Chapter 5: Synchronous Sequential Logic

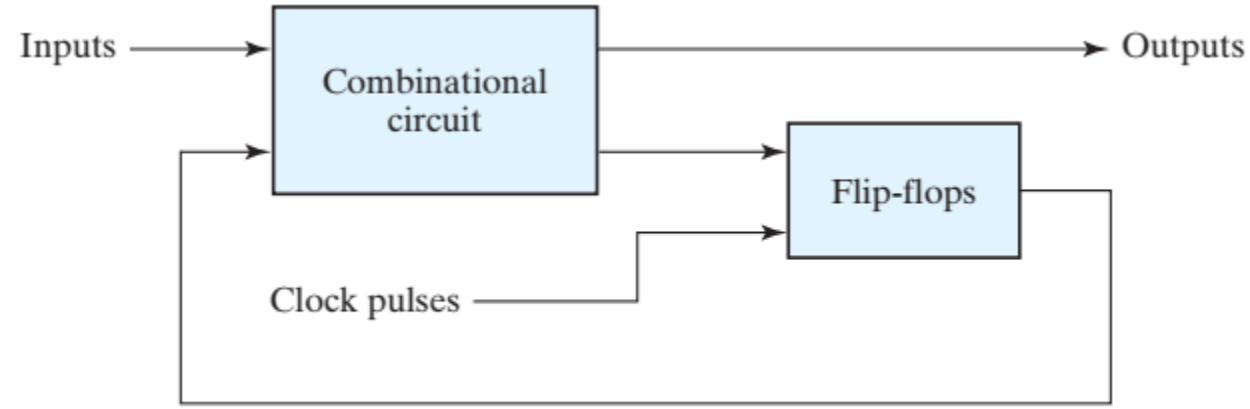
Chapter 6: Registers and Counters

# CSE 231: Digital Logic Design

Section 4
Summer 2020
TnF

# Synchronous Sequential Logic

- A synchronous sequential circuit is a system whose behavior can be defined from the knowledge of its signals at discrete instants of time.
- A synchronous sequential circuit employs signals that affect the storage elements at only discrete instants of time.

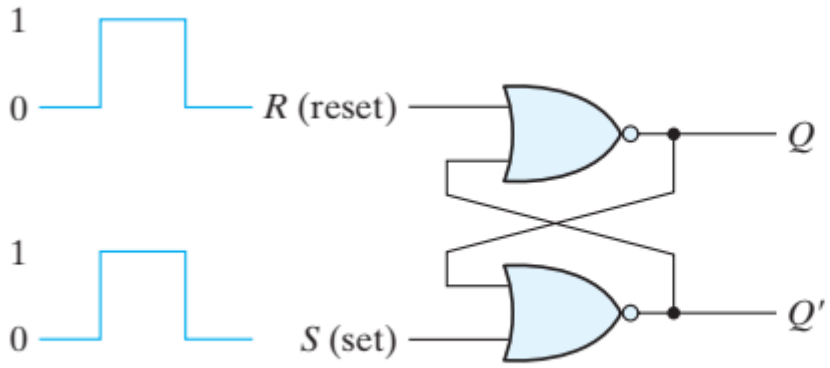


(a) Block diagram

- The storage elements (memory) used in clocked sequential circuits are called flipflops. A flip-flop is a binary storage device capable of storing one bit of information. In a stable state, the output of a flip-flop is either 0 or 1.

# STORAGE ELEMENTS : LATCHES

- A storage element in a digital circuit can maintain a binary state indefinitely.
- Storage elements that operate with signal levels are referred to as latches; those controlled by a clock transition are flip-flops.
- ***SR* Latch**



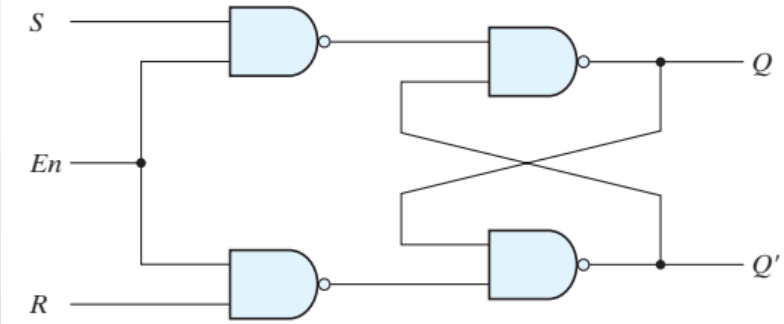

(a) Logic diagram

| $S$ | $R$ | $Q$ | $Q'$ | |
|---|---|---|---|---|
| 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

# STORAGE ELEMENTS : Flipflop

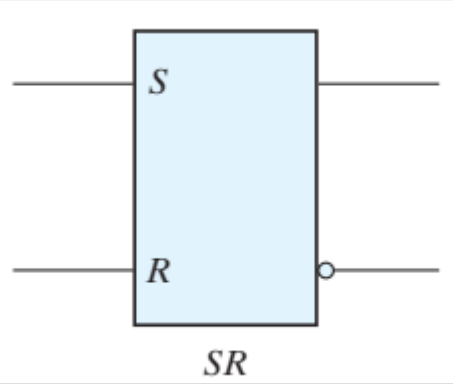
- The state of a latch or flip-flop is switched by a change in the control input. This momentary change is called a trigger, and the transition it causes is said to trigger the flip-flop.
- SR Flipflop



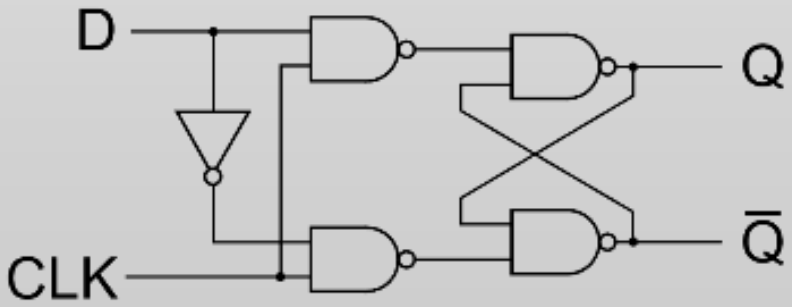

(a) Logic diagram

| *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

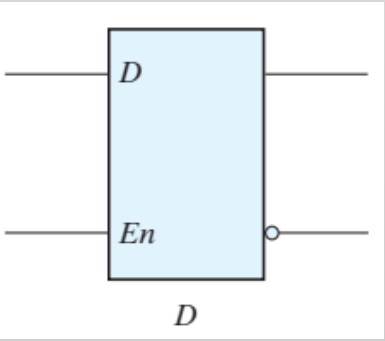



- D flipflop



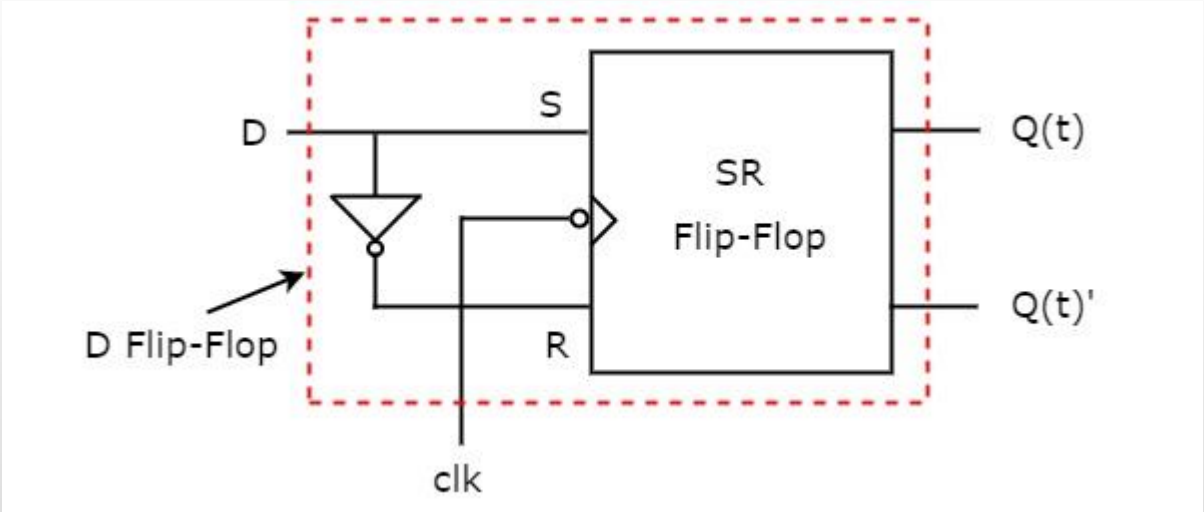

| *En* | *D* | Next state of *Q* |
|---|---|---|
| 0 | X | No change |
| 1 | 0 | *Q* = 0; reset state |
| 1 | 1 | *Q* = 1; set state |

