

CS2022: 數位系統設計

Synchronous Sequential Logic

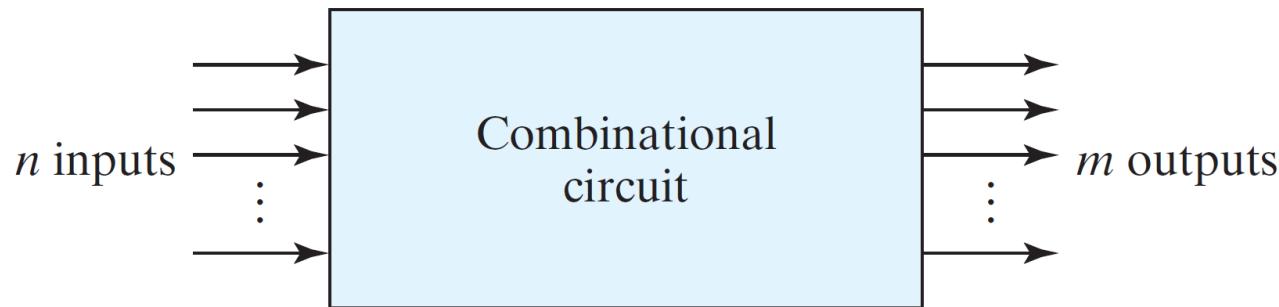
Outline

- **Introduction**
- **Sequential Circuits**
- **Storage Element: Latches**
- **Storage Element: Flip-Flops**
- **Analysis of Clocked Sequential Circuits**
- **State Reduction and Assignment**
- **Design Procedure**

Introduction

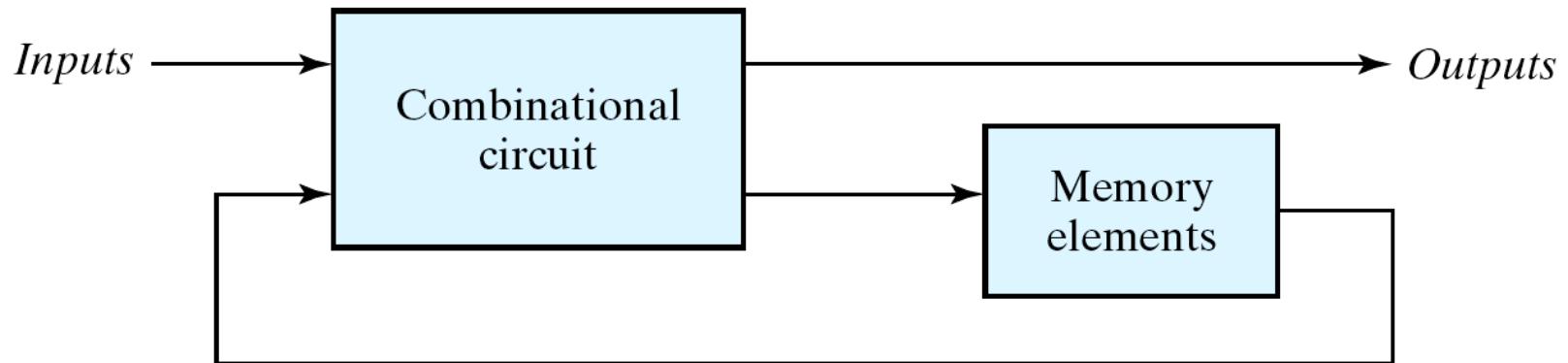
□ Combinational circuits

- ◆ Contains no memory elements
- ◆ The outputs depends on the inputs



Sequential Circuits

■ Sequential circuits



- ◆ A feedback path
- ◆ The state of the sequential circuit
- ◆ $(\text{inputs}, \text{current state}) \Rightarrow (\text{outputs}, \text{next state})$
- ◆ Synchronous: the transition happens at discrete instants of time
- ◆ Asynchronous: at any instant of time

Synchronous Sequential Circuits

■ Synchronous sequential circuits

- ◆ A master-clock generator to generate a periodic train of clock pulses
- ◆ The clock pulses are distributed throughout the system
- ◆ Clocked sequential circuits
- ◆ Most commonly used
- ◆ Seldom instability problems
- ◆ The memory elements: flip-flops
 - » Each binary cell is capable of storing one bit of information
 - » Two outputs: one for the normal value and one for the complement value
 - » Maintain a binary state indefinitely until directed by an input signal to switch state



Duty cycle of a clock (pulse train)

Clocked Sequential Circuits

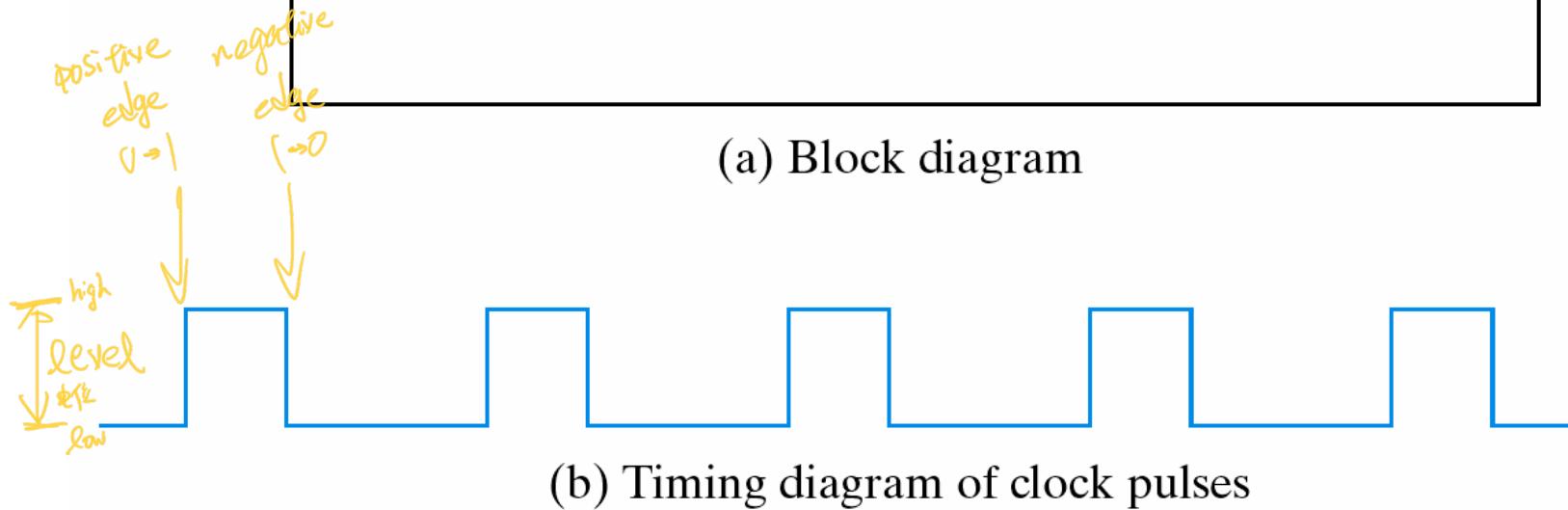
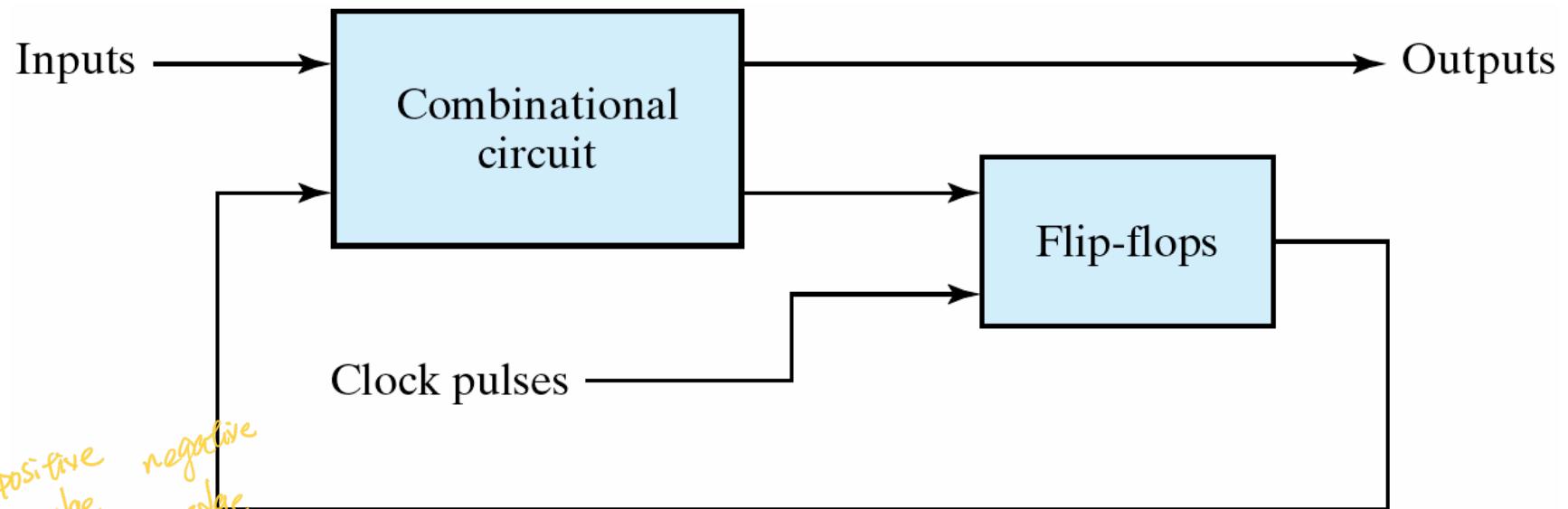


Fig. 2 Synchronous clocked sequential circuit

Latches

Latches
是不同时钟的锁

- ❑ Latches are level-sensitive devices
- ❑ Flip-flops are edge-sensitive devices
- ❑ Latches are asynchronous sequential circuits
 - ◆ State changes whenever inputs change
 - ◆ Building blocks of flip-flops
 - ◆ Not practical for use in synchronous sequential circuits

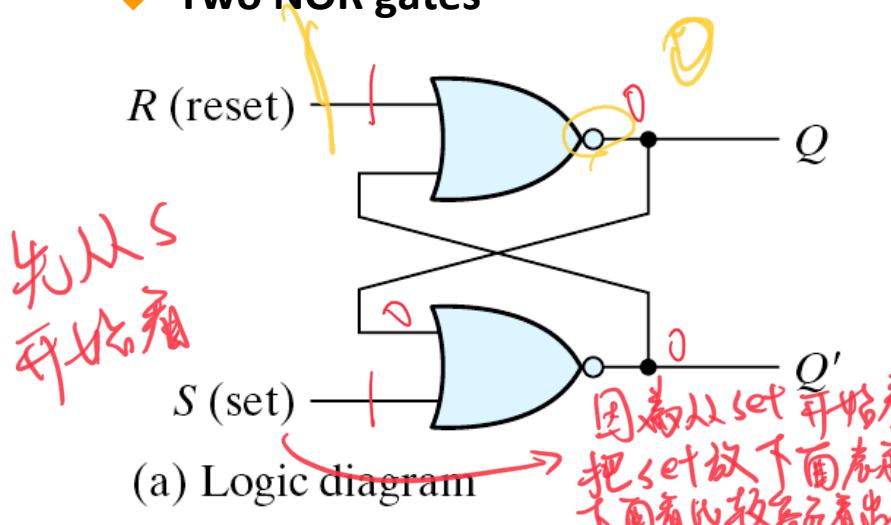
实存

同步时

SR Latch with NOR Gates

SR latch with NOR gates

- Two NOR gates



$Q = 1$ = set Mode
 $Q' = 1$ = Reset Mode

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

禁止的

- More complicated types can be built upon it

- Cross-coupled connection

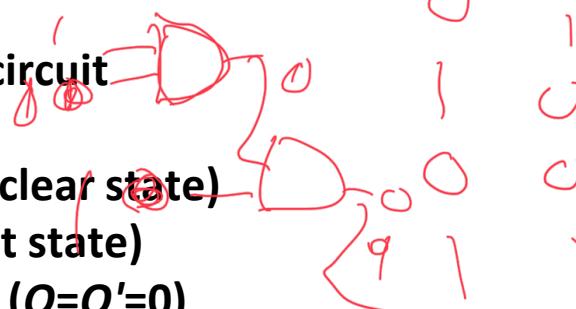
- An asynchronous sequential circuit

- $(S, R) = (0, 0)$: no operation

$(S, R) = (0, 1)$: reset ($Q=0$, the clear state)

$(S, R) = (1, 0)$: set ($Q=1$, the set state)

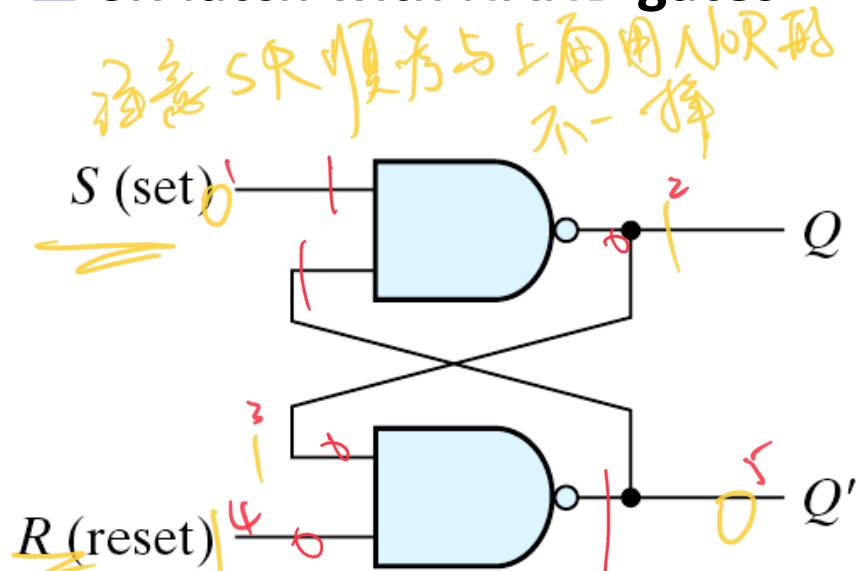
$(S, R) = (1, 1)$: forbidden state ($Q=Q'=0$)



