CHAPTER-5

Synchronous Sequential Logic

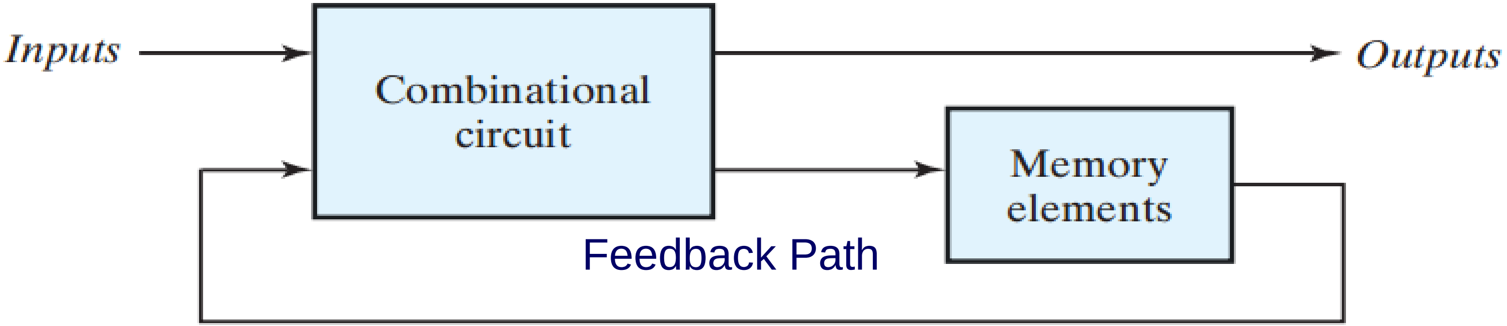
Digital Design (with an introduction to the Verilog HDL) 6th Edition,
M. Morris Mano, Michael D. Ciletti

Outline

- Sequential Circuit
- Storage Elements: Latches
- Storage Elements: Flip-flops
- Analysis of Clocked Sequential Circuits
- HDL models (Skipped)
- State Reduction and Assignment
- Design Procedure

Sequential Circuits

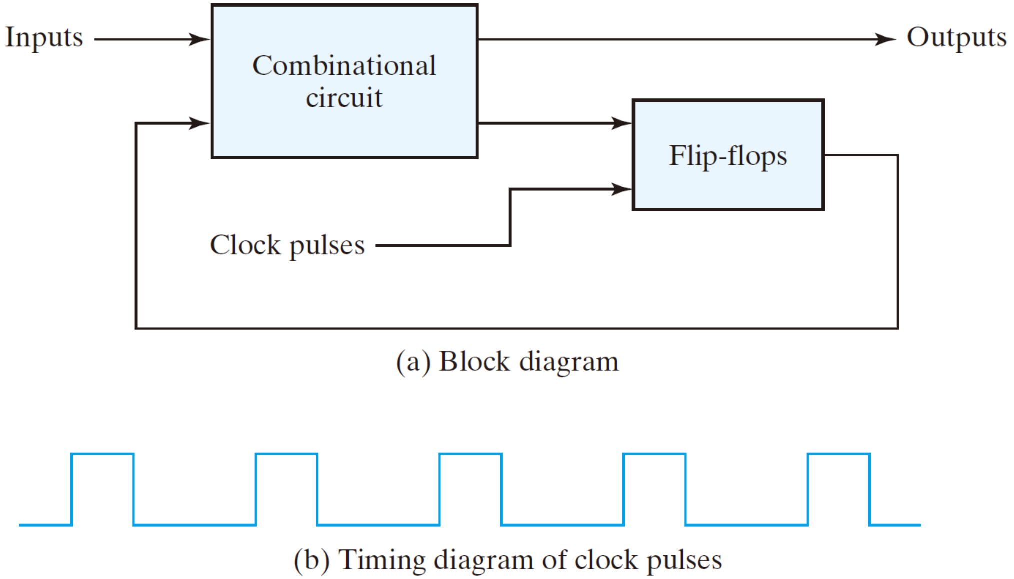
- Sequential circuits employ storage elements or feedback path besides logic gates. The outputs depend on both memory values (in the past) and present values of inputs.



- Inputs, present state
- Outputs, next state
- A sequential circuit is specified by a time sequence of inputs, outputs, and internal states.

Synchronous and Asynchronous

- A synchronous sequential circuit: behavior defined at determined discrete instants of time.
- A asynchronous sequential circuit: at any instant of time.
- Synchronization is achieved by a timing device called a clock generator, commonly denoted by “clock” or “clk”.



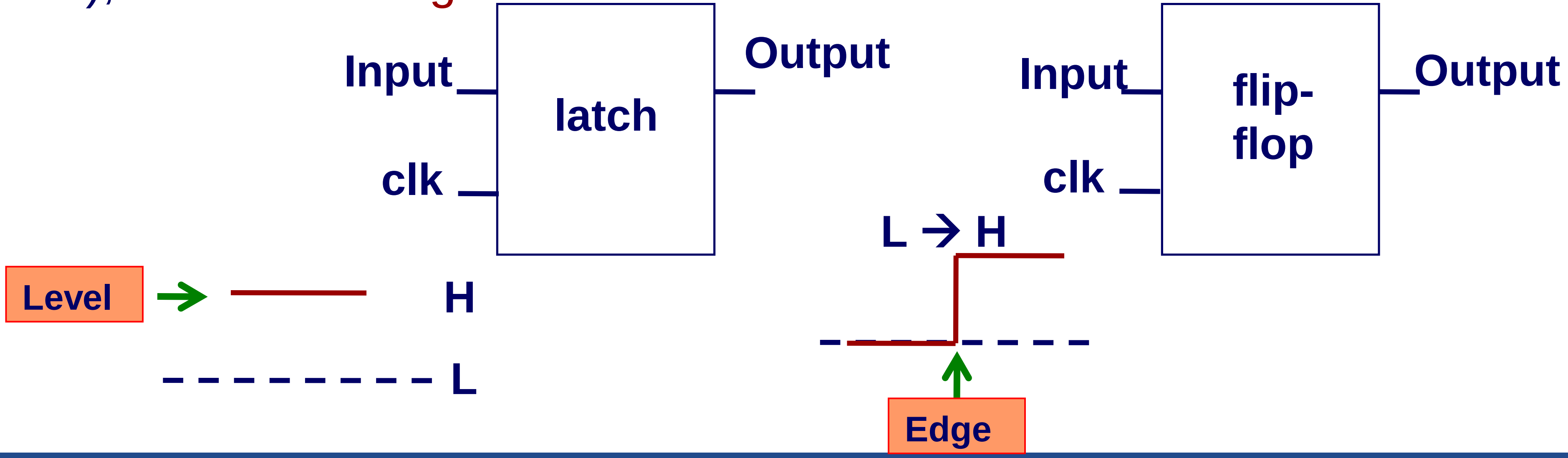
(a) Block diagram

(b) Timing diagram of clock pulses

FIGURE 5.2
Synchronous clocked sequential circuit

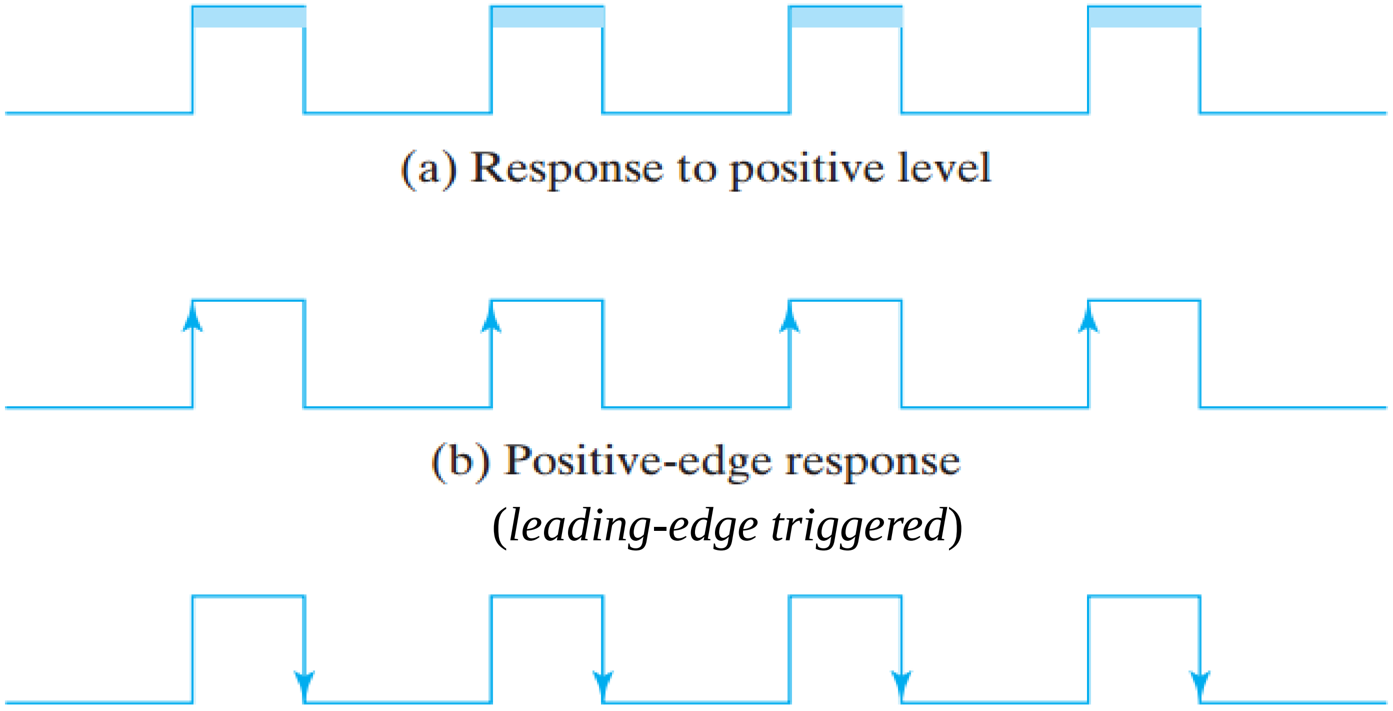
Storage Elements

- An electronic device that can be used to store one bit of information.
- Two main types of storage elements
  - latch
  - flip-flop
- Latch: the storage element is controlled by signal levels (usually H or L), also called level sensitive devices.
- Flip-flop: the storage element is controlled by a clock transition (i.e. L→H or H→L), also called edge-sensitive devices.



Storage Elements

- **Trigger**
  - The state of a latch or flip-flop is switched by a change in the control input.
- Level triggered → latches
- Edge triggered → flip flops



(a) Response to positive level

Level triggered

(b) Positive-edge response

(*leading-edge triggered*)

Edge triggered

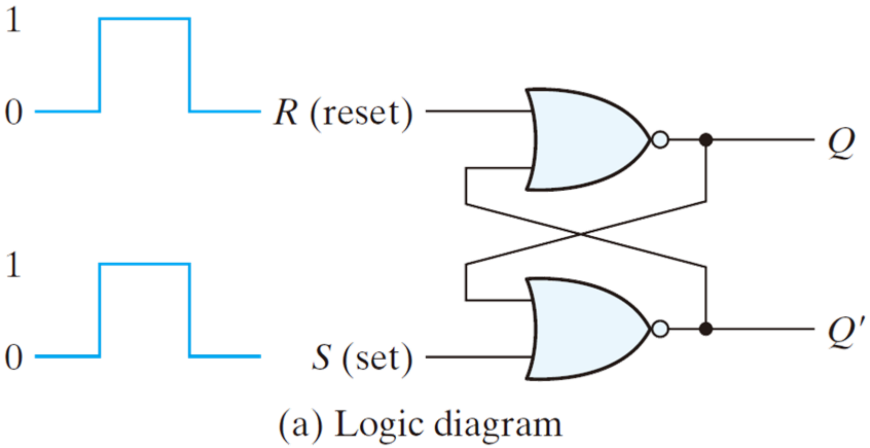
(c) Negative-edge response

(*trailing-edge triggered*)

Edge triggered

SR Latches with NOR Gates

- The SR latch is a circuit with two cross-coupled NOR gates or two cross-coupled NAND gates, and two inputs labeled *S* for set and *R* for reset.
- *Q* and *Q*’ are normally the complement of each other. *Q* = 1 and *Q*’ = 0, the latch is said to be in the set state. *Q* = 0 and *Q*’ = 1, it is in the reset state.
- The SR latches constructed with two cross-coupled NOR gates are active-high.



(a) Logic diagram

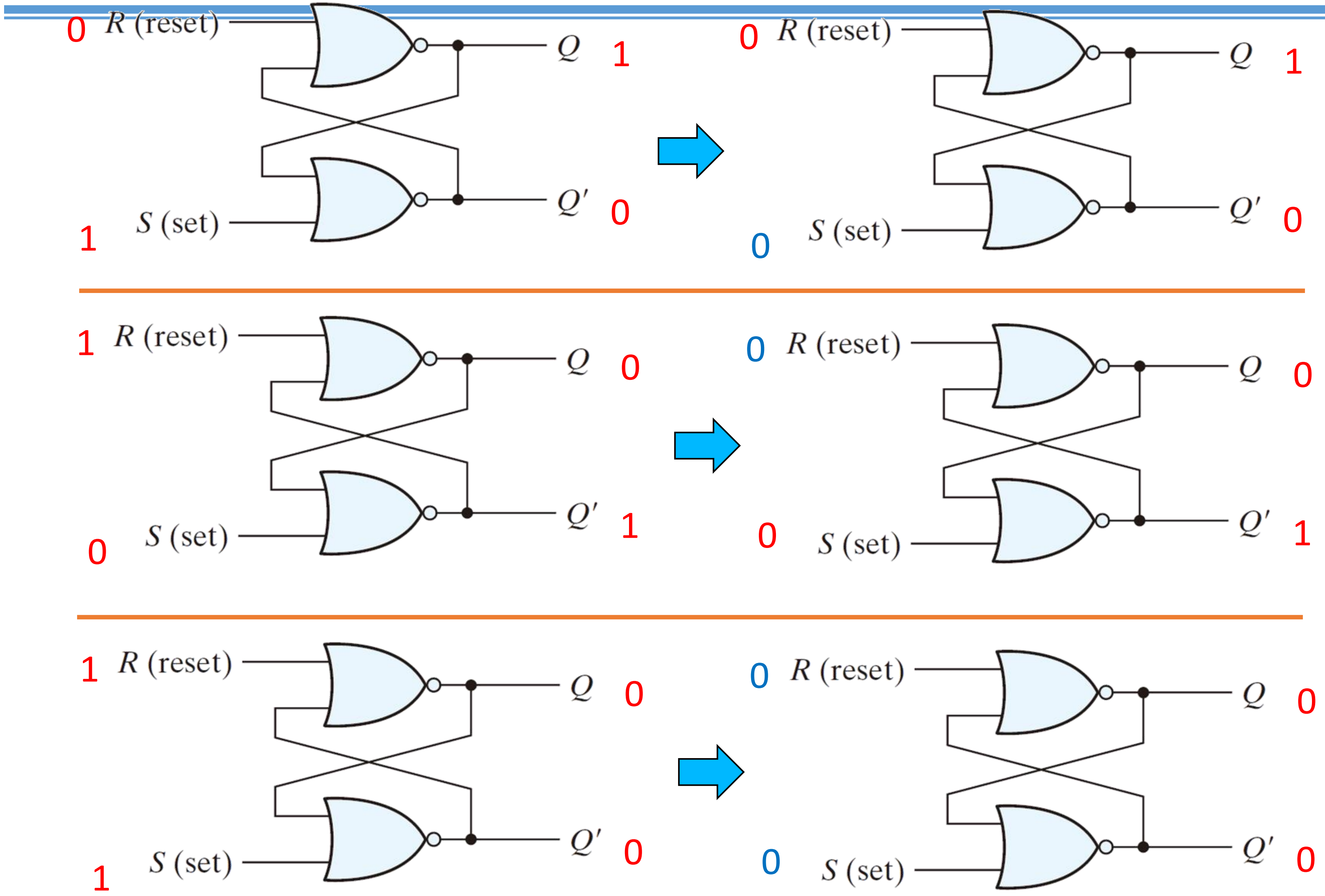
S	R	Q	Q'	
1	0	1	0	
0	0	1	0	(after $S = 1, R = 0$)
0	1	0	1	
0	0	0	1	(after $S = 0, R = 1$)
1	1	0	0	(forbidden)

(b) Function table

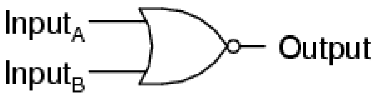
FIGURE 5.3
SR latch with NOR gates

- **S=1, R=0 → set state (Q will become 1)**
- **S=0, R=1 → reset state (Q will become 0)**
- **When S=0 and R=0 → keep the current value**

SR Latches with NOR Gates



NOR gate



A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0

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)

