$A(t + 1) = Ax + Bx$

$B(t + 1) = A'x$

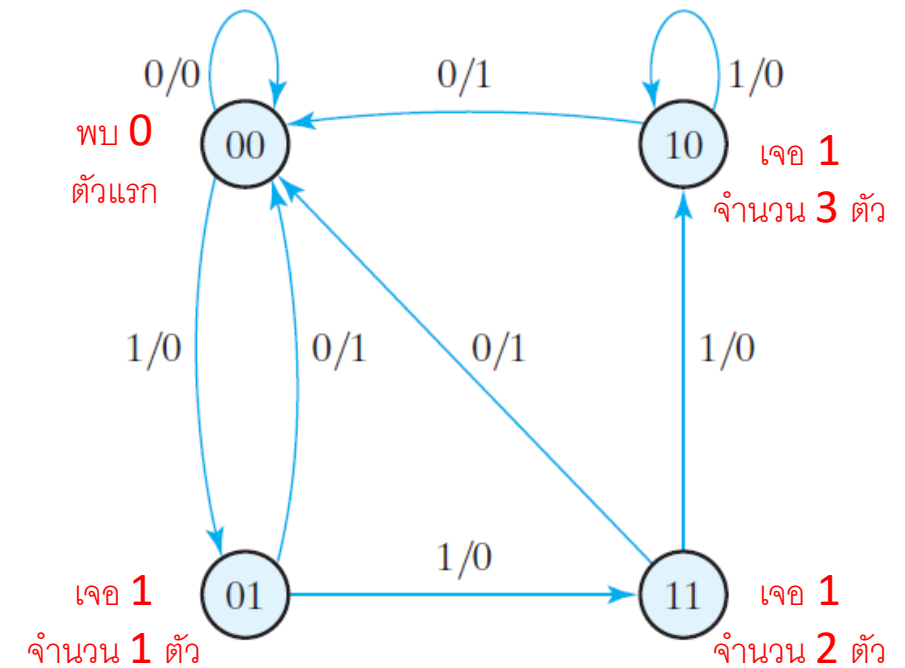
$y = (A + B)x'$

อินพุต x ต้อง stable ถ้าอินพุตเปลี่ยน เอาพุต y จะเปลี่ยนตามกันที

Table 5.3*Second Form of the State Table*

Present State		Next State				Output	
		$x = 0$		$x = 1$		$x = 0$	$x = 1$
<i>A</i>	<i>B</i>	<i>A</i>	<i>B</i>	<i>A</i>	<i>B</i>	<i>y</i>	<i>y</i>
0	0	0	0	0	1	0	0
0	1	0	0	1	1	1	0
1	0	0	0	1	0	1	0
1	1	0	0	1	0	1	0

แบบ Mealy: output ขึ้นกับ input
เขียน output ไว้บนเส้นเชื่อม

**FIGURE 5.16**

State diagram of the circuit of Fig. 5.15

state diagram นี้ น่าจะหา 0 ตัวแรกหลังจาก (1+) แต่มี redundant states

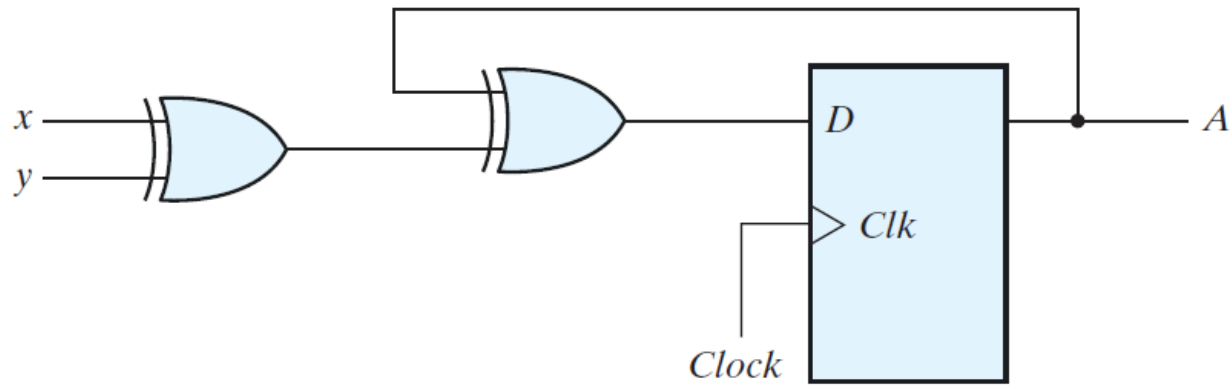
Analysis with D Flip-Flops

$D_A = A \oplus x \oplus y$
 $A(t + 1) = A \oplus x \oplus y$

Odd function

รับ input ที่ละ 2 บิต

output แปลตาม state ไม่ขึ้นกับ input
ตัวอย่างนี้ให้ output = state เลย ง่ายดี
กรณีทั่วไป output = f(state, input)

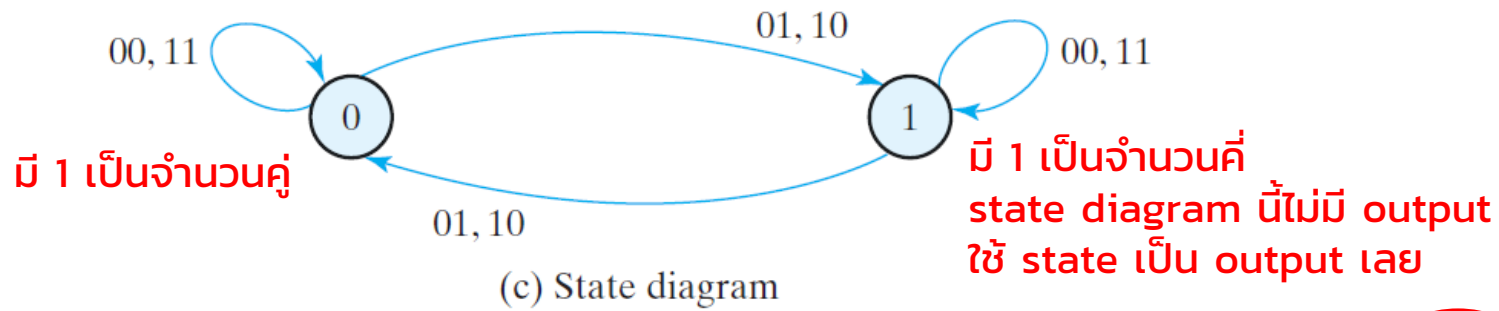


(a) Circuit diagram

Present state	Inputs		Next state
A	x	y	A
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

(b) State table

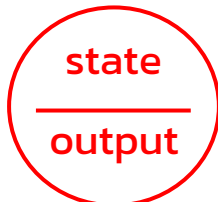
แบบ Moore: output ขึ้นกับ state เท่านั้น ไม่ขึ้นกับ input



(c) State diagram

FIGURE 5.17
Sequential circuit with D flip-flop

กรณีทั่วไปต้องวาด state diagram
แบบ Moore โดยระบุทั้ง state และ output



Analysis with JK Flip-Flops

$J_A = B \quad K_A = Bx'$

$J_B = x' \quad K_B = A'x + Ax' = A \oplus x$

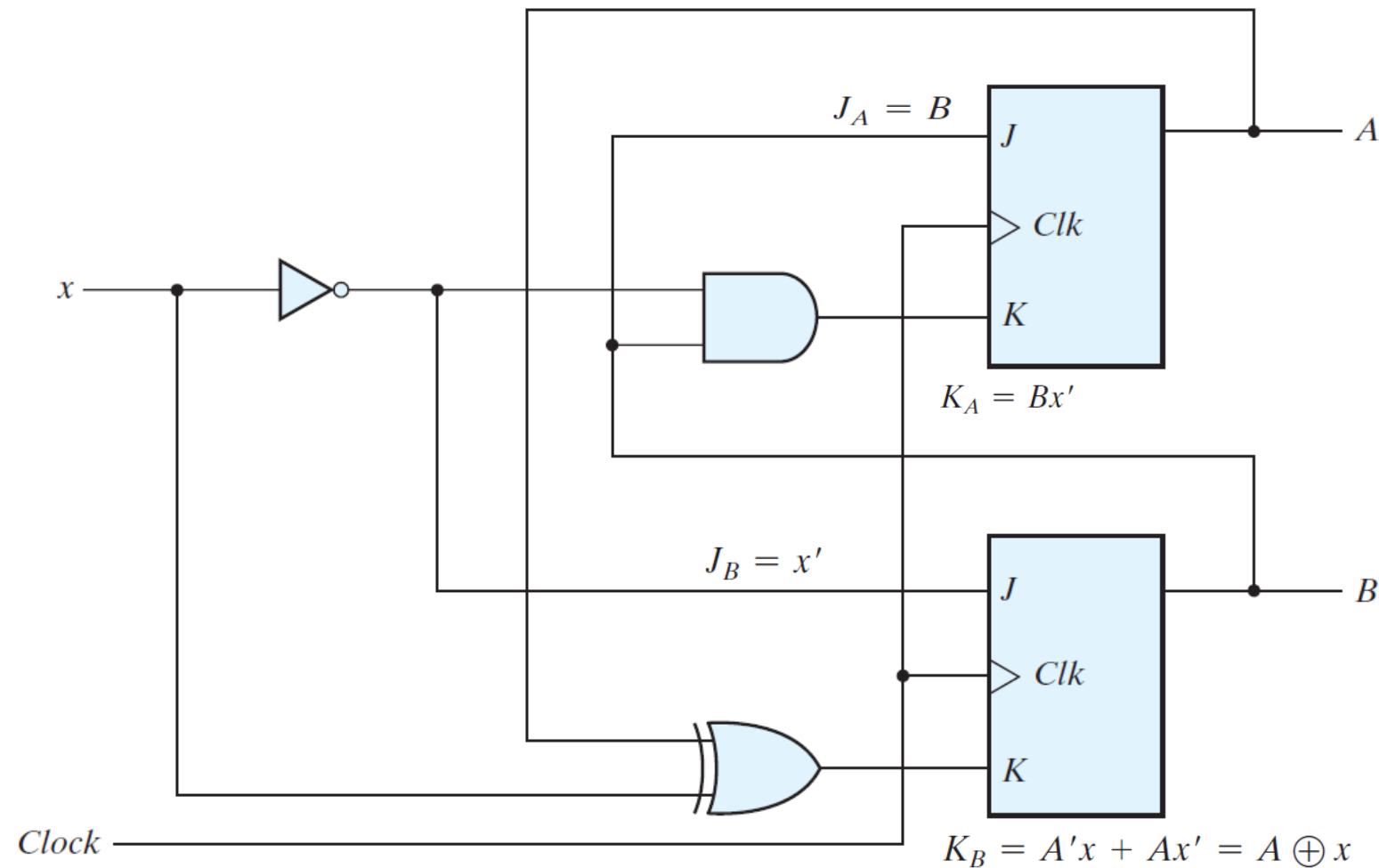


FIGURE 5.18
Sequential circuit with JK flip-flop

$$J_A = B \quad K_A = Bx'$$

$$J_B = x' \quad K_B = A'x + Ax' = A \oplus x$$

J	K	$Q(t + 1)$	
0	0	$Q(t)$	No change
0	1	0	Reset
1	0	1	Set
1	1	$Q'(t)$	Complement

$$A(t + 1) = JA' + K'A$$

$$B(t + 1) = JB' + K'B$$

$$Q(t + 1) = JQ' + K'Q$$

$$A(t + 1) = BA' + (Bx')'A = A'B + AB' + Ax$$

$$B(t + 1) = x'B' + \underbrace{(A \oplus x)'}_{}B = B'x' + \underbrace{ABx + A'Bx'}_{}B$$