(b) Function table



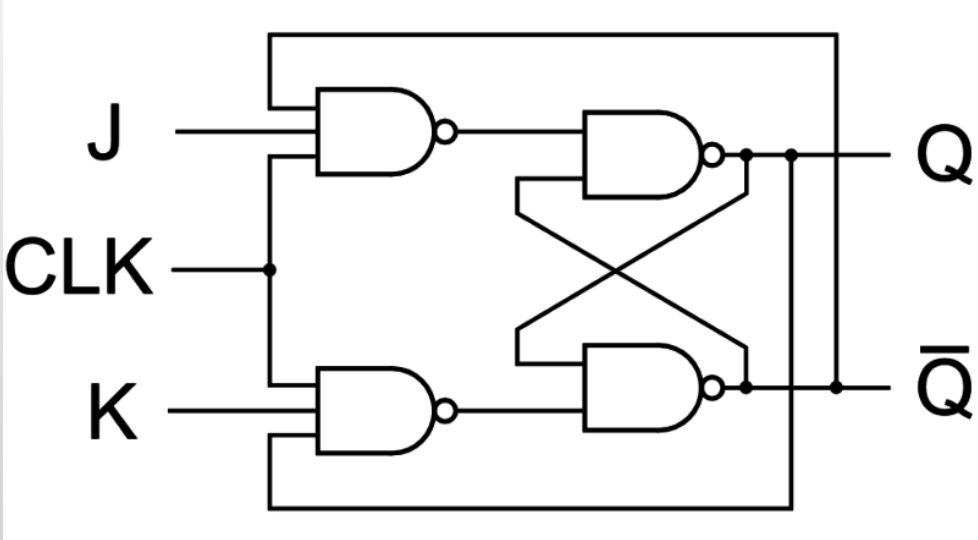
# D flipflop from SR flipflop

- The input is connected to S input and inverted to R input.



# J-K flipflop

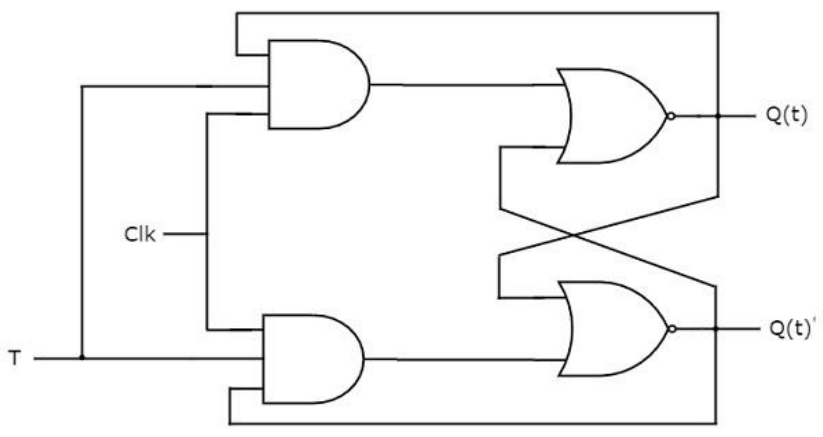
- This circuit has two inputs J & K and two outputs Q(t) & Q(t)′. The operation of JK flip-flop is similar to SR flip-flop. Here, we considered the inputs of SR flip-flop as S = J Q(t)′ and R = KQ(t) in order to utilize the modified SR flip-flop for 4 combinations of inputs.
- The following table shows the Logic diagram & state table of JK flip-flop.



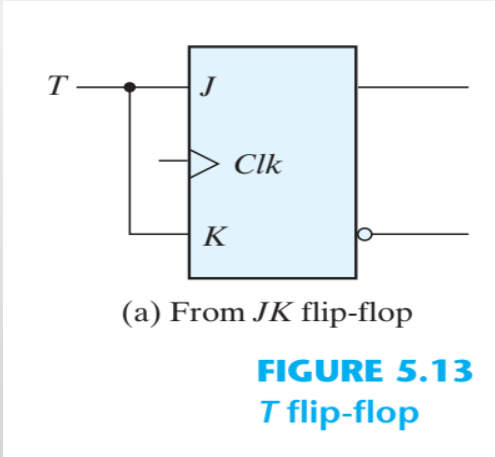

| J | K | Q $t+1$ |
|---|---|---|
| 0 | 0 | Q $t$ |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | Q $t$ ' |

# T flipflop (Toggle flipflop)

- T flip-flop is the simplified version of JK flip-flop. It is obtained by connecting the same input 'T' to both inputs of JK flip-flop. It operates with only positive clock transitions or negative clock transitions. The circuit diagram of T flip-flop is shown in the following figure.
- This circuit has single input T and two outputs Qt & Qt'. The operation of T flip-flop is same as that of JK flip-flop last (11) combination.




| Inputs | Present State | Next State |
|---|---|---|
| T | Qt | Qt+1 |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

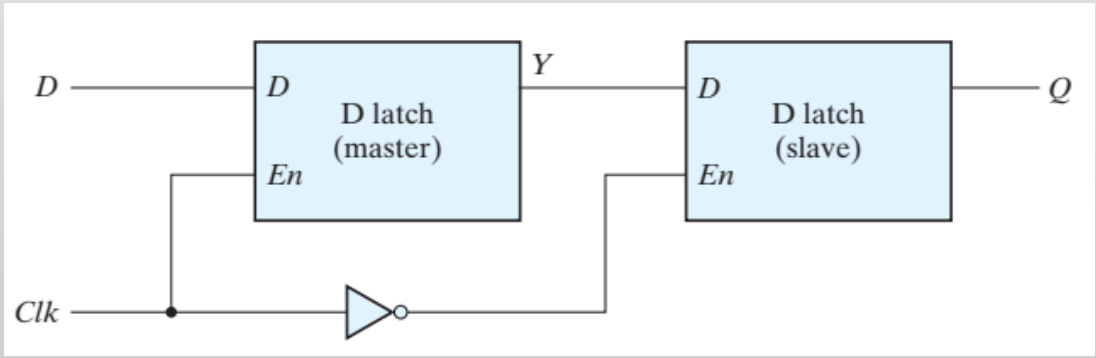



(a) From *JK* flip-flop

**FIGURE 5.13**
***T* flip-flop**

# Master–slave D Flip-Flop

- The behavior of the master–slave flip-flop just described dictates that (1) the output may change only once, (2) a change in the output is triggered by the negative edge of the clock, and (3) the change may occur only during the clock's negative level.
- The value that is produced at the output of the flip-flop is the value that was stored in the master stage immediately before the negative edge occurred.
- A change in the output of the flip-flop can be triggered only by and during the transition of the clock from 1 to 0.



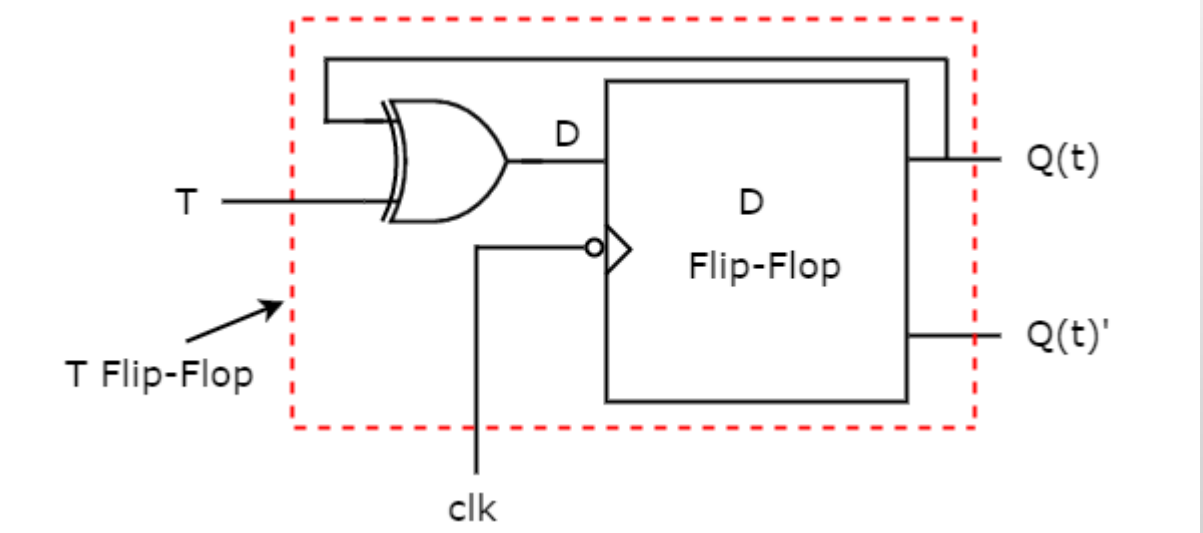
# Flip-flop conversion: D to T

- From the above table, we can directly write the Boolean function of D as below.