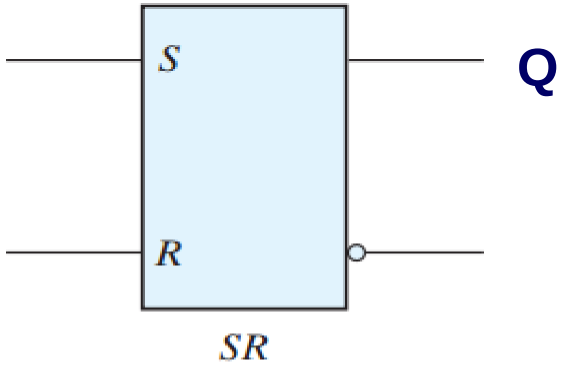
(b) Function table

- S=1, R=0 → set state (Q will become to 1)
- S=0, R=1 → reset state (Q will become to 0)
- When S=0 and R=0 → keep the current value

Not mutually complementary

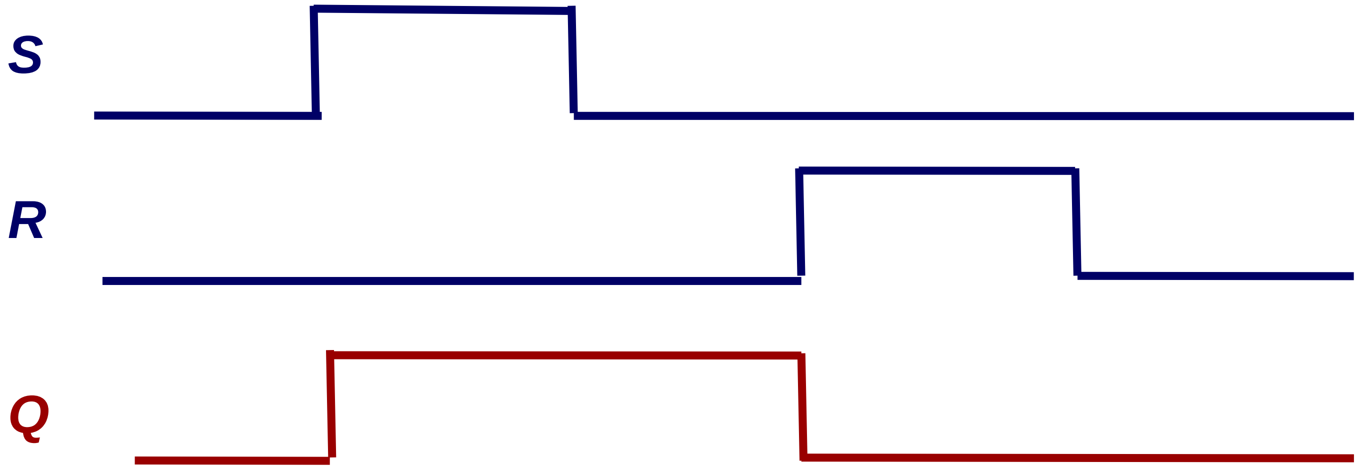
Forbidden!!!

SR Latch Timing Waveform



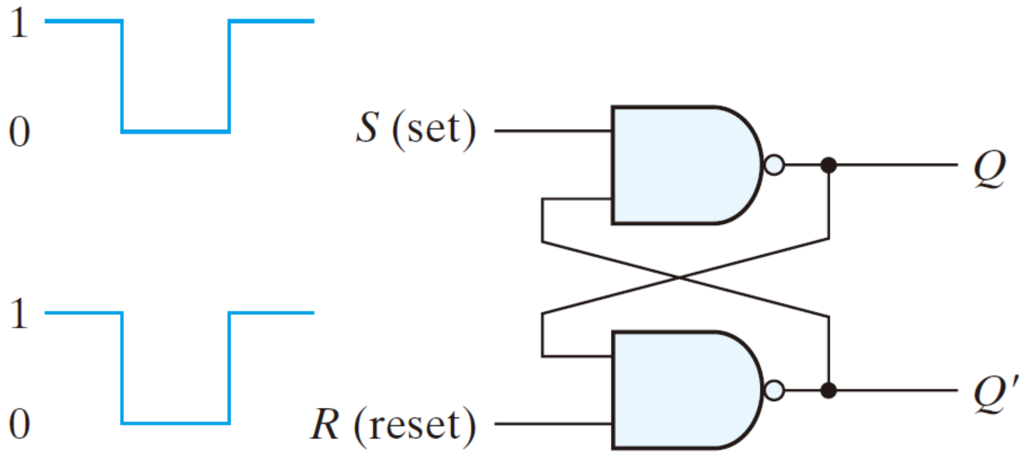
Assume initial value $Q_{init} = 0$

S	*R*	*Q*
1	0	1
0	1	0
0	0	Hold



SR Latches with NAND Gates

- The SR latches constructed with two cross-coupled NAND gates are active-low
  - S=1, R=O → reset state (Q will become to O)
  - S=O, R=1 → set state (Q will become to 1)
  - S=1, R=1 → unchanged



(a) Logic diagram

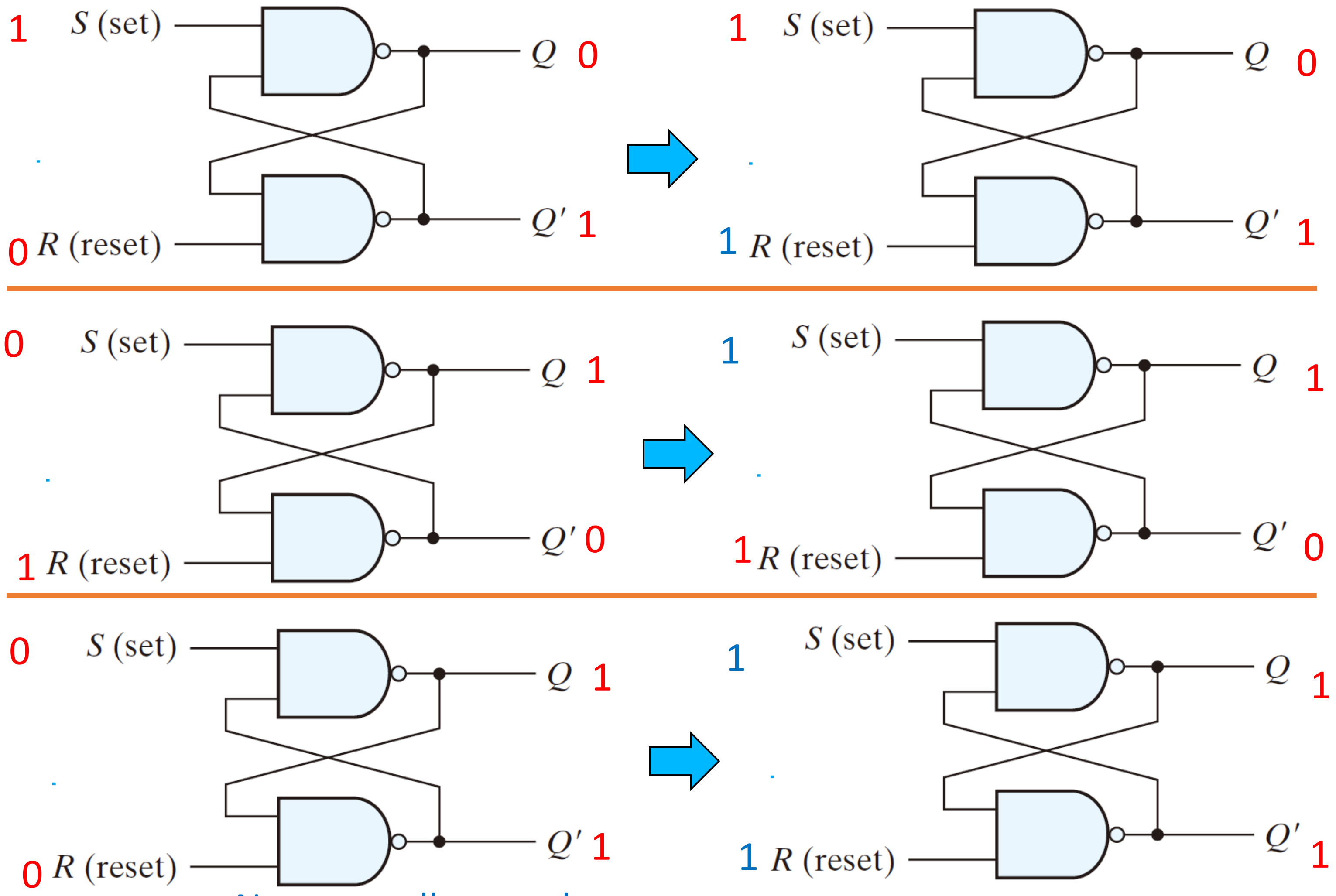
S	R	Q	Q'	
1	0	0	1	
1	1	0	1	(after $S = 1, R = 0$)
0	1	1	0	
1	1	1	0	(after $S = 0, R = 1$)
0	0	1	1	(forbidden)

(b) Function table

FIGURE 5.4
SR **latch with NAND gates**

What is the difference between SR latches with NOR gates and NAND gates?

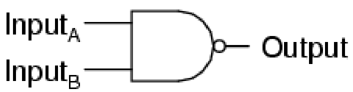
SR Latches with NAND Gates



Not mutually complementary

Forbidden!!!

NAND gate



A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0

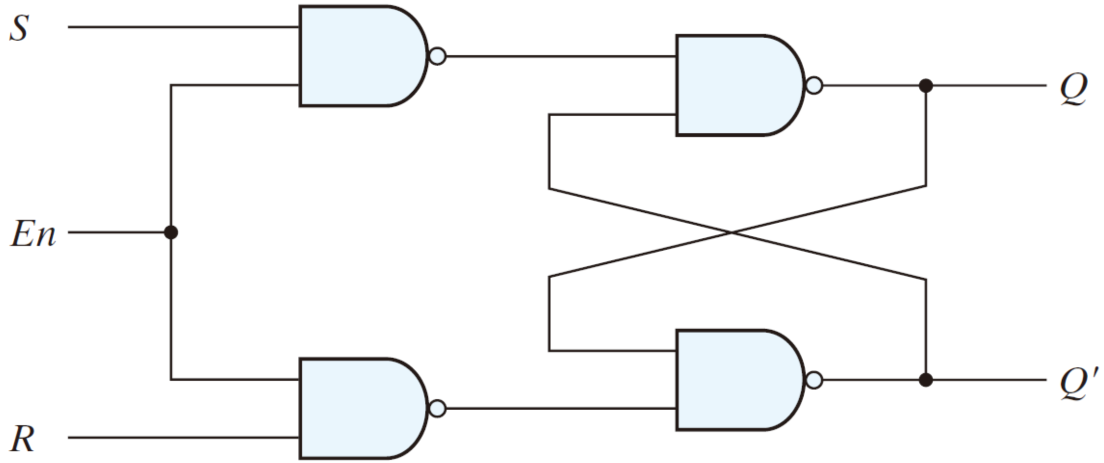
S	R	Q	Q'	
1	0	0	1	
1	1	0	1	(after $S = 1, R = 0$)
0	1	1	0	
1	1	1	0	(after $S = 0, R = 1$)
0	0	1	1	(forbidden)

(b) Function table

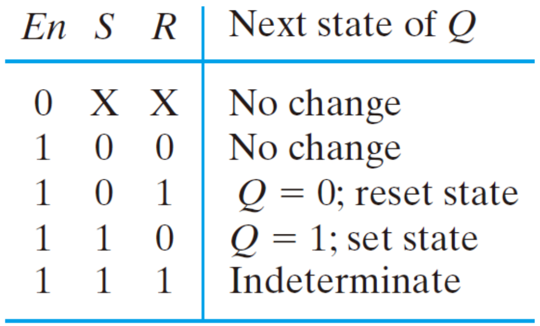
- S=1, R=O → reset state (Q will become to 0)
- S=O, R=1 → set state (Q will become to 1)
- When S=1 and R=1 → keep the current value

SR Latch with Control Input

- Add an additional control input to determine when the state of the latch can be changed
- En = 0: S and R are disabled (no change at outputs)
- En = 1: S and R are active-high



(a) Logic diagram



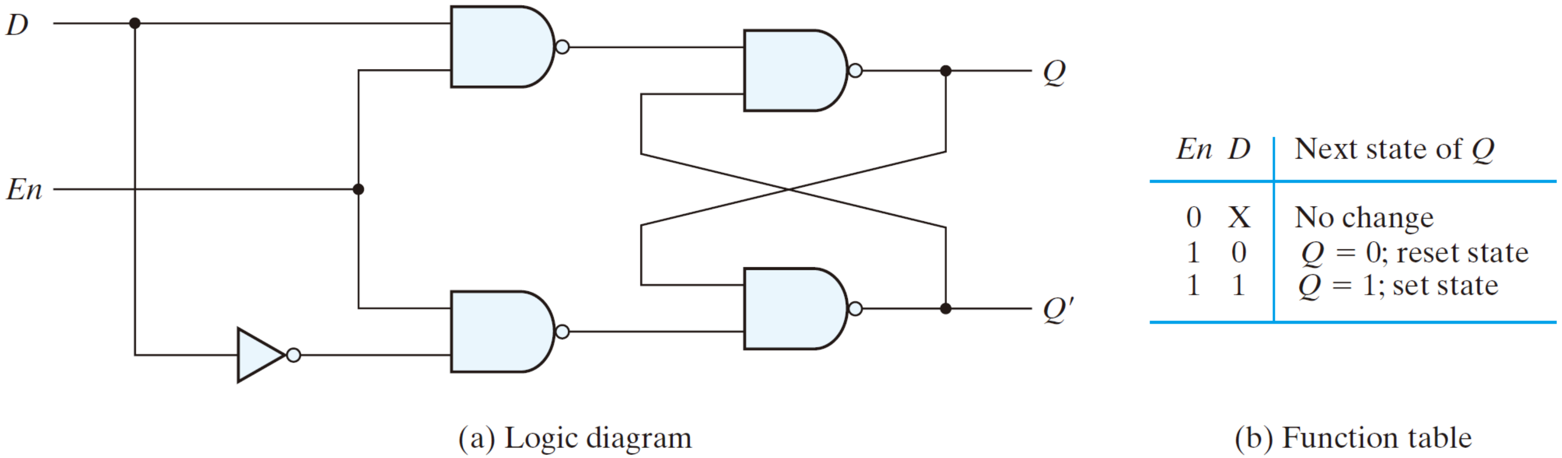
En	*S*	*R*	Next state of *Q*
0	X	X	No change
1	0	0	No change
1	0	1	$Q = 0$; reset state
1	1	0	$Q = 1$; set state
1	1	1	Indeterminate

(b) Function table

FIGURE 5.5
SR latch with control input

D Latch (Transparent Latch)

- Eliminate the indeterminate state in the SR latches
- D latch has only two inputs: D (data) and En (control)
- Use the value of D to set the output value
- The D input goes directly to the S input and its complement is applied to the R input
  - D=1 → S=1, R=0 → Q=1