$$= (Ax' + A'x)'B$$

$$= (Ax')'(A'x)'B$$

$$= (A' + x)(A + x')B$$

$$= (A'A + A'x' + xA + xx')B$$

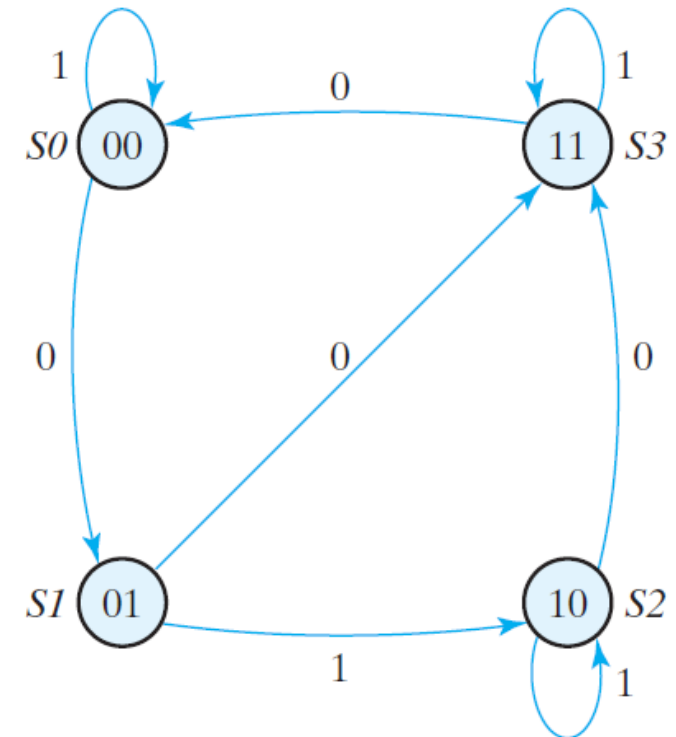
$$= (A'x' + xA)B$$

$$= ABx + A'Bx'$$

Table 5.4*State Table for Sequential Circuit with JK Flip-Flops*

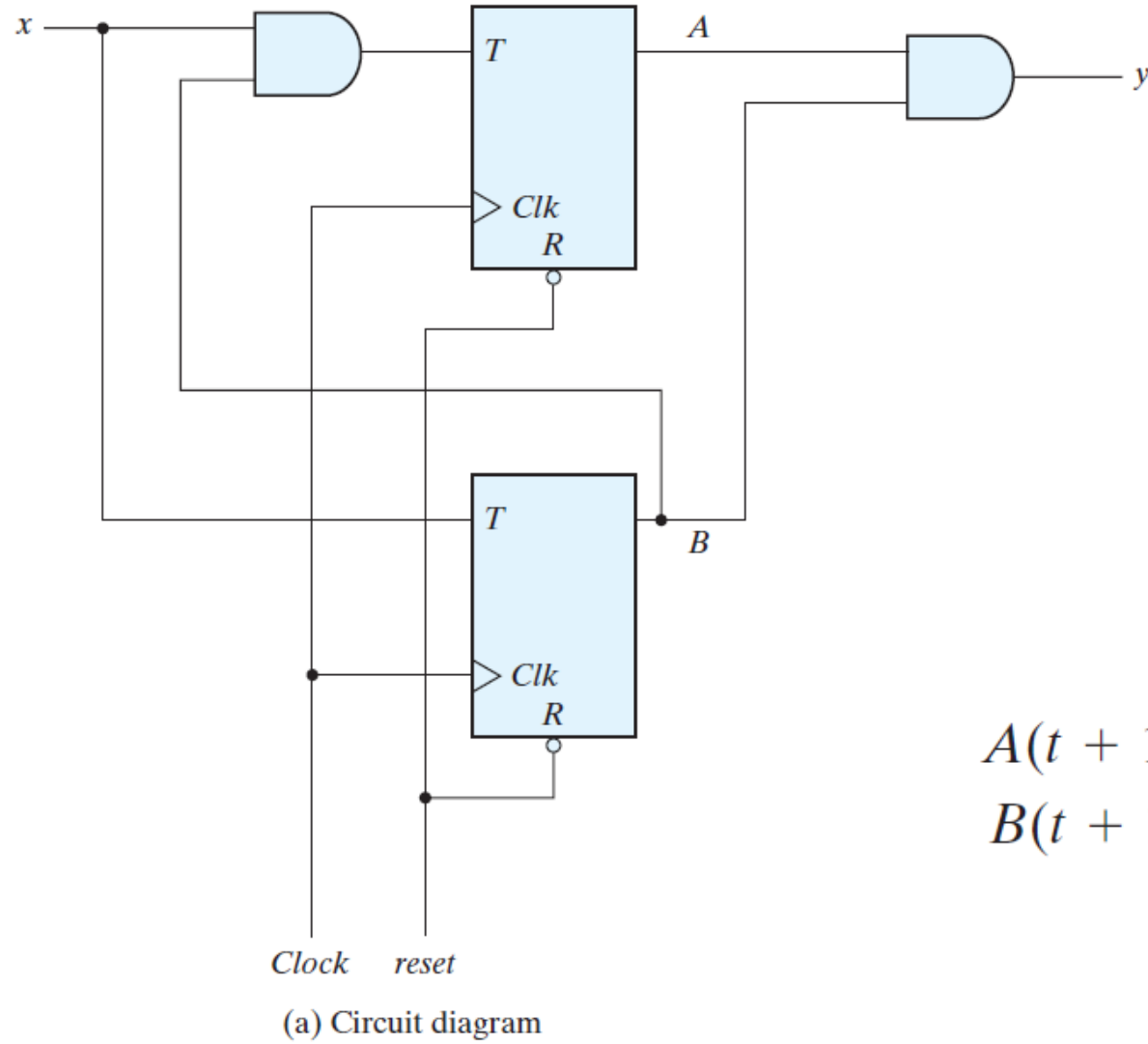
Present State		Input	Next State	
A	B		A	B
0	0	0	0	1
0	0	1	0	0
0	1	0	1	1
0	1	1	1	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	0
1	1	1	1	1

state 00 คือ จำนวน 0 ในอินพุต
เป็น 0, 3, 6, 9, ... มี redundant state



แบบ Moore
ไม่มี output

Analysis with *T* Flip-Flops



$$Q(t + 1) = T \oplus Q = T'Q + TQ'$$

$$T_A = Bx$$

$$T_B = x$$

$$y = AB$$

$$A(t + 1) = (Bx)'A + (Bx)A' = AB' + Ax' + A'Bx$$

$$B(t + 1) = x \oplus B$$

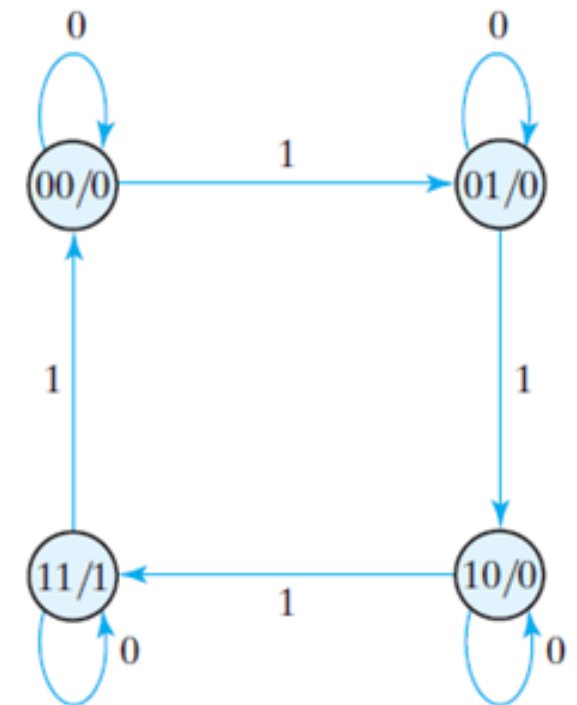
FIGURE 5.20

Sequential circuit with *T* flip-flops (Binary Counter)

Table 5.5
State Table for Sequential Circuit with T Flip-Flops

Present State		Input	Next State		Output
A	B		A	B	
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	1	0	0
1	0	0	1	0	0
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	0	0	0

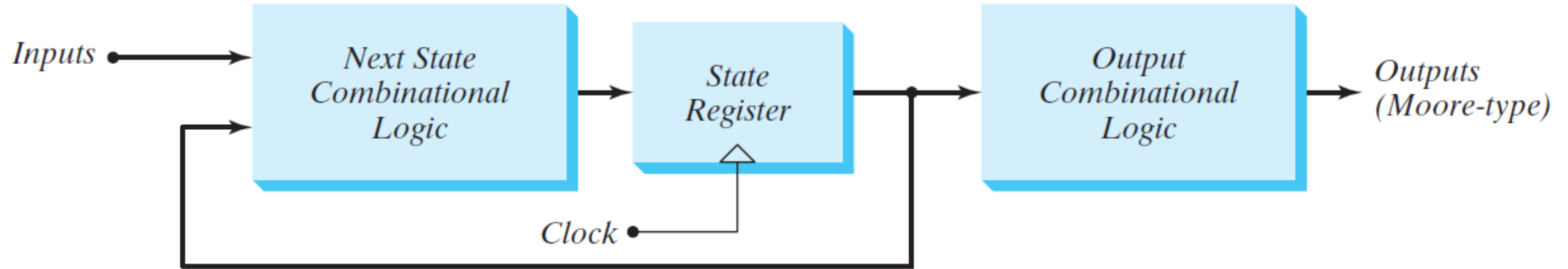
state 00 คือ จำนวน 1 ในอินพุต
 เป็น 0, 4, 8, 12, ...



แบบ Moore
 มี output

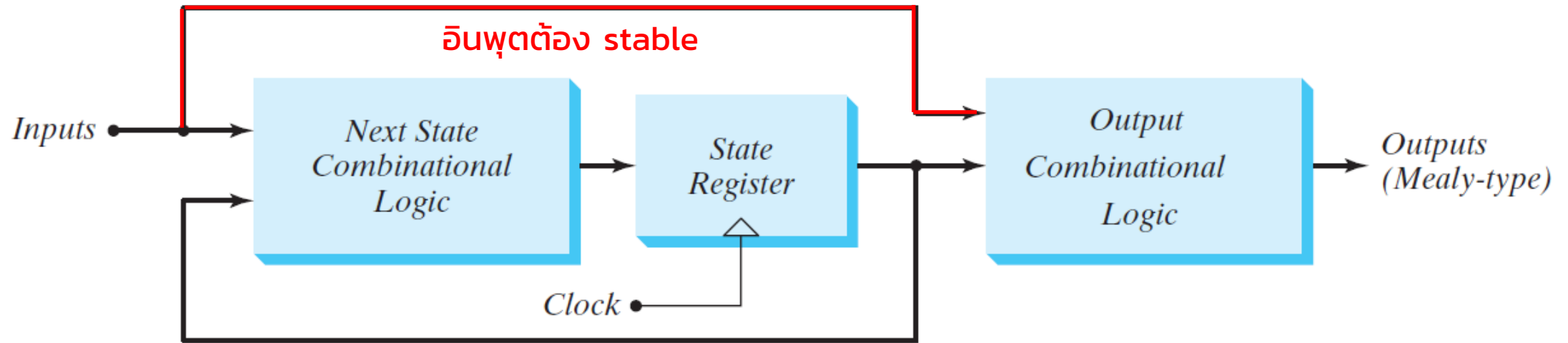
หลัง positive edge ไปแล้ว
อินพุตไม่ต้อง stable