$$D=T \oplus Q(t)$$

- So, we require a two input Exclusive-OR gate along with D flip-flop. The circuit diagram of T flip-flop is shown in the following figure.

| T flip-flop input | Present State | Next State | D flip-flop input |
|---|---|---|---|
| **T** | **Q**t() | **Q(**t+1) | **D** |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |

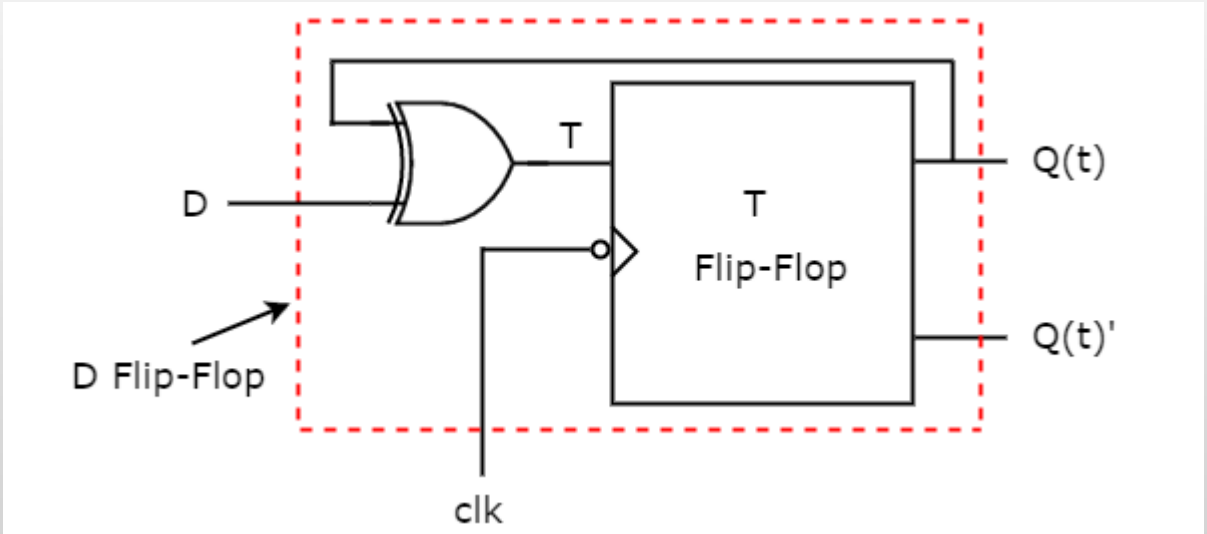


# Flip-flop conversion: T to D

- Here, the given flip-flop is T flip-flop and the desired flip-flop is D flip-flop. Therefore, consider the characteristic table of D flip-flop and write down the excitation values of T flip-flop. From the bellow table, we can directly write the Boolean function of T as below.

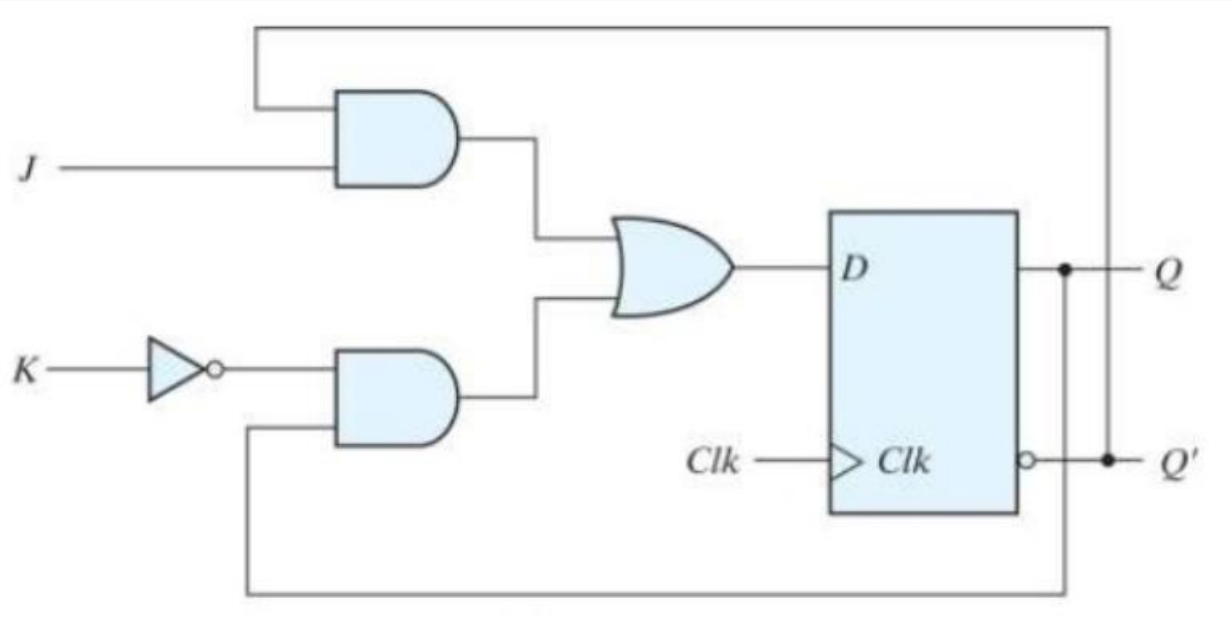
$$T=D \oplus Q(t)$$

| D flip-flop input | Present State | Next State | T flip-flop input |
|---|---|---|---|
| **D** | **Q**tt | **Q**t+1t+1 | **T** |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 |



# Flip-flop conversion: D to JK

- The JK flip-flop is a D flip-flop with gates as shown:



# Flip-flops characteristic equations

- Characteristic equations can be derived from characteristic table:

**D flip-flop:** $Q(t + 1) = D$

Which states that the next state of the output is simply equal to the input D in the present state

**JK flip-flop:** $Q(t + 1) = JQ' + K'Q$

Where Q is the value of the flip-flop prior to the application of the clock edge.

**T flip-flop:** $Q(t + 1) = T \oplus Q = TQ' + T'Q$

Not

# Flip-flop excitation tables

- During Sequential Circuit design, we need tables that lists the required input combinations for a given change in state. such table is called fliplflop Excitation table.

| SR Flip-flop | | | |
|---|---|---|---|
| Q(t) | Q(t+1) | S | R |
| 0 | 0 | 0 | X |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | X | 0 |