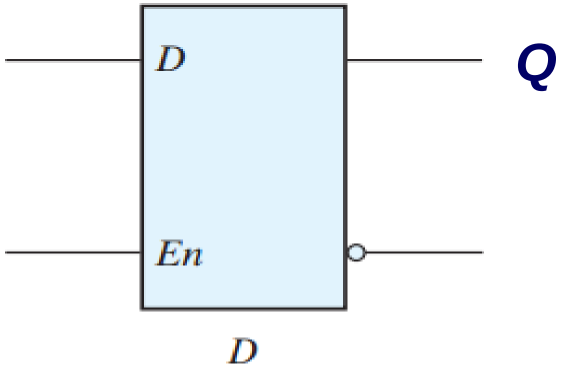
(a) Logic diagram

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

(b) Function table

FIGURE 5.6
D latch

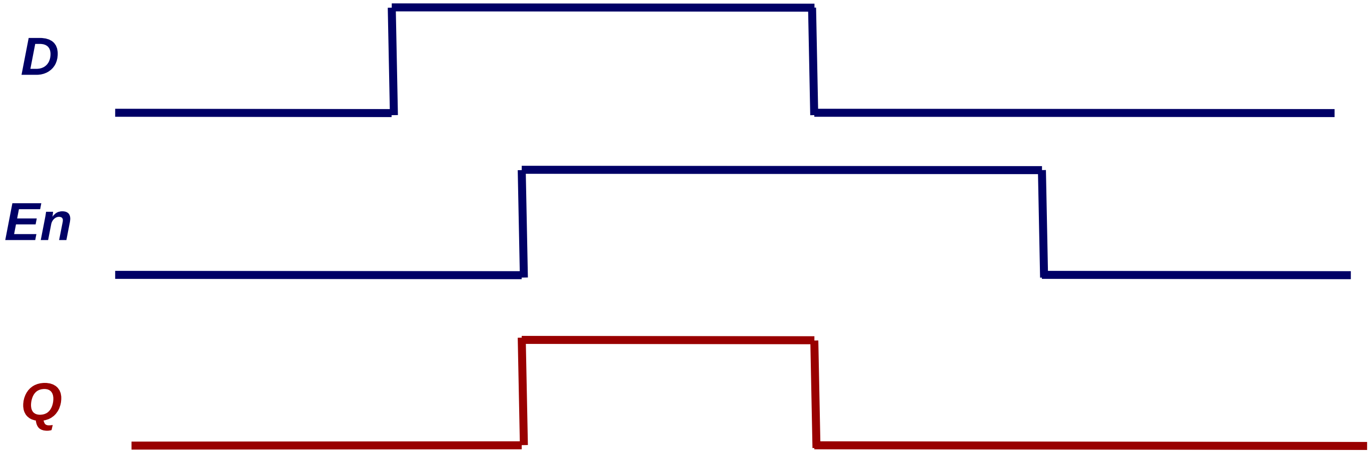
D Latch Timing Waveform



Assume initial value $Q_{init} = 0$

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

(b) Function table



Graphic symbols for Latches

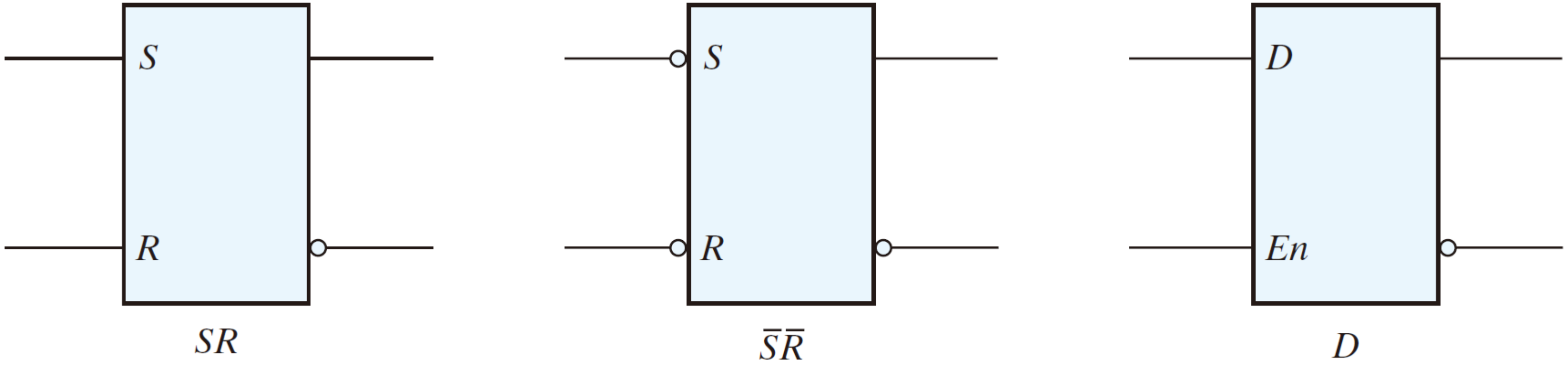
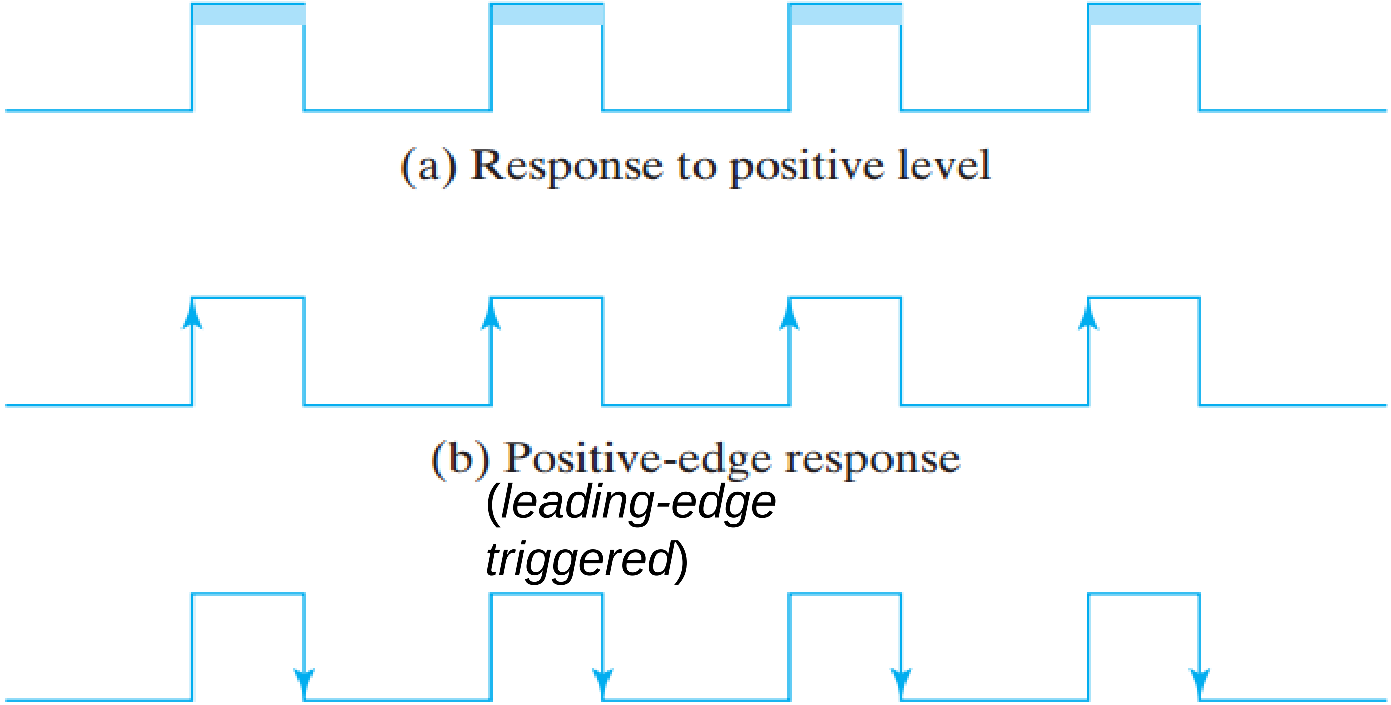


FIGURE 5.7
Graphic symbols for latches

- You should know the operation in your mind when seeing the block diagrams of latches.
- SR latch: (*S*,*R*) = (1,0) (0,1) (0,0)
- S’R’ latch: (*S*,*R*) = (0,1) (1,0) (1,1)
  - In the case of a NAND gate latch, bubbles are added to the inputs to indicate that setting and resetting occur with a logic-0 signal.
- D latch: *En*, *D*

Storage Elements: Flip-flops

- The state of a latch or flip-flop is switched by a change in the control input. This momentary change is called a trigger, and the transition it causes is said to trigger the flip-flop.



(a) Response to positive level

Level triggered

(b) Positive-edge response

(*leading-edge triggered*)

Edge triggered

(c) Negative-edge response

(*trailing-edge triggered*)

Edge triggered

Negative Edge-Triggered D Flip-flops

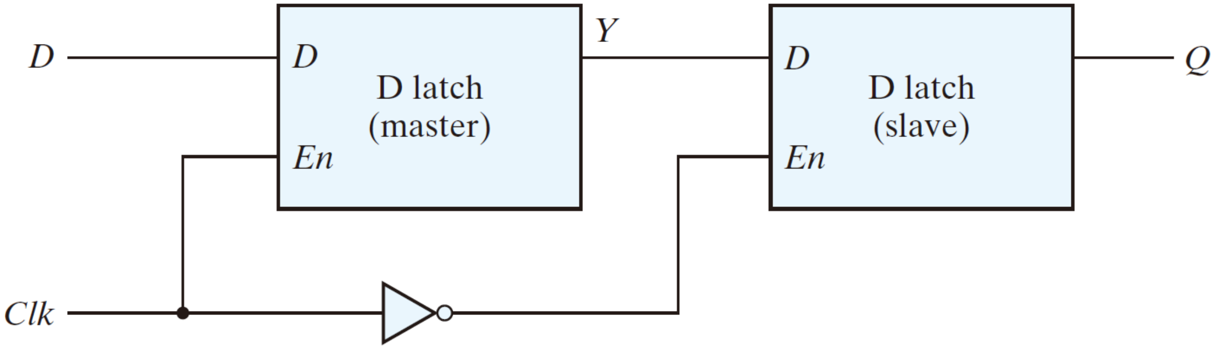
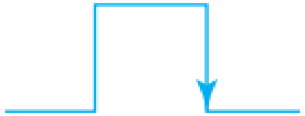


FIGURE 5.9
Master–slave *D* flip-flop



- Constructed with two D latches and an inverter
- The first latch (master) is enabled when CLK=1
  - the data from the external D input are transferred to the master
- The second latch (slave) is enabled when CLK=0
  - When the clock pulse returns to 0, the master is disabled and is isolated from the D input.
  - At the same time, the slave is enabled and the value of Y is transferred to the output of the flip-flop at Q.
- The circuit samples the D input and changes its output Q only at the negative-edge of the controlling clock

Negative Edge-Triggered D Flip-flops

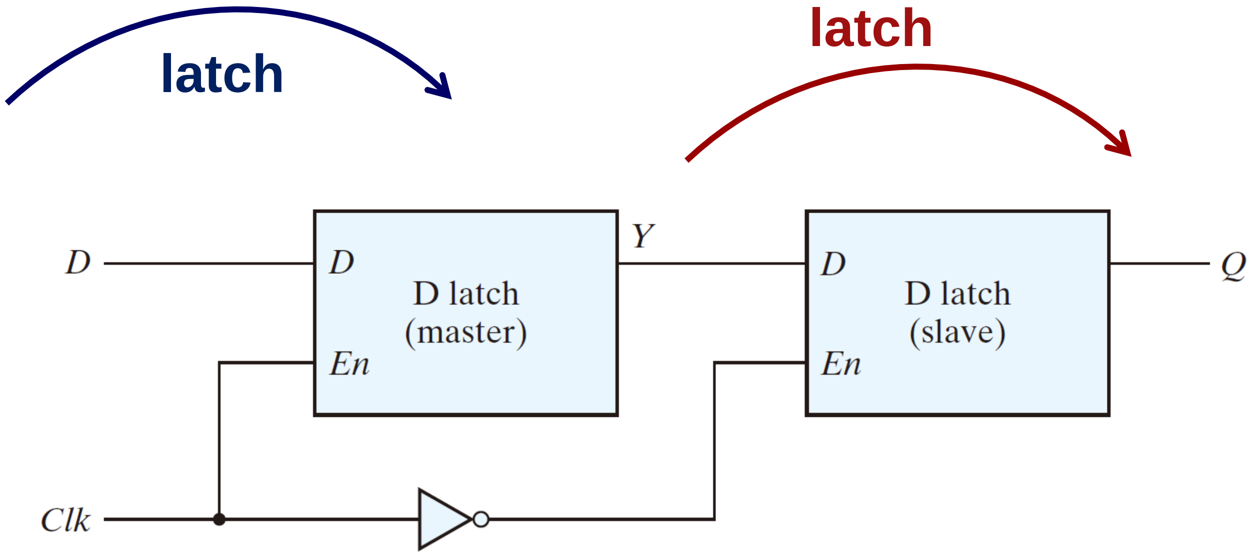
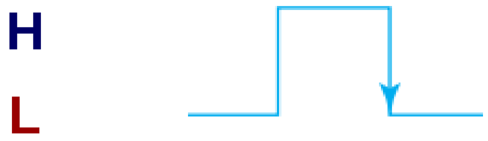
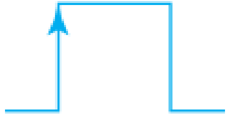


FIGURE 5.9
Master–slave *D* flip-flop



- ➢ Only the transition from high to low, the input *D* can be passed over the cascaded latches
- ➢ Negative edge-triggered
- ➢ How do you change it to positive edge-triggered?



Positive Edge-Triggered D Flip-flops

- An edge-triggered D flip-flop uses three SR latches
- Two latches respond to the external D (data) and Clk (clock) inputs.
- The third latch provides the outputs for the flip-flop.
- When the input clock in the positive-edge-triggered flip-flop makes a positive transition, the value of D is transferred to Q.
- A negative transition of the clock (i.e., from 1 to 0) does not affect the output, nor is the output affected by changes in D when Clk is in the steady logic-1 level or the logic-0 level.
- Hence, this type of flip-flop responds to the transition from 0 to 1 and nothing else.

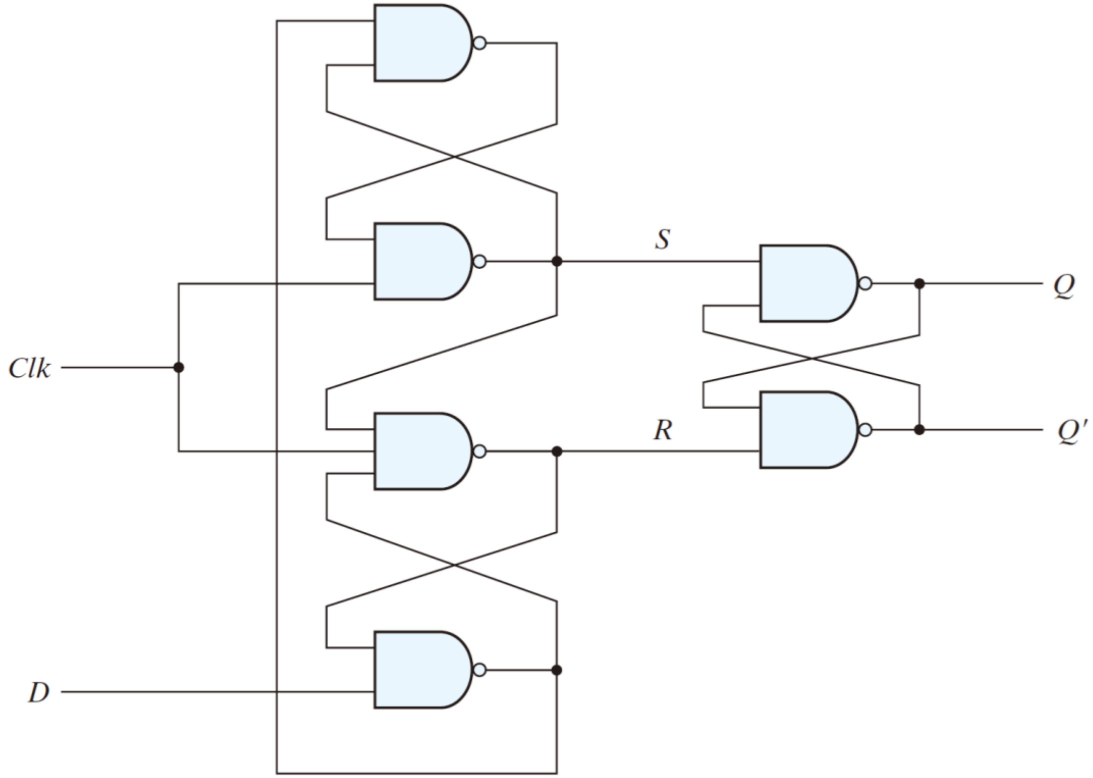


FIGURE 5.10
D-type positive-edge-triggered flip-flop

Positive Edge-Triggered D Flip-flops

- Clk=0; S and R are maintained at 1::Q and Q' in the present state
- If D=0; and Clk=0 to 1:: R=0 which makes Q=0 (reset)
- If there is a change in the input if clk is 1 :: Flip-flop is locked.
- Clk=0::R=1:: No change sin the output
- If D=1; Clk=0 to 1, S=0 and Q=1