Moore Machine

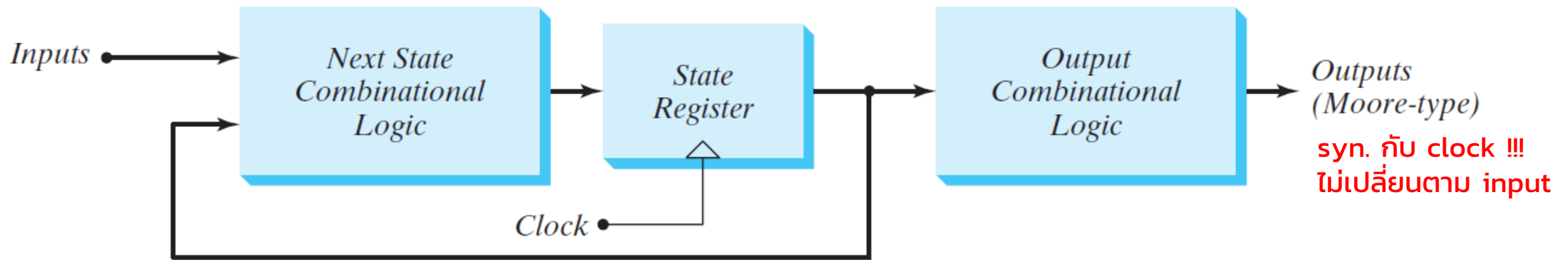


Mealy Machine

อินพุตต้อง stable

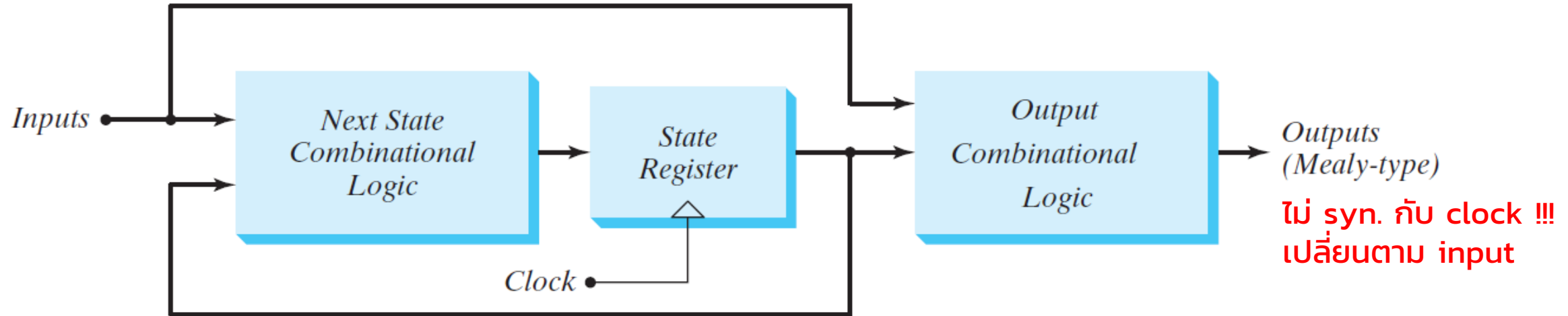


Moore Machine

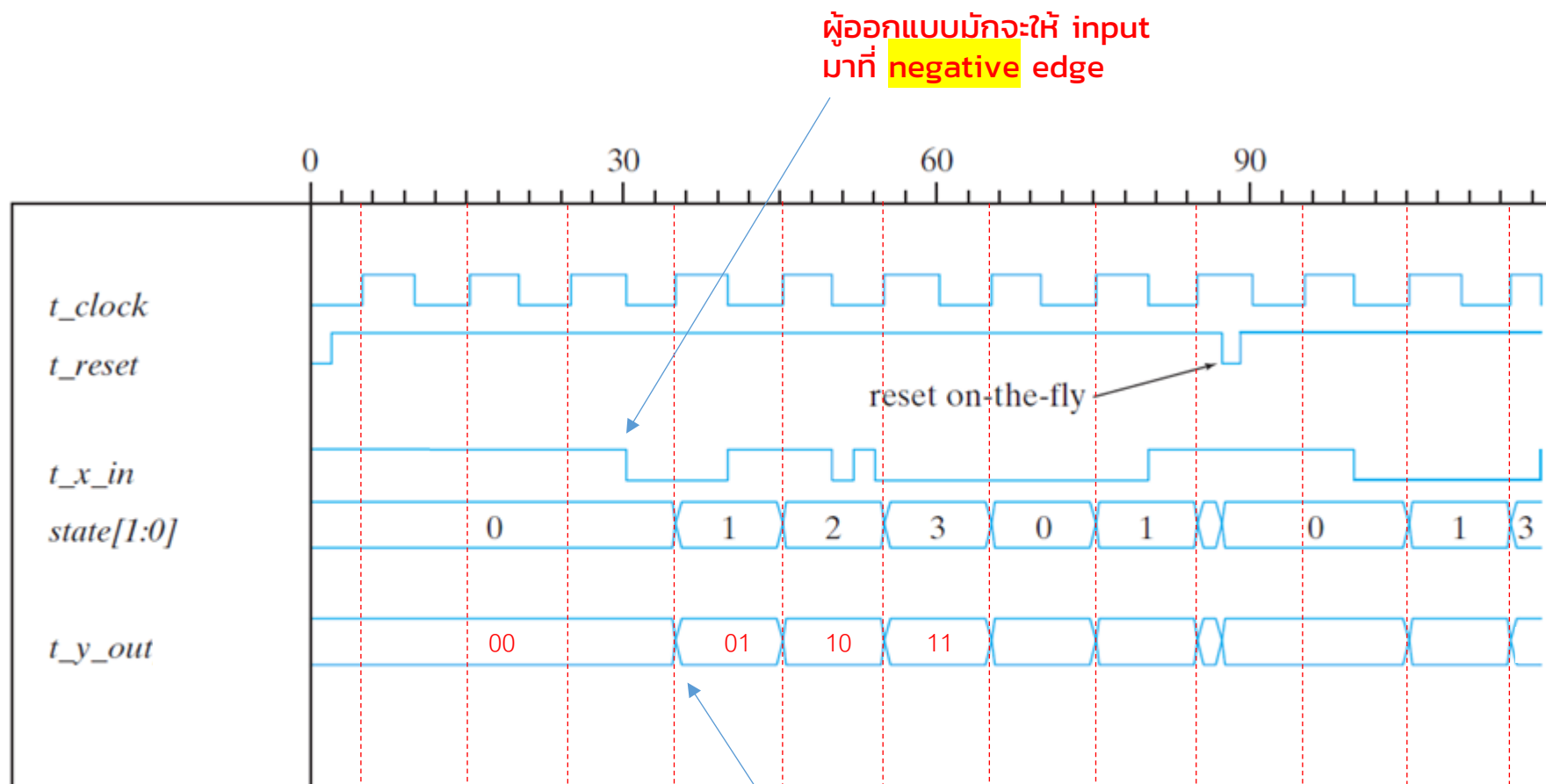


In a Moore model, the outputs of the sequential circuit are synchronized with the clock, because they depend only on flip-flop outputs that are synchronized with the clock. In a Mealy model, the outputs may change if the inputs change during the clock cycle. Moreover, the outputs may have momentary false values because of the delay encountered from the time that the inputs change and the time that the flip-flop outputs change. In order to synchronize a Mealy-type circuit, the inputs of the sequential circuit must be synchronized with the clock and the outputs must be sampled immediately before the clock edge. The inputs are changed at the inactive edge of the clock to ensure that the inputs to the flip-flops stabilize before the active edge of the clock occurs. Thus, **the output of the Mealy machine is the value that is present immediately before the active edge of the clock.**

Mealy Machine



In a Moore model, the outputs of the sequential circuit are synchronized with the clock, because they depend only on flip-flop outputs that are synchronized with the clock. In a Mealy model, the outputs may change if the inputs change during the clock cycle. Moreover, the outputs may have momentary false values because of the delay encountered from the time that the inputs change and the time that the flip-flop outputs change. In order to synchronize a Mealy-type circuit, the inputs of the sequential circuit must be synchronized with the clock and the outputs must be sampled immediately before the clock edge. The inputs are changed at the inactive edge of the clock to ensure that the inputs to the flip-flops stabilize before the active edge of the clock occurs. Thus, the output of the Mealy machine is the value that is present immediately before the active edge of the clock.



Moore Machine: Zero Detector

state 00 คือ จำนวน 0
ในอินพุตเป็น 0, 3, 6, 9, ...
มี redundant states

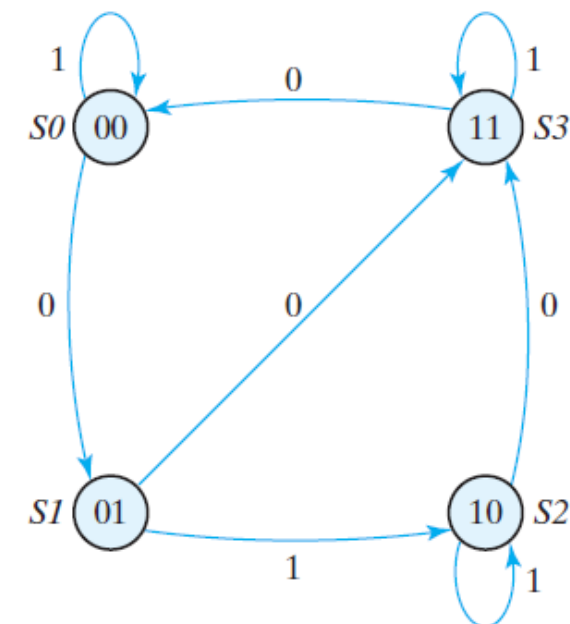
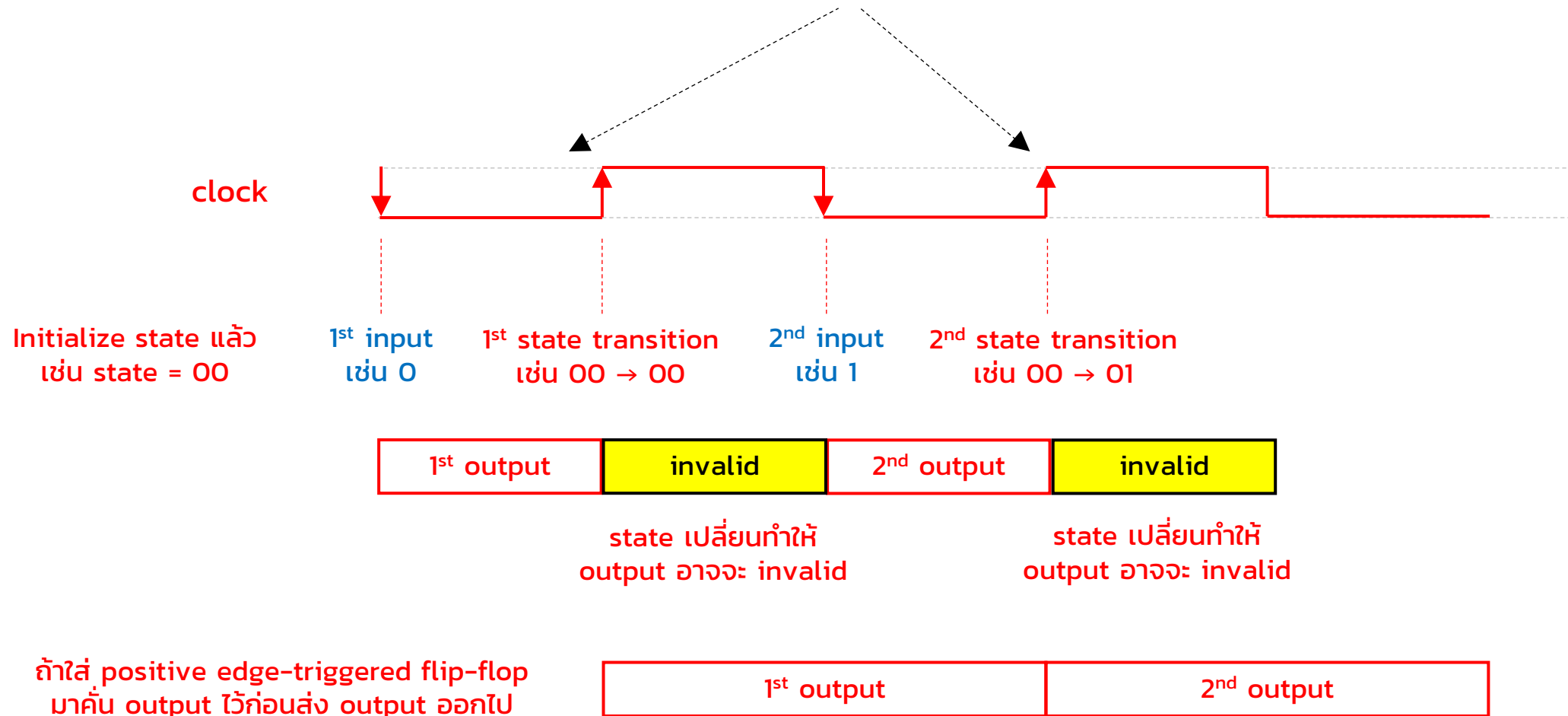


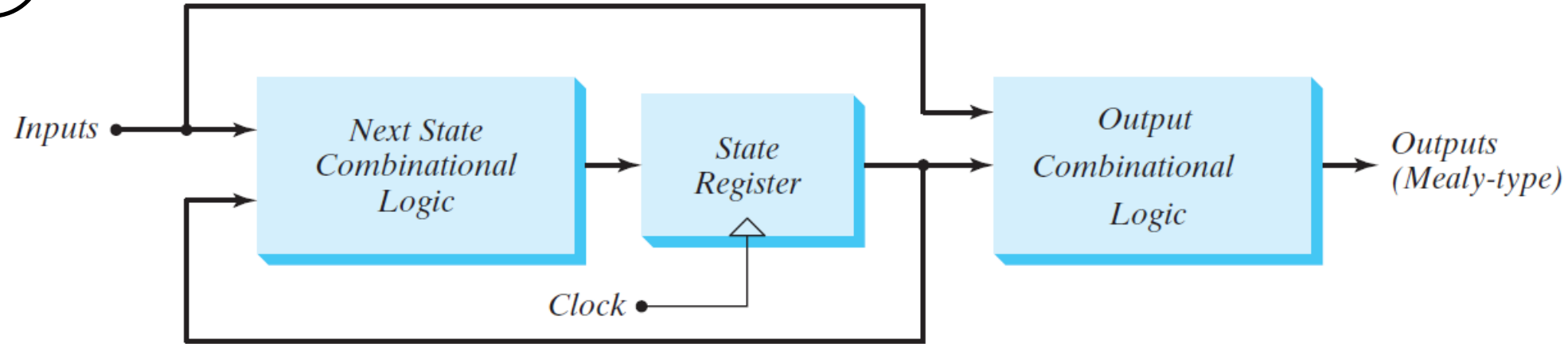
Figure 5.19

แบบ Mealy

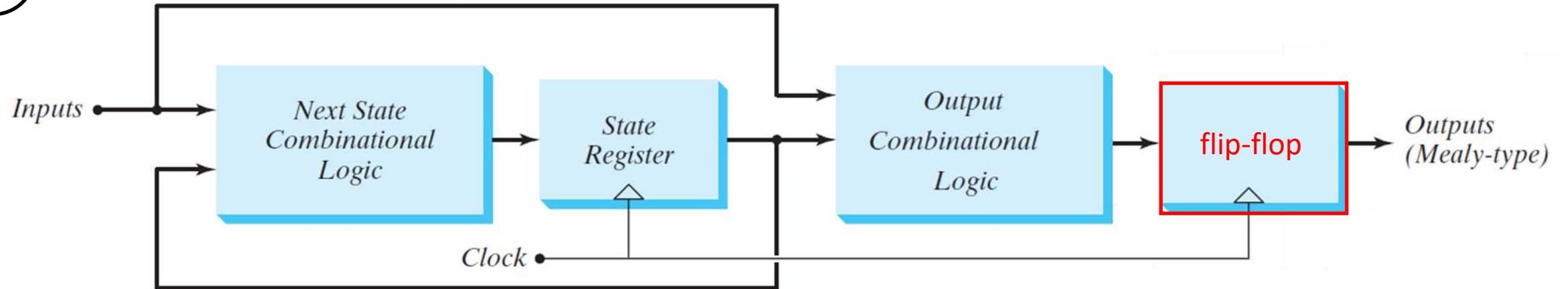
The output is the value that is present immediately before the active edge of the clock (เนื่องจาก propagation delay)