S	R	Q	\bar{Q}
0	1	0	1
1	0	1	0
0	0	0	1
1	1	0	0

SR Latch with NAND Gates

■ SR latch with NAND gates



(a) Logic diagram

R是置0
S是置1

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

Fig. 4 SR latch with NAND gates

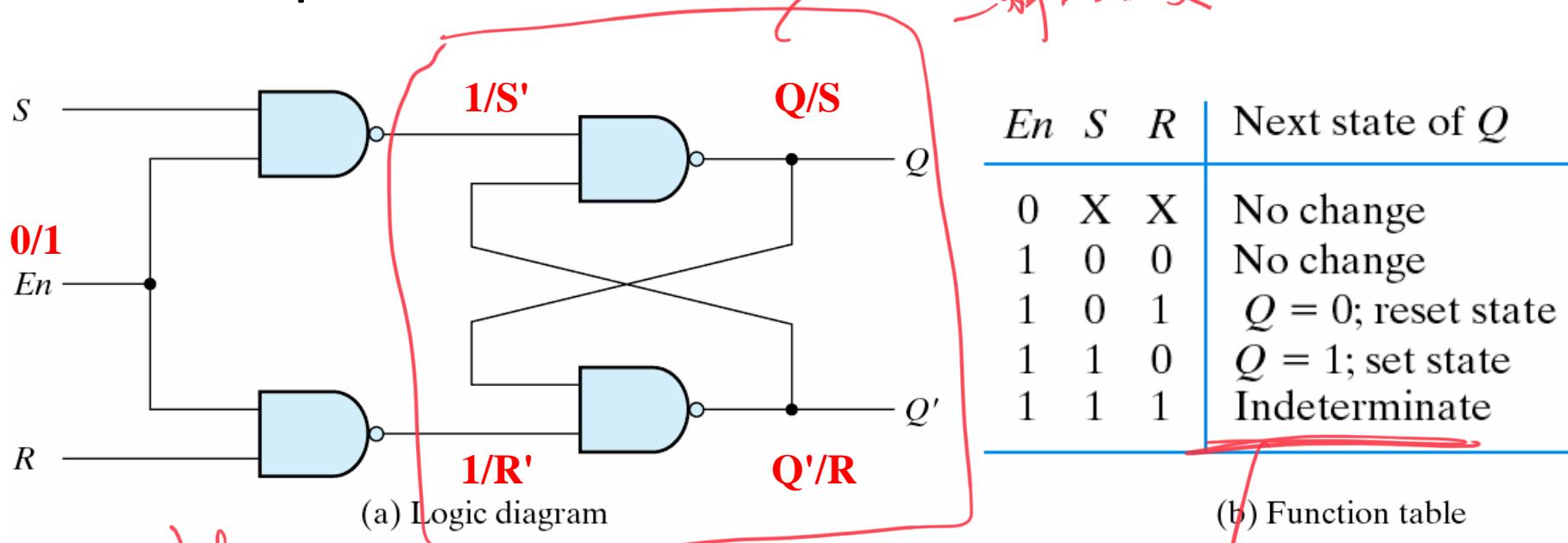


STUDENT

SR Latch with Control

SR latch with control input

- ◆ $En=0$: no change
- ◆ $En=1$: operates as normal SR latch



enable
就可以维持不变

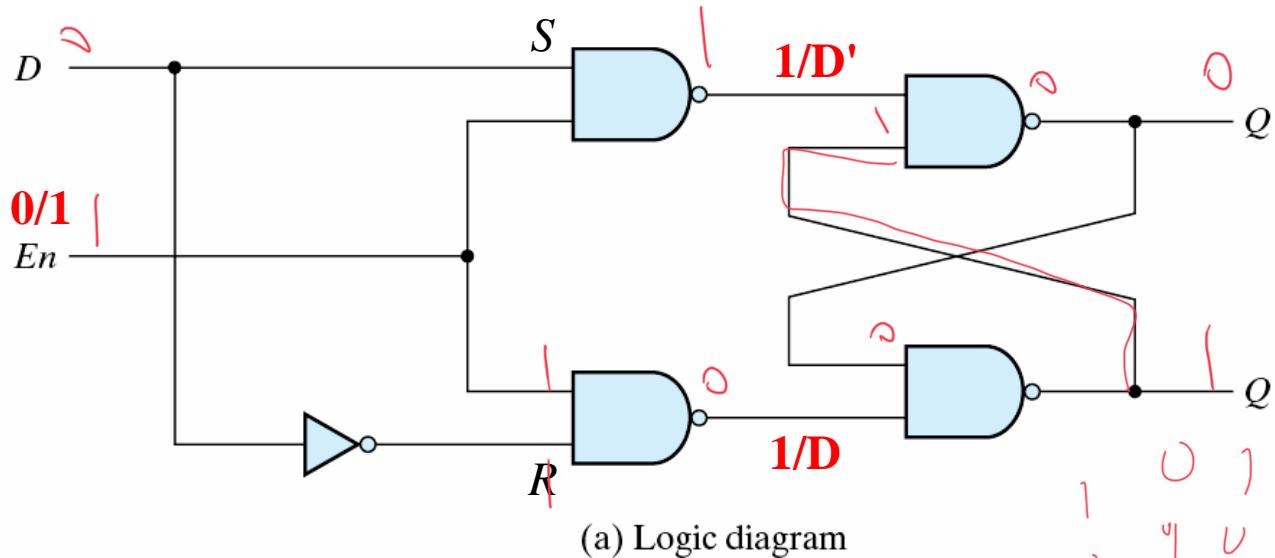
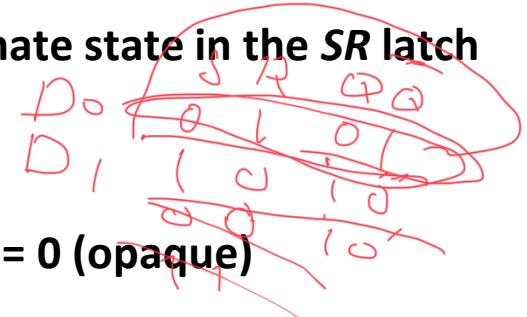
Fig. 5 SR latch with control input

无法决定(禁止)

D Latch (Transparent Latch)

D Latch

- ◆ Eliminate the undesirable conditions of the indeterminate state in the SR latch
- ◆ D: data
- ◆ Gated D-latch
- ◆ $D \Rightarrow Q$ when $En = 1$ (transparent); no change when $En = 0$ (opaque)

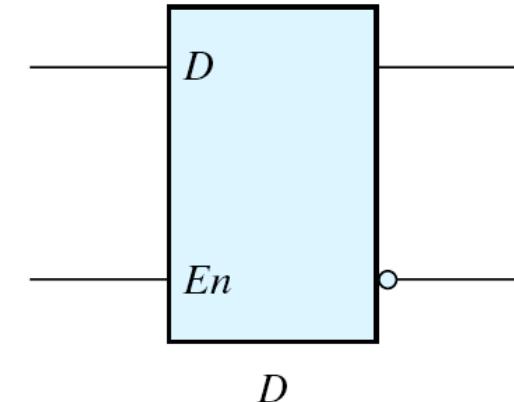
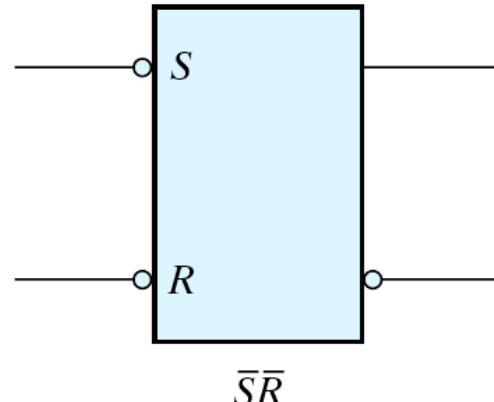
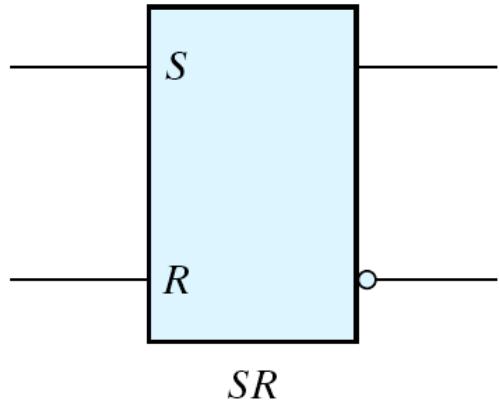


En	D	Next state of Q
0	X	No change
1	0	$Q = 0$; reset state
1	1	$Q = 1$; set state

1	0	1	0	1	1	0	1
1	1	1	1	0	0	1	1
0	0	0	0	0	0	0	0
1	1	0	1	1	0	1	0
1	0	1	0	0	1	0	1

Fig. 6 D latch

Various Graphic Symbols for Latches



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)

SR latch with NOR gates

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)

SR latch with NAND gates

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

D latch with enable

Fig. 7 Graphic symbols for latches

Flip-Flops

- **Trigger**
 - ◆ The state of a latch/flip-flop is switched by a change of the control input
- **Level triggered – Latch**
 - ◆ The state transition starts as soon as the clock is during logic 1 (positive level-sensitive) or logic 0 (negative level-sensitive)
- **Edge triggered – Flip-Flop**
 - ◆ The state transition starts only at positive (positive edge-triggered) or negative edge (negative edge-triggered) of the clock signal

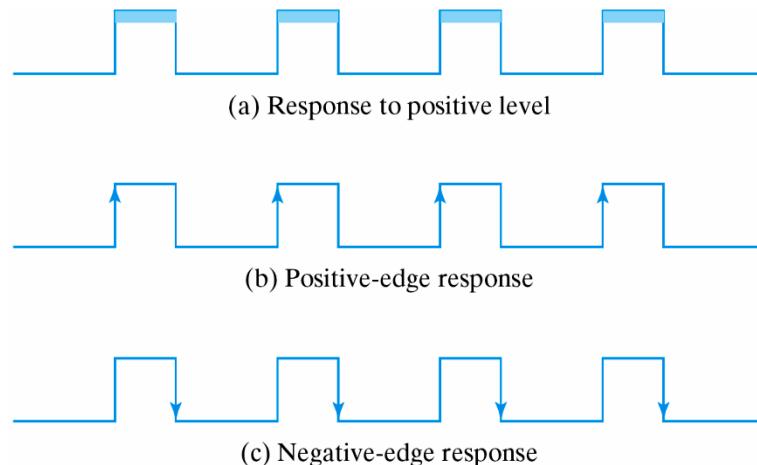
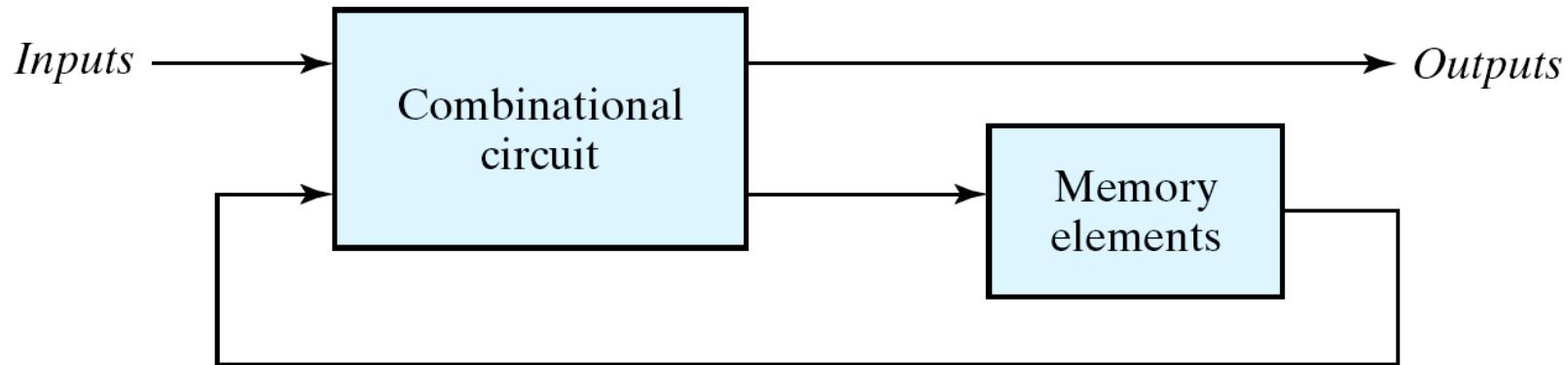


Fig. 8 Clock response in latch and flip-flop

Level-Triggered vs. Edge-Triggered



□ Level-triggered

- ◆ The feedback path may cause instability problem

□ Edge-triggered

- ◆ The state transition happens only at the edge
- ◆ Eliminate the multiple-transition problem

Master-Slave D Flip-Flop

■ Master-slave *D* flip-flop

- ◆ Two separate latches
 - » A master latch (positive-level triggered)
 - » A slave latch (negative-level triggered)
 - » Negative-edge-triggered flip-flop

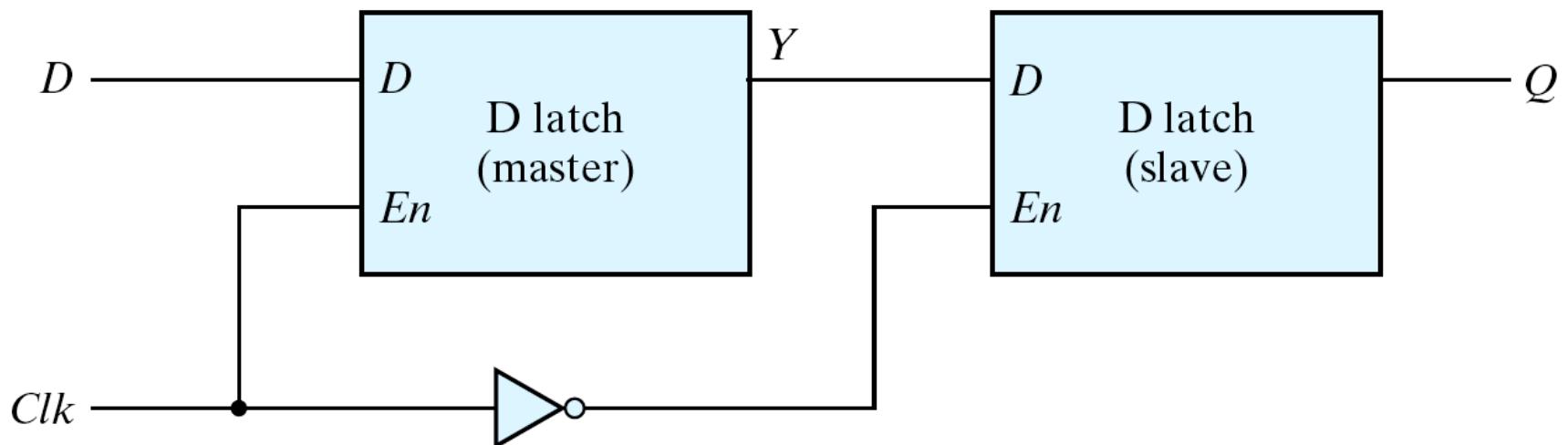


Fig. 9 Master-slave D flip-flop



Edge-Triggered D Flip-Flop (1/4)