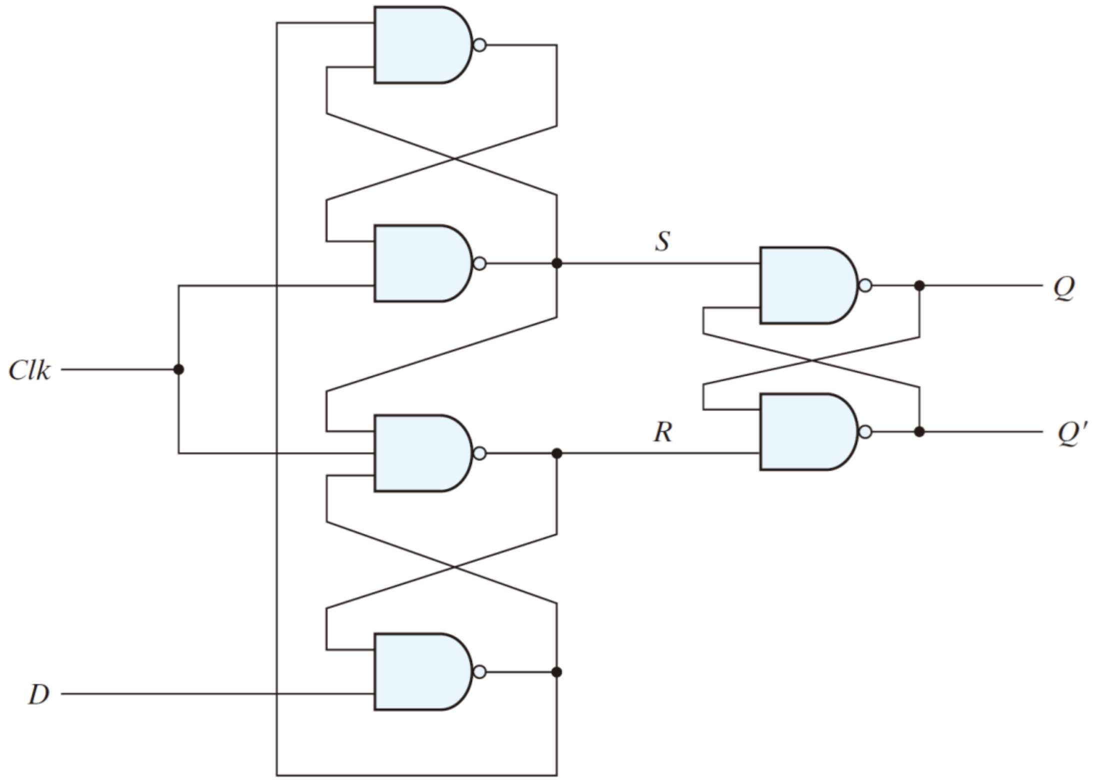
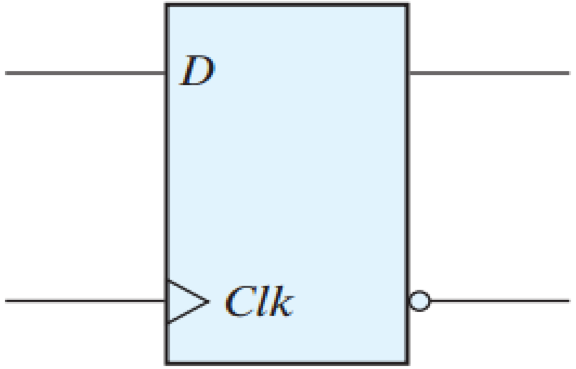


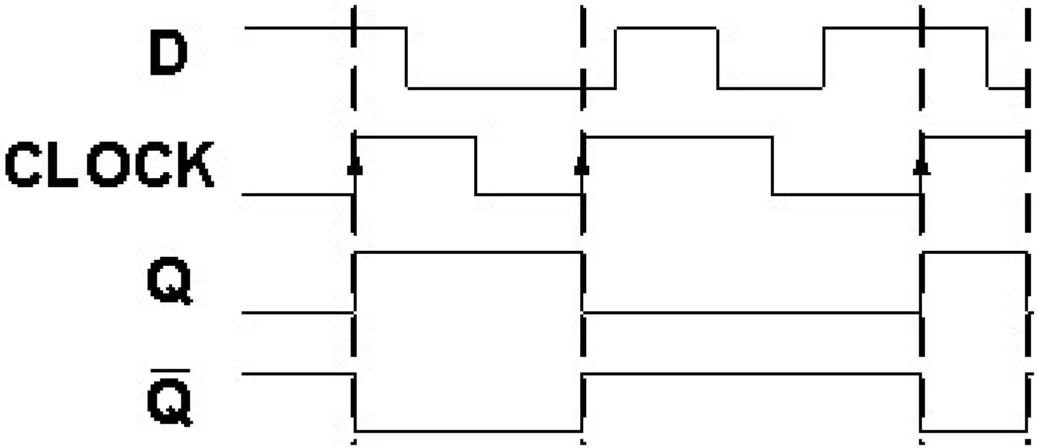
FIGURE 5.10
D-type positive-edge-triggered flip-flop

D Flip-flop

***D* Flip-Flop**

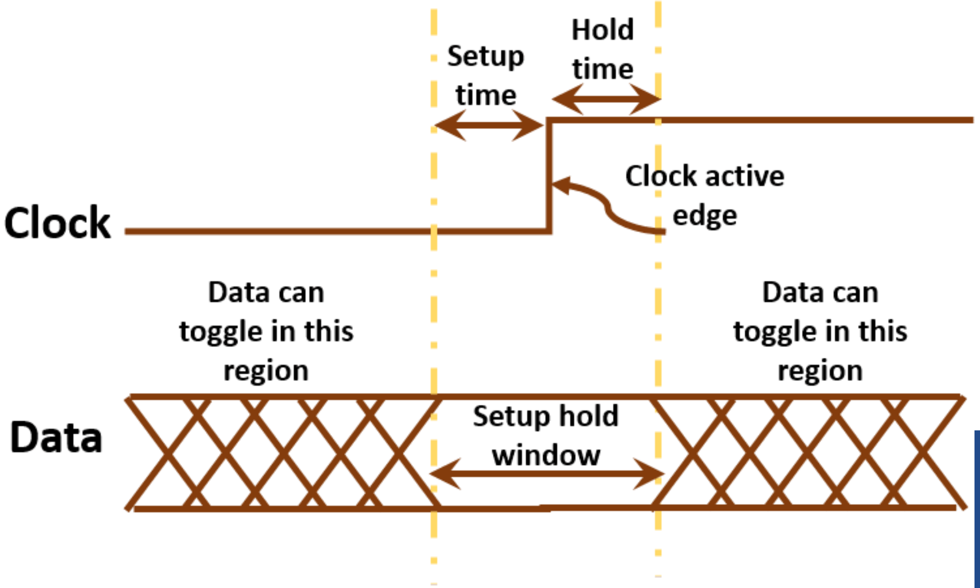
D	*Q*(*t* + 1)	
0	0	Reset
1	1	Set





Setup Time and Hold Time

- The timing of the response of a flip-flop to input data and to the clock must be taken into consideration when one is using edge-triggered flip-flops.
- There is a minimum time called the *setup time* during which the *D* input must be maintained at a constant value prior to the occurrence of the clock transition.
- Similarly, there is a minimum time called the *hold time* during which the *D* input must not change after the application of the positive transition of the clock.
- The propagation delay time of the flip-flop is defined as the interval between the trigger edge and the stabilization of the output to a new state.



DFF (D Flip-Flop) Symbol

- The dynamic indicator (>) denotes the fact that the flip-flop responds to the edge transition of the clock.
- A bubble (o) outside the block adjacent to the dynamic indicator designates a negative edge trigger.

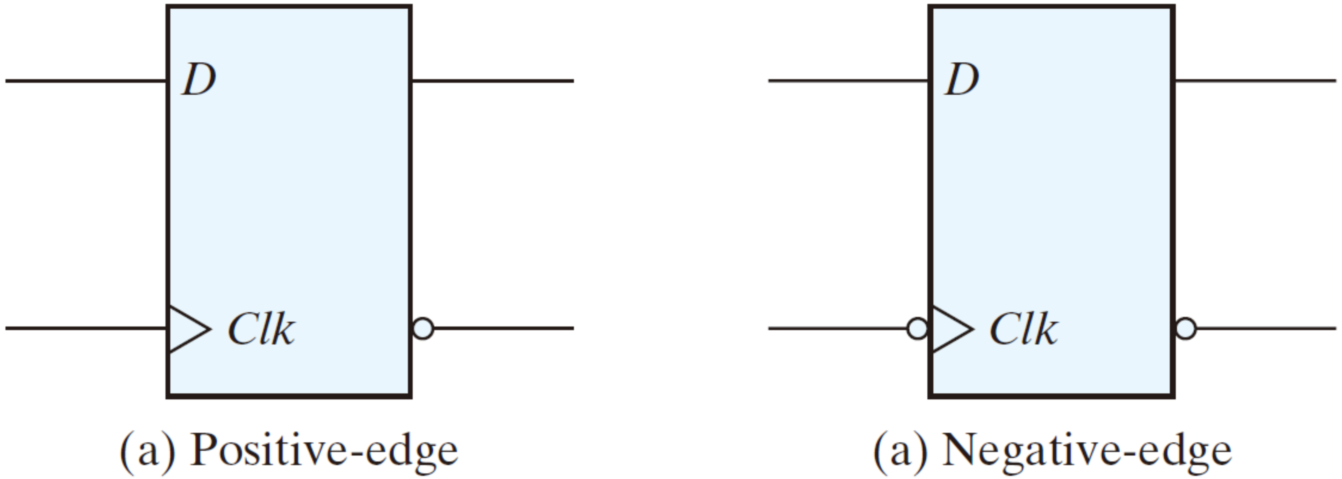
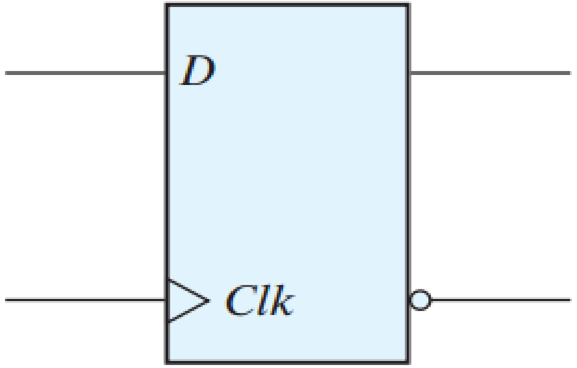


FIGURE 5.11
Graphic symbol for edge-triggered *D* flip-flop

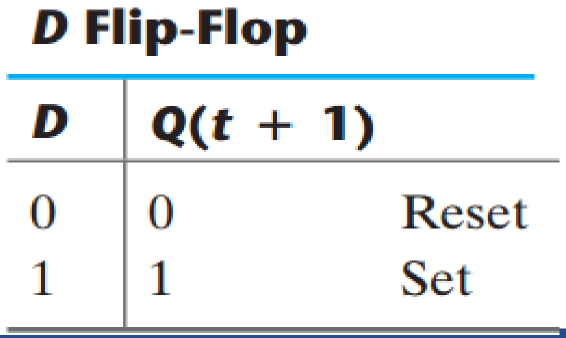
Characteristic Table

- **A characteristic table defines the logical properties of a flip-flop by describing its operation in tabular form.**
- **The table defines the next state as a function of the inputs and the present state.**
- Q(t) refers to the present state (i.e., the state present prior to the application of a clock edge).

  Q(t + 1) is the next state one clock period later.
- Note that the clock edge input is not included in the characteristic table, but is implied to occur between times t and t + 1.
  - Q(t) ➔ state of the flip-flop immediately before the clock edge
  - Q(t + 1) ➔ state that results from the clock transition



Characteristic Table



***D* Flip-Flop**

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

Characteristic Equation

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

Other Flip-Flops

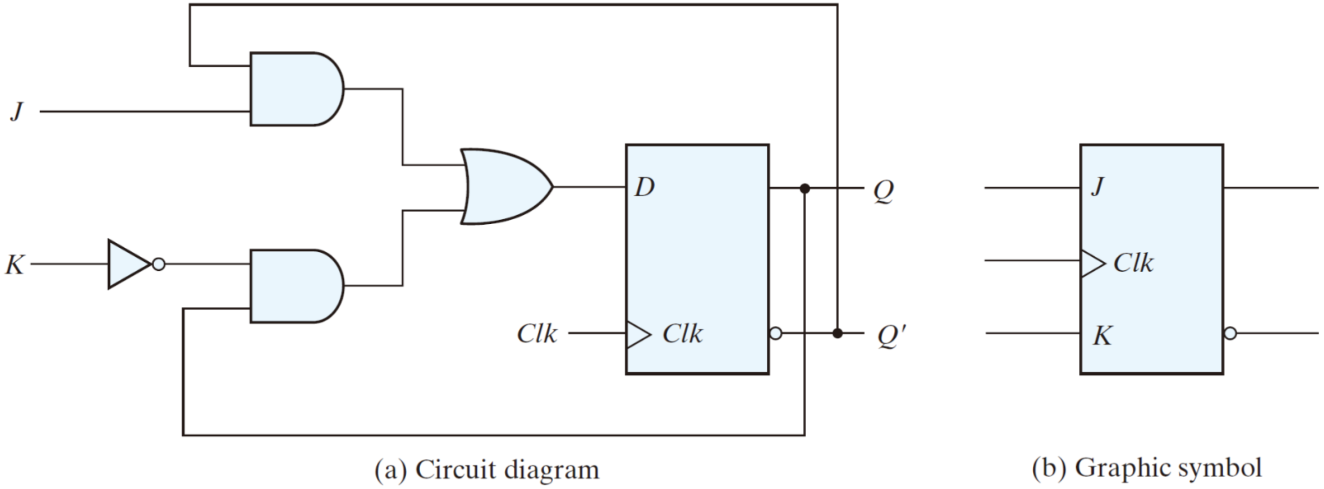
- JK and T flip-flops
- There are three operations that can be performed with a flip-flop:
  - Set it to 1
  - reset it to 0
  - Complement its output
- With only a single input, the **D flip-flop** can set or reset the output, depending on the value of the D input immediately before the clock transition.
- Synchronized by a clock signal, the **JK flip-flop** has two inputs and performs all three operations.
- The **T (toggle) flip-flop** is a complementing flip-flop.

JK Flip-Flops

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

- JK flip-flop function: The J input sets the flip-flop to 1, the K input resets it to 0, and when both inputs are enabled, the output is complemented.

$$D = JQ' + K'Q$$



(a) Circuit diagram

(b) Graphic symbol

FIGURE 5.12
JK **flip-flop**

(1). (J=1 , K=0) ➔ $D = JQ' + K'Q = 1 \cdot Q' + 1 \cdot Q = Q' + Q = 1$

(2). (J=0 , K=1) ➔ $D = JQ' + K'Q = 0 \cdot Q' + 0 \cdot Q = 0 + 0 = 0$

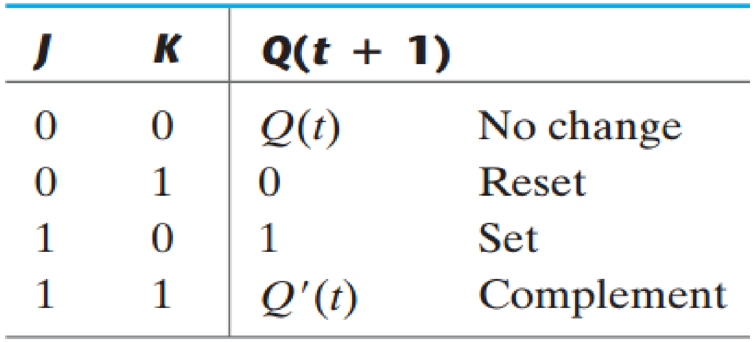
(3). (J=1 , K=1) ➔ $D = JQ' + K'Q = 1 \cdot Q' + 0 \cdot Q = Q'$

(4). (J=0 , K=0) ➔ $D = JQ' + K'Q = 0 \cdot Q' + 1 \cdot Q = Q$ **, No change**

Characteristic Table

JK Flip-Flop

Characteristic Table

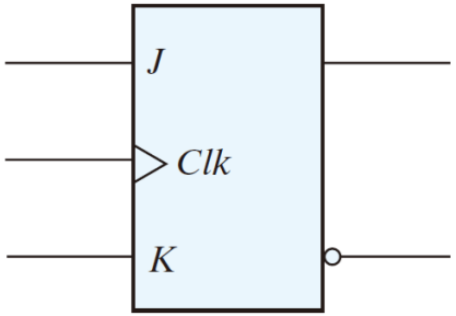


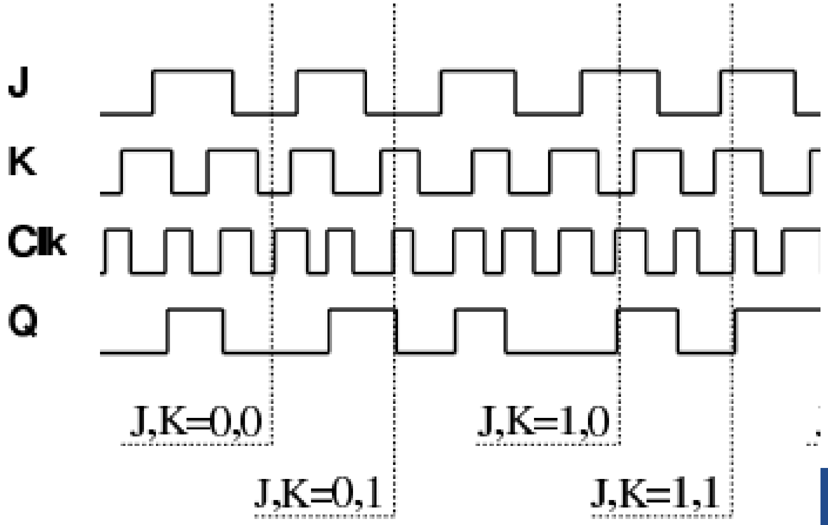
J	K	$Q(t+1)$	
0	0	$Q(t)$	No change
0	1	0	Reset
1	0	1	Set
1	1	$Q'(t)$	Complement

Characteristic Equation

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

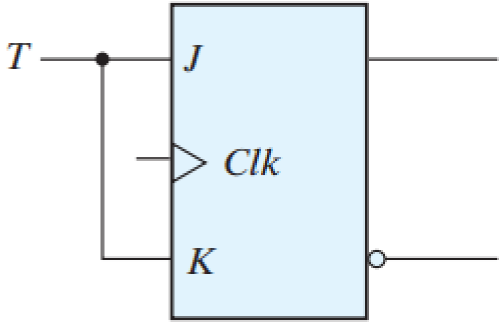
Graphic symbol





T Flip-Flops

- **T flip-flop: “toggle”** flip-flop → complementing flip-flop
- Constructed with JK flip-flop when inputs J and K are tied together
- *When* $T = 0$ ($J = K = 0$), a clock edge does not change the output.
- When $T = 1$ ($J = K = 1$), a clock edge complements the output.