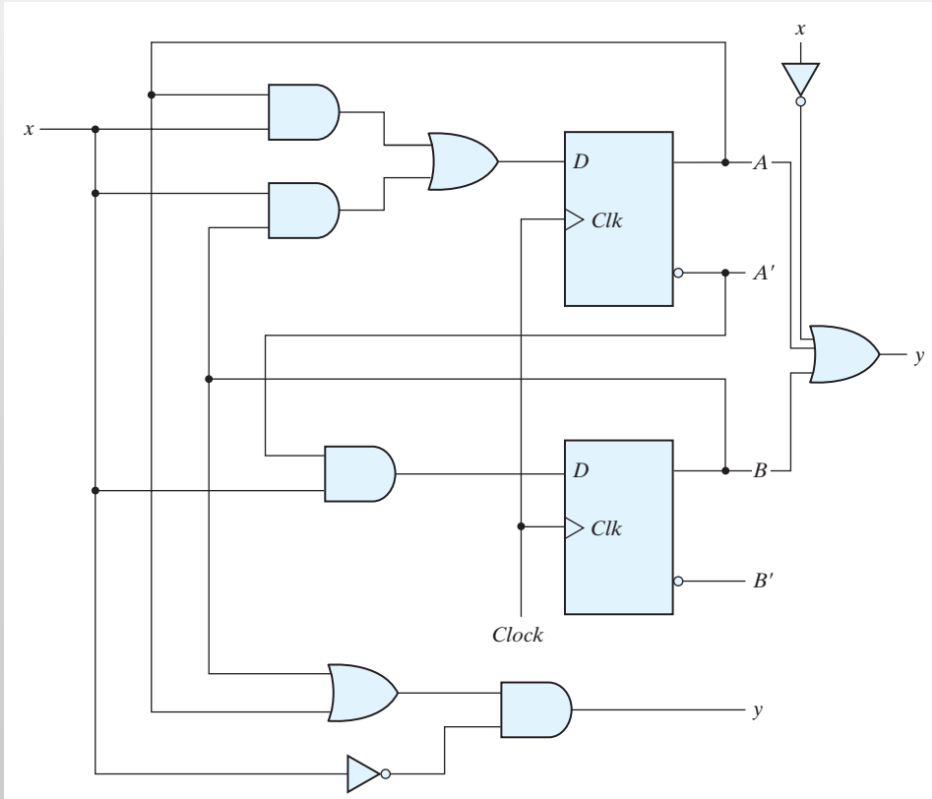
| D Flip-flop | | |
|---|---|---|
| Q(t) | Q(t+1) | DR |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

| JK flip-flop | | | |
|---|---|---|---|
| Q(t) | Q(t+1) | J | K |
| 0 | 0 | 0 | x |
| 0 | 1 | 1 | x |
| 1 | 0 | x | 1 |
| 1 | 1 | x | 0 |

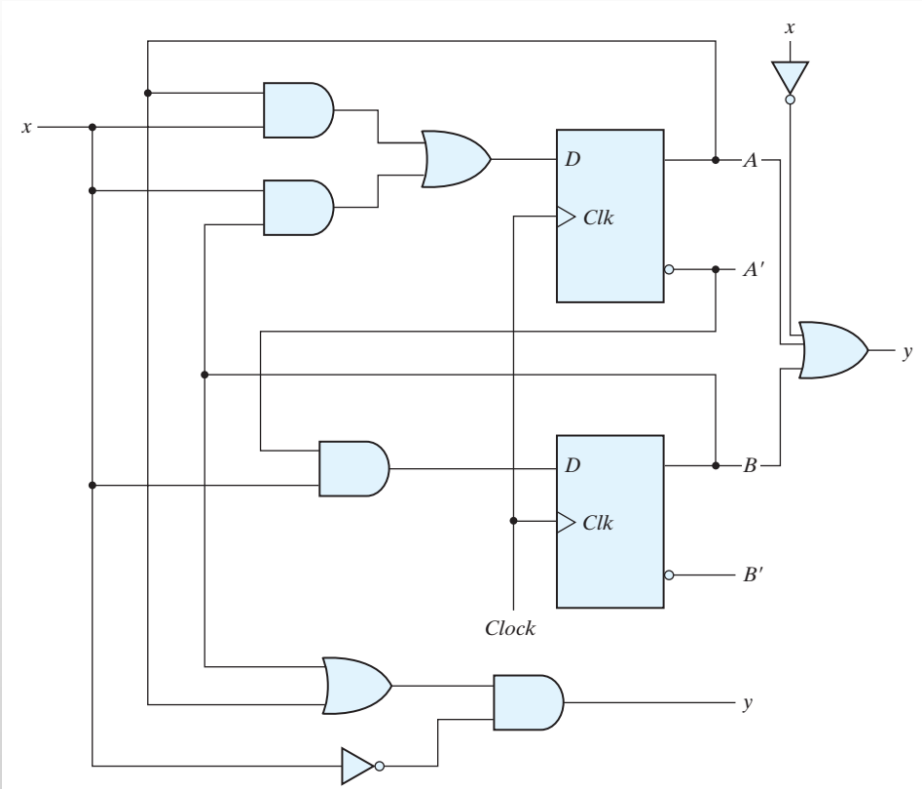
| T flip-flop | | |
|---|---|---|
| Q(t) | Q(t+1) | DR |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

# Analysis of a Clocked Sequential Circuit

- Analysis of Clocked Sequential Circuits consists of obtaining a table or diagram for the time sequence of inputs, outputs, and internal states.
- It is also possible to write a Boolean expression that describes the behavior of the sequential circuit.
- A logic diagram is recognized as a clocked sequential circuit if it contains flip-flops (of any type) with clock inputs. Example is given bellow:



# Analysis of a Clocked Sequential Circuit




Example of Sequential Circuit 1

- 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. The state equations/input equation for the example on the left are:

$$D_A = A(t)x + B(t)x$$
$$D_B = A'(t)x$$

- The present-state equation of the output can be expressed as:

$$A(t+1) = A(t)x + B(t)x \quad [= D_A]$$
$$B(t+1) = A'(t)x \quad [= D_B]$$
$$y(t) = A(t) + B(t) + x'$$

The (t+1) represent next state, and (t) represents present state.

- As the example circuit used "D" flipflop, and the characteristic of it is:

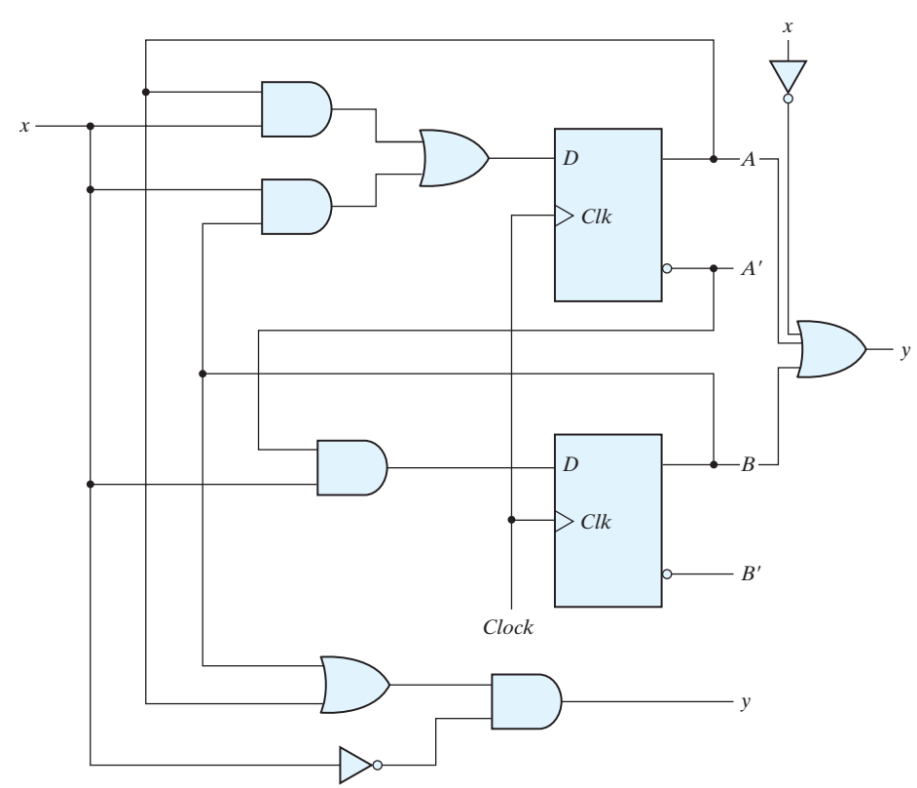
$$D = A(t+1)$$

- Thus state input and output equations are the same:

$$A(t+1) = D_A$$
$$B(t+1) = D_B$$

For other flipflops the input and output equations may differ.