□ Edge-triggered flip-flop

- ◆ The state changes during a clock-pulse transition

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

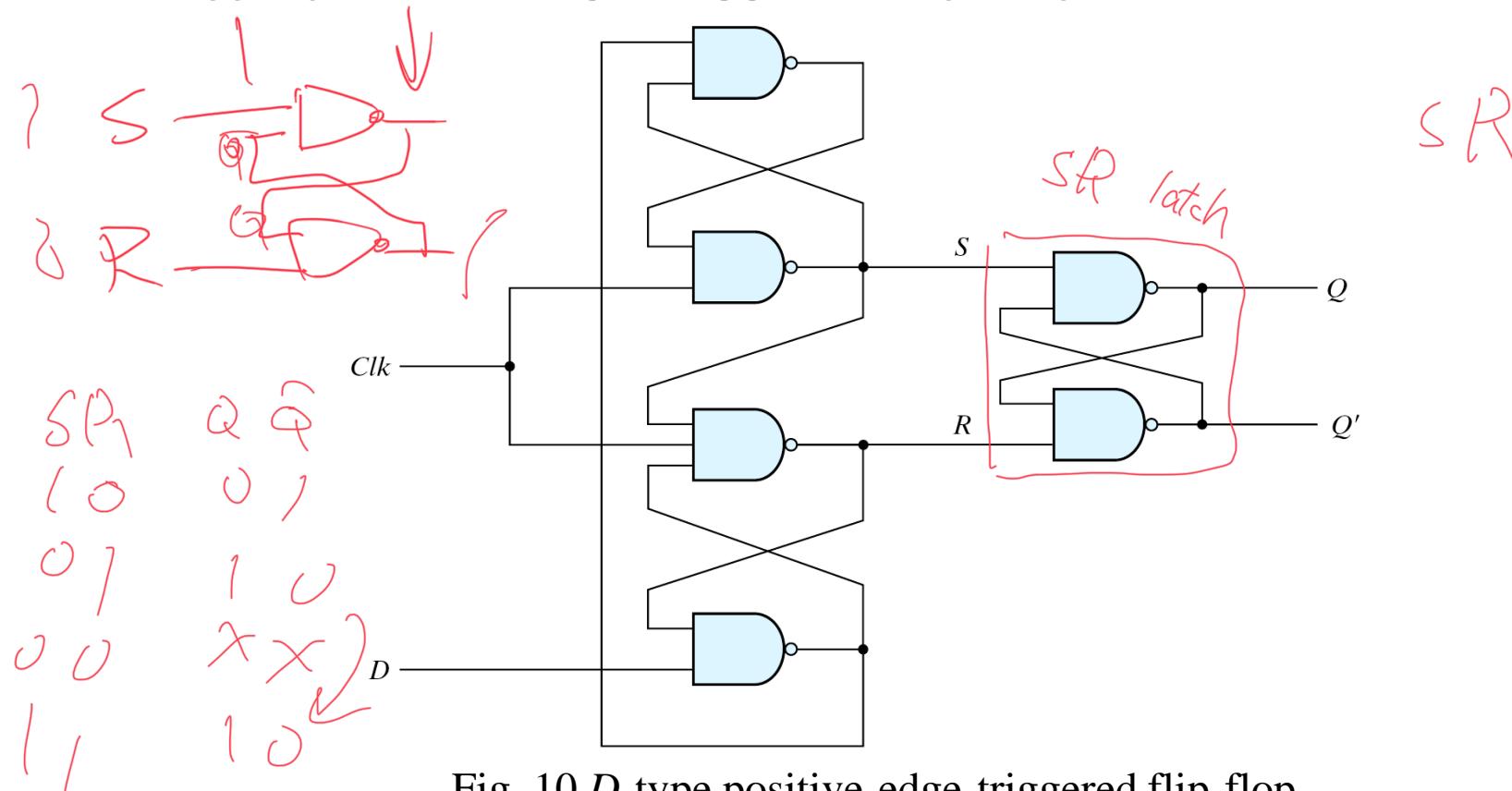


Fig. 10 D-type positive-edge-triggered flip-flop
Synchronous Sequential Logic-16

Edge-Triggered D Flip-Flop (2/4)

- ◆ Three basic *SR* latches

- » $(S, R) = (0, 1)$: $Q = 1$
- » $(S, R) = (1, 0)$: $Q = 0$
- » $(S, R) = (1, 1)$: no operation
- » $(S, R) = (0, 0)$: should be avoided

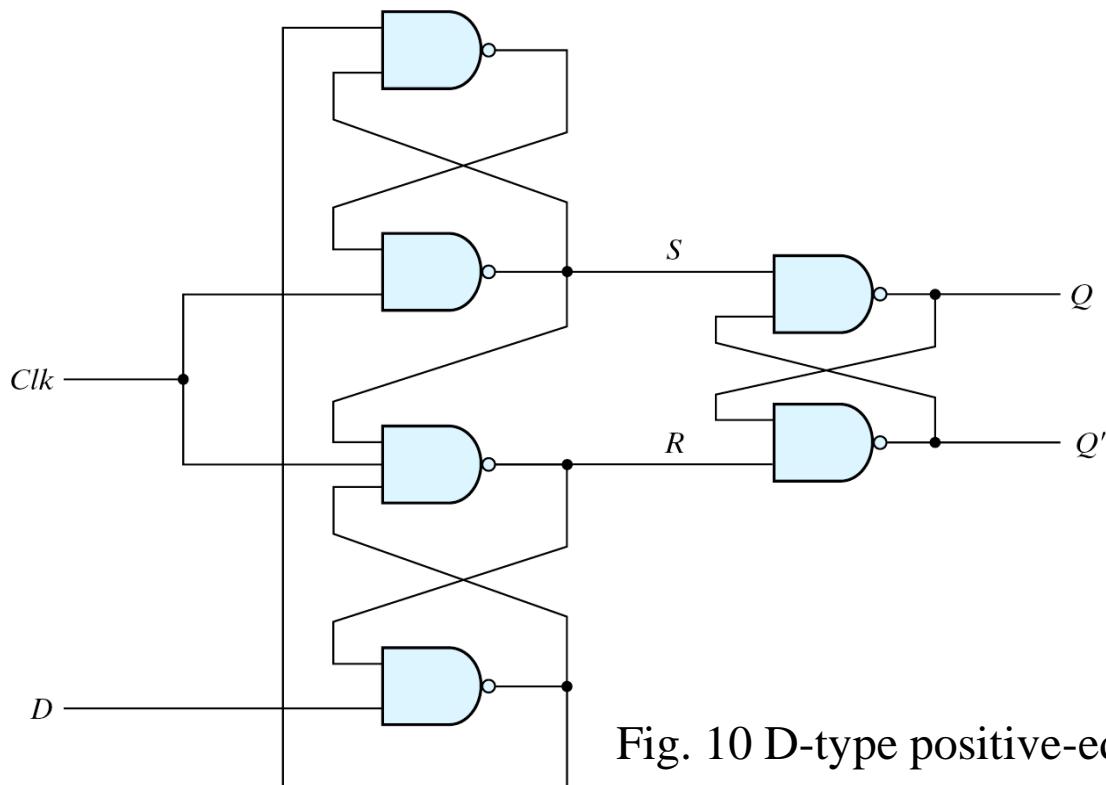
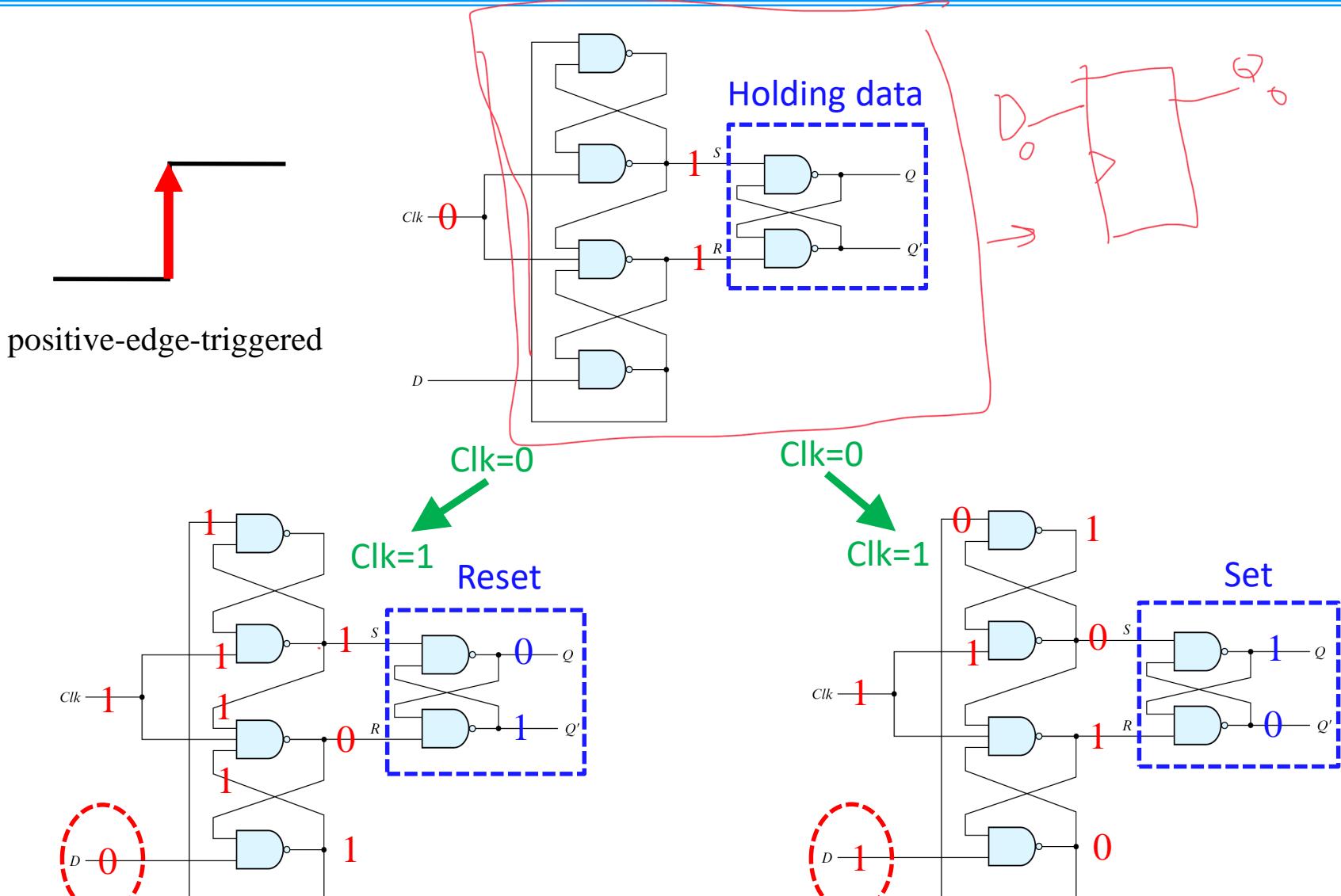


Fig. 10 D-type positive-edge-triggered flip-flop

Edge-Triggered D Flip-Flop (3/4)



Edge-Triggered D Flip-Flop (4/4)

■ Summary

- ◆ $CLK = 0$: $(S, R) = (1, 1)$, no state change
- ◆ $CLK = \uparrow$: state change once
- ◆ $CLK = 1$ and \downarrow : state holds
- ◆ Eliminate the feedback problems in sequential circuits

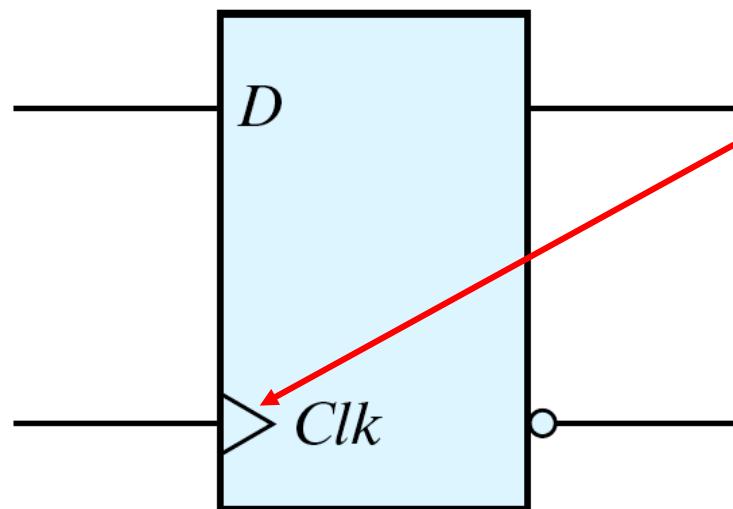
■ All flip-flops must make their transitions at the same time

- ◆ Clock Tree Synthesis: clock signal is distributed evenly to all sequential elements (i.e., flip-flops) in a design
- ◆ Objective: minimize skew and latency

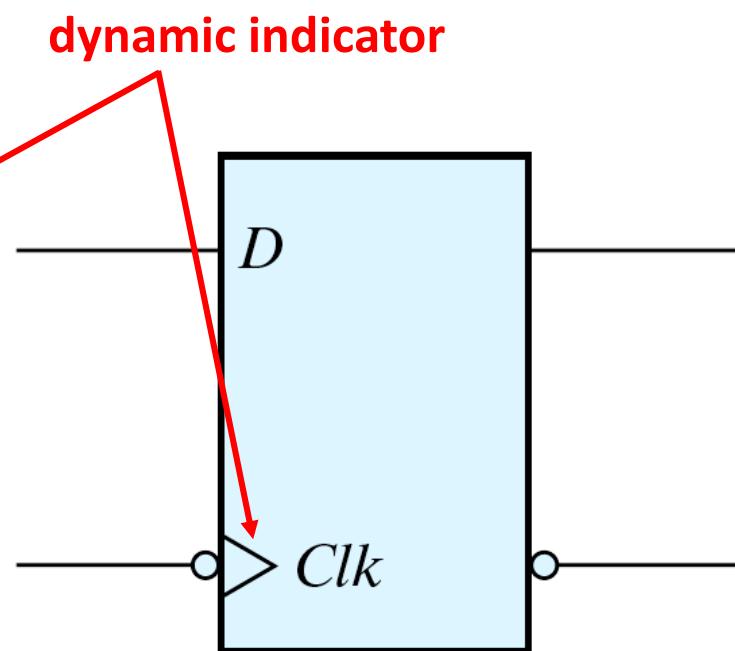
D Flip-Flops Graphic Symbols

■ The edge-triggered D flip-flops

- ◆ The most economical and efficient
- ◆ Positive-edge and negative-edge



(a) Positive-edge



(a) Negative-edge

Fig. 11 Graphic symbols for edge-triggered D flip-flop

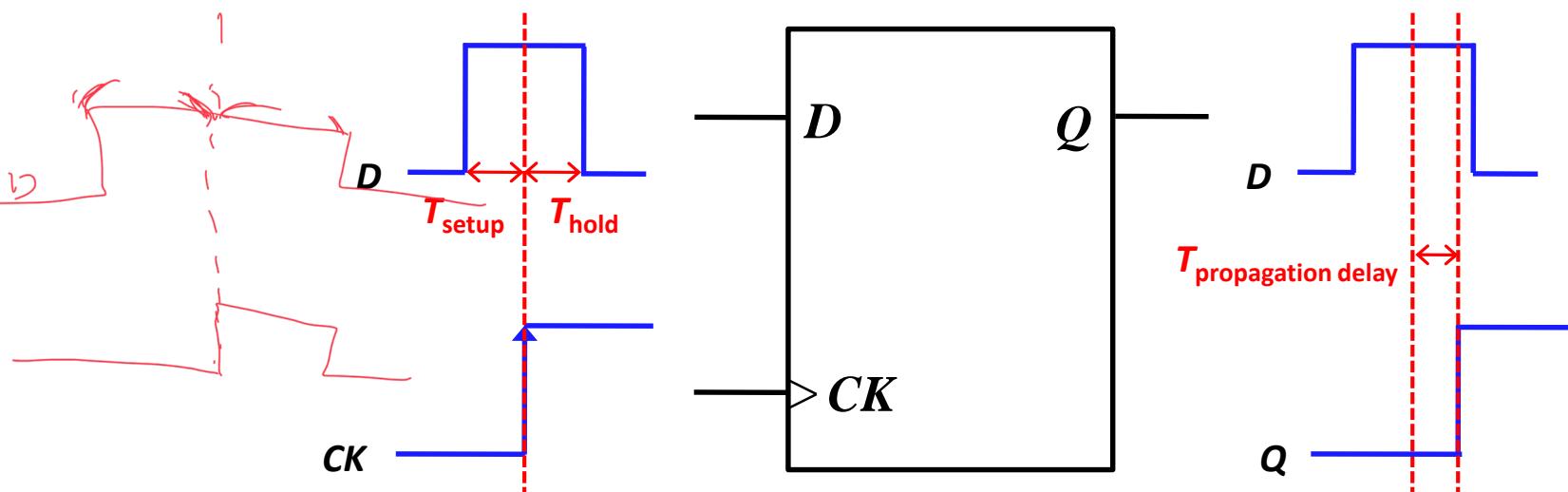
Setup and Hold Time for Flip-Flop

□ The setup time

- ◆ D input must be maintained at a constant value prior to the application of the positive CP pulse
- ◆ Data to the internal latches