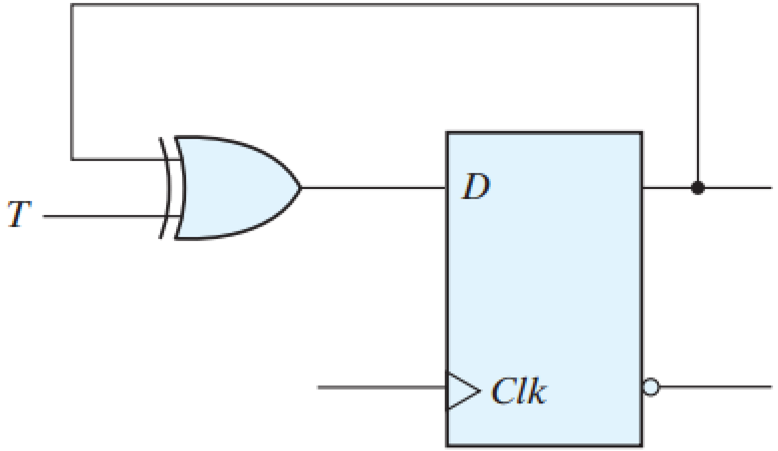
(a) From *JK* flip-flop

$$(3).\ (J{=}1\ ,\ K{=}1) \Rightarrow\ D = JQ' + K'Q = 1\cdot Q' + 0\cdot Q = Q'$$

$$(4).\ (J{=}0\ ,\ K{=}0) \Rightarrow\ D = JQ' + K'Q = 0\cdot Q' + 1\cdot Q = Q$$

T Flip-Flops

- T flip-flop can be constructed with a D flip-flop and an exclusive-OR gate



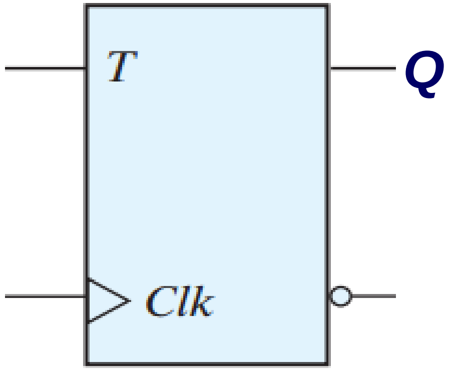
(b) From *D* flip-flop

$$\Rightarrow D = T \oplus Q = TQ' + T'Q$$

- When T = 0, D = Q → no change
- When T = 1, D = Q’ → complement

Characteristic Table

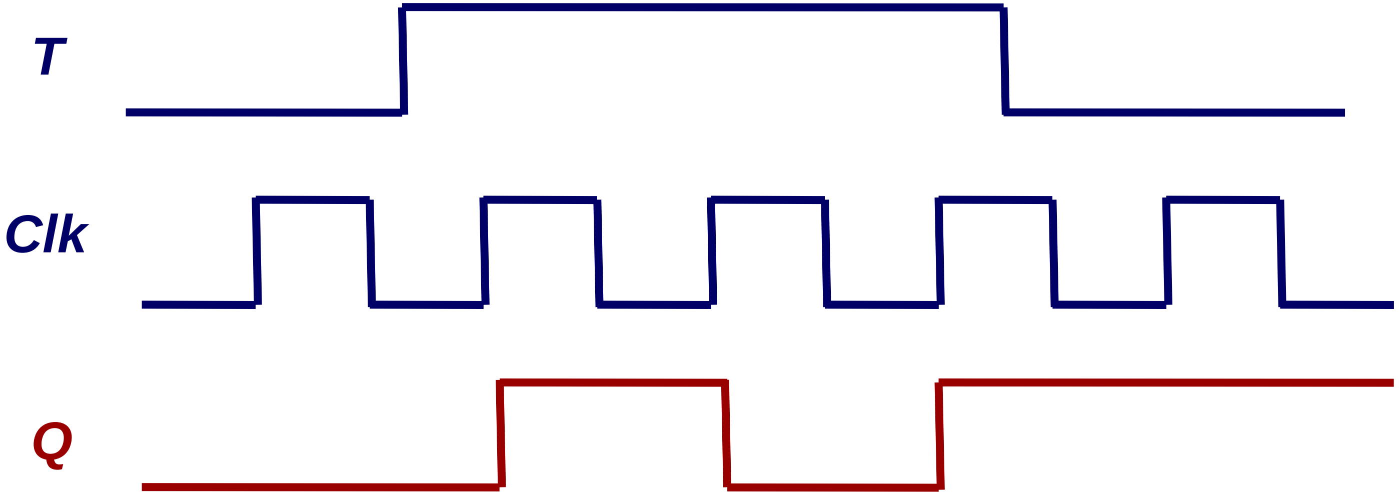
T Flip-Flop



Assume initial value $Q_{init} = 0$

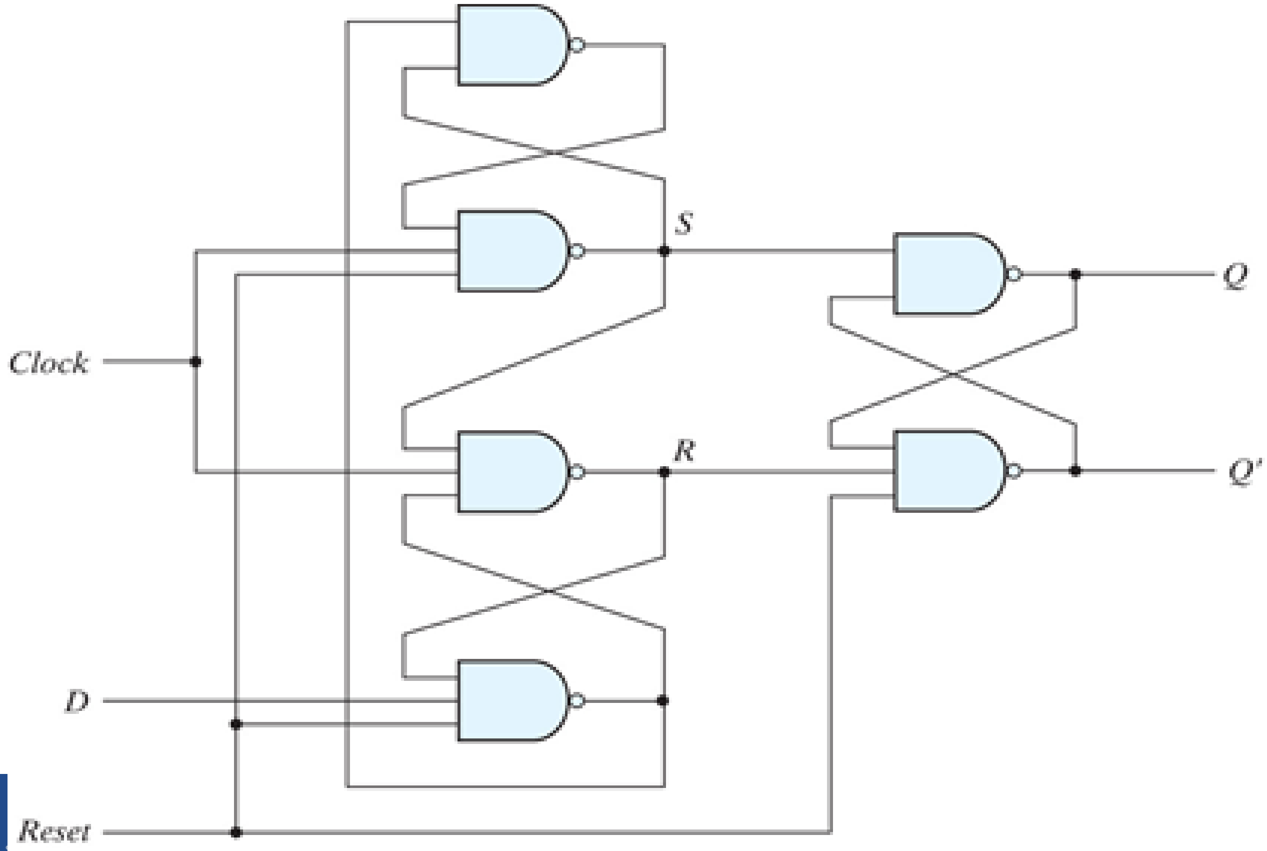
T	$Q(t+1)$	
0	$Q(t)$	No change
1	$Q'(t)$	Complement

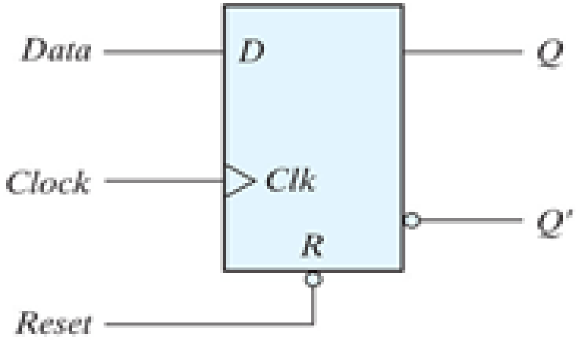
$$Q(t+1) = T \oplus Q = TQ' + T'Q$$



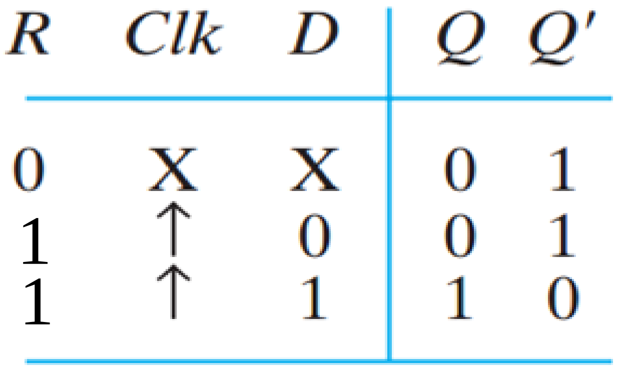
DFF with Asynchronous Reset

- Some flip-flops have asynchronous inputs that are used to force the flip-flop to a particular state independently of the clock.
- The input that sets the flip-flop to 1 is called preset or direct set.
- The input that clears the flip-flop to 0 is called clear or direct reset.





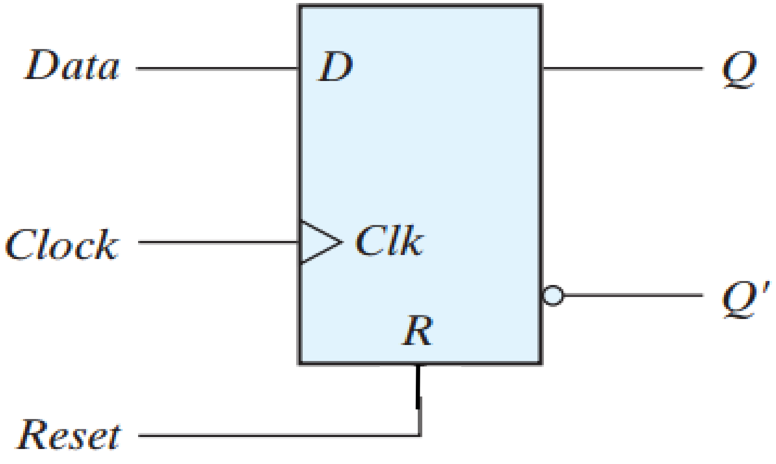
(b) Graphic symbol



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

DFF with Asynchronous Reset

- Asynchronous reset is used to force the flip-flop to reset to 0, independently of the clock.



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

- Active high reset: a circuit is reset when *R* = “1”.
- Active low reset: a circuit is reset when *R* = “0”.

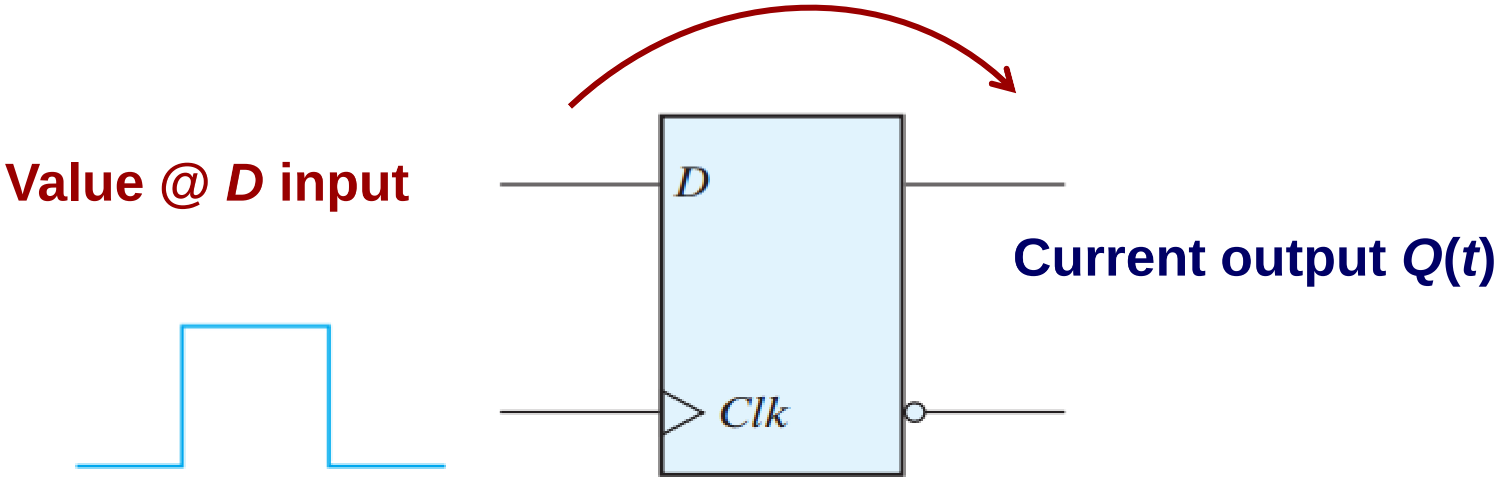
Analysis of Clocked Sequential Circuits

- Analysis describes what a given circuit will do under certain operating conditions.
- The behavior of a clocked sequential circuit is determined from the inputs, the outputs, and the state of its flip-flops.
- The analysis of a sequential circuit consists of obtaining a table or a diagram for the time sequence of inputs, outputs, and internal states.

Analysis of Clocked Sequential Circuits

- The next state of a flip-flop is determined by
  - Input values
  - Current state of the FF
  - Clock trigger

input value determines $Q(t + 1)$



next clock edge comes

State Equations (Transition Equations)

- The behavior of a clocked sequential circuit can be described algebraically by means of state equations.
- A state equation (also called a transition equation) specifies the next state as a function of the present state and inputs.
- A state equation is an algebraic expression that specifies the condition for a flip-flop state transition.