# Analysis of a Clocked Sequential Circuit



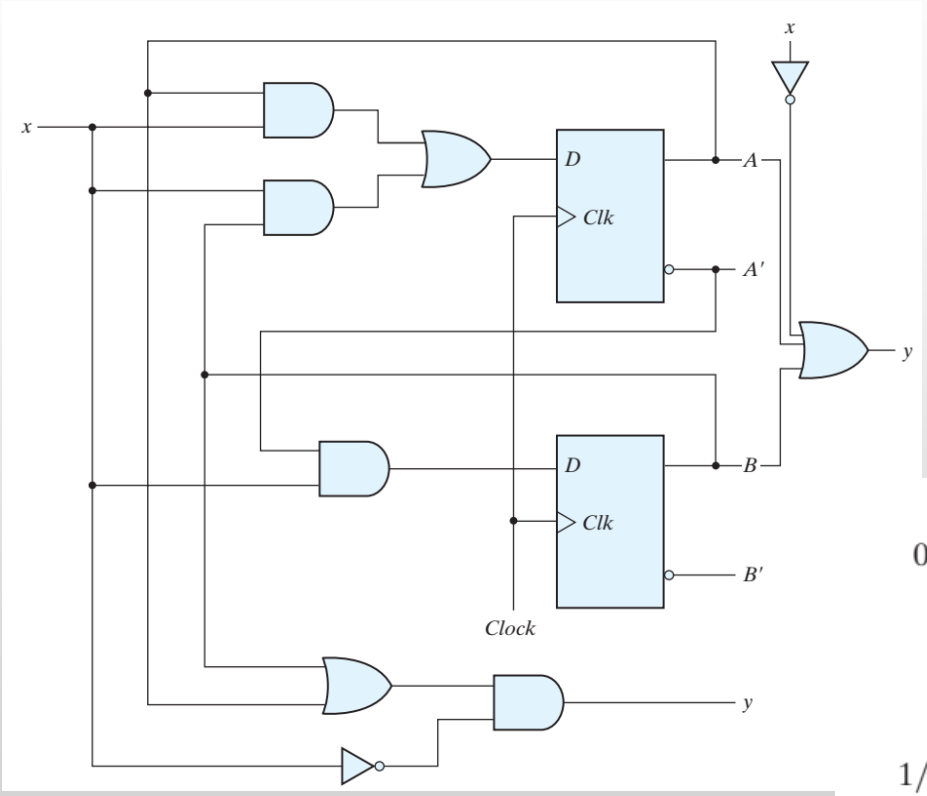

Example of Sequential Circuit 1

- The time sequence of inputs, outputs, and flip-flop states can be enumerated in a state table (sometimes called a transition table).
- The table consists of four sections labeled present state, input, next state, and output.
- The state table of the example is shown bellow:

State Table/Characteristic Table

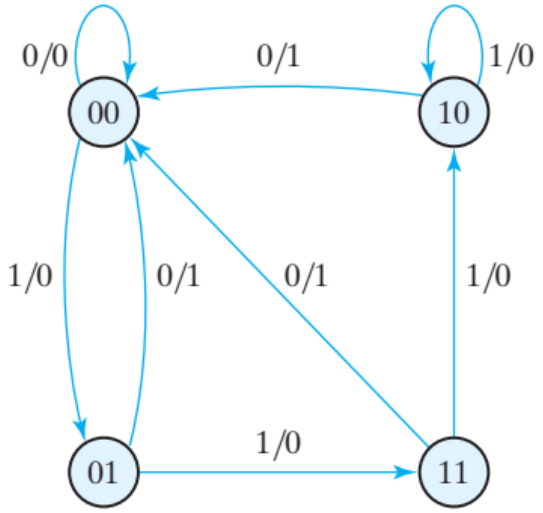
| Present State | | Input | Next State | | Output |
|---|---|---|---|---|---|
| A | B | x | A | B | y |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 |

# Analysis of a Clocked Sequential Circuit




Example of Sequential Circuit 1

- The information available in a state table can be represented graphically in the form of a state diagram. In this type of diagram, a state is represented by a circle, and the (clock-triggered) transitions between states are indicated by directed lines connecting the circles.




State diagram

State Table/Characteristic Table

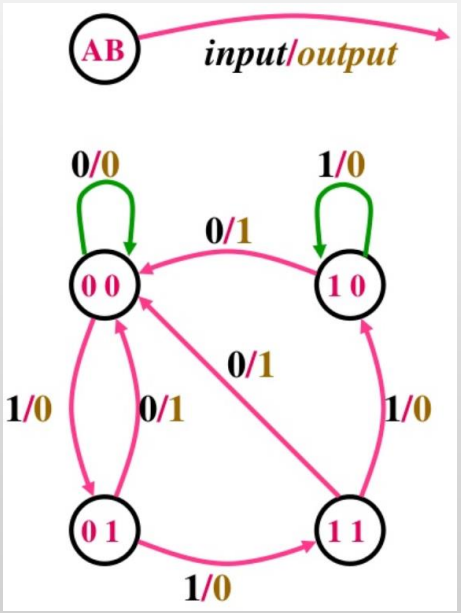
| Present State | | Input | Next State | | Output |
|---|---|---|---|---|---|
| A | B | x | A | B | y |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 |

# Analysis of a Clocked Sequential Circuit (cont.)

- For a given Sequential circuit the analysis steps will include:

  Circuit diagram ⟶ Equations ⟶ State table ⟶ State diagram

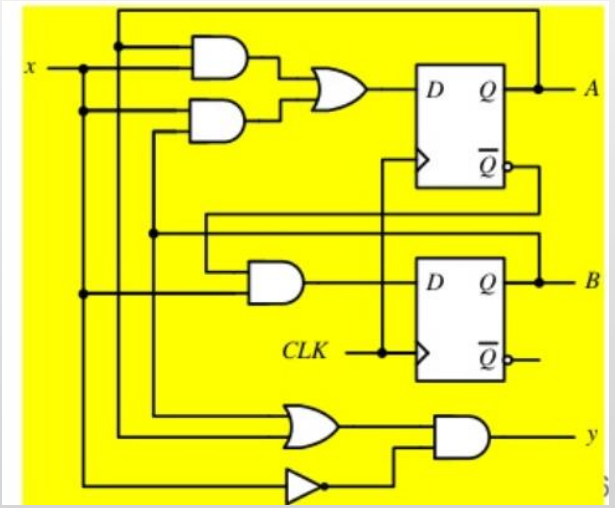
- For a given state diagram, the steps will be other way around. As example of analysis of sequential circuit from state diagram is shown bellow:




| 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 |

Input
$D_A = A(t)x + B(t)x$
$D_B = A'(t)x$

Output equation:
$y(t) = A(t) + B(t) + x'$



# Analysis of a Clocked Sequential Circuit (cont.): Example 2

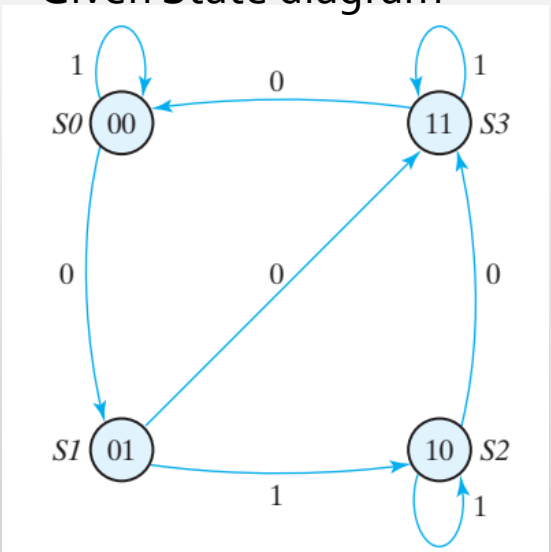
- For a given state table with JK flip-flop.