□ The hold time

- ◆ D input must not change after the application of the positive CP pulse
- ◆ Clock to the internal latch



JK Flip-Flop

JK flip-flop

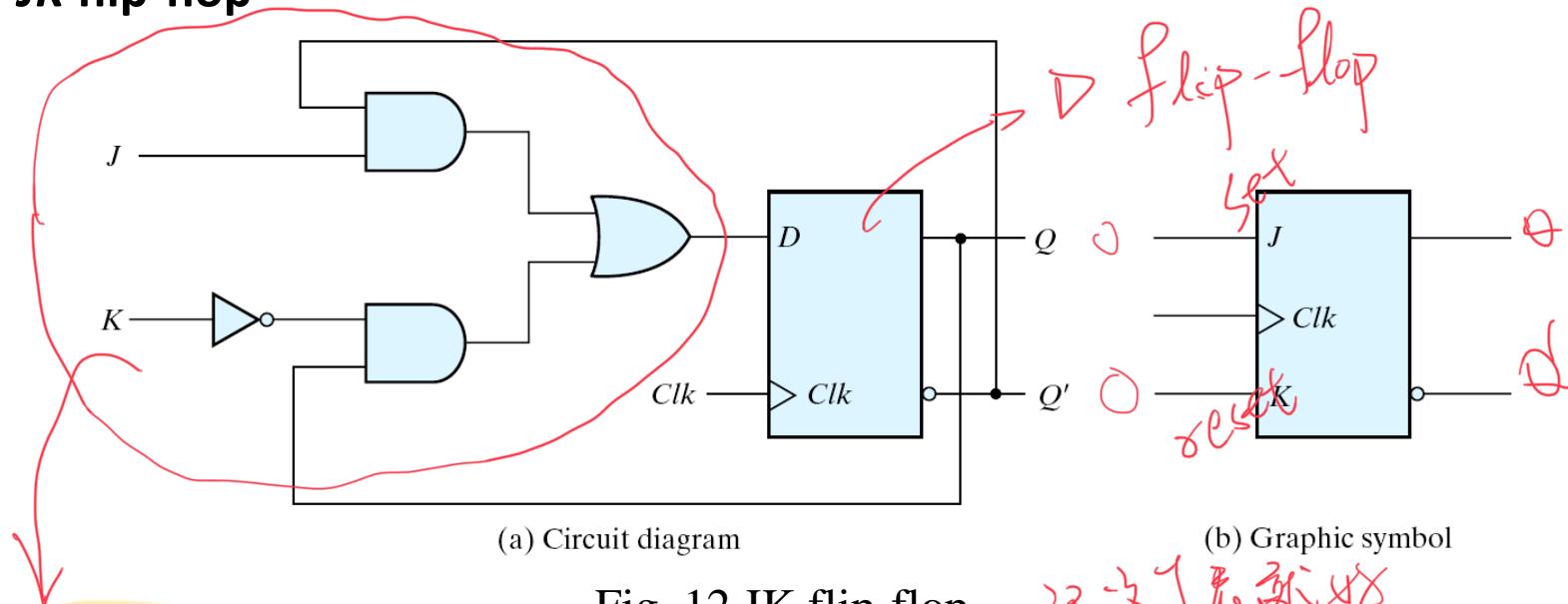


Fig. 12 JK flip-flop

$D = JQ' + K'Q$

- ◆ $J=0, K=0: D=Q$, no change
- ◆ $J=0, K=1: D=0 \Rightarrow Q=0$
- ◆ $J=1, K=0: D=1 \Rightarrow Q=1$
- ◆ $J=1, K=1: D=Q' \Rightarrow Q=Q'$

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

記憶用

這四種狀態

這四種狀態



STUDENT

T Flip-Flop

■ T (toggle) flip-flop

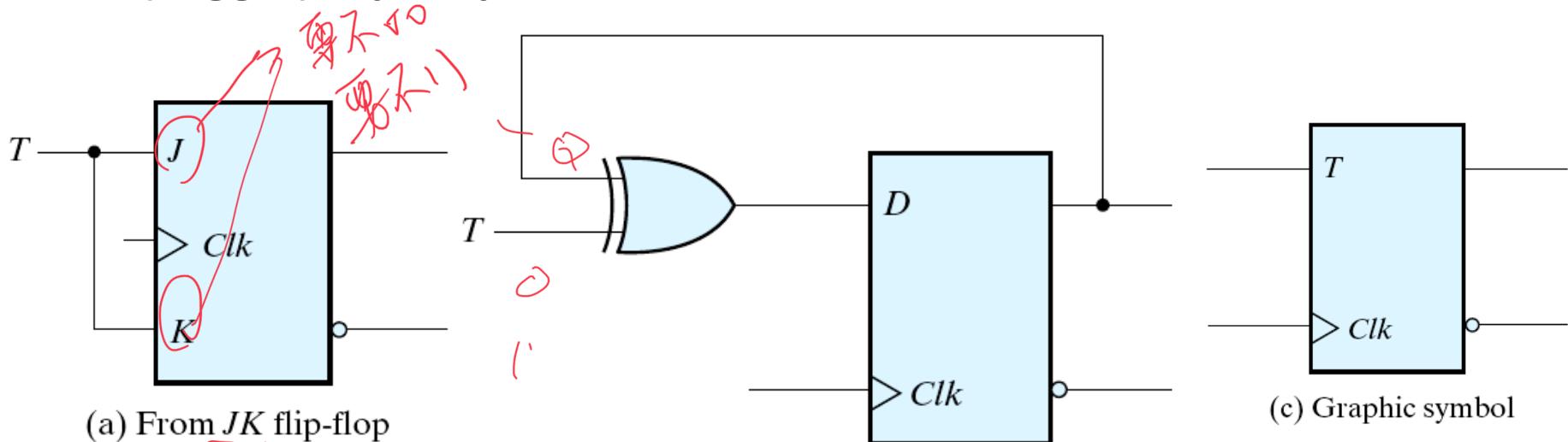


Fig. 13 T flip-flop

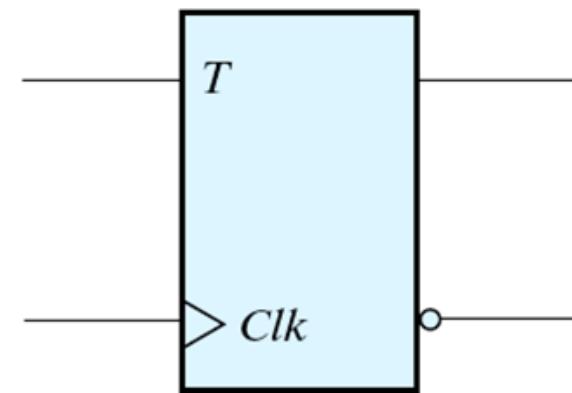
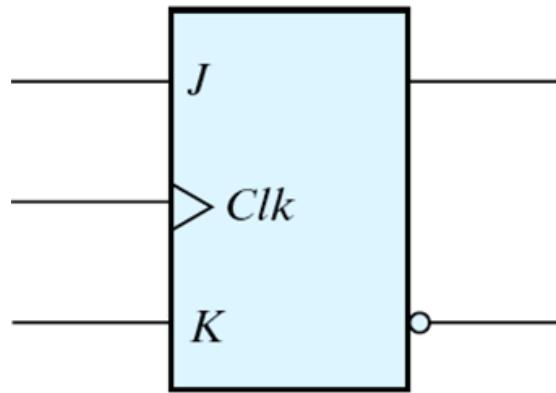
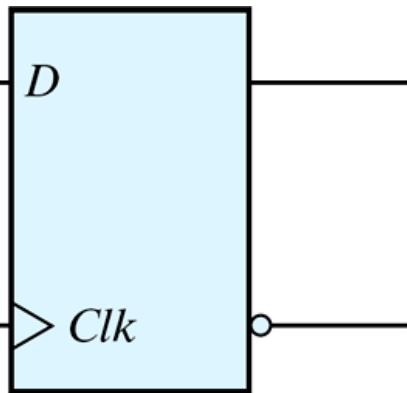
- ◆ $D = T \oplus Q = TQ' + T'Q$
 - » $T=0: D=Q$, no change
 - » $T=1: D=Q' \Rightarrow Q=Q'$

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

支教重慶

Characteristic Tables

Characteristic tables



D Flip-Flop

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

JK Flip-Flop

J	K	Q(t + 1)
0	0	$Q(t)$
0	1	0
1	0	1
1	1	$Q'(t)$

Annotations: A red bracket under the first two rows is labeled D . A red bracket under the last two rows is labeled T . Red arrows point from the $Q(t)$ and $Q'(t)$ labels to the corresponding output lines in the block diagrams above.

T Flip-Flop

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

Important

Characteristic Equations

Characteristic equations

D flip-flop

$$\Rightarrow Q(t+1) = D' \cdot 0 + D \cdot 1 = D$$

JK flip-flop

$$\Rightarrow Q(t+1) = J' \cdot K' \cdot Q(t) + J' \cdot K \cdot 0 + J \cdot K' \cdot 1 + J \cdot K \cdot Q'(t) = J \cdot Q' + K' \cdot Q$$

T flop-flop

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

$$\widehat{JQ} + \widehat{KQ}$$



D Flip-Flop

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

JK Flip-Flop

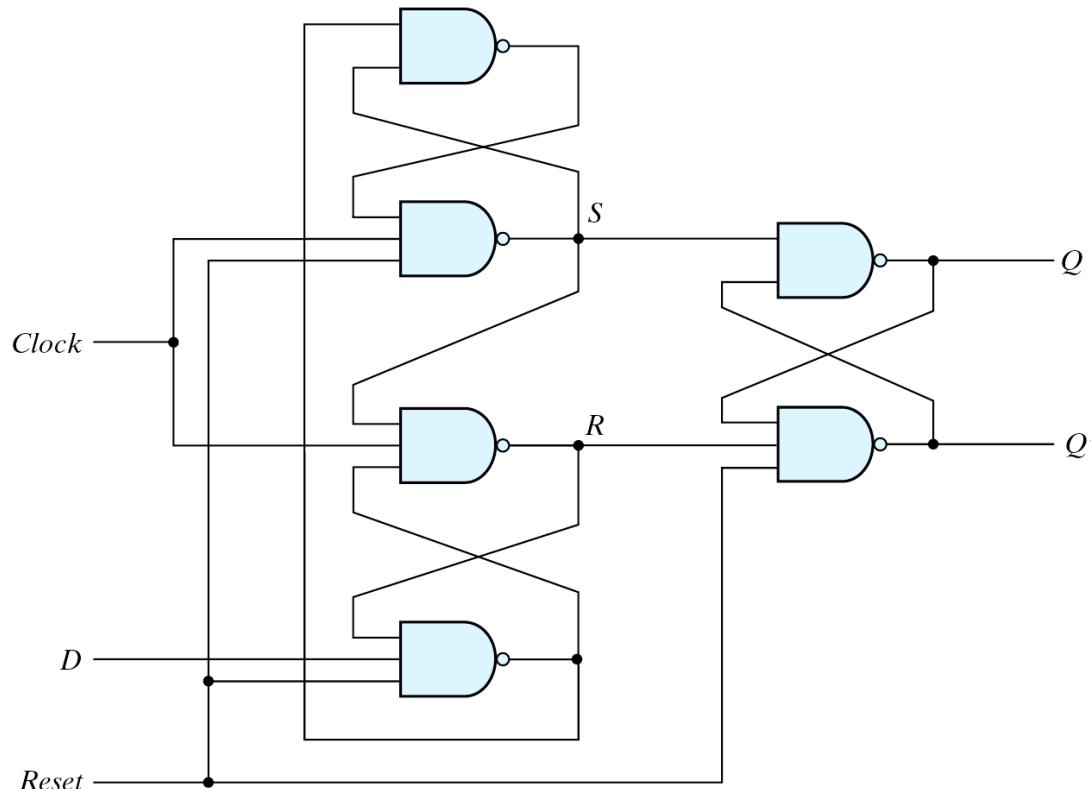
J	K	Q(t + 1)	
0	0	$Q(t)$	No change
0	1	0	Reset
1	0	1	Set
1	1	$Q'(t)$	Complement

T Flip-Flop

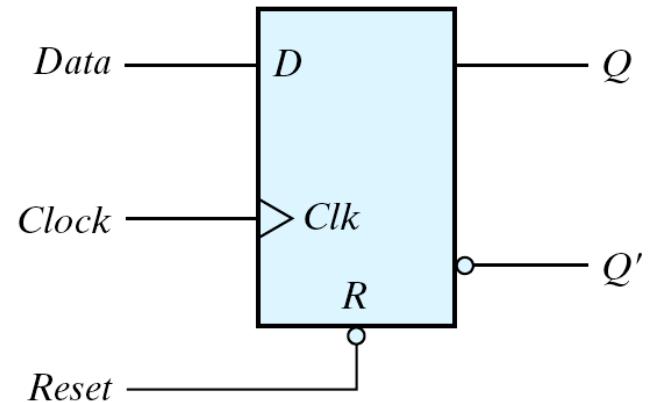
T	Q(t + 1)
0	$Q(t)$ No change
1	$Q'(t)$ Complement

Direct Inputs

■ Asynchronous set (preset) / asynchronous reset (clear)



(a) Circuit diagram



(b) Graphic symbol

R	Clk	D	Q	Q'
0	X	X	0	1
1	↑	0	0	1
1	↑	1	1	0

(c) Function table

Fig. 14 D flip-flop with asynchronous reset

Analysis of Clocked Sequential Circuits

□ Analysis for sequential circuit

- ◆ (inputs, current state) \Rightarrow (output, next state)
- ◆ State equation (also called transition equation) specifies the next state as a function of the present states and inputs
- ◆ State table (also called transition table) enumerates all present states and inputs
- ◆ State diagram – graphical form of a state table
- ◆ Flip-flop input equation – combinational function to flip-flop input
- ◆ Mealy and Moore machines