State Equations (Transition Equations)

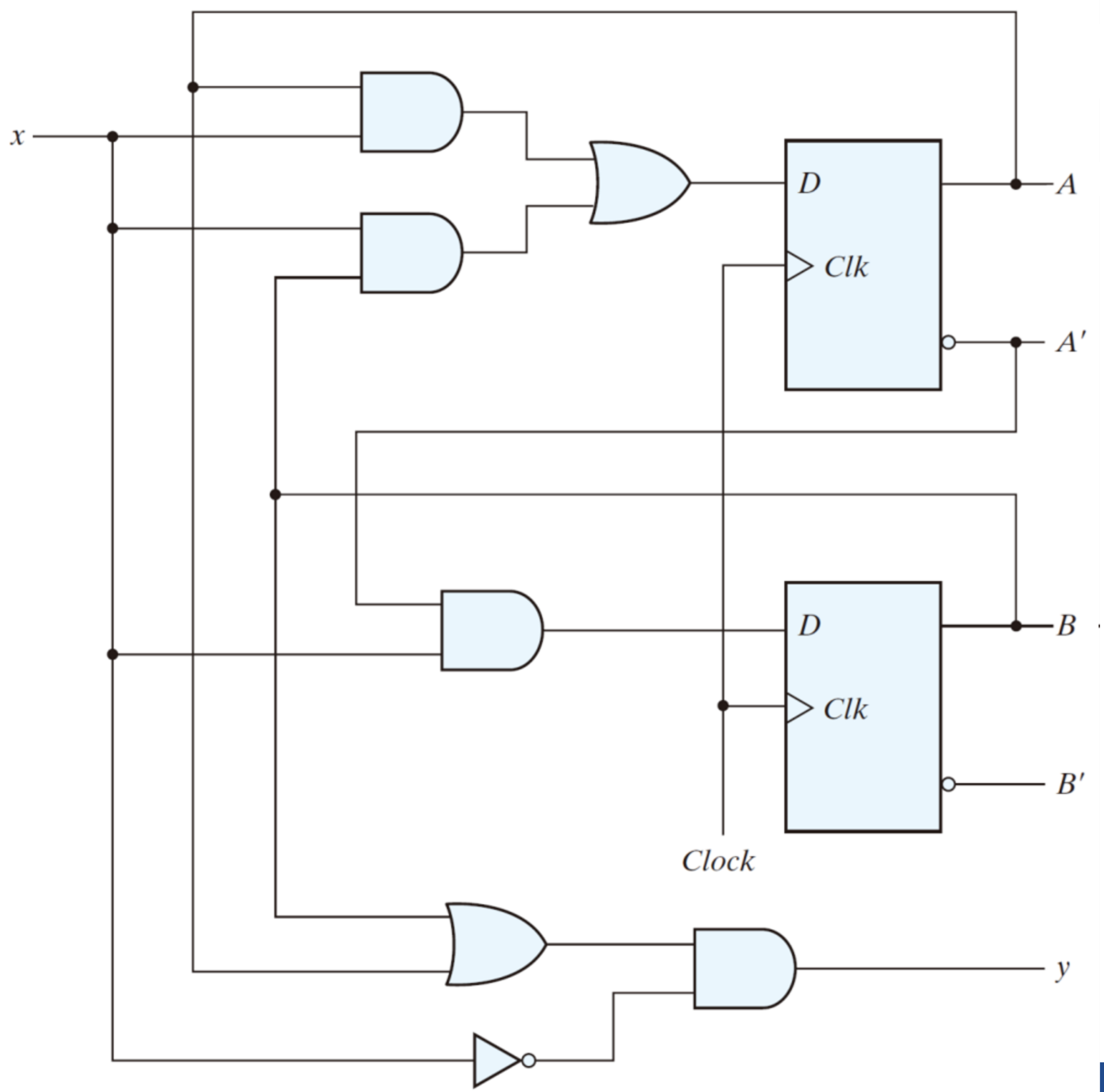


FIGURE 5.15
Example of sequential circuit

D input of a flip-flop determines the value of the next state

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

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

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

State Equations

- The left side of the equation, with (t + 1), denotes the next state of the flip-flop one clock edge later.
- The right side of the equation is a Boolean expression that specifies the present state and inputs.

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

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

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

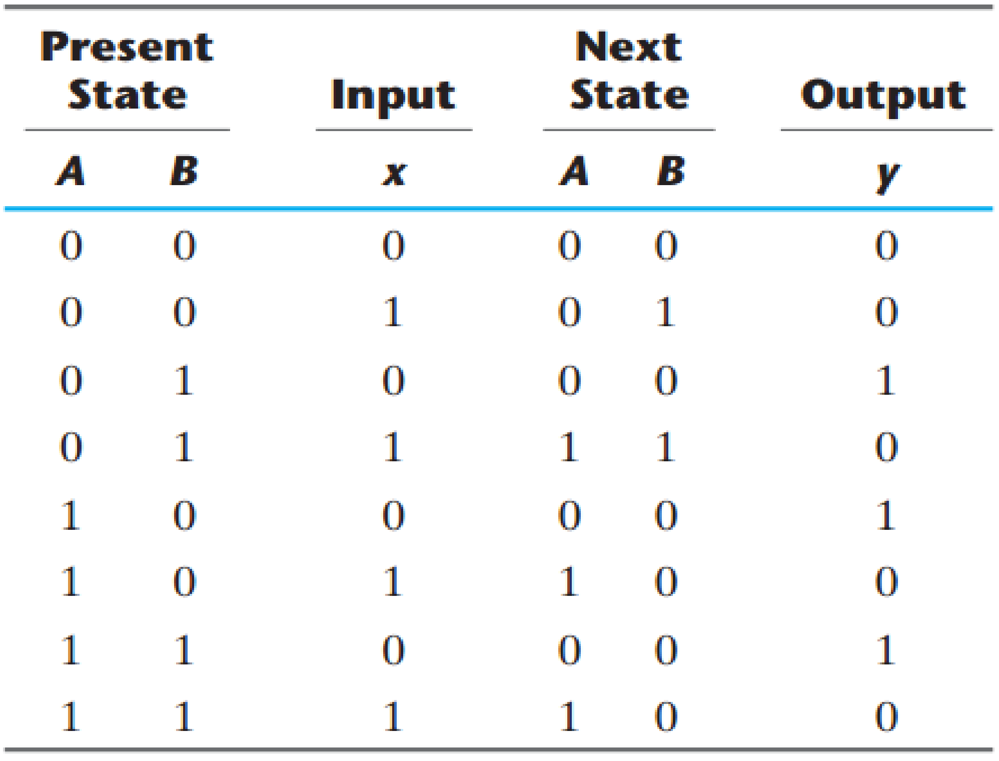
$$A(t+1) = Ax + Bx$$

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

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

State Table

- The time sequence of inputs, outputs, and flip-flop states can be enumerated in a state table (or transition table).
- The table consists of four sections labeled present state, input, next state, and output.
- The derivation of a state table requires listing all possible binary combinations of present states and inputs.



Present State		Input	Next State		Output
A	*B*	*x*	*A*	*B*	*y*
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

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

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

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

State Table

- It is sometimes convenient to express the state table in a slightly different form having only three sections: present state, next state, and output.

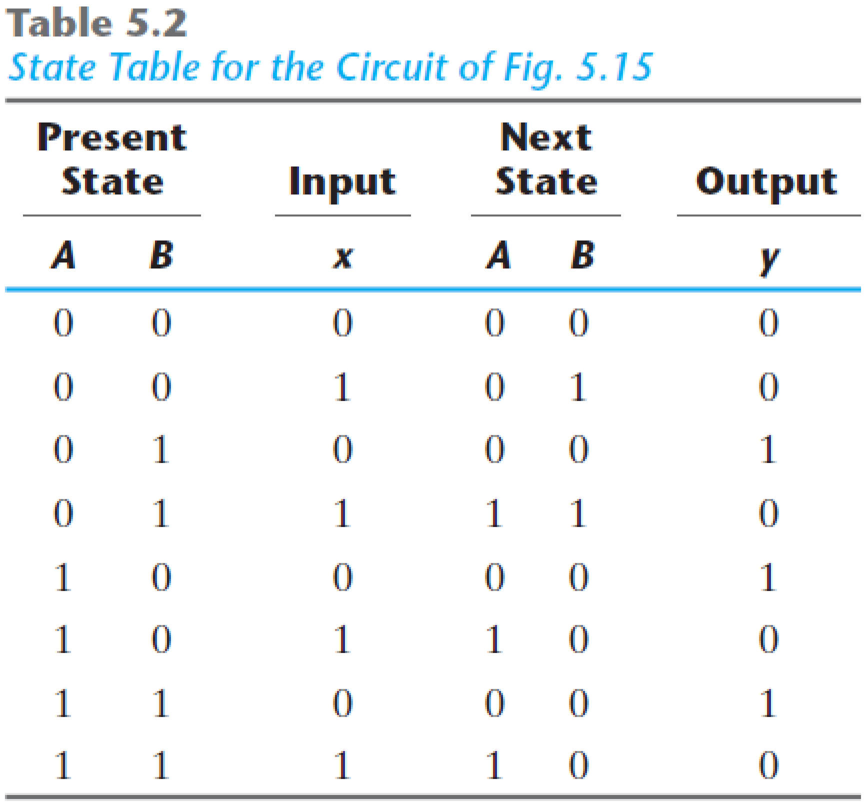


Table 5.2
State Table for the Circuit of Fig. 5.15

Present State		Input	Next State		Output
A	*B*	*x*	*A*	*B*	*y*
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

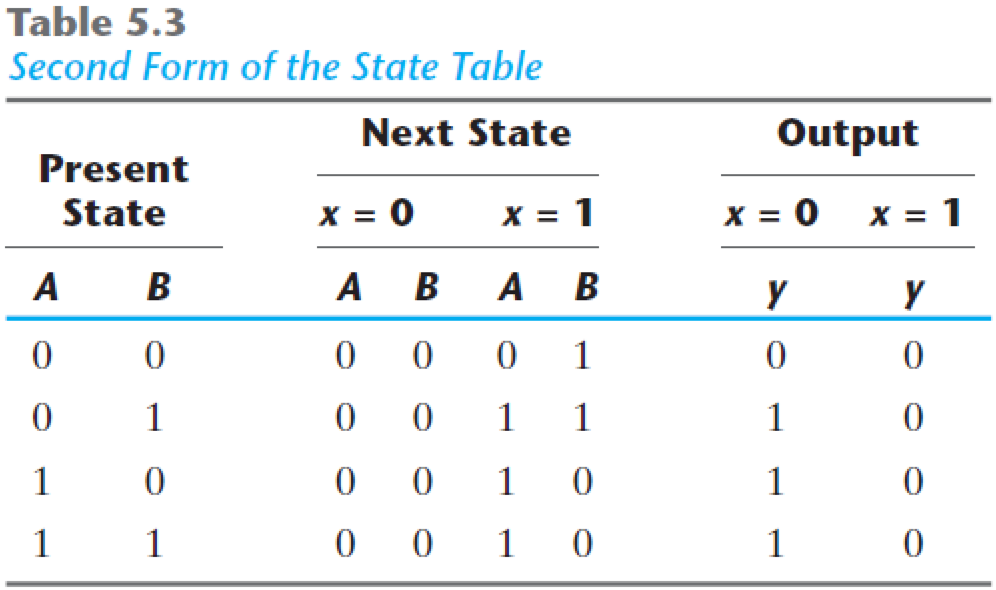


Table 5.3
Second Form of the State Table

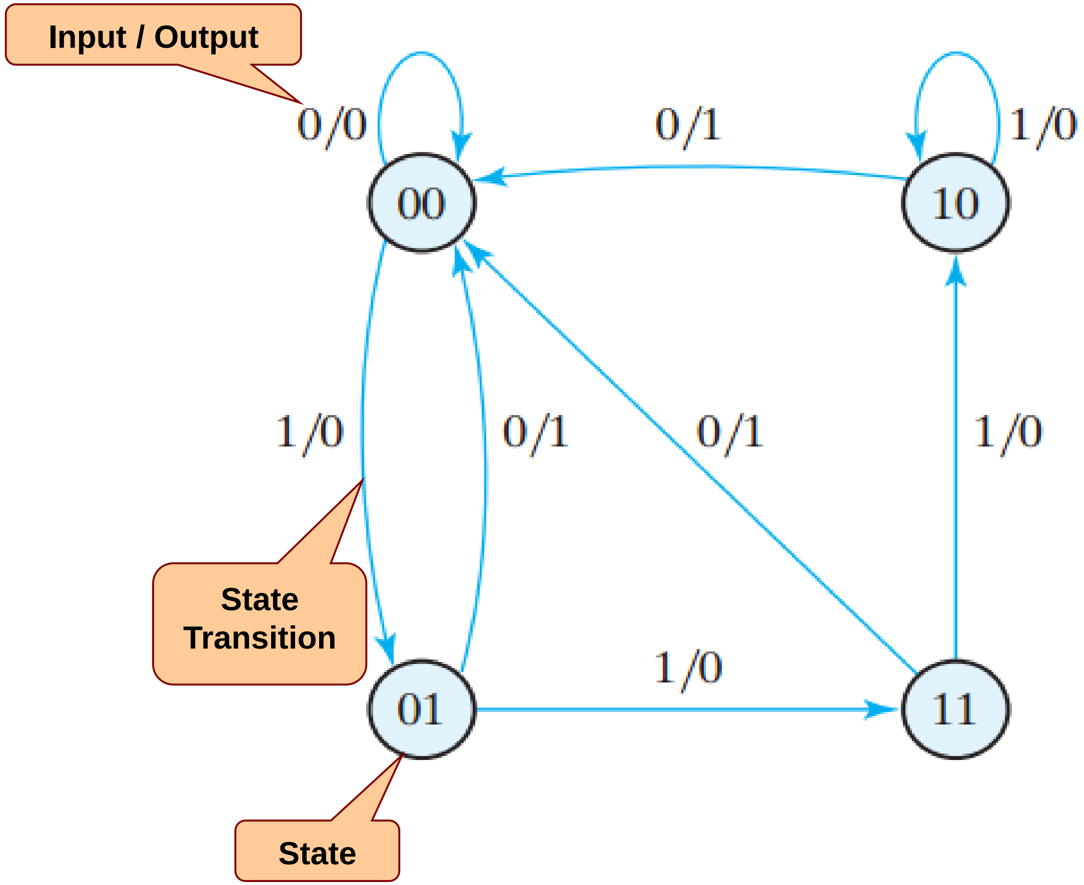
Present State		Next State				Output	
		x = 0		*x* = 1		*x* = 0	*x* = 1
A	*B*	*A*	*B*	*A*	*B*	*y*	*y*
0	0	0	0	0	1	0	0
0	1	0	0	1	1	1	0
1	0	0	0	1	0	1	0
1	1	0	0	1	0	1	0

State Diagram

- The information available in a state table can be represented graphically in the form of a state diagram.

Table 5.2
State Table for the Circuit of Fig. 5.15

Present State		Input	Next State		Output
A	*B*	*x*	*A*	*B*	*y*
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

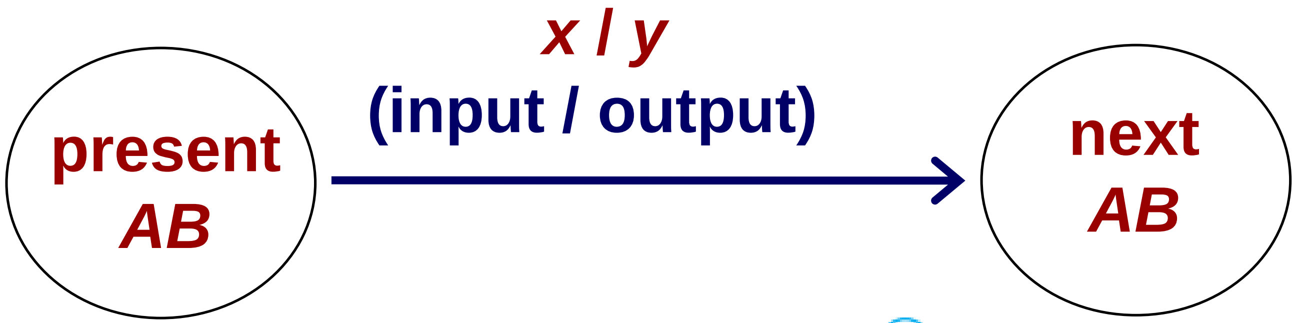


Input / Output

State Transition

State

State Diagram



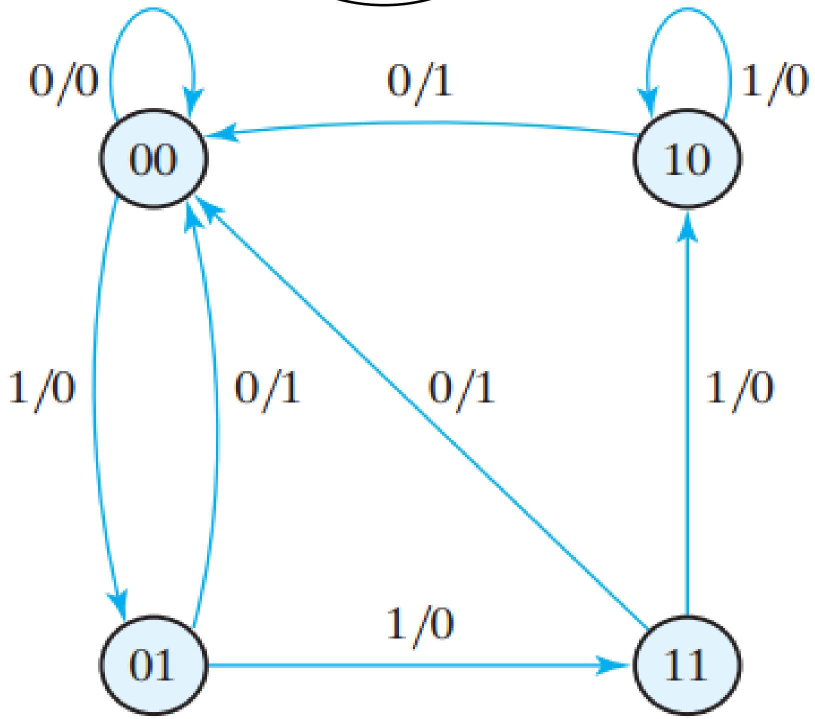
x / *y*

(input / output)