1

Mealy Machine

2

Mealy Machine

1

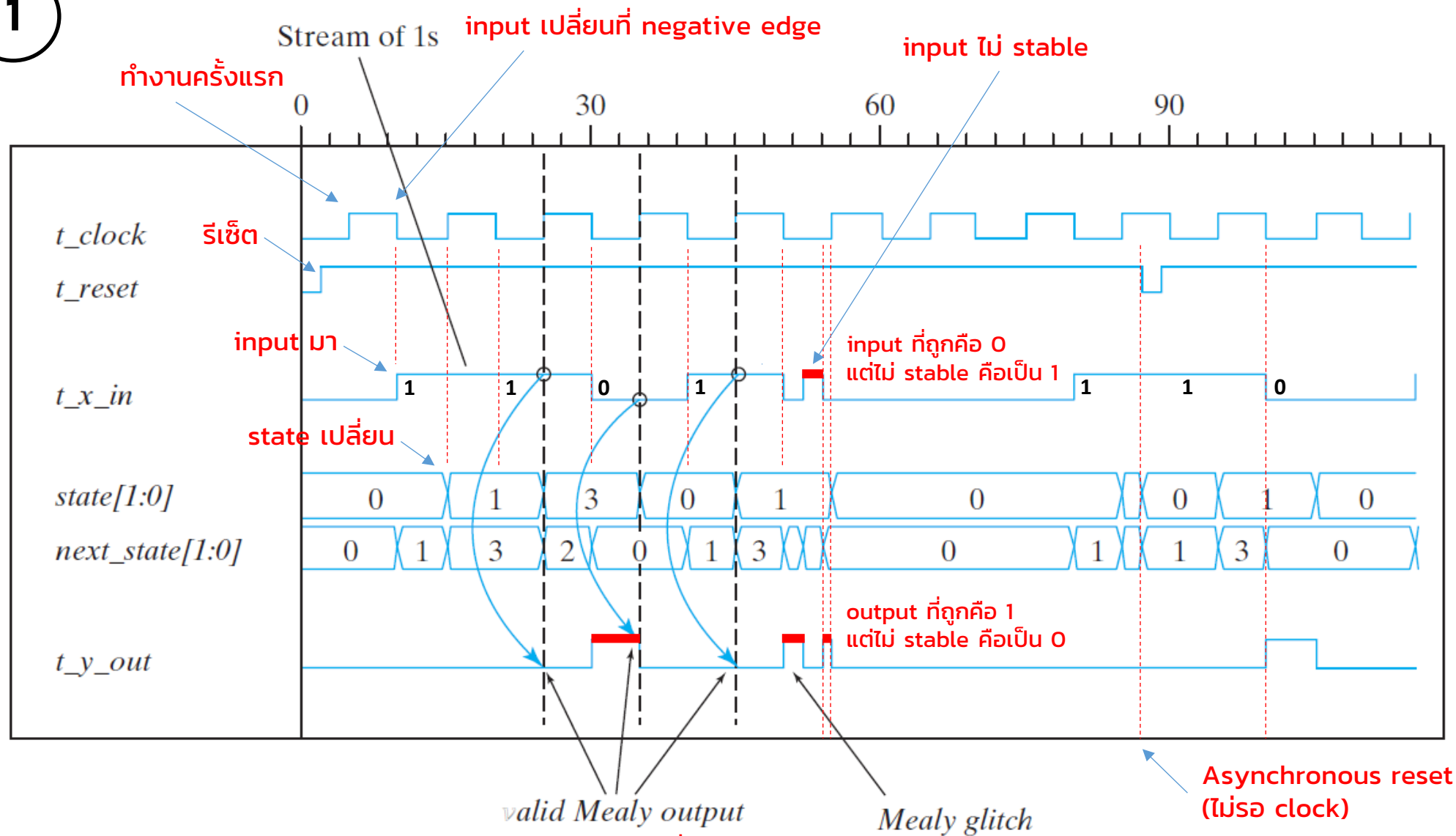


FIGURE 5.22

Simulation output of Mealy_Zero_Detector

หา 0 ตัวแรก ที่ตามหลัง 1+ มี redundant states

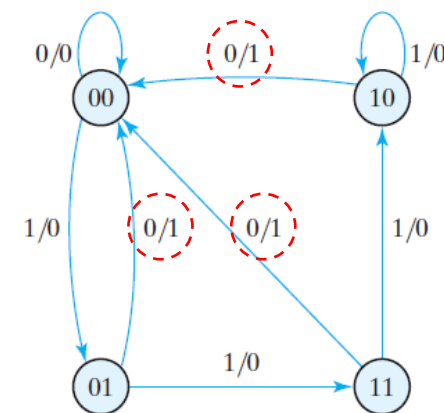
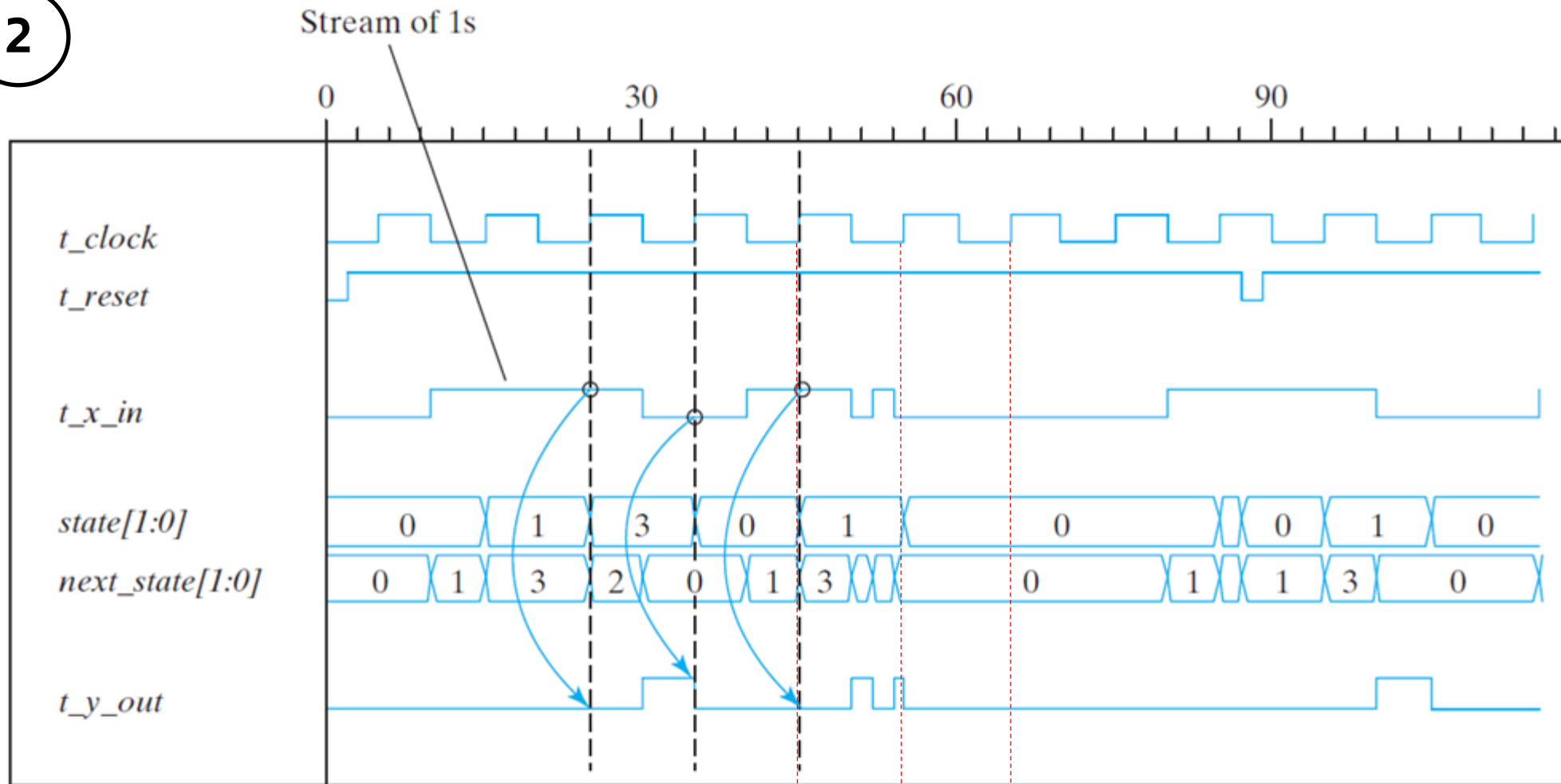


Figure 5.16

2



$t_{y_{out}}$ (แบบ 2)

สังเกตว่า

output จะออกมาในเวลาที่ถูกต้องพร้อมกับ next state และ stable อยู่ 1 clock cycle แม้ว่า input จะไม่ stable

หา 0 ตัวแรก ที่ตามหลัง 1+ มี redundant states

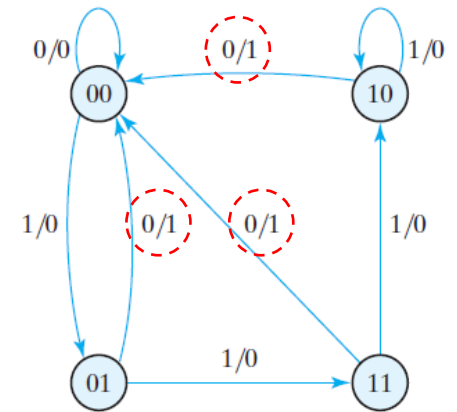
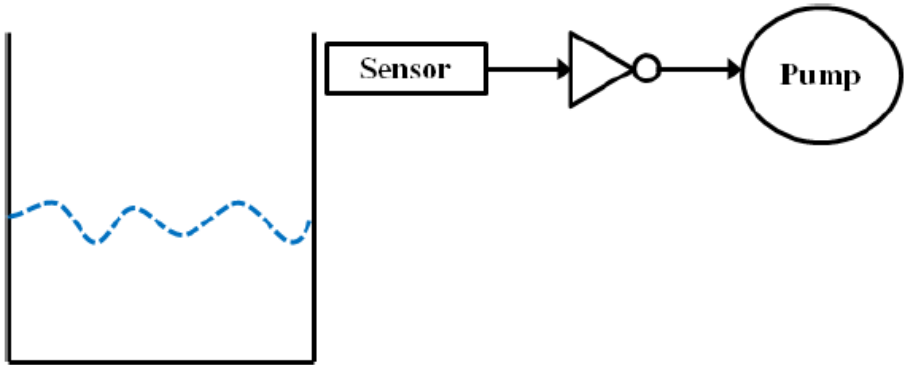