*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 |

state diagram from
the given Characteristic table

Given State diagram



# Analysis of a Clocked Sequential Circuit (cont.): Example 2

*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 |

| | B′x′ | B′x | Bx | Bx′ |
|---|---|---|---|---|
| A′ | 0 | 0 | 1 | 1 |
| A | 0 | 0 | 1 | 1 |

$J_A = B$

| | B′x′ | B′x | Bx | Bx′ |
|---|---|---|---|---|
| A′ | 0 | 0 | 0 | 1 |
| A | 0 | 0 | 0 | 1 |

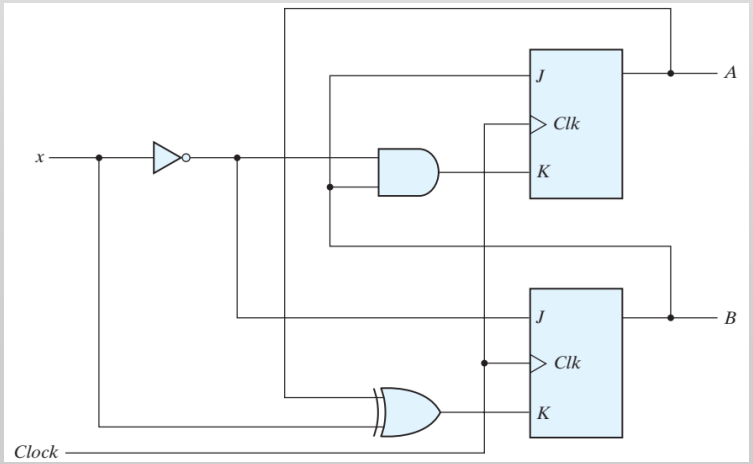
$k_A = Bx'$

| | B′x′ | B′x | Bx | Bx′ |
|---|---|---|---|---|
| A′ | 1 | 0 | 0 | 1 |
| A | 1 | 0 | 0 | 1 |

$J_B = x'$

| | B′x′ | B′x | Bx | Bx′ |
|---|---|---|---|---|
| A′ | 0 | 1 | 1 | 0 |
| A | 1 | 0 | 0 | 1 |

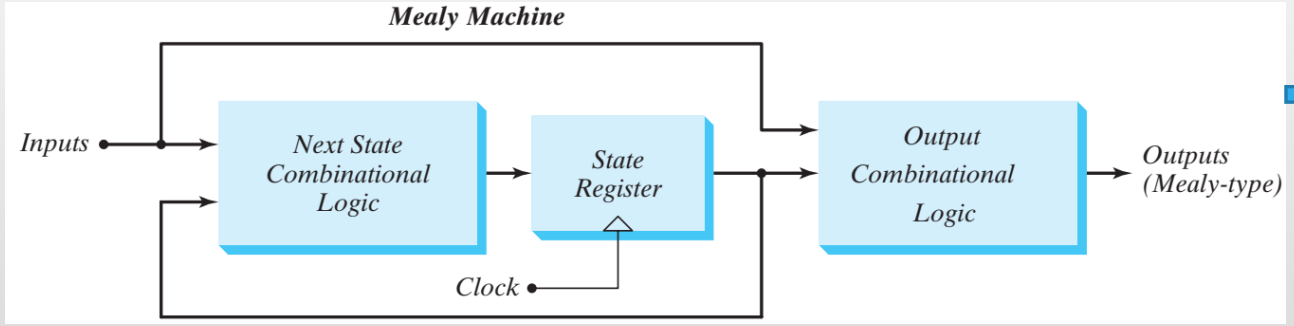
$k_B = A'x + Ax' = A \oplus x$

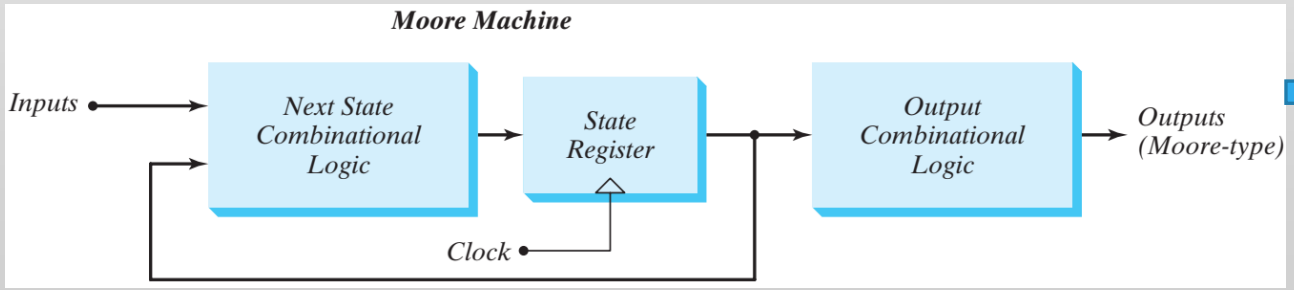


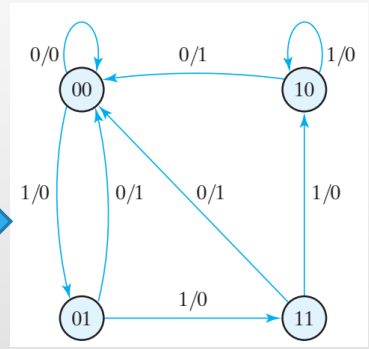

Circuit from derived equations

# Mealy and Moore Models of Finite State Machines

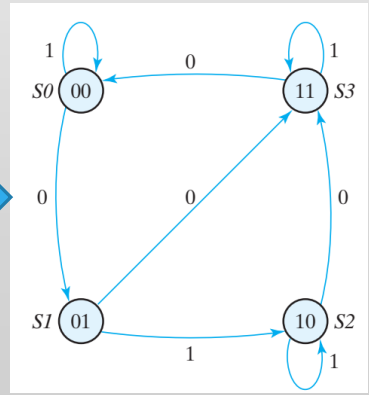
- The most general model of a sequential circuit has inputs, outputs, and internal states.
- It is customary to distinguish between two models of sequential circuits: the Mealy model and the Moore model.
- They differ only in the way the output is generated.



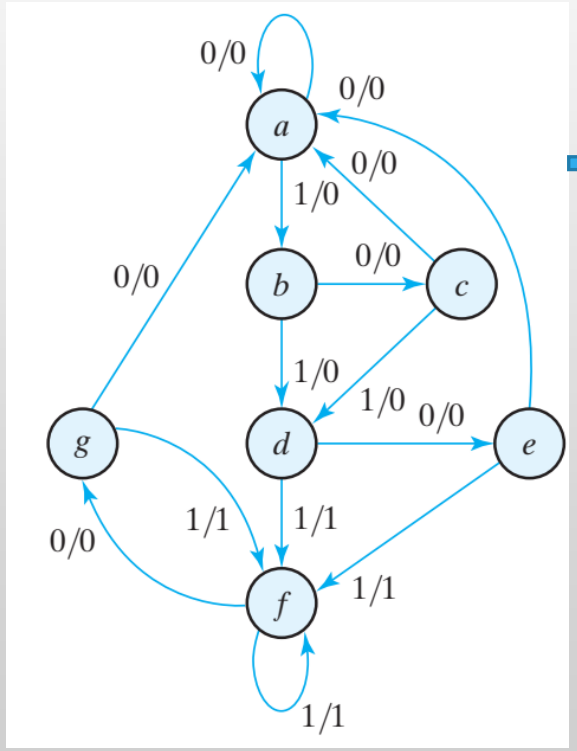

Example 1




Example 2

# STATE REDUCTION AND ASSIGNMENT

- The reduction in the number of flip-flops in a sequential circuit is referred to as the state-reduction problem. State-reduction algorithms are concerned with procedures for reducing the number of states in a state table, while keeping the external input–output requirements unchanged. We will illustrate the state-reduction procedure with an example.




*State Table*

| Present State | Next State $x=0$ | Next State $x=1$ | Output $x=0$ | Output $x=1$ |
|---|---|---|---|---|
| $a$ | $a$ | $b$ | 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 |

*Reducing the State Table*

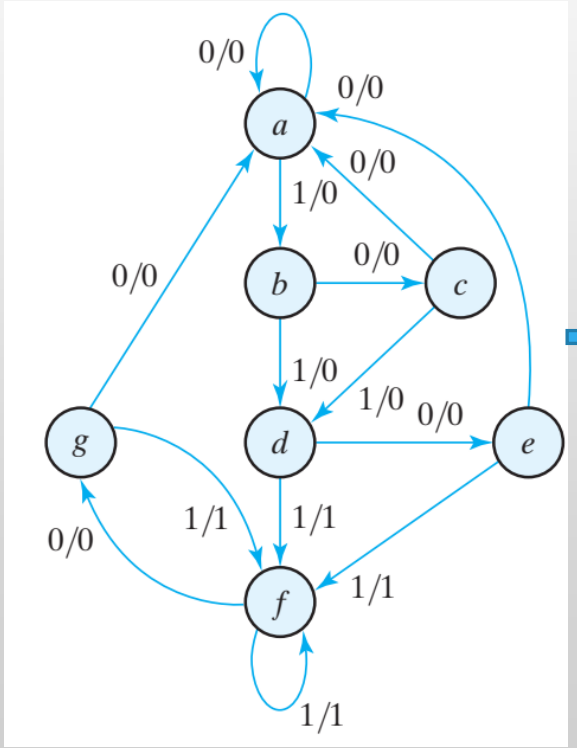
| Present State | Next State $x=0$ | Next State $x=1$ | Output $x=0$ | Output $x=1$ |
|---|---|---|---|---|
| $a$ | $a$ | $b$ | 0 | 0 |
| $b$ | $c$ | $d$ | 0 | 0 |
| $c$ | $a$ | $d$ | 0 | 0 |
| $d$ | $e$ | $f$ | 0 | 1 |
| $e$ | $a$ | $f$ | 0 | 1 |
| $f$ | $e$ | $f$ | 0 | 1 |

*Reduced State Table*

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

# STATE REDUCTION AND ASSIGNMENT

- The reduction in the number of flip-flops in a sequential circuit is referred to as the state-reduction problem. State-reduction algorithms are concerned with procedures for reducing the number of states in a state table, while keeping the external input–output requirements unchanged. We will illustrate the state-reduction procedure with an example.