present *AB*

next *AB*

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



State Table & State Diagram

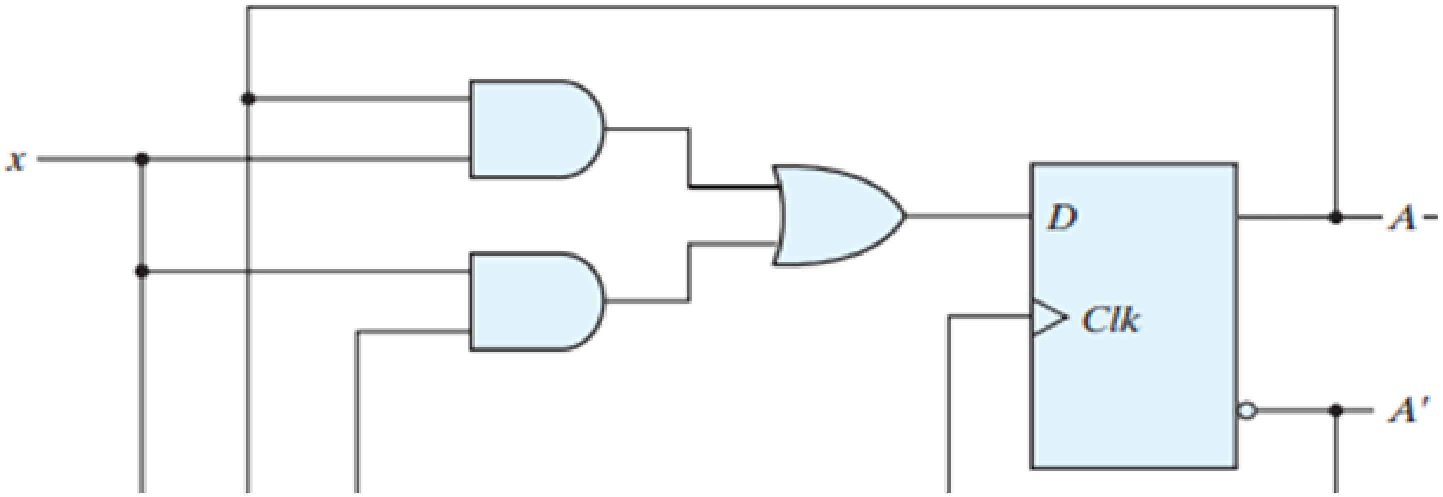
- **Summarized procedure:**

Circuit diagram → Equations – State table → State diagram

- There is no difference between a state table and a state diagram.
- The state table is easier to derive from a given logic diagram and the state equation.
- The state diagram is the form more suitable for human interpretation of the circuit's operation.
  - Easier for an HDL model in the form of gate-level description or in the form of a behavior description
  - Easier for the design of an automatic synthesis tool

Flip-Flop Input Equations (Excitation Equations)

- **The part of the circuit that generates the inputs to flip-flops is described algebraically by a set of Boolean functions called flip-flop input equations.**
- **The flip-flop input equations can be used to obtain state equations.**



$$D_A = Ax + Bx$$

D flip-flop with output A

Output Equations

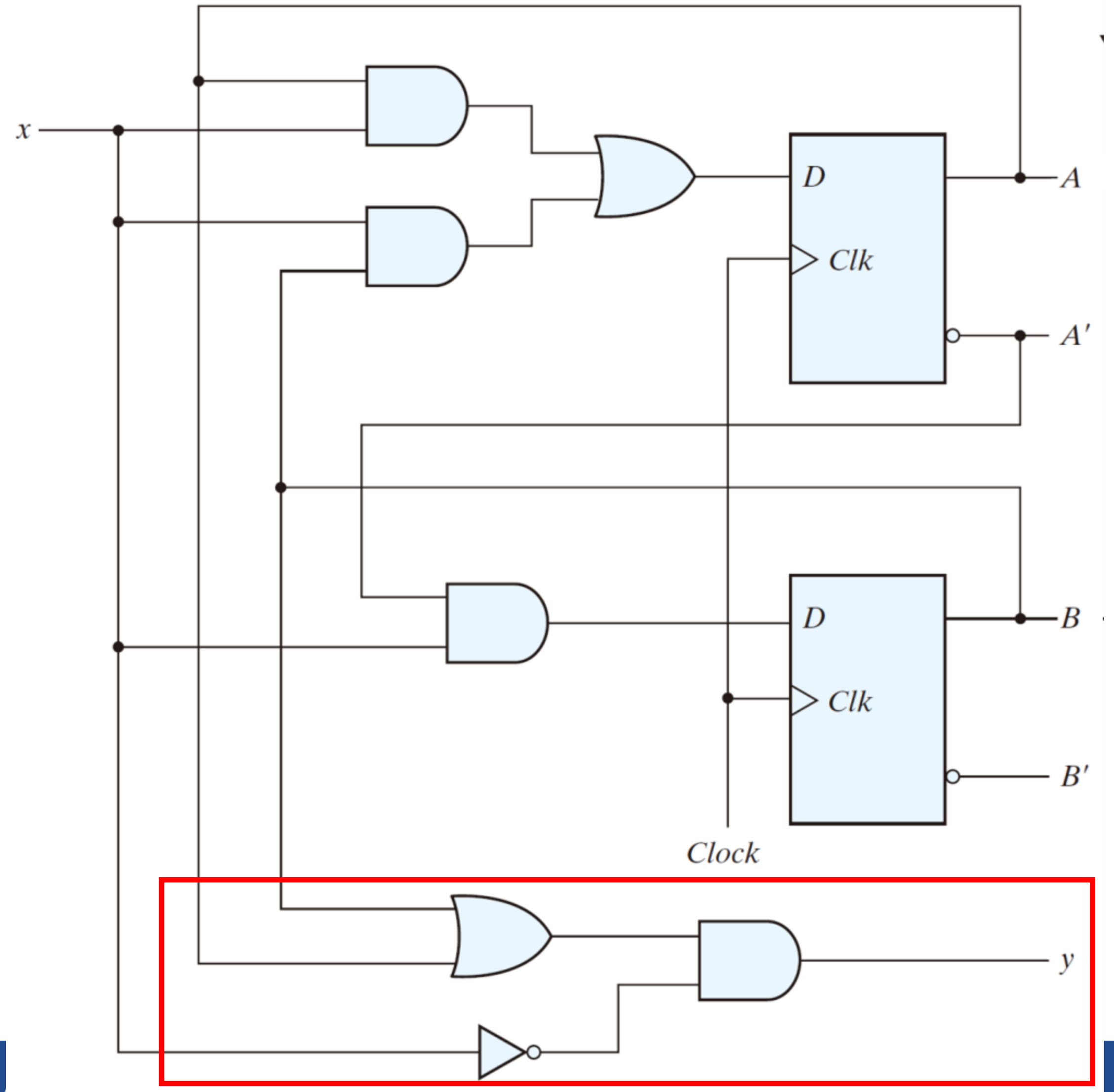


FIGURE 5.15
Example of sequential circuit

- External outputs are described algebraically by a set of Boolean functions called output equations.

$$D_A = Ax + Bx$$
$$D_B = A'x$$
$$y = (A + B)x'$$

- The Boolean expressions associated with the input/output equations specify the combinational circuit part of the sequential circuit

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

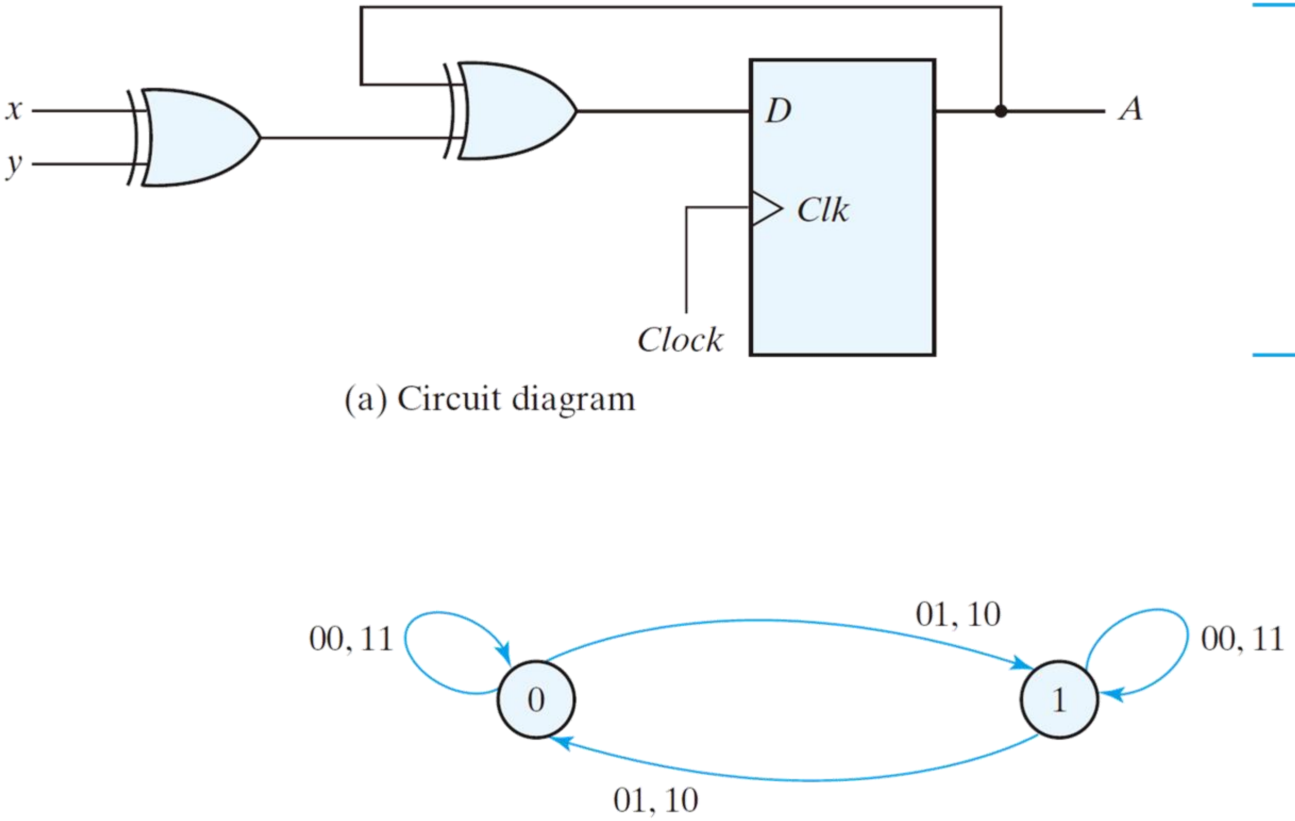
Analysis with DFF

- **Input equation**

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

The state equation

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



(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

(c) State diagram

FIGURE 5.17
Sequential circuit with *D* flip-flop

Analysis with JK Flip-Flop

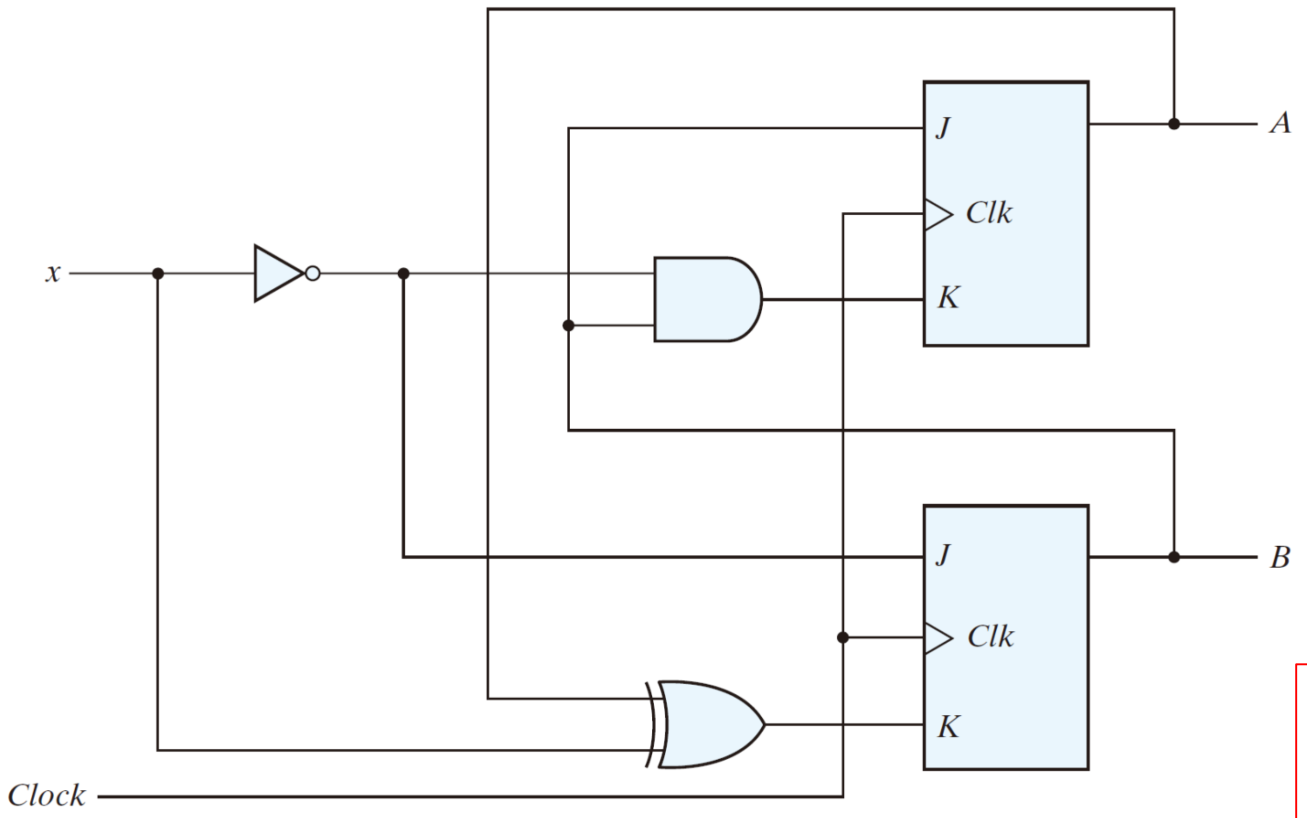


FIGURE 5.18
Sequential circuit with *JK* flip-flop

Characteristic equation for JK flip-flop:

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

Write state equations

Analysis with JK Flip-Flop

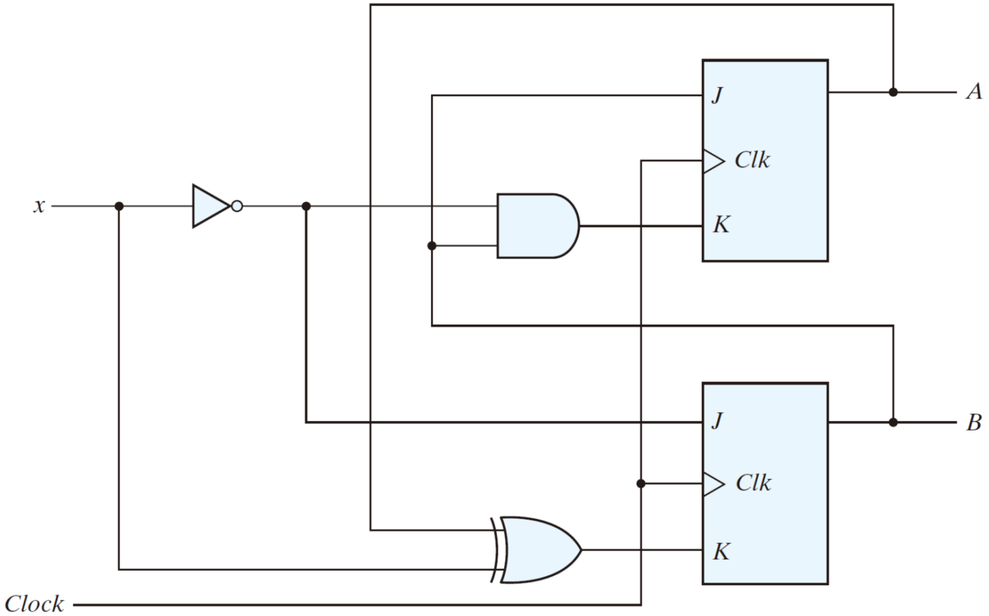


FIGURE 5.18
Sequential circuit with *JK* flip-flop

Characteristic equation for JK flip-flop:

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

- **Characteristic equations of a JK flip-flop.**

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

- **Input equations of the circuit**

$$J_A = B \quad K_A = Bx'$$
$$J_B = x' \quad K_B = A'x + Ax' = A \oplus x$$

- **State equations of the circuit**

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

Analysis with JK Flip-Flop

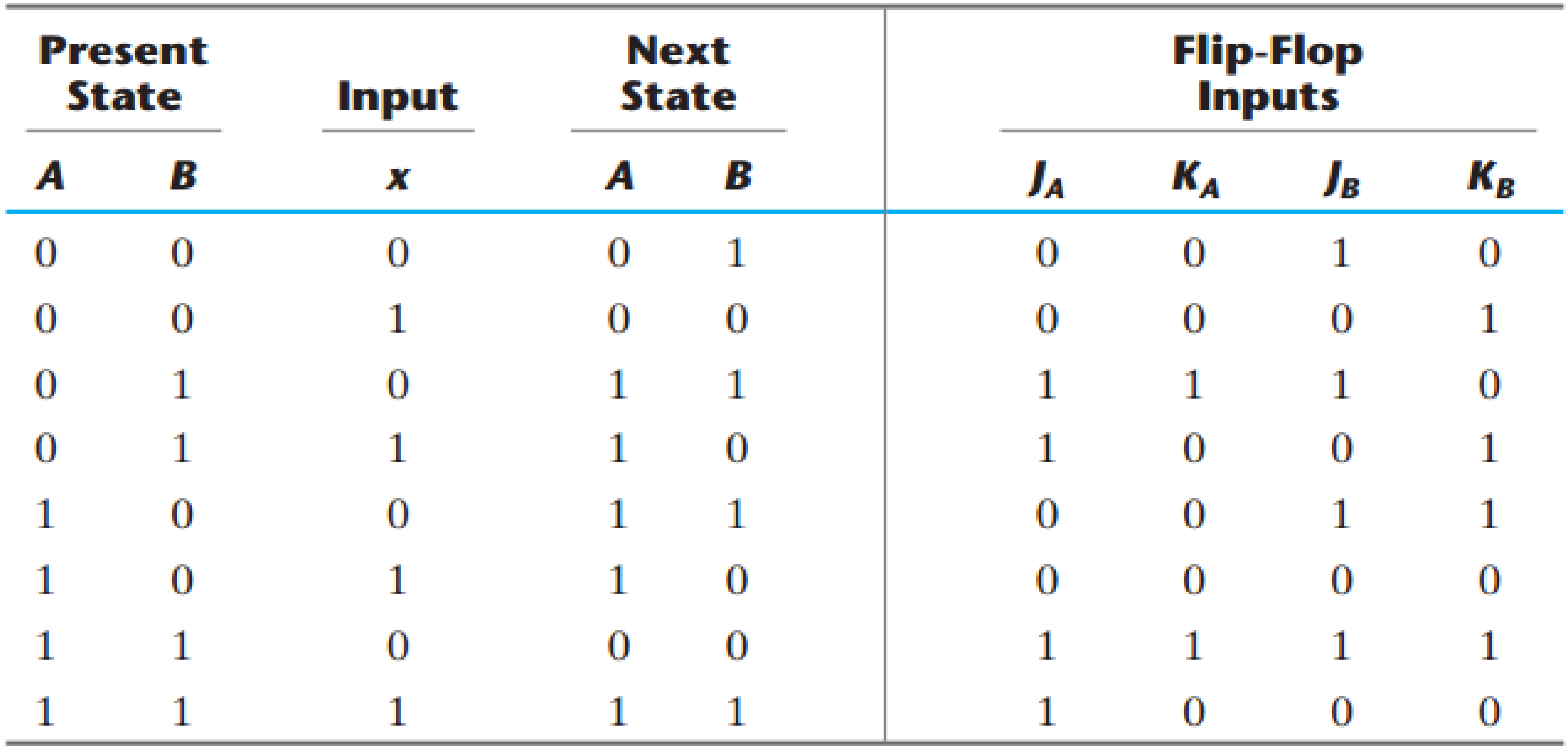
$$A(t+1) = BA' + (Bx')'A = A'B + AB' + Ax$$

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

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

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

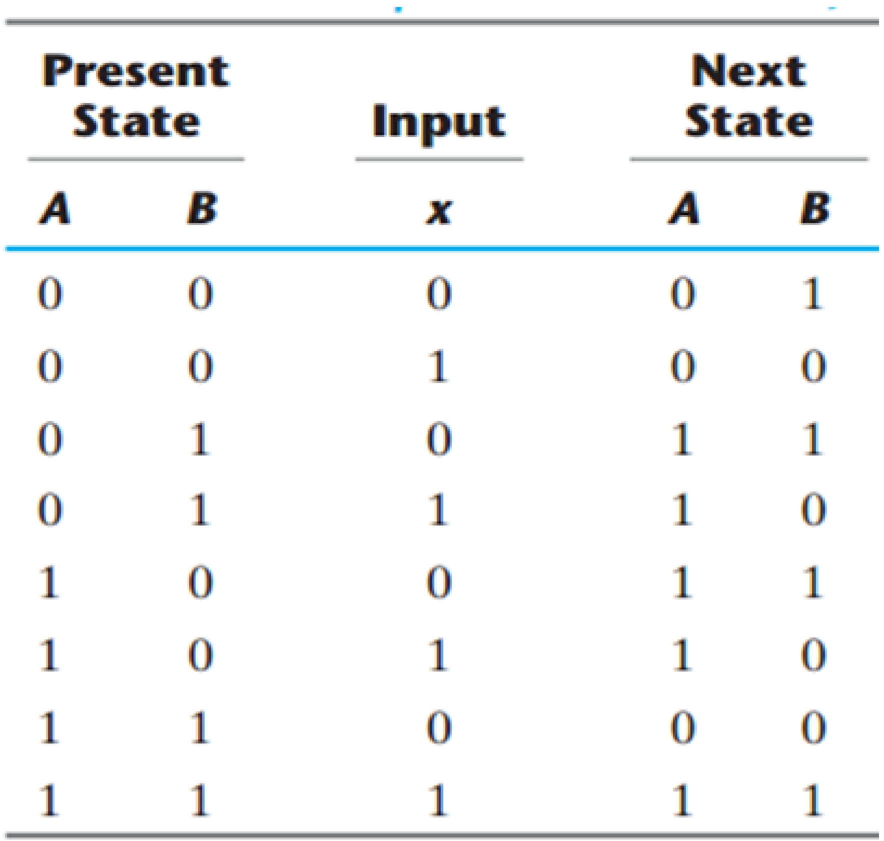
State Table for Sequential Circuit with JK Flip-Flops



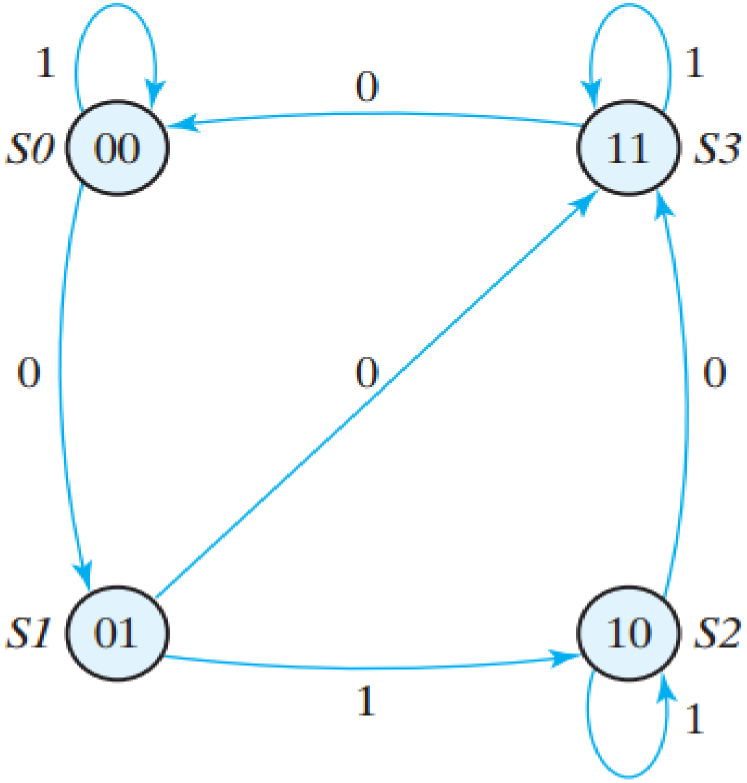
Present State		Input	Next State		Flip-Flop Inputs			
A	B	x	A	B	J_A	K_A	J_B	K_B
0	0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0	1
0	1	0	1	1	1	1	1	0
0	1	1	1	0	1	0	0	1
1	0	0	1	1	0	0	1	1
1	0	1	1	0	0	0	0	0
1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	0	0	0

Analysis with JK Flip-Flop

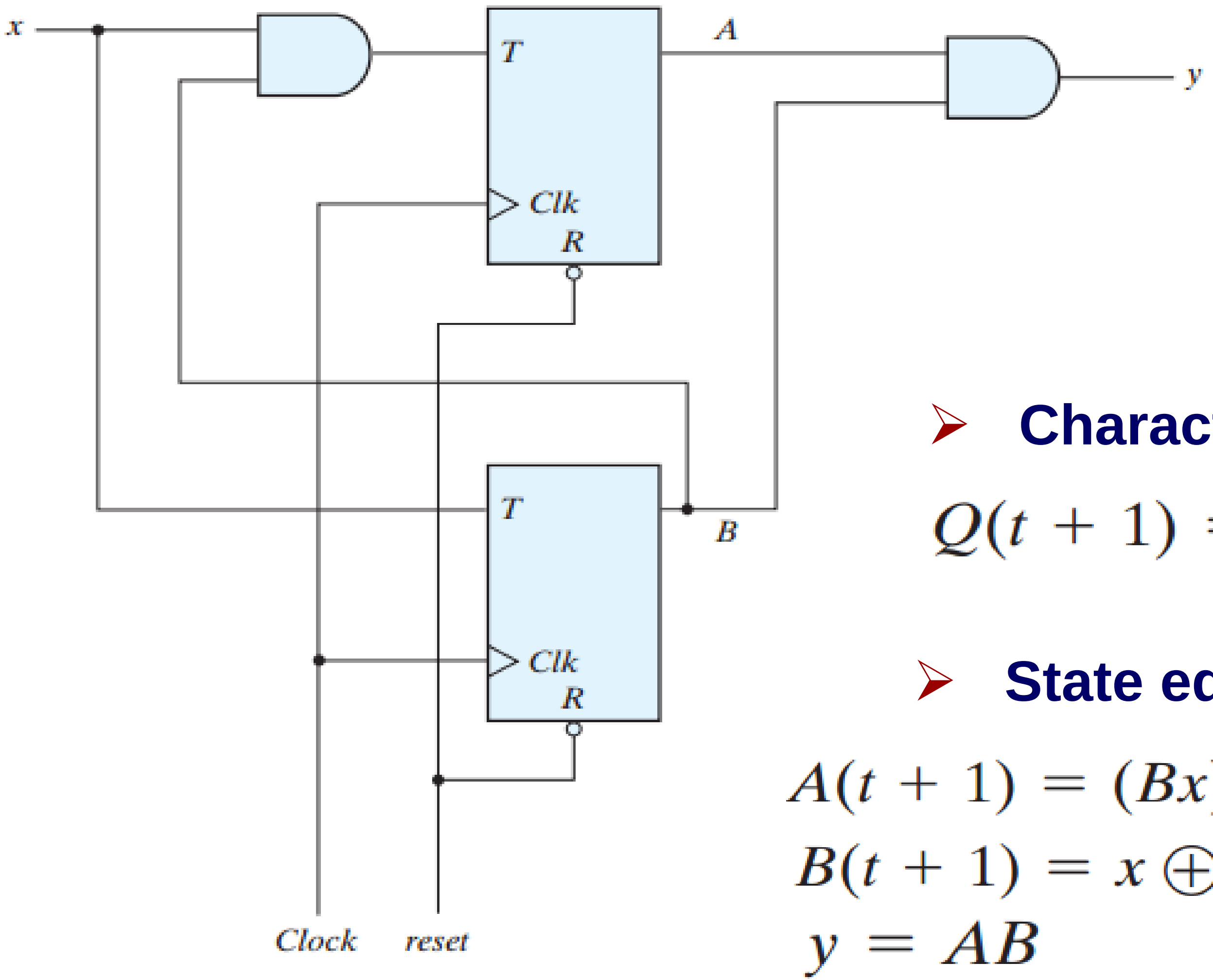
State Diagram of the Circuit



Present State		Input	Next State	
A	*B*	*x*	*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



Analysis with TFF



- **Input equations**

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

- **Characteristic equations**

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

- **State equations of the circuit**

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

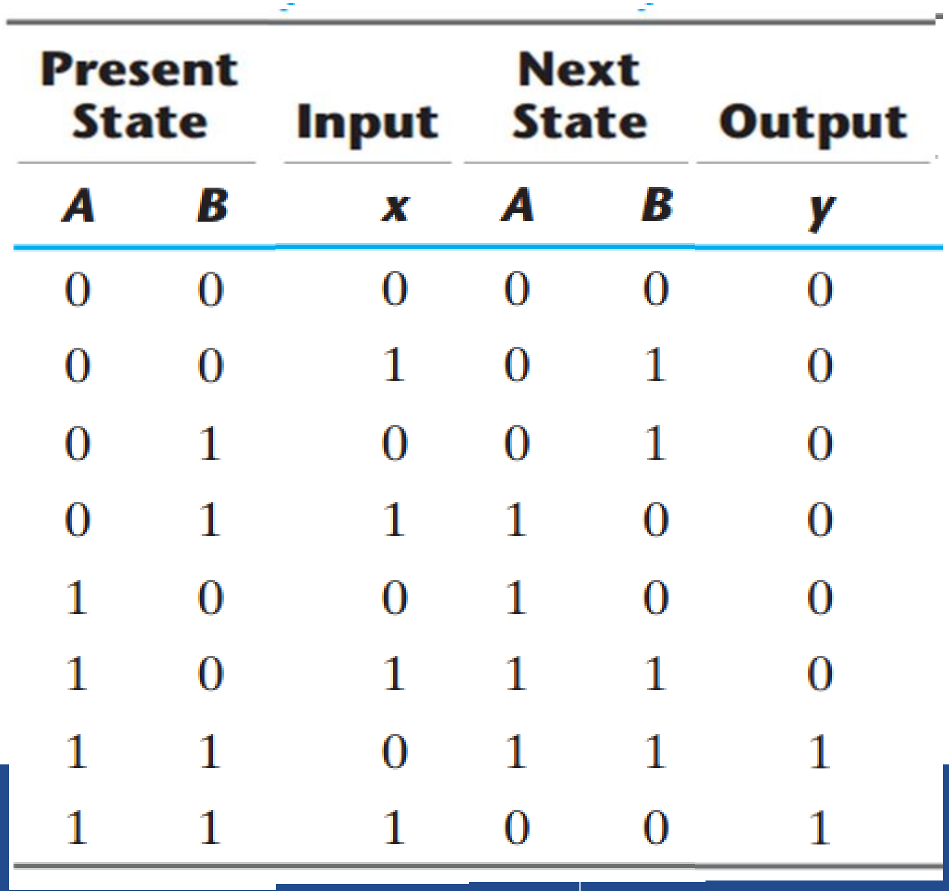
Analysis with TFF

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

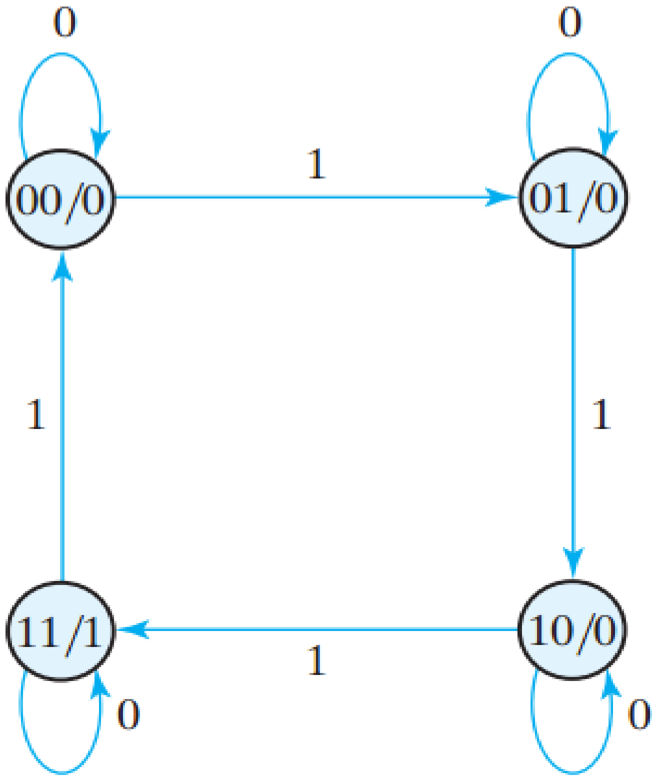
$$B(t+1) = x \oplus B$$

$$y = AB$$

State Table



Present State		Input	Next State		Output
A	*B*	*x*	*A*	*B*	*y*
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



- As long as input *x* is equal to 1, the circuit behaves as a binary counter.
- Here, the output depends on the present state only and is independent of the input.
- The two values inside each circle and separated by a slash are for the present state and output.