State Equations

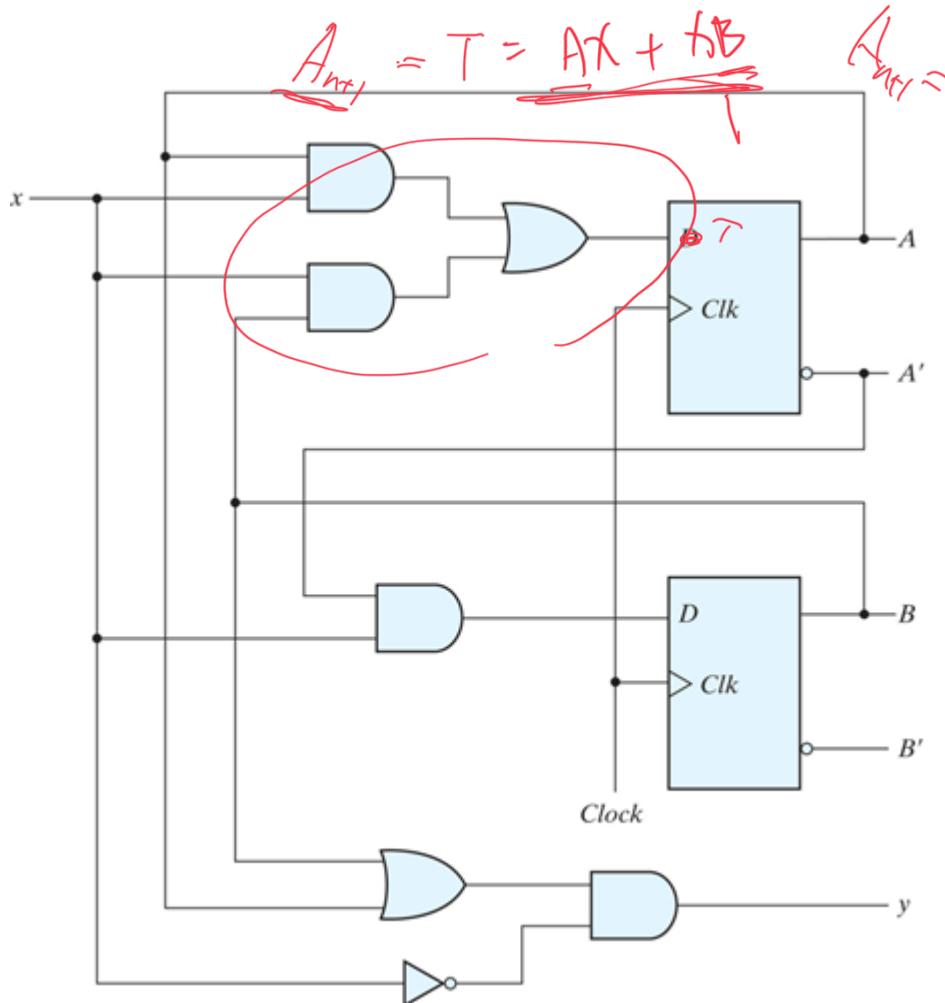


Fig. 15 Example of sequential circuit

State equations

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

A compact form

- ◆ $A(t+1) = Ax + Bx$
- ◆ $B(t+1) = A'x$

The output equation

- ◆ $y(t) = (A(t)+B(t))x'(t)$
- ◆ $y = (A+B)x'$

练习：把D改成T $T \oplus Q = Q$

$$T \oplus Q = Q$$

$$Ans = A(t+1) = (A(t) \cdot (Q(t) + B(t) \cdot X(t))) \oplus A(t)$$

State Table (1/2)

■ State table (transition table)

◆ = State equations

Table 5.2

State Table for the Circuit of Fig. 5.15

$$A(t+1) = Ax + Bx$$
$$B(t+1) = A'x$$
$$y = Ax' + Bx'$$

Present State		Input <i>x</i>	Next State		Output <i>y</i>
<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	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

$$Y = A \cdot B$$

State Table (2/2)

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

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

$$y = Ax' + Bx'$$

Table 5.3
Second Form of the State Table

Present State	Next State				Output	
	$x = 0$		$x = 1$		y	
A	B	A	B	A	B	y
0	0	0	0	0	1	0
0	1	0	0	1	1	0
1	0	0	0	1	0	1
1	1	0	0	1	0	0

Table 5.2
State Table for the Circuit of Fig. 5.15

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

State Diagram

State diagram

- ◆ A circle: a state
- ◆ A directed line connecting the circles: transition between the states
 - » Each directed line is labeled "inputs/outputs"

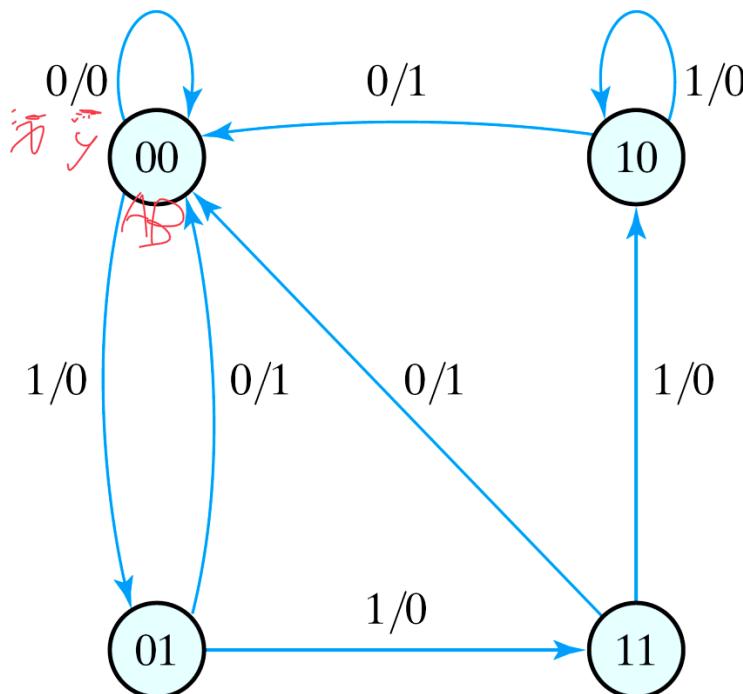


Table 5.3
Second Form of the State Table

Present State		Next State		Output	
A	B	x = 0	x = 1	x = 0	x = 1
		A	B	A	B
0	0	0	0	0	0
0	1	0	0	1	0
1	0	0	0	1	0
1	1	0	0	1	0

Fig. 16 State diagram of the circuit of Fig. 15

- ◆ Circuit diagram \Leftrightarrow state equation \Leftrightarrow state table \Leftrightarrow state diagram

Flip-Flop Input Equations

■ The part of circuit that generates the inputs to flip-flops

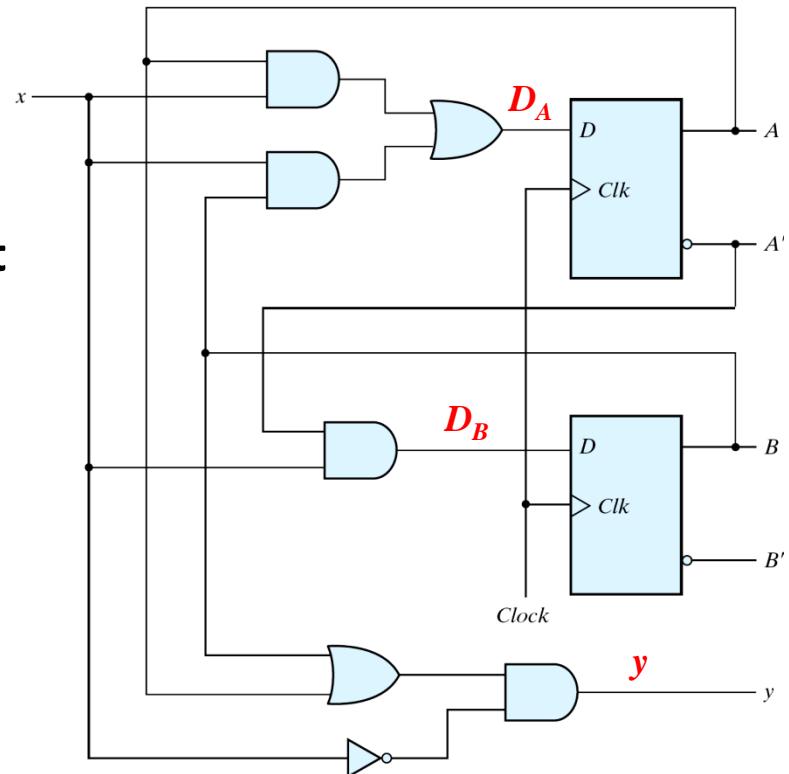
- ◆ Also called excitation functions
- ◆ $D_A = Ax + Bx$
- ◆ $D_B = A'x$

■ The output equations

- ◆ To fully describe the sequential circuit
- ◆ $y = Ax' + Bx'$

■ Note

- ◆ $Q(t+1) = D_Q$



Pitfall: flip-flop input equation \neq state equation! (except D flip-flop)

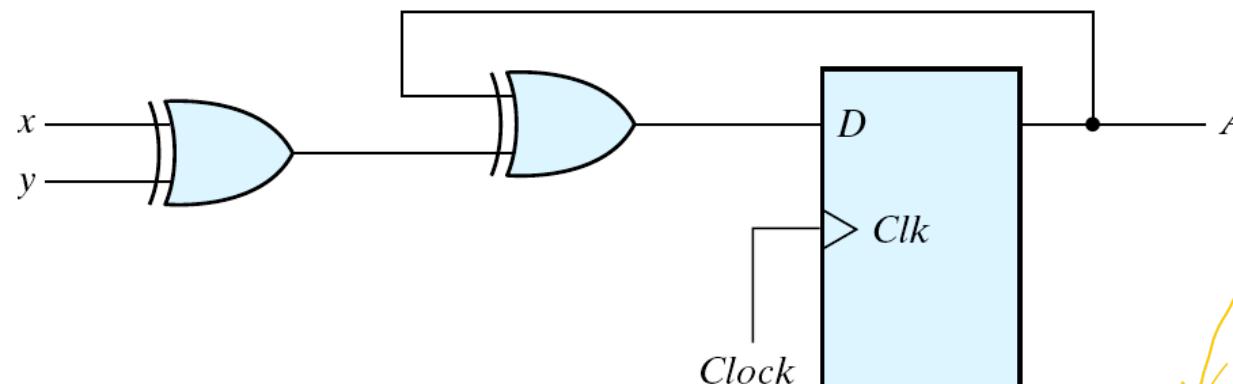
Analysis with D FFs

□ The input equation

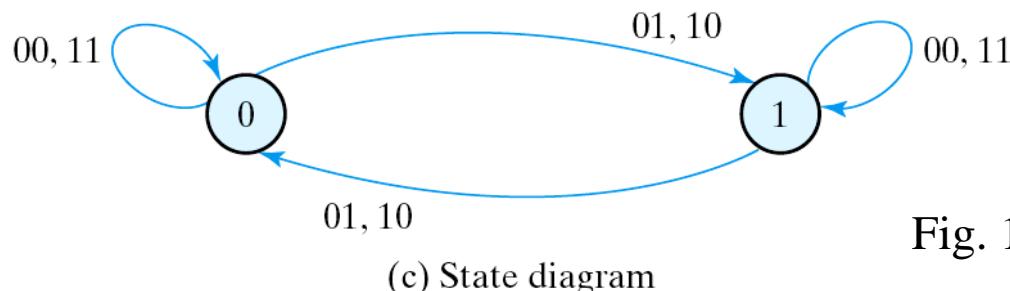
$$\diamond D_A = A \oplus x \oplus y$$

□ The state equation

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



(a) Circuit diagram



(c) State 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



Fig. 17 Sequential circuit with D flip-flop

Analysis with Other Flip-flops

□ The next-state can be derived by following procedures

1. Determine the flip-flop input equations in terms of the present state and input variables
2. List the binary values of each input equation
3. Use the corresponding flip-flop characteristic table to determine the next-state values in the state table

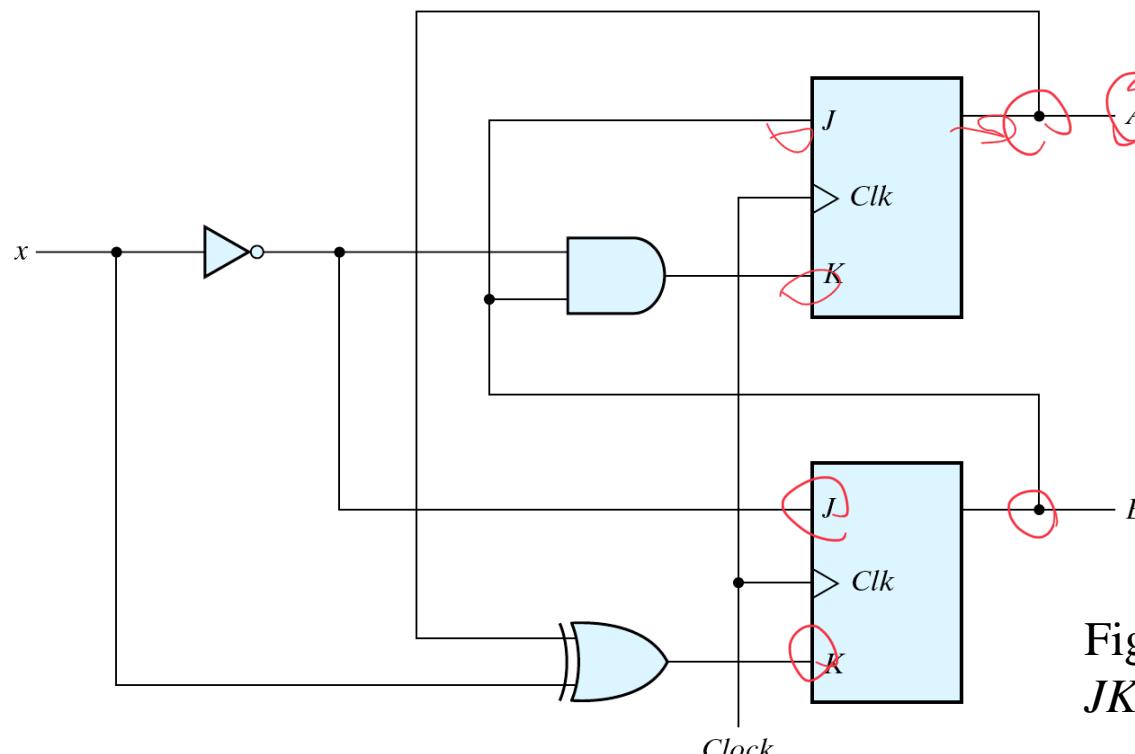
- OR -

1. Determine the flip-flop input equations in terms of the present state and input variables
 2. Substitute the input equations into the flip-flop characteristic equation to obtain the state equation
 3. Use the corresponding state equations to determine the next-state values in state table
- 逐字逐句 (逐字逐句) 2
特徵*

Analysis with JK flip-flops (1/3)

■ An sequential circuit with JK FFs

- ◆ Determine the flip-flop input function in terms of the present state and input variables
- ◆ Used the corresponding flip-flop characteristic table to determine the next state



$$\begin{aligned} J_A &= \bar{A}(t) + \bar{x}A(t) \\ &= A(t+1) \end{aligned}$$

$$\begin{aligned} J_A &= B, \quad K_A = Bx' \\ J_B &= x', \quad K_B = A'x + Ax' \end{aligned}$$

Fig. 18 Sequential circuit with JK flip-flop

Analysis with JK flip-flops (2/3)

◆ $J_A = B, K_A = Bx'$

◆ $J_B = x', K_B = A'x + Ax'$

◆ Derive the state table

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

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

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

$$\begin{aligned}
 A_{t+1} &= \overline{J_A}A + \overline{K_A}A \\
 B_{t+1} &= \overline{J_B}x + \overline{K_B}x' \\
 &= \overline{J_A}A + \overline{K_A}A \\
 &= B \cdot \overline{x} + B' \cdot x' \cdot A
 \end{aligned}$$

◆ Or, derive the state equations using characteristic equation

Analysis with JK flip-flops (3/3)