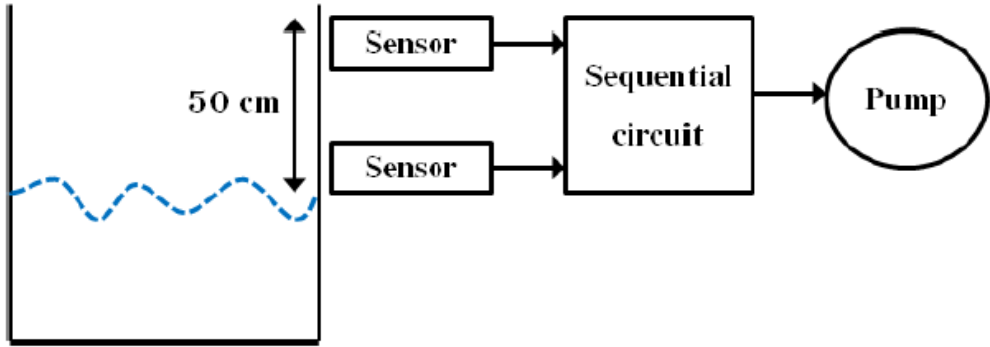
Figure 5.16

ปั้มน้ำ: ลองเขียน (finite-state machine) FSM แบบ Moore และ Merely เพื่อควบคุมการทำงานของปั้มน้ำ ให้ปั้มน้ำทำงานต่อเนื่อง ไม่ติด ๆ ดับ ๆ

มีน้ำ sensor = 1 1 ปั้มน้ำทำงาน
ไม่มีน้ำ sensor = 0 0 ปั้มน้ำหยุดทำงาน

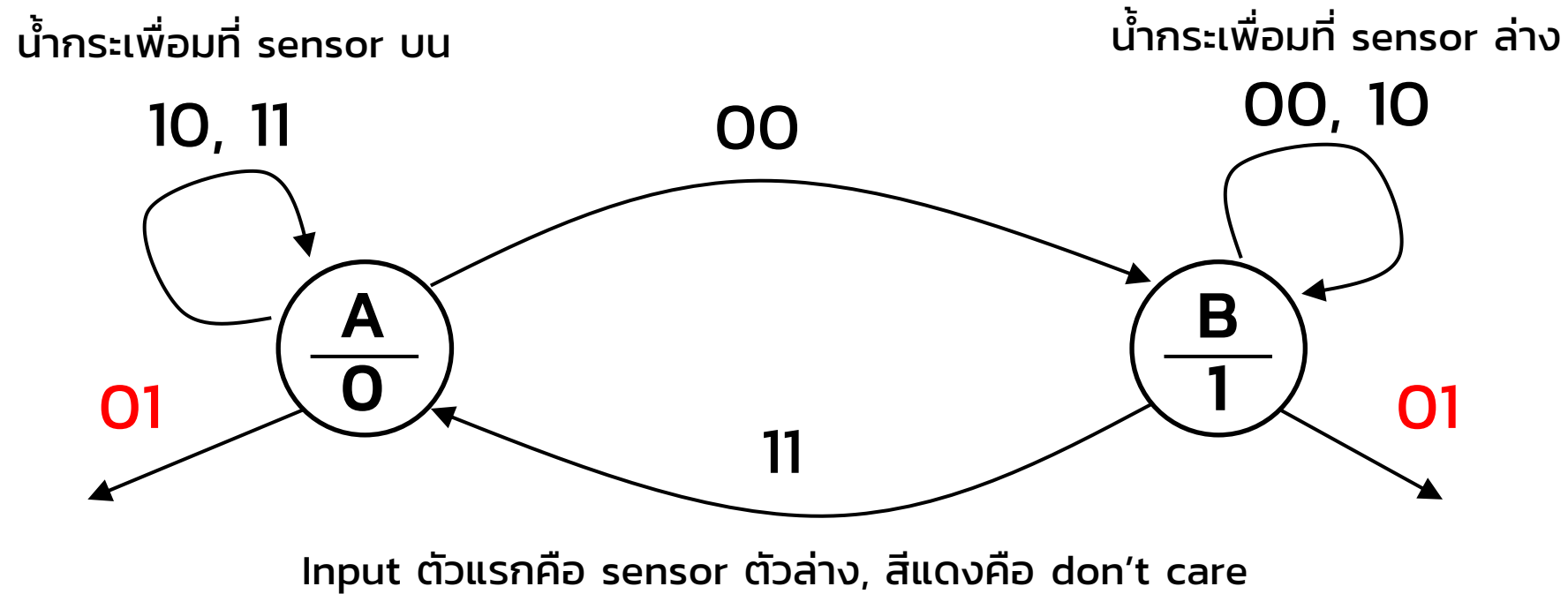


รูปที่ 5.11 ระบบควบคุมปั้มน้ำด้วยวงจรเชิงผสม



รูปที่ 5.12 ระบบควบคุมปั้มน้ำด้วยวงเชิงลำดับ

แบบ Moore

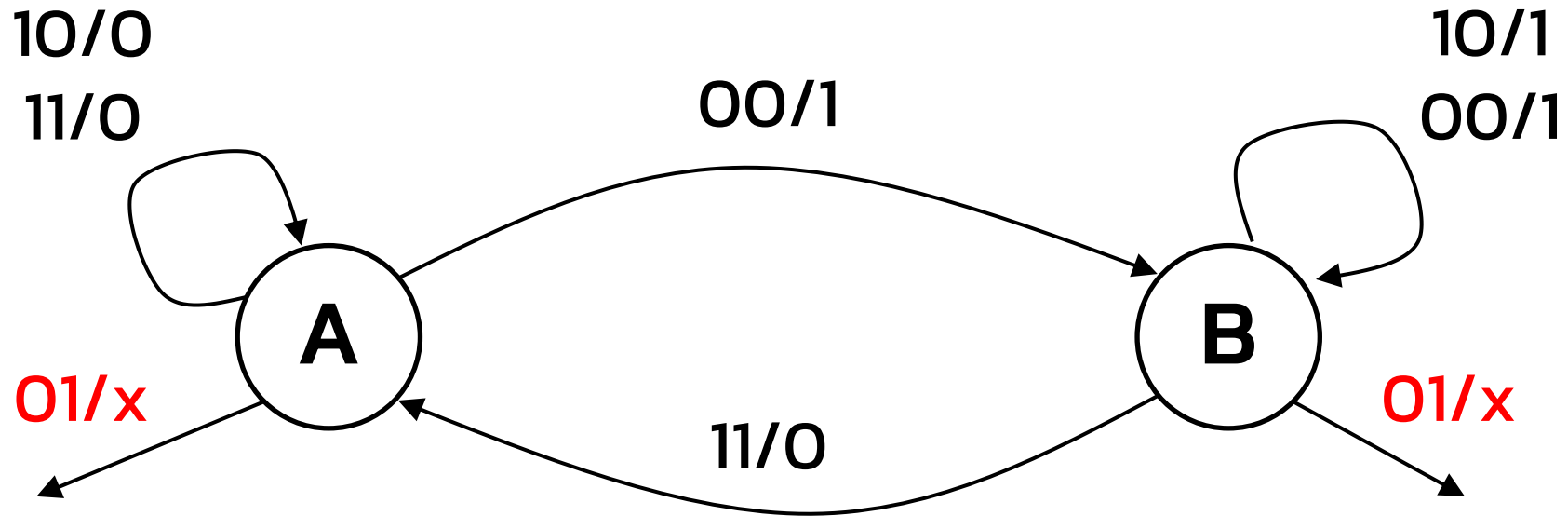


ความหมายของแต่ละ State

A คือ น้ำเต็มถึง สูงกว่า sensor ทั้ง 2 ตัว

B คือ น้ำน้อย ต่ำกว่า sensor ทั้ง 2 ตัว

แบบ Mealy



Input ตัวแรกคือ sensor ตัวล่าง, สีแดงคือ don't care

ความหมายของแต่ละ State

A คือ น้ำเต็มถัง สูงกว่า sensor ทั้ง 2 ตัว

B คือ น้ำน้อย ต่ำกว่า sensor ทั้ง 2 ตัว

STATE REDUCTION AND ASSIGNMENT

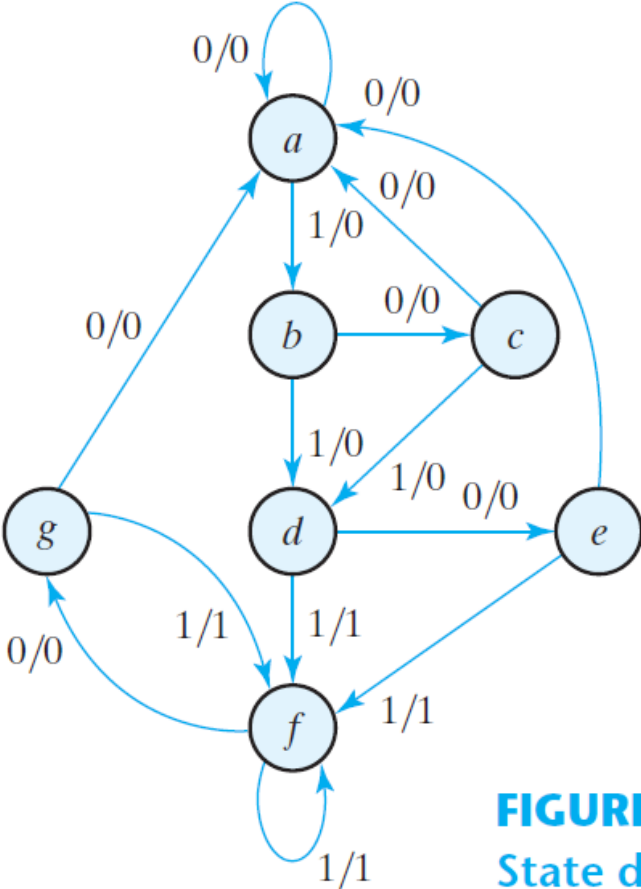


FIGURE 5.25
State diagram

state	<i>a</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>f</i>	<i>g</i>	<i>f</i>	<i>g</i>	<i>a</i>
input	0	1	0	1	0	1	1	0	1	0	0	
output	0	0	0	0	0	1	1	0	1	0	0	

Table 5.6
State Table

Present State	Next State		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
<i>a</i>	<i>a</i>	<i>b</i>	0	0
<i>b</i>	<i>c</i>	<i>d</i>	0	0
<i>c</i>	<i>a</i>	<i>d</i>	0	0
<i>d</i>	<i>e</i>	<i>f</i>	0	1
<i>e</i>	<i>a</i>	<i>f</i>	0	1
<i>f</i>	<i>g</i>	<i>f</i>	0	1
<i>g</i>	<i>a</i>	<i>f</i>	0	1

record ซ้ำ
แสดงว่า state
e กับ g เหมือนกัน

Table 5.7
Reducing the State Table

Present State	Next State		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
<i>a</i>	<i>a</i>	<i>b</i>	0	0
<i>b</i>	<i>c</i>	<i>d</i>	0	0
<i>c</i>	<i>a</i>	<i>d</i>	0	0
<i>d</i>	<i>e</i>	<i>f</i>	0	1
<i>e</i>	<i>a</i>	<i>f</i>	0	1
<i>f</i>	<i>e</i>	<i>f</i>	0	1

เอา state g ออกทั้ง record
และ replace g ในตารางด้วย e

Table 5.7
Reducing the State Table

Present State	Next State		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
-----▶ d	e	f	0	1
e	a	f	0	1
-----▶ f	e	f	0	1

Table 5.8
Reduced State Table

Present State	Next State		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	d	0	1
e	a	d	0	1

Table 5.8
Reduced State Table

Present State	Next State		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	d	0	1
e	a	d	0	1

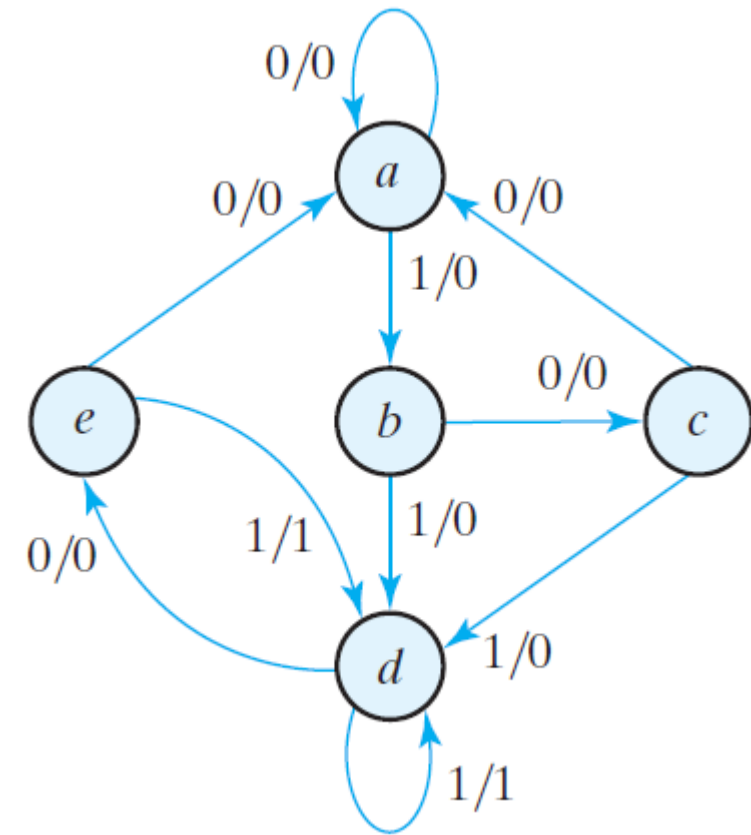


FIGURE 5.26
Reduced state diagram

State Assignment

Table 5.9
Three Possible Binary State Assignments

State	Assignment 1, Binary	Assignment 2, Gray Code	Assignment 3, One-Hot
a	000	000	00001
b	001	001	00010
c	010	011	00100
d	011	010	01000
e	100	110	10000

assignment ที่ต่างกัน ทำให้ได้วงจรต่างกัน และมีขนาดไม่เท่ากัน เช่น one-hot อาจจะใช้จำนวนบิต (state) มากกว่า แต่วงจรมีขนาดเล็กกว่า เป็นต้น