*State Table*

| Present State | Next State $x=0$ | Next State $x=1$ | Output $x=0$ | Output $x=1$ |
|---|---|---|---|---|
| $a$ | $a$ | $b$ | 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~~ |

*Reducing the State Table*

| Present State | Next State $x=0$ | Next State $x=1$ | Output $x=0$ | Output $x=1$ |
|---|---|---|---|---|
| $a$ | $a$ | $b$ | 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~~ |

*Reduced State Table*

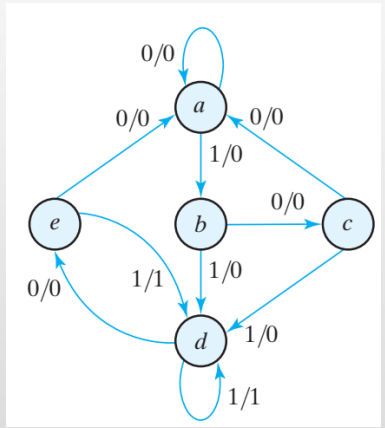
| Present State | Next State $x=0$ | Next State $x=1$ | Output $x=0$ | Output $x=1$ |
|---|---|---|---|---|
| $a$ | $a$ | $b$ | 0 | 0 |
| $b$ | $c$ | $d$ | 0 | 0 |
| $c$ | $a$ | $d$ | 0 | 0 |
| $d$ | $e$ | $d$ | 0 | 1 |
| $e$ | $a$ | $d$ | 0 | 1 |

# STATE REDUCTION AND ASSIGNMENT

- Reduced state table & state diagram
- As we have 5 states after reduction we need minimum 5 binary combinations to represent them.

*Reduced State Table*

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




*Binary State Assignments*

| State | Assignment Binary |
|---|---|
| $a$ | 000 |
| $b$ | 001 |
| $c$ | 010 |
| $d$ | 011 |
| $e$ | 100 |

*Reduced State Table with Binary Assignment 1*

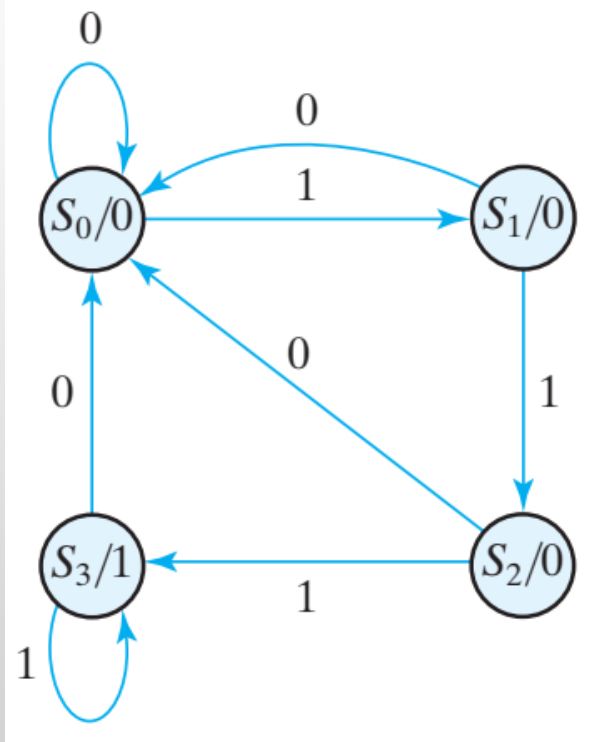
| Present State | Next State $x = 0$ | Next State $x = 1$ | Output $x = 0$ | Output $x = 1$ |
|---|---|---|---|---|
| 000 | 000 | 001 | 0 | 0 |
| 001 | 010 | 011 | 0 | 0 |
| 010 | 000 | 011 | 0 | 0 |
| 011 | 100 | 011 | 0 | 1 |
| 100 | 000 | 011 | 0 | 1 |

# DESIGN PROCEDURE

- The design of a clocked sequential circuit starts from a set of specifications and culminates in a logic diagram or a list of Boolean functions from which the logic diagram can be obtained.
- The procedure for designing synchronous sequential circuits can be summarized by a list of recommended steps:
  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.

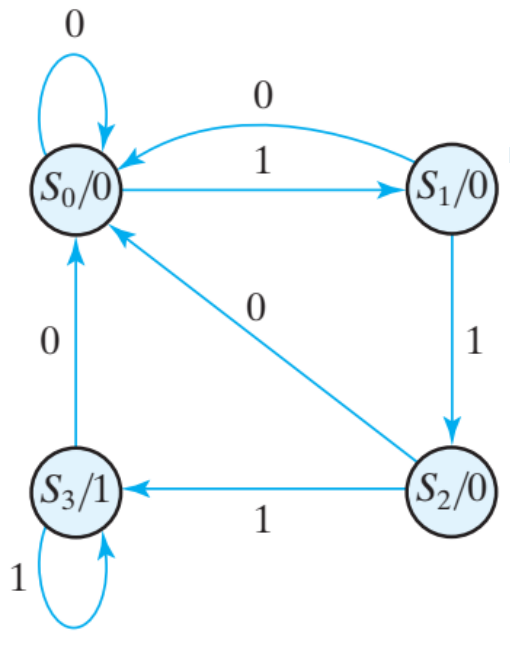
# Design a sequence detector

- Suppose we wish to design a circuit that detects a sequence of three or more consecutive 1's in a string of bits coming through an input line.
- The state diagram for this type of circuit is shown on the right.
- It is derived by starting with state S0, the reset state. If the input is 0,
- the circuit stays in S0, but if the input is 1, it goes to state S1 to
- indicate that a 1 was detected. If the next input is 1, the change is to
- state S2 to indicate the arrival of two consecutive 1's, but if the input
- is 0, the state goes back to S0. The third consecutive 1 sends the circuit to
- state S3. If more 1's are detected, the circuit stays in S3. Any 0 input sends
- the circuit back to S0. In this way, the circuit stays in S3 as long as there
- are three or more consecutive 1's received. This is a Moore model sequential
- circuit, since the output is 1 when the circuit is in state S3 and is 0 otherwise.



# Design a sequence detector

- Once the state diagram has been derived, the rest of the design follows a straight forward.




State Assignment:

$S_0$=00

$S_1$=01

$S_2$=10

$S_3$=11

*State Table for Sequence Detector*

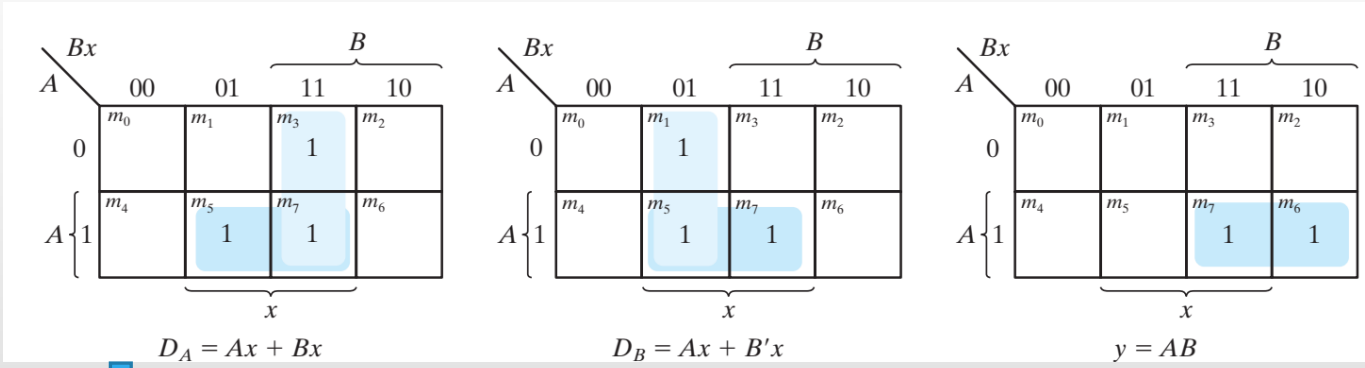
| Present State | | Input | Next State | | Output |
|---|---|---|---|---|---|
| *A* | *B* | *x* | *A* | *B* | *y* |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |

# Design a sequence detector

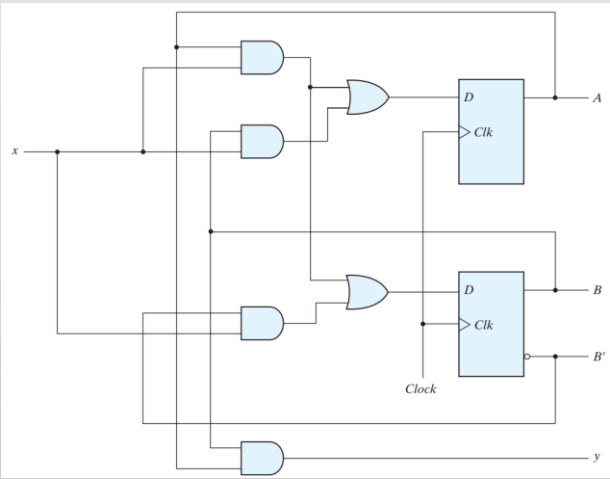
- Once the state diagram has been derived, the rest of the design follows a straight forward. We decided to make the circuit using "D" flipflop.

*State Table for Sequence Detector*

| 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 | 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 |




$$D_A = Ax + Bx$$
$$D_B = Ax + B'x$$
$$y = AB$$




Logic diagram of a Moore-type sequence detector

# Analysis of a Clocked Sequential Circuit: Example 3

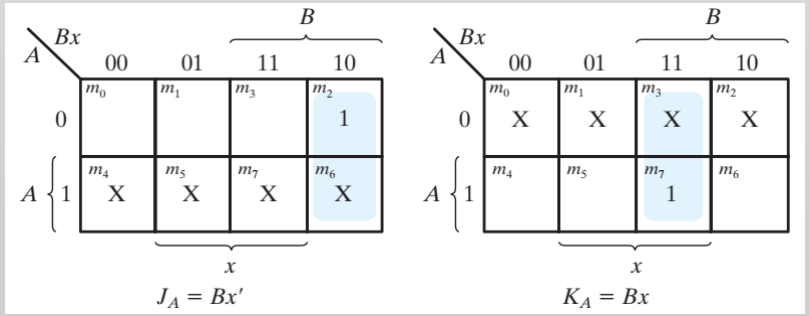
- A state table is given and the sequential circuit has to be build with JK flipflop.

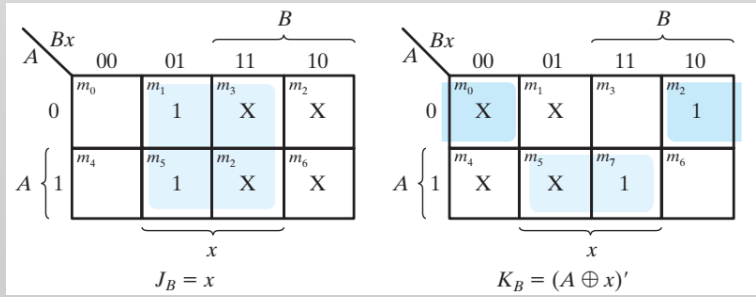
| Present State | | Input | Next State | |
|---|---|---|---|---|
| A | B | x | A | B |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 |

Using excitation table of JK flipflop to get the input equations

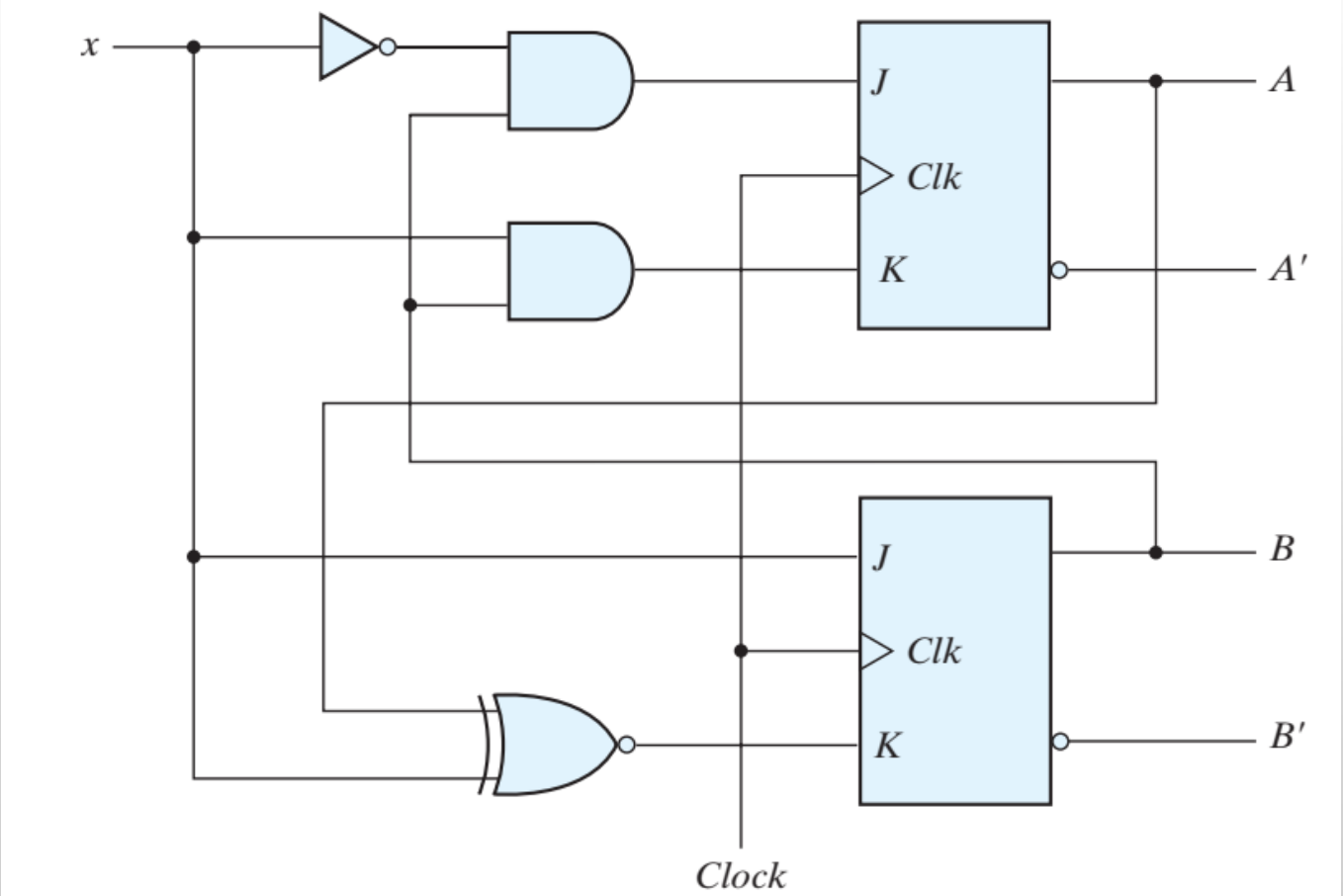
*State Table and JK Flip-Flop Inputs*

| 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 | 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 |



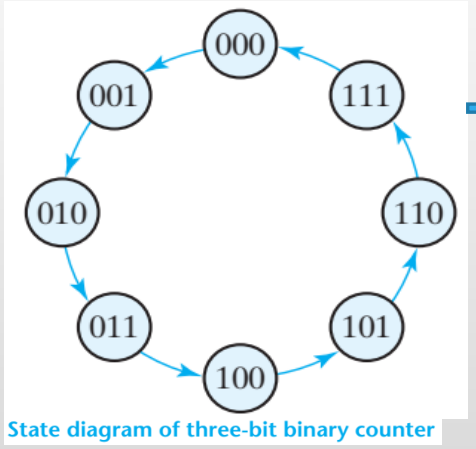



# Analysis of a Clocked Sequential Circuit: Example 3



# Analysis of a Clocked