State transition diagram

State equation for A and B:

$$J_A = B, K_A = Bx'$$

$$J_B = x', K_B = A'x + Ax'$$

$$A(t+1) = J_A A' + K_A' A$$

$$B(t+1) = J_B B' + K_B' B$$

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

$$B(t+1) = x'B' + (A \oplus x)'B = B'x' + ABx + A'Bx'$$

Table 5.4

State Table for Sequential Circuit with JK Flip-Flops

Present State		Input x	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

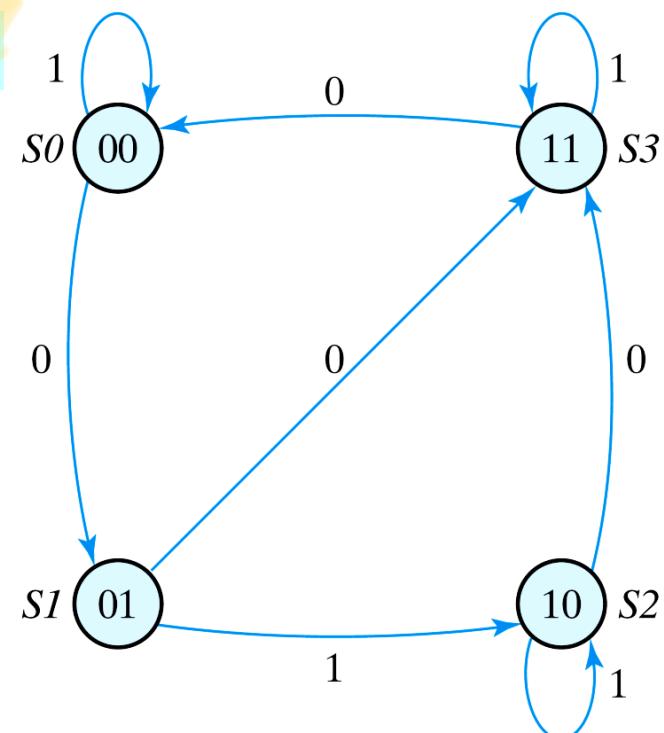


Fig. 19 State diagram of the circuit of Fig. 18

Analysis with T Flip-Flops (1/2)

The input and output functions

- ◆ $T_A = Bx$
- ◆ $T_B = x$
- ◆ $y = AB$

$A\backslash B$	00	01	10	11
0	00	01	10	11
1	11	10	01	00

The characteristic equation

- ◆ $Q(t+1) = T \oplus Q = T'Q + TQ'$

The state equations

- ◆ $A(t+1) = (Bx)'A + (Bx)A' = AB' + Ax' + A'Bx$
- ◆ $B(t+1) = x \oplus B$

$$A(t+1) = \underline{A(t)} \oplus T$$

$$B(t+1) = \underline{B(t)} \oplus T$$

(B(t) · T)
t で 等しい

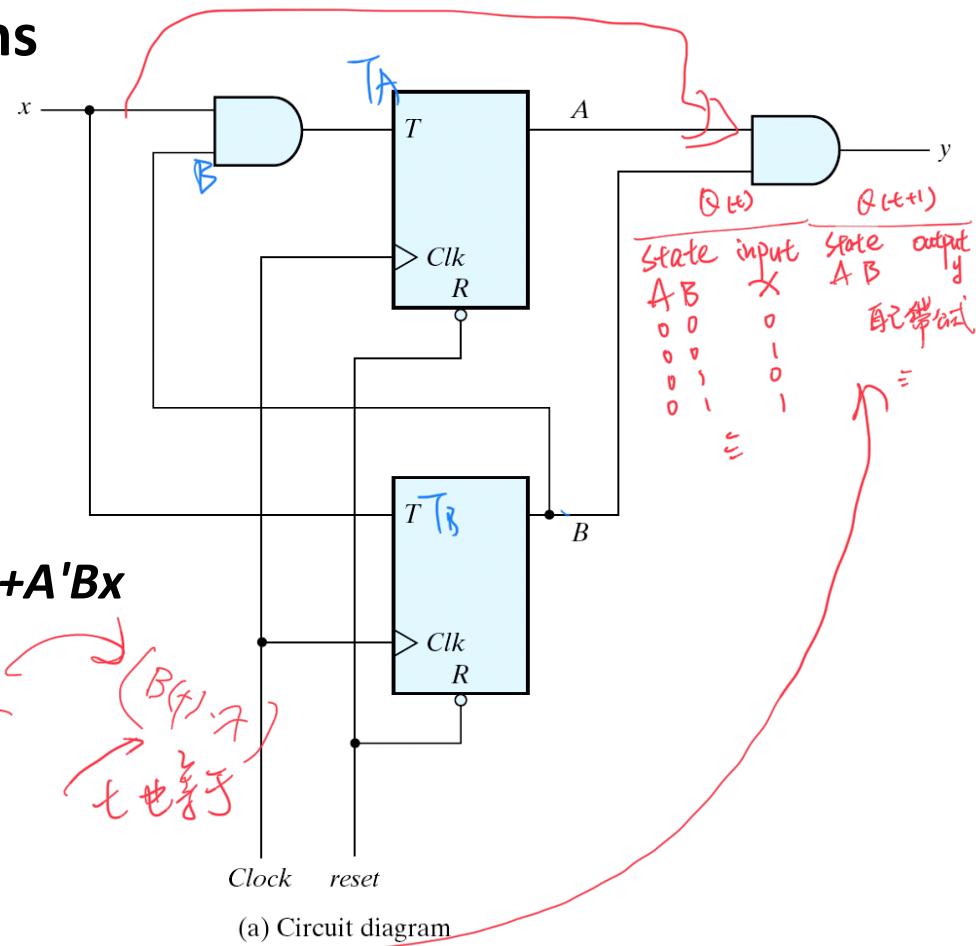


Fig. 20 Sequential circuit with T flip-flop

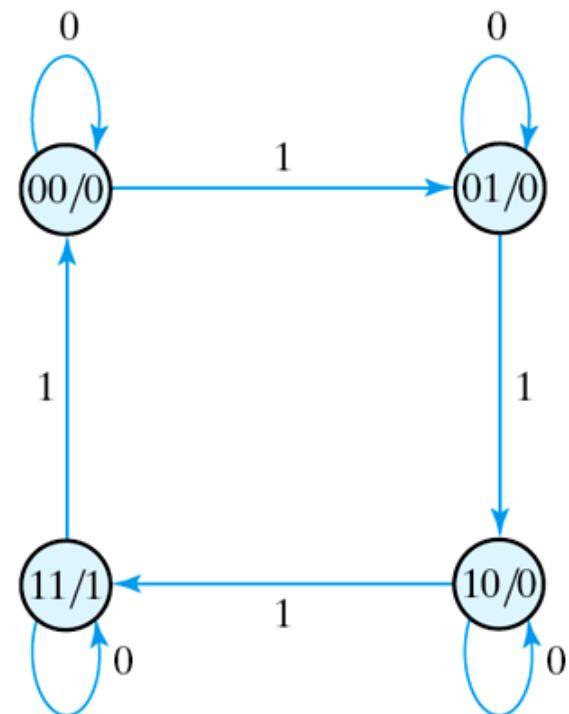
Analysis with T Flip-Flops (2/2)

- ◆ $A(t+1) = AB' + Ax' + A'Bx$
- ◆ $B(t+1) = x \oplus B$
- ◆ $y = AB$

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

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

Present State		Input x	Next State		Output y
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	0
1	1	0	1	1	1
1	1	1	0	0	1



Mealy and Moore Models (1/2)

- **The Mealy model:** the outputs are functions of both the present state and inputs (Fig. 15)
 - ◆ The outputs may change if the inputs change during the clock pulse period
 - » The outputs may have momentary false values unless the inputs are synchronized with the clocks
- **The Moore model:** the outputs are functions of the present state only (Fig. 20)
 - ◆ The outputs are synchronous with the clocks

Mealy and Moore Models (2/2)

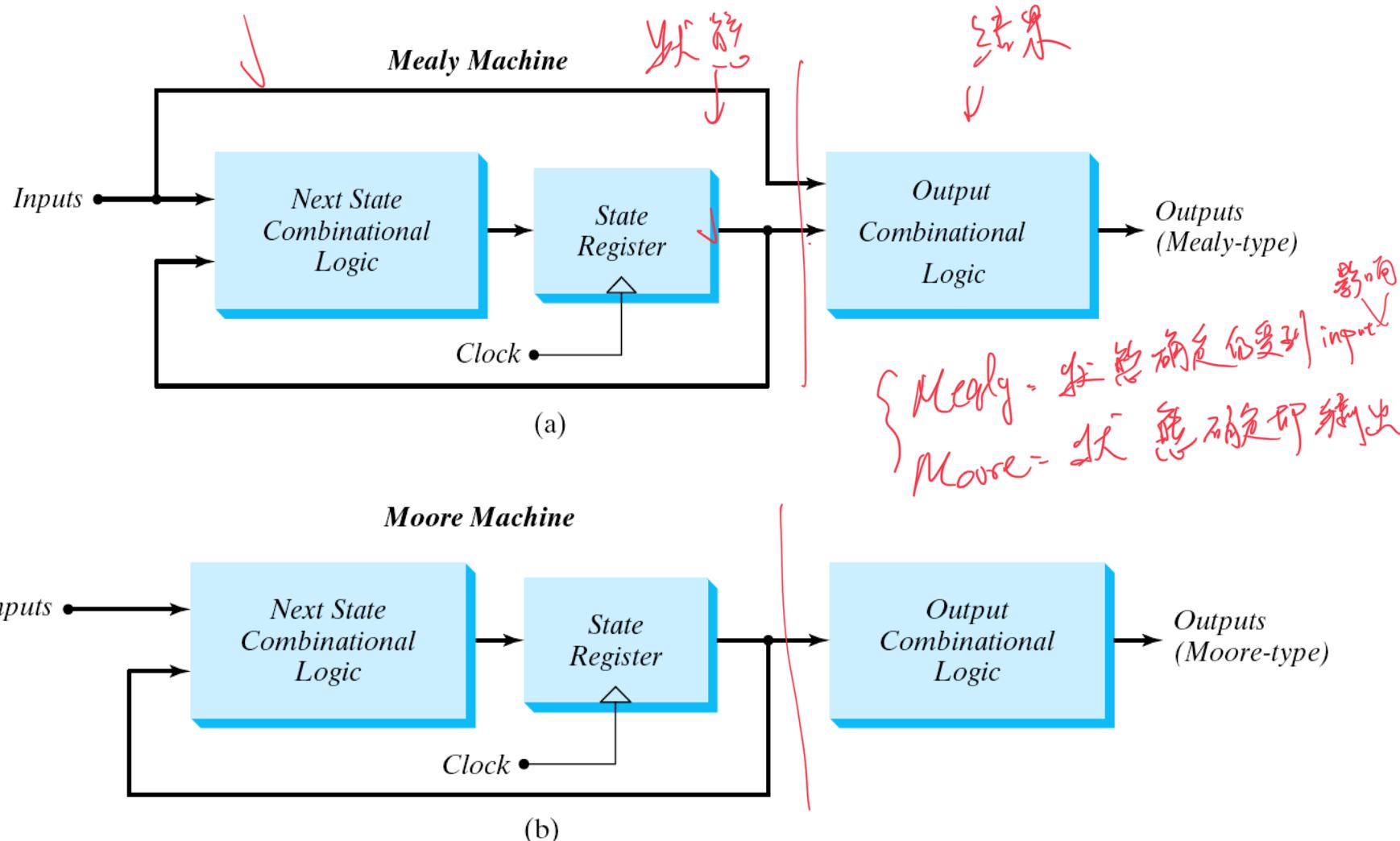


Fig. 21 Block diagrams of Mealy and Moore state machines

State Reduction and Assignment

■ State Reduction

- ◆ A reduction in the number of states may result in a reduction in the number of flip-flops and gates
- ◆ Only the input-output sequences are important
- ◆ Two circuits are equivalent
 - » Have identical outputs for all input sequences
 - » The number of states is not important

State Reduction (1/5)

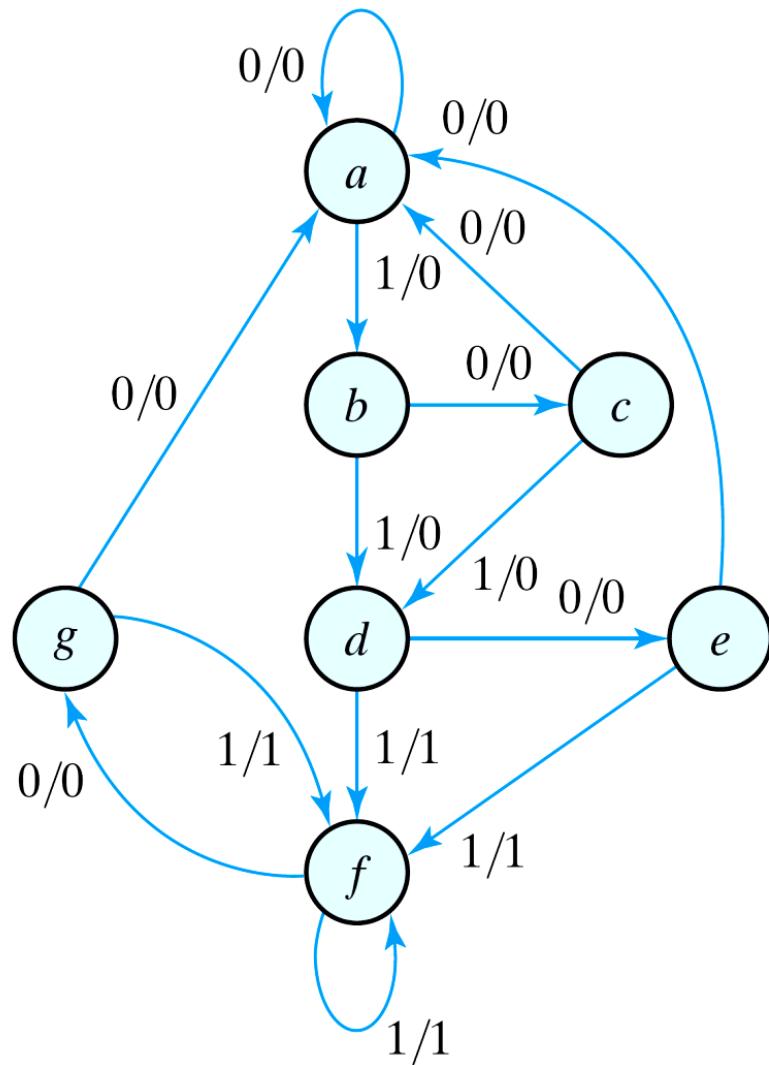


Fig. 25 State diagram

State:	a	a	b	c	d	e	f	f	g	f	g	a
Input:	0	1	0	1	0	1	1	0	1	0	0	0
Output:	0	0	0	0	0	1	1	0	1	0	0	0