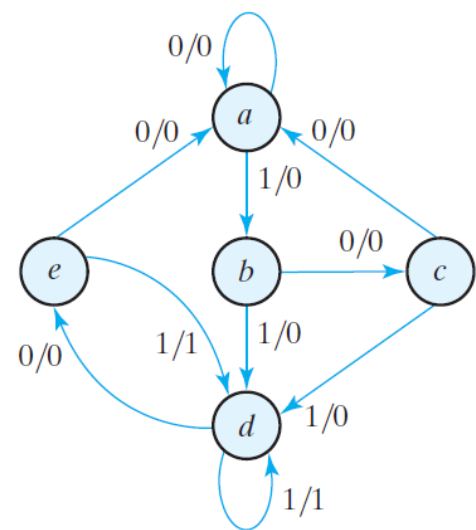
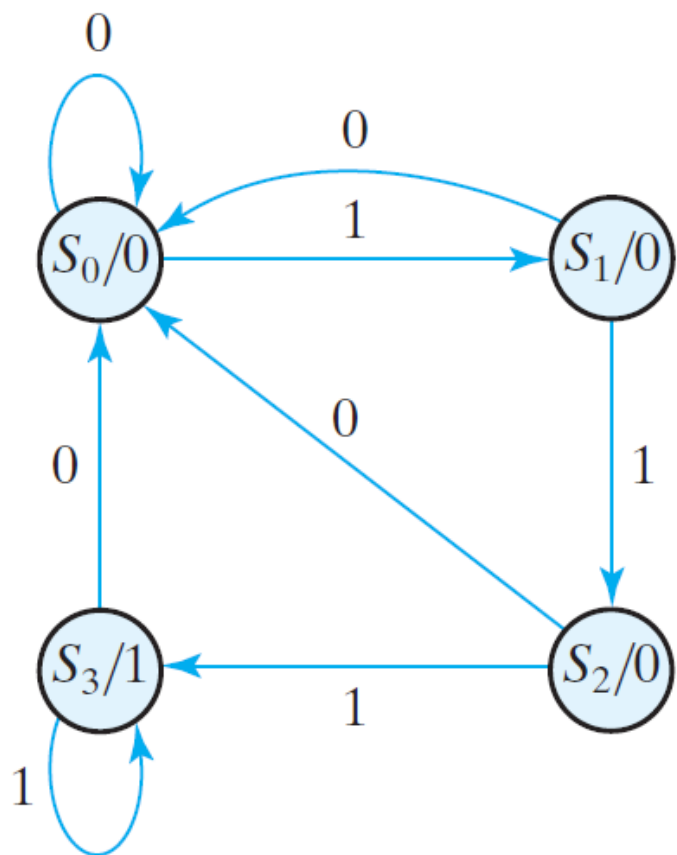


Table 5.10
Reduced State Table with Binary Assignment 1

Present State		Next State		Output	
		x = 0	x = 1	x = 0	x = 1
a	000	000	001	0	0
b	001	010	011	0	0
c	010	000	011	0	0
d	011	100	011	0	1
e	100	000	011	0	1

DESIGN PROCEDURE

1. From the word description and specifications of the desired operation, derive a state diagram for the circuit.
2. Reduce the number of states if necessary.
3. Assign binary values to the states.
4. Obtain the binary-coded state table.
5. Choose the type of flip-flops to be used.
6. Derive the simplified flip-flop input equations and output equations.
7. Draw the logic diagram.



เช็คว่ามี subsequence 111 ใน input หรือไม่?

IN	0	1	0	0	1	1	0	1	0	1	0	1	1	1	1	1	1	0	1	0	1	
OUT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0

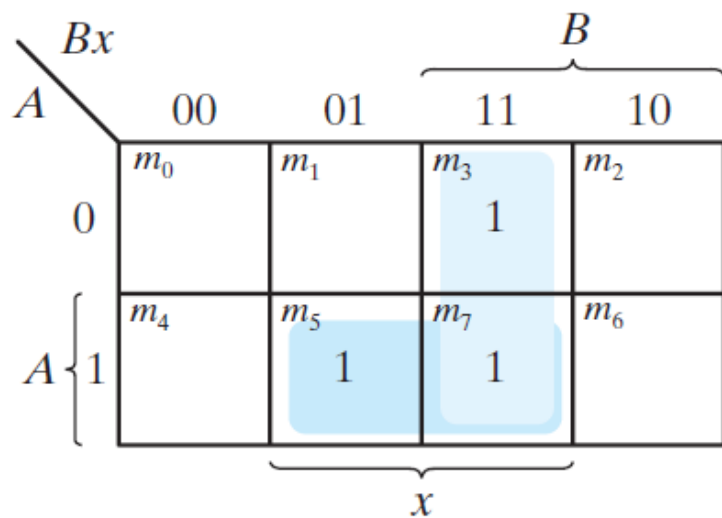
FIGURE 5.27

State diagram for sequence detector

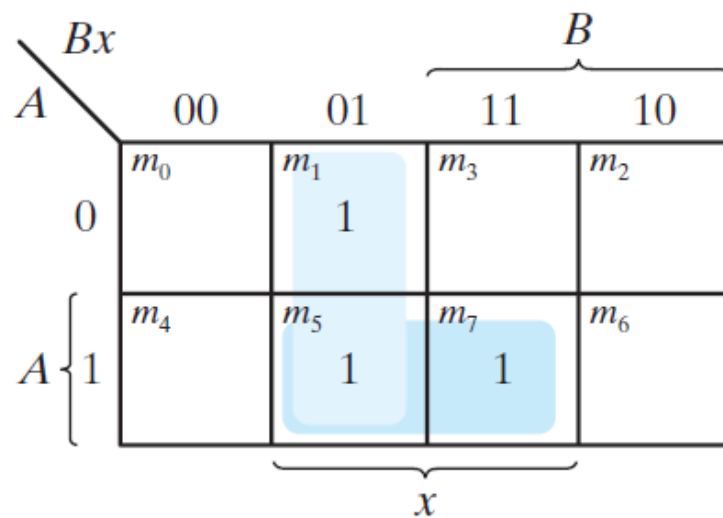
Table 5.11
State Table for Sequence Detector

Present State		Input	Next State		Output
<i>A</i>	<i>B</i>		<i>A</i>	<i>B</i>	
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1

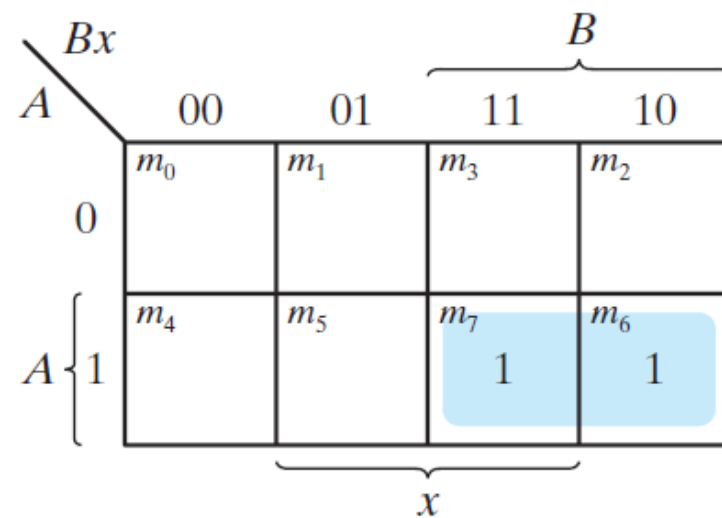
State assignment 000 binary



$$D_A = Ax + Bx$$



$$D_B = Ax + B'x$$



$$y = AB$$

$$A(t + 1) = D_A(A, B, x) = \Sigma(3, 5, 7)$$

$$B(t + 1) = D_B(A, B, x) = \Sigma(1, 5, 7)$$

$$y(A, B, x) = \Sigma(6, 7)$$

$$D_A = Ax + Bx$$

$$D_B = Ax + B'x$$

$$y = AB$$

$$D_A = Ax + Bx$$

$$D_B = Ax + B'x$$

$$y = AB$$

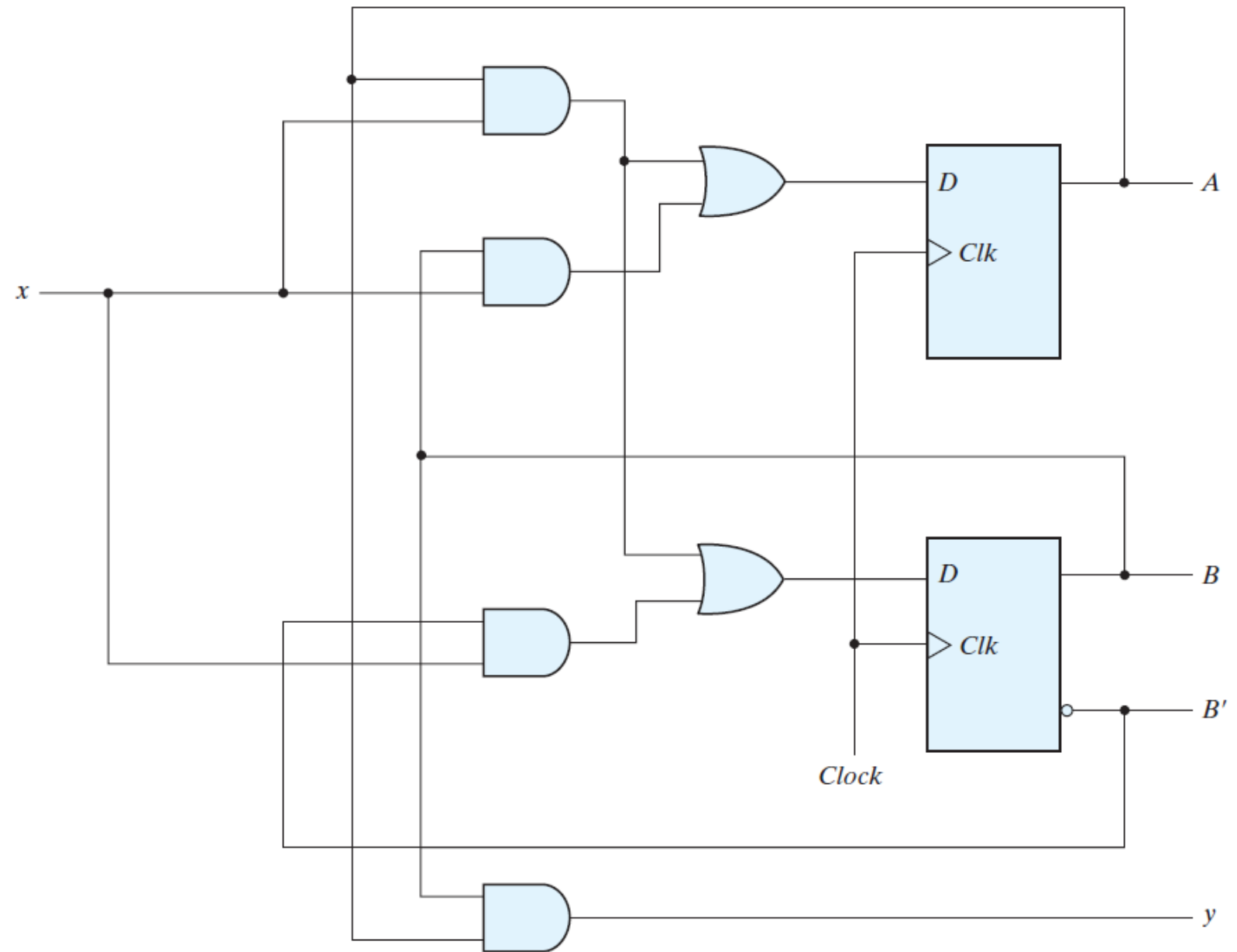
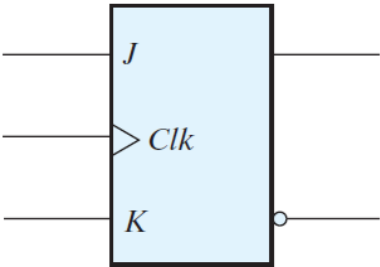


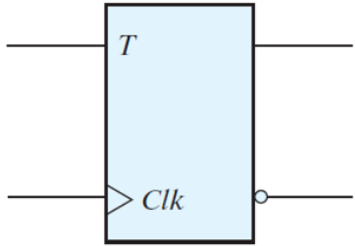
FIGURE 5.29

Logic diagram of a Moore-type sequence detector

ถ้าไม่ใช่ D flip-flop ?



JK Flip-Flop			
J	K	Q(t + 1)	
0	0	Q(t)	No change
0	1	0	Reset
1	0	1	Set
1	1	Q'(t)	Complement



T Flip-Flop		
T	Q(t + 1)	
0	Q(t)	No change
1	Q'(t)	Complement

Table 5.12
Flip-Flop Excitation Tables

Q(t)	Q(t = 1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

(a) JK Flip-Flop

Q(t)	Q(t = 1)	T
0	0	0
0	1	1
1	0	1
1	1	0

(b) T Flip-Flop

JK และ T ทำได้แค่ set, reset, ฟังก์ชันที่ relative กับ previous state ไม่เหมือน D ที่เราสร้างฟังก์ชัน next state ได้ตามใจ

Table 5.13
State Table and JK Flip-Flop Inputs

$Q(t)$	$Q(t = 1)$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Present State		Input	Next State		Flip-Flop Inputs			
A	B		A	B	J_A	K_A	J_B	K_B
0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1

โจทย์ใหม่แล้ว ใช้ state diagram อะไรก็ไม่รู้ (ไม่สำคัญ)

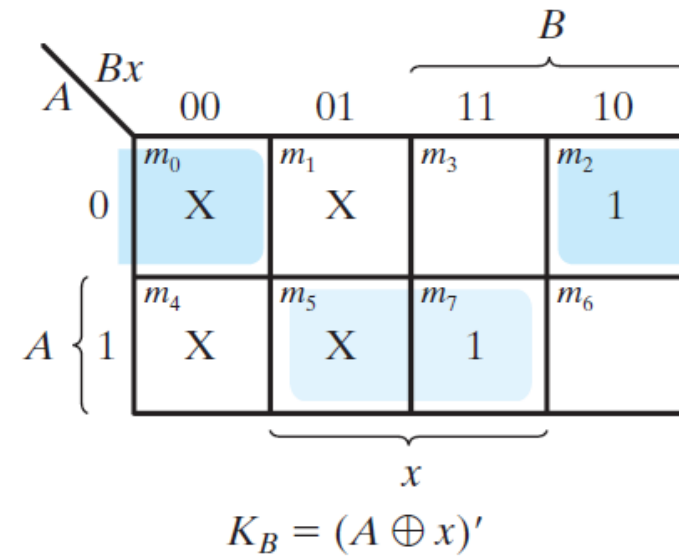
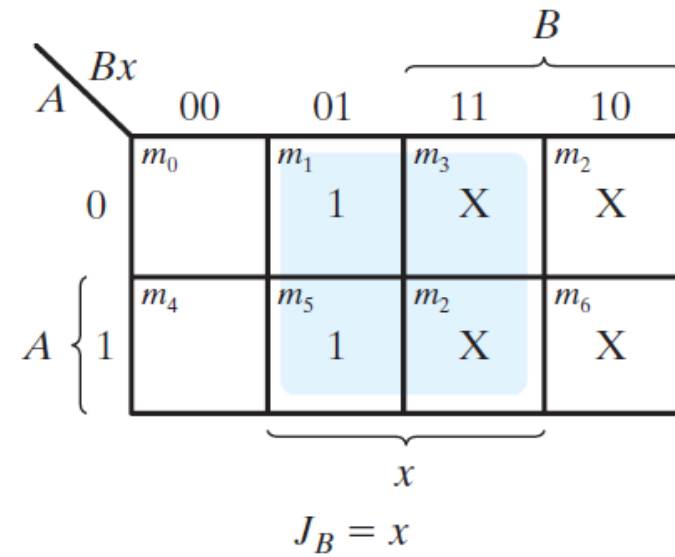
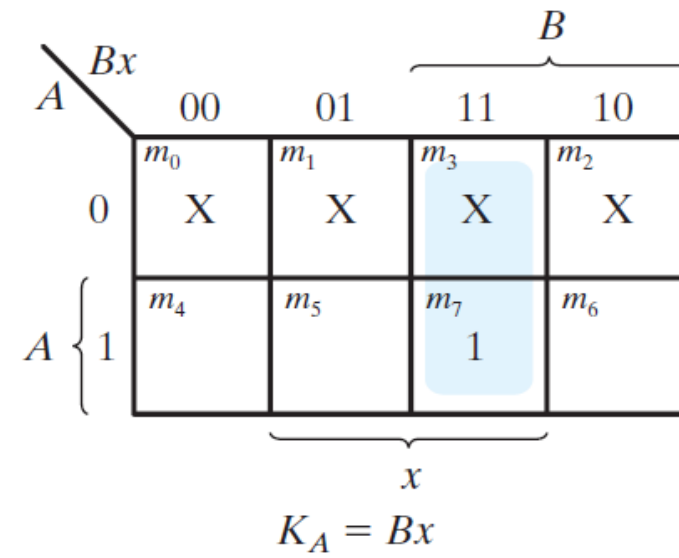
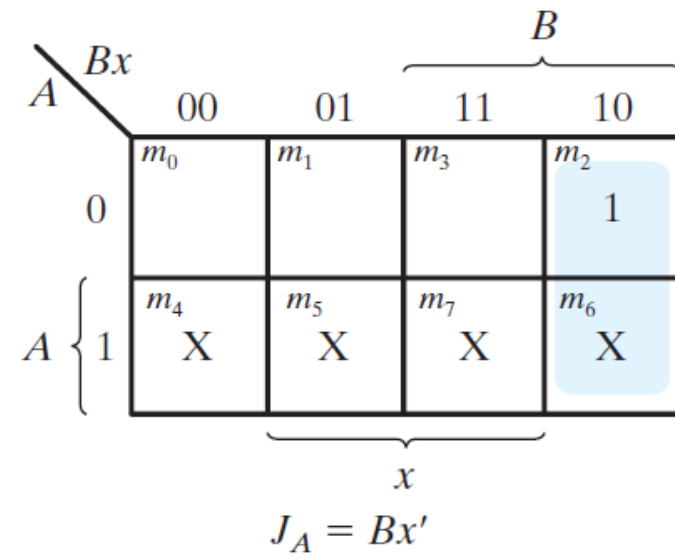
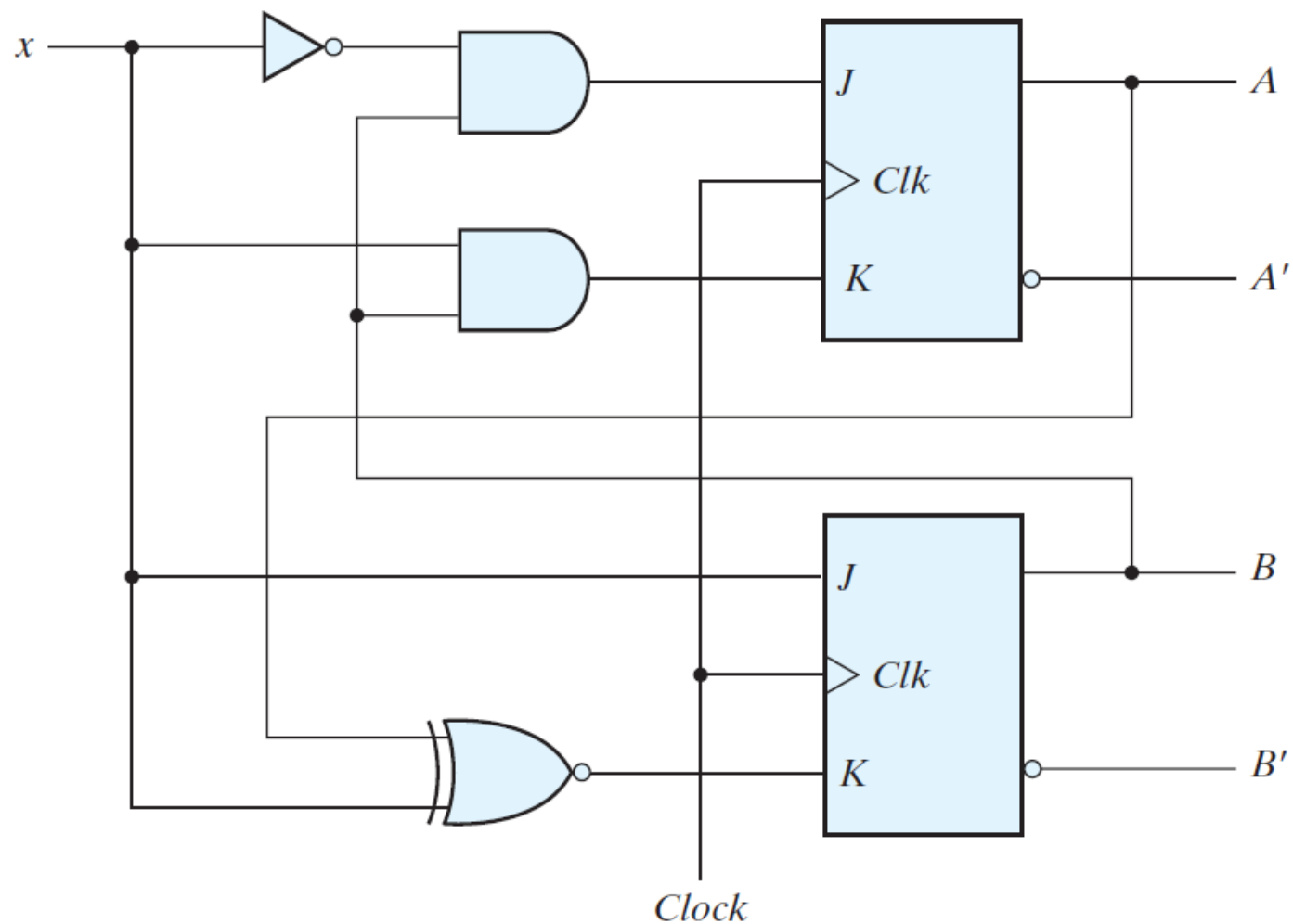