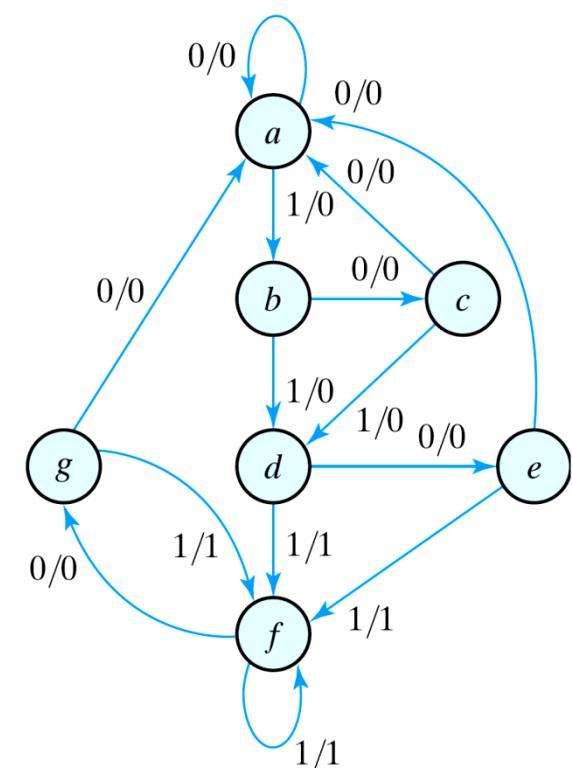
State Reduction (2/5)

■ Equivalent states

- ◆ Two states are said to be equivalent
 - » For each member of the set of inputs, they give exactly the same output and send the circuit to the same state or to an equivalent state
 - » One of them can be removed

Table 5.6
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	g	f	0	1
g	a	f	0	1



State Reduction (3/5)

Reducing the state table

- ◆ $e = g$ (remove g);
- ◆ $d = f$ (remove f);

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

State Reduction (4/5)

- ◆ The reduced finite state machine

Table 5.8
Reduced 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>d</i>	0	1
<i>e</i>	<i>a</i>	<i>d</i>	0	1

State: a a b c d e d d e d e a
Input: 0 1 0 1 0 1 1 0 1 0 0
Output: 0 0 0 0 0 1 1 0 1 0 0

State Reduction (5/5)

- ◆ The checking of each pair of states for possible equivalence can be done systematically
- ◆ The unused states are treated as don't-care condition \Rightarrow fewer combinational gates

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

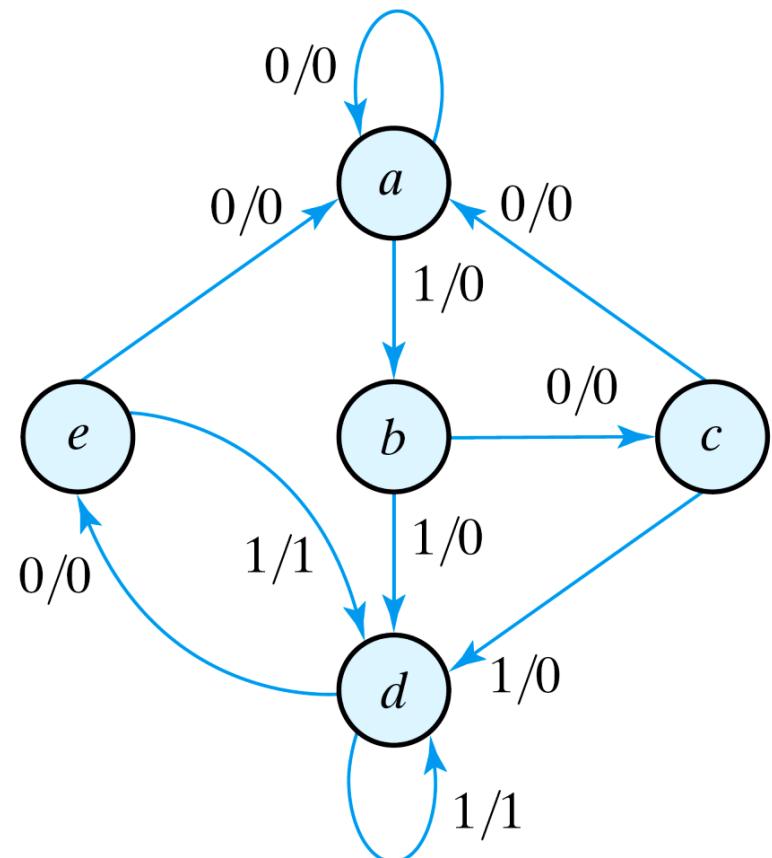


Fig. 26 Reduced State diagram

State Assignment (1/2)

■ State assignment

- ◆ To minimize the cost of the combinational circuits
- ◆ Various state assignments
 - » m states require at least n -bits, where $2^n > m$

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

State Assignment (2/2)

- ◆ Any binary number assignment is satisfactory as long as each state is assigned a unique number
- ◆ There is no simple state-encoding procedure which guarantees a minimum-cost or minimum-delay combinational circuits

Table 5.10

Reduced State Table with Binary Assignment 1

Present State	Next State		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
000	000	001	0	0
001	010	011	0	0
010	000	011	0	0
011	100	011	0	1
100	000	011	0	1

Design Procedure

■ Design Procedure for sequential circuit

- ◆ The word description of the circuit behavior to get a state diagram
- ◆ State reduction if necessary
- ◆ Assign binary values to the states
- ◆ Obtain the binary-coded state table
- ◆ Choose the type of flip-flops
- ◆ Derive the simplified flip-flop input equations and output equations
- ◆ Draw the logic diagram

Synthesis Using D flip-flops (1/4)

- An example state diagram and state table: design a circuit to detect a sequence of three or more consecutive 1's

連續的

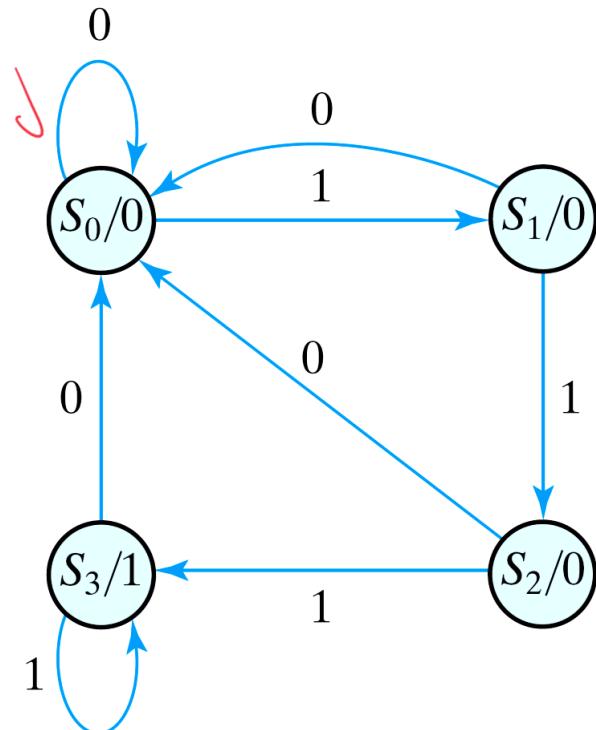


Fig. 27 State diagram for sequence detector

Table 5.11
State Table for Sequence Detector

Present State		Input <i>x</i>	Next State		Output <i>y</i>
<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

Synthesis Using D flip-flops (2/4)

Table 5.11

State Table for Sequence Detector

FF input

Present State		Input <i>x</i>	Next State		Output <i>y</i>
<i>A</i>	<i>B</i>		<i>A</i>	<i>B</i>	
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	2	0	0
0	1	1	3	1	0
1	0	0	4	0	0
1	0	1	5	1	0
1	1	0	6	0	1
1	1	1	7	1	1

- The flip-flop input equations

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

- The output equation

- ◆ $y(A, B, x) = \Sigma(6, 7)$

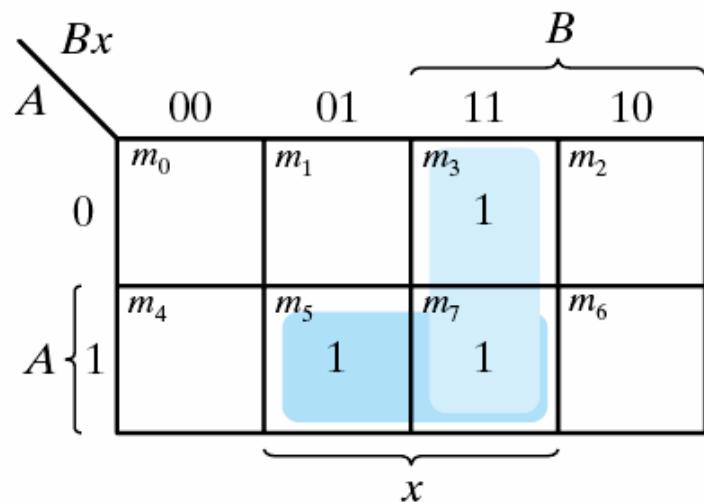
- Logic minimization using the K-map

- ◆ $D_A = Ax + Bx$

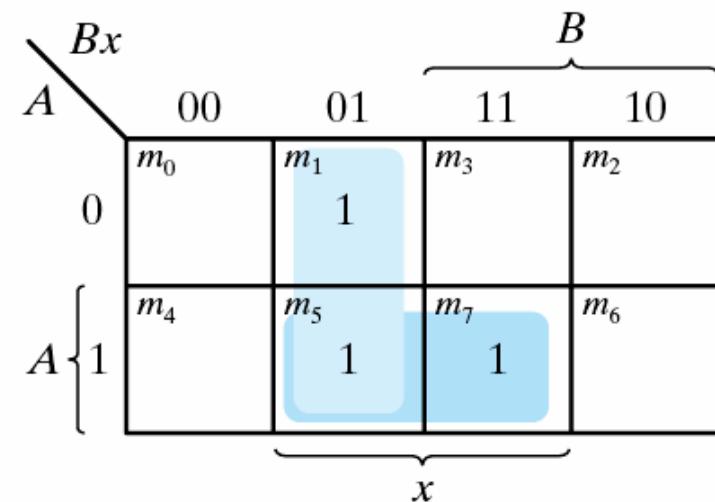
- ◆ $D_B = Ax + B'x$

- ◆ $y = AB$

Synthesis Using D flip-flops (3/4)



$$D_A = Ax + Bx$$



$$D_B = Ax + B'x$$

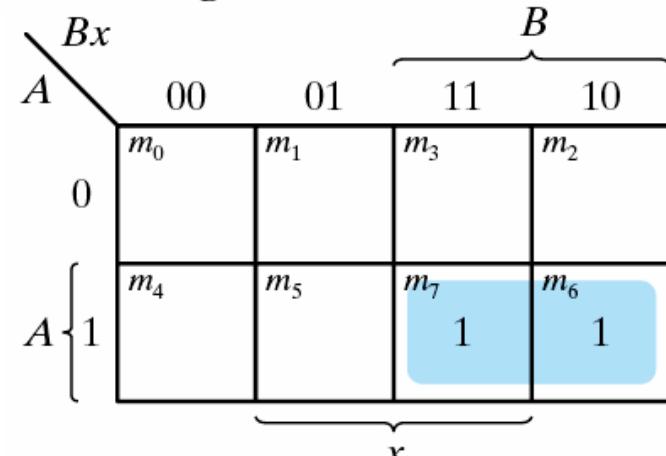


Fig. 28 Maps for sequence detector

Synthesis Using D flip-flops (4/4)

The logic diagram of sequence detector

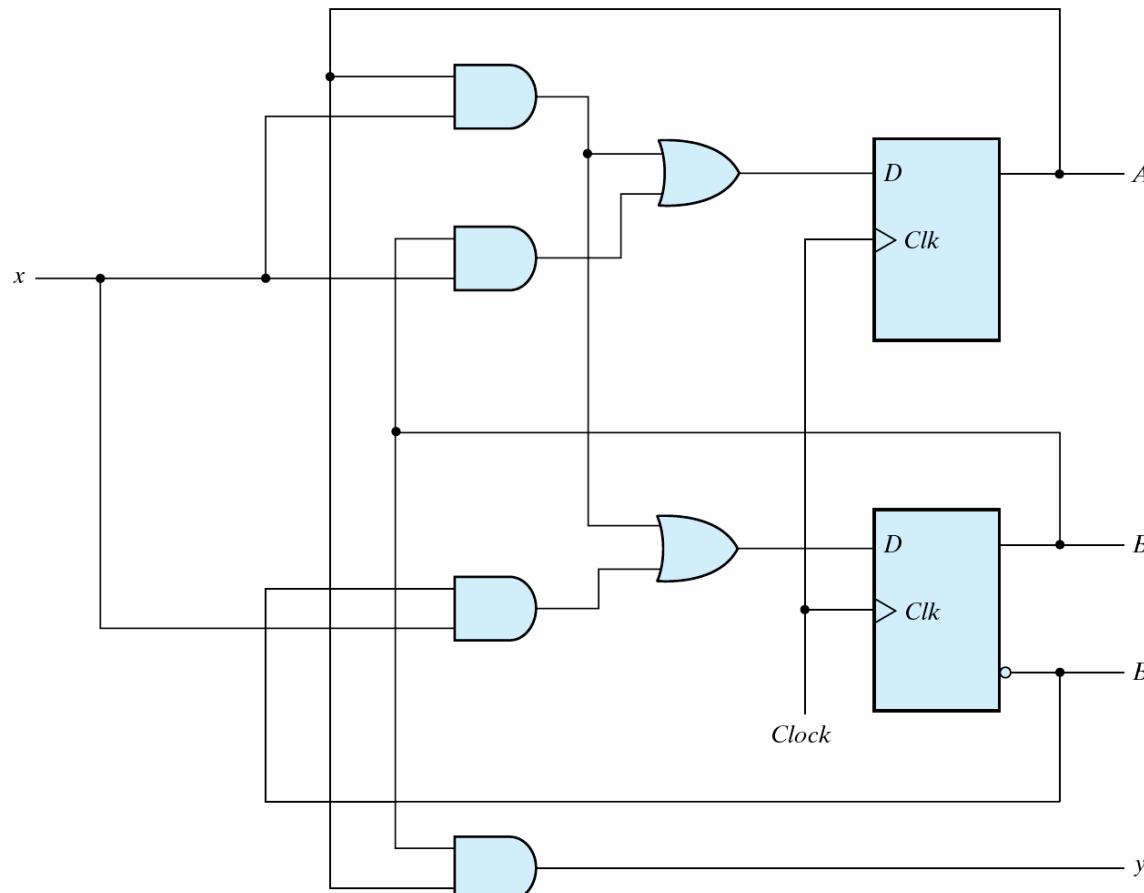


Fig. 29 Logic diagram of sequence detector

Excitation Tables

A state diagram \Rightarrow flip-flop input functions

- ◆ Straightforward for D flip-flops
- ◆ We need excitation tables for JK and T flip-flops

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	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

$Q(t)$	$Q(t + 1)$	T
0	0	0
0	1	1
1	0	1
1	1	0

(b) T



Synthesis Using JK FFs (1/3)

- The same procedure with D FFs, but an extra excitation table must be used
- The state table and JK flip-flop inputs

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 x	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

Synthesis Using JK FFs (2/3)