FIGURE 5.30
Maps for J and K input equations



$$J_A = Bx'$$

$$K_A = Bx$$

$$J_B = x$$

$$K_B = (A \oplus x)'$$

FIGURE 5.31

Logic diagram for sequential circuit with J/K flip-flops

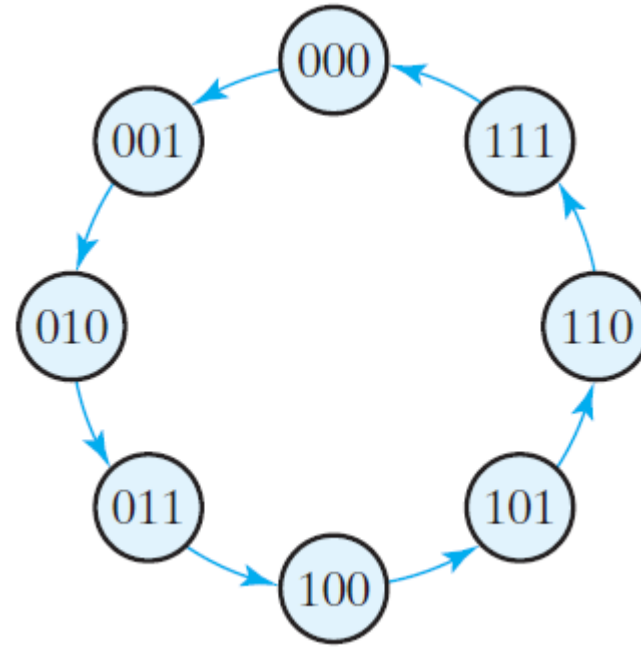
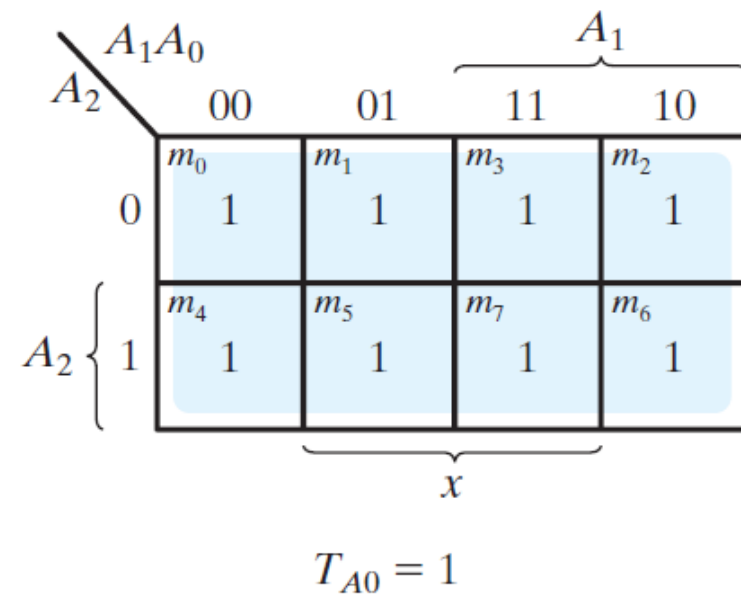
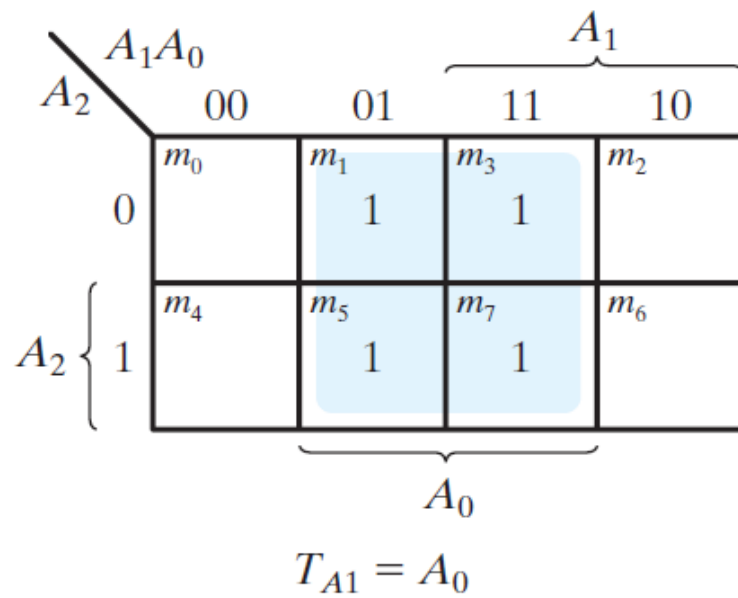
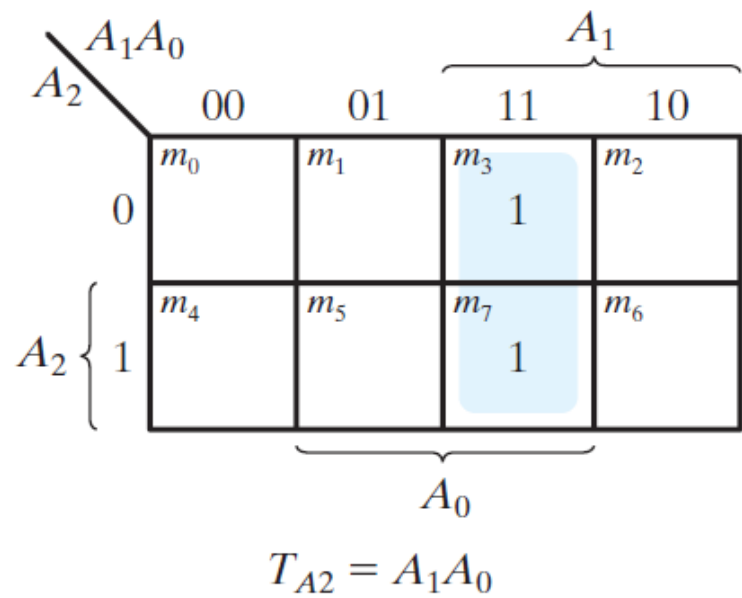


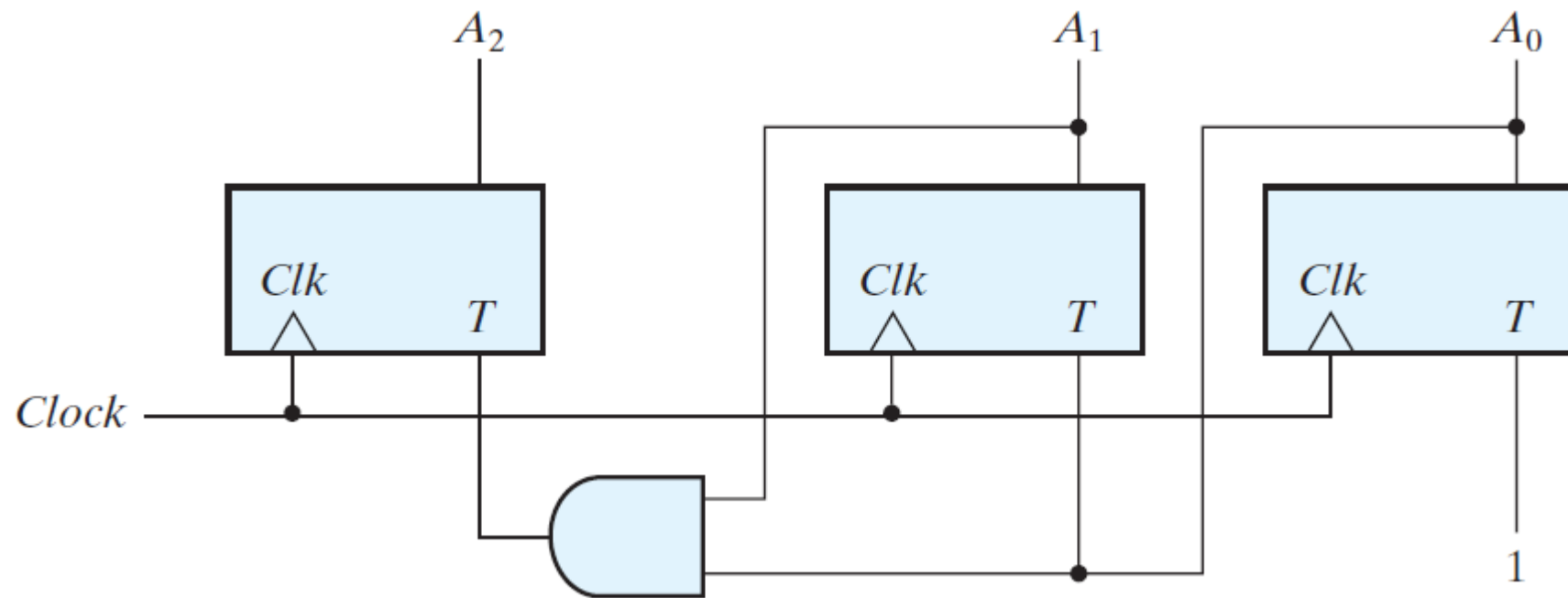
FIGURE 5.32

State diagram of three-bit binary counter

Table 5.14
State Table for Three-Bit Counter

Present State			Next State			Flip-Flop Inputs		
A_2	A_1	A_0	A_2	A_1	A_0	T_{A2}	T_{A1}	T_{A0}
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	1	1
1	1	1	0	0	0	1	1	1





$$T_{A2} = A_1 A_0$$

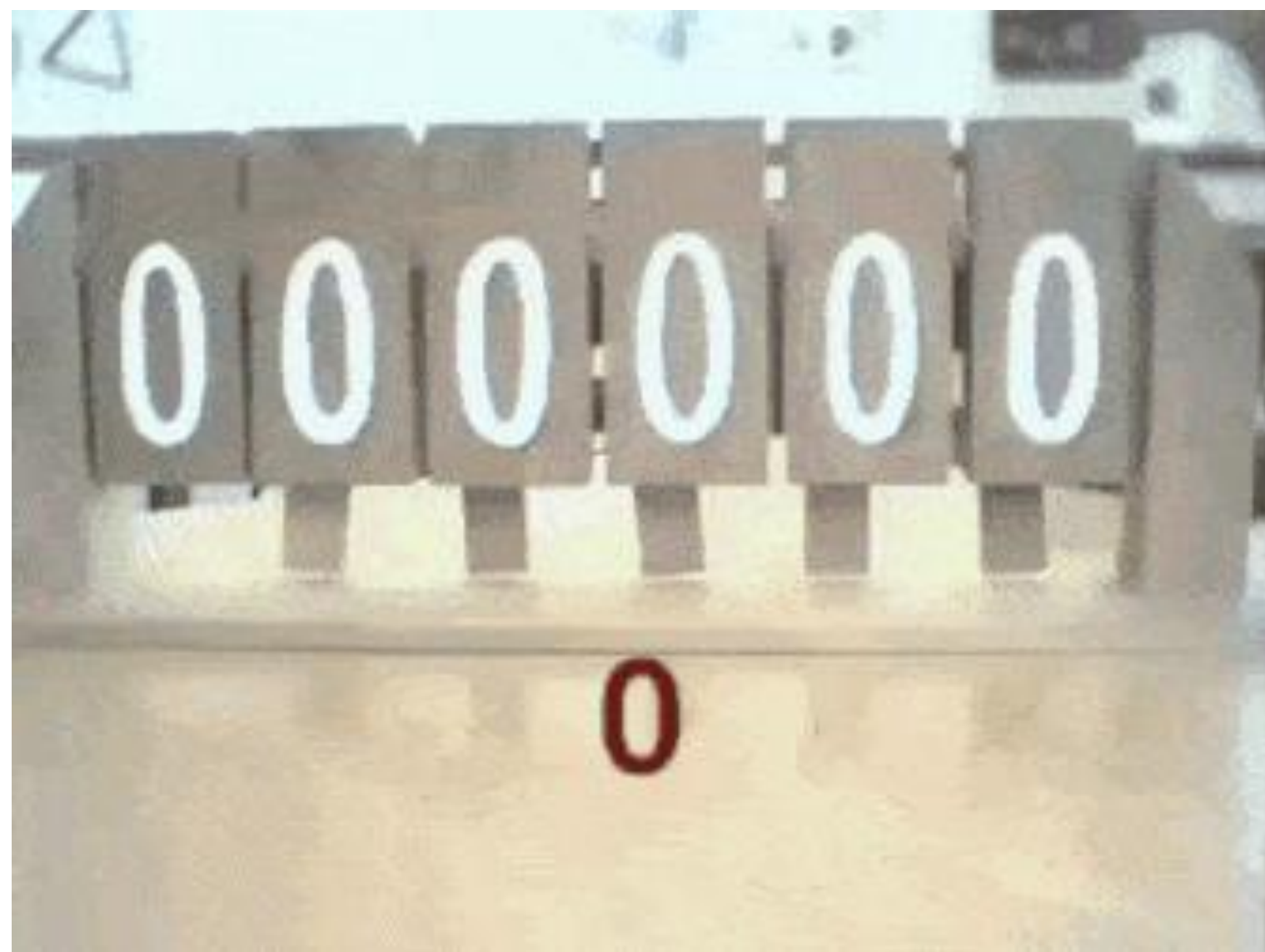
$$T_{A1} = A_0$$

$$T_{A0} = 1$$

ให้ flip บิต (invert)
เมื่อบิตทางขวาเป็น 1 หมุน

FIGURE 5.34

Logic diagram of three-bit binary counter



PROBLEMS

- 5.1** The D latch of Fig. 5.6 is constructed with four NAND gates and an inverter. Consider the following three other ways for obtaining a D latch. In each case, draw the logic diagram and verify the circuit operation.
- (a) Use NOR gates for the SR latch part and AND gates for the other two. An inverter may be needed.
 - (b) Use NOR gates for all four gates. Inverters may be needed.
 - (c) Use four NAND gates only (without an inverter). This can be done by connecting the output of the upper gate in Fig. 5.6 (the gate that goes to the SR latch) to the input of the lower gate (instead of the inverter output).

5.2 Construct a JK flip-flop using a D flip-flop, a two-to-one-line multiplexer, and an inverter. (HDL—see Problem 5.34.)

5.3 Show that the characteristic equation for the complement output of a JK flip-flop is

$$Q'(t + 1) = J'Q' + KQ$$

5.4 A PN flip-flop has four operations: clear to 0, no change, complement, and set to 1, when inputs P and N are 00, 01, 10, and 11, respectively.

- (a) Tabulate the characteristic table.
- (b)* Derive the characteristic equation.
- (c) Tabulate the excitation table.
- (d) Show how the PN flip-flop can be converted to a D flip-flop.

5.5 Explain the differences among a truth table, a state table, a characteristic table, and an excitation table. Also, explain the difference among a Boolean equation, a state equation, a characteristic equation, and a flip-flop input equation.