- ◆ $J_A = Bx'$; $K_A = Bx$
- ◆ $J_B = x$; $K_B = (A \oplus x)'$

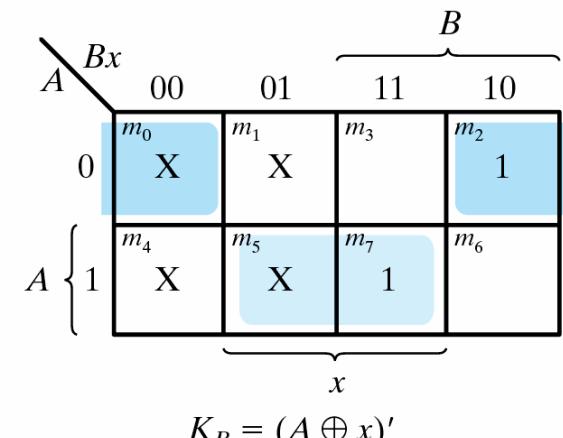
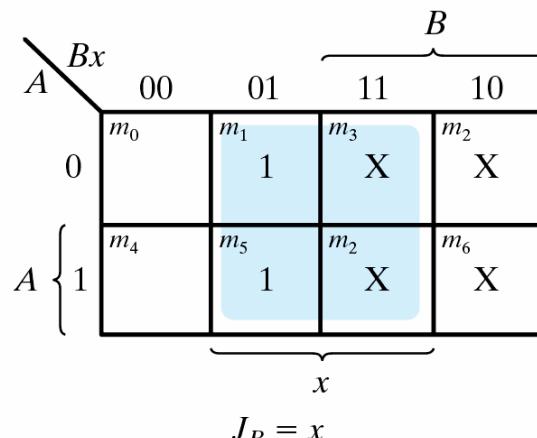
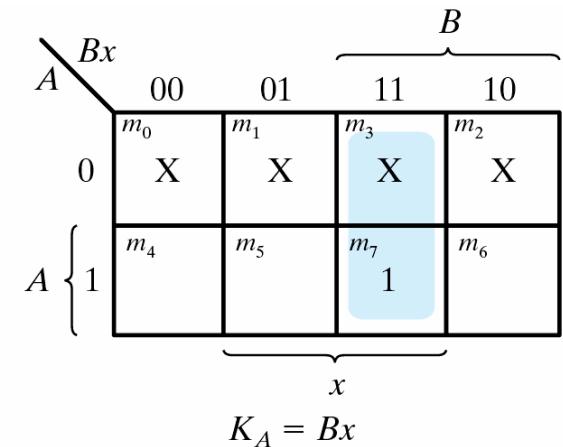
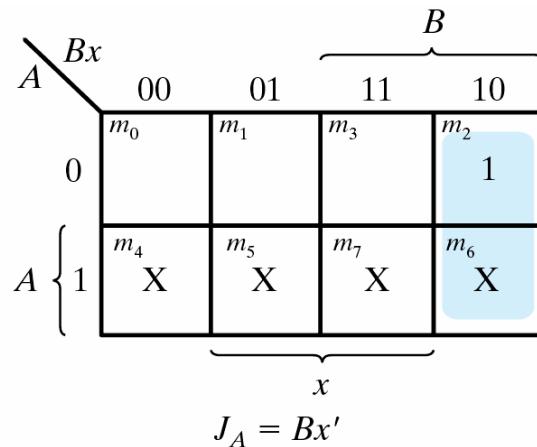


Fig. 30 Maps for J and K
input equations

Synthesis Using JK FFs (3/3)

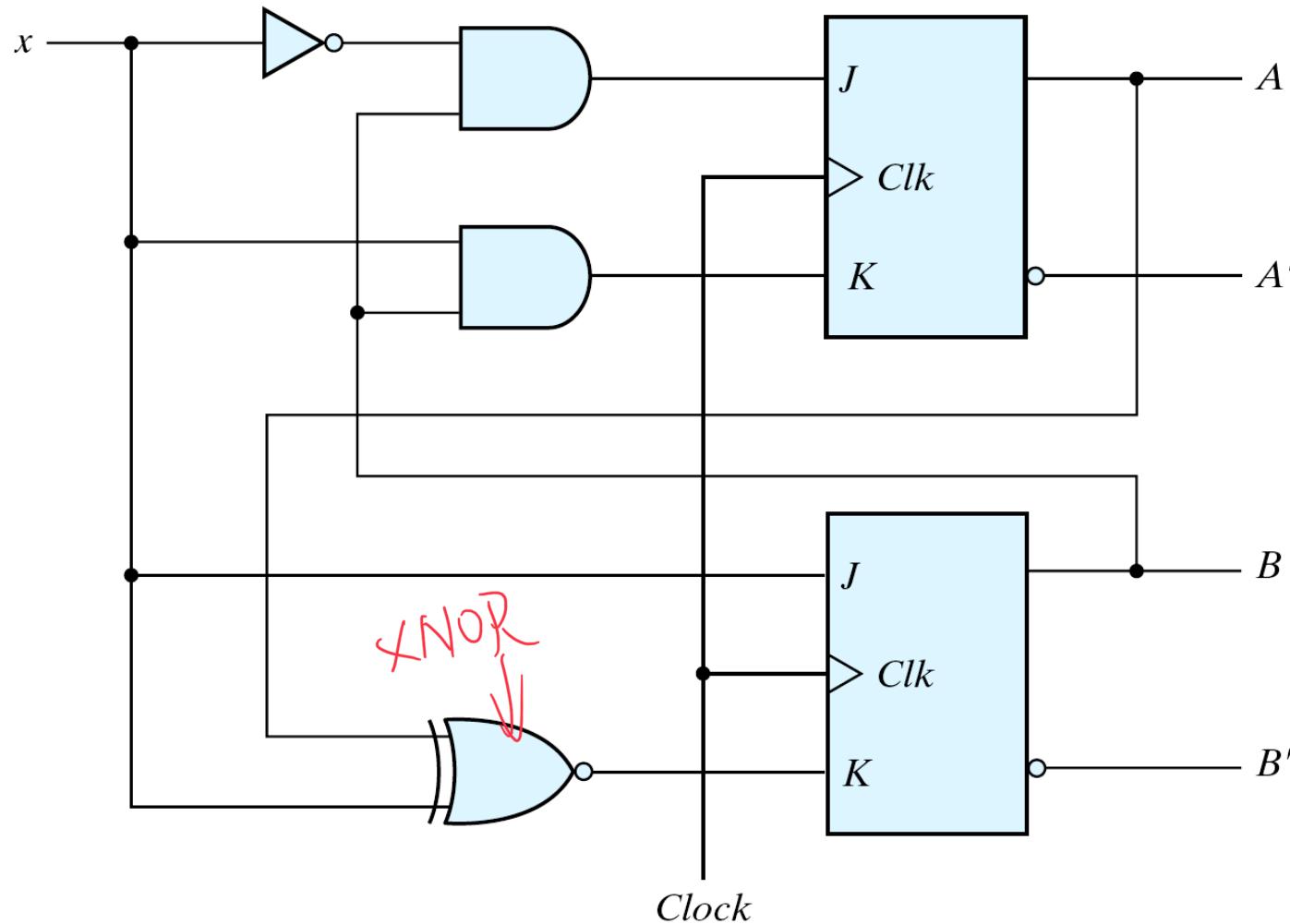


Fig. 31 Logic diagram for sequential circuit with JK flip-flops

Synthesis Using T flip-flops (1/4)

■ An n -bit binary counter

- ◆ The state diagram
- ◆ No inputs (except for the clock input)

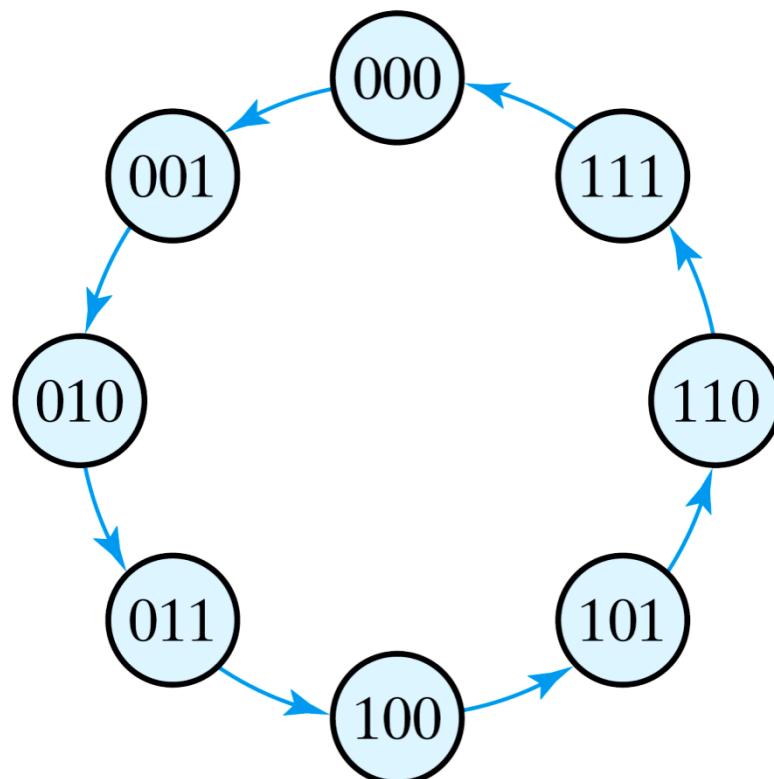


Fig. 32 State diagram of three-bit binary counter

Synthesis Using T flip-flops (2/4)

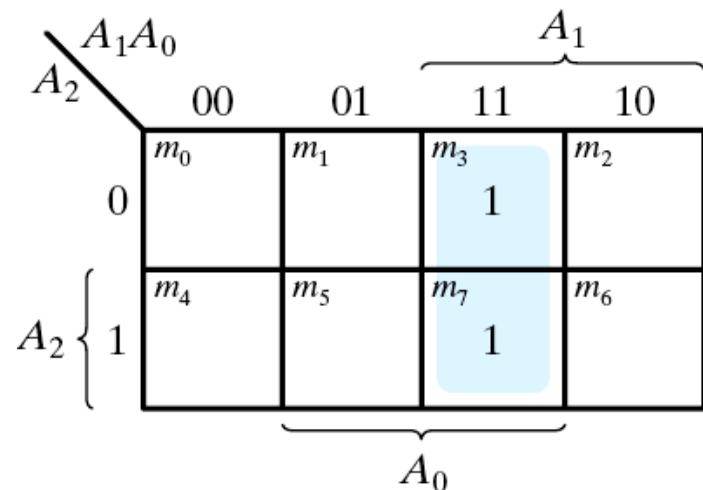
□ The state table and the flip-flop inputs

$Q(t)$	$Q(t + 1)$	T
0	0	0
0	1	1
1	0	1
1	1	0

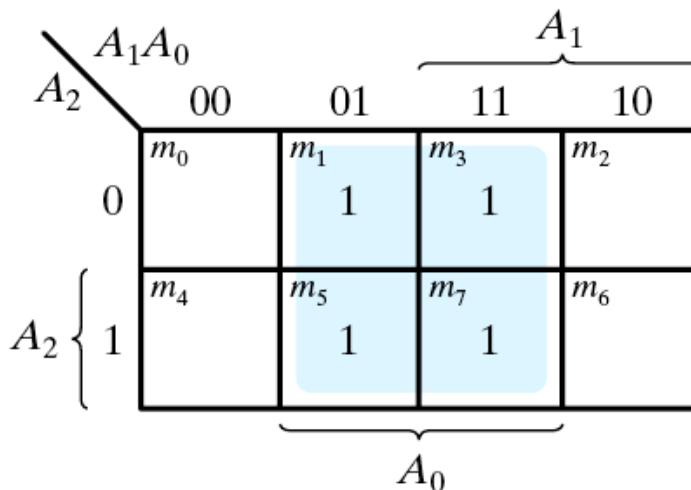
Table 5.14
State Table for Three-Bit Counter

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	0	1
1	1	1	0	0	0	1	1	1

Synthesis Using T flip-flops (3/4)



$$T_{A2} = A_1A_0$$



$$T_{A1} = A_0$$

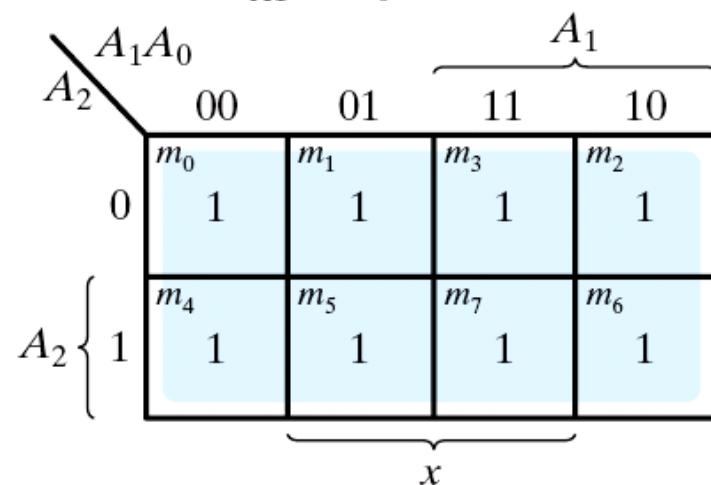


Fig. 33 Maps of three-bit binary counter

$$T_{A0} = 1$$

Synthesis Using T flip-flops (4/4)

Logic simplification using the K-map

- ◆ $T_{A2} = A_1 A_0$
- ◆ $T_{A1} = A_0$
- ◆ $T_{A0} = 1$

The logic diagram

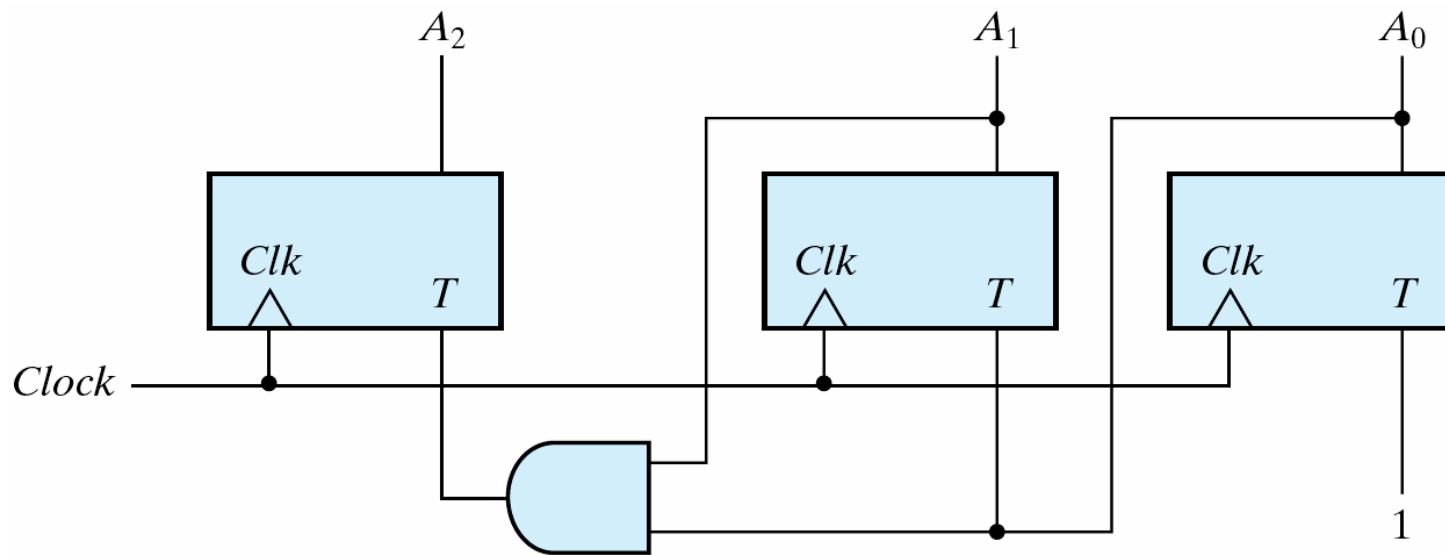


Fig. 34 Logic diagram of three-bit binary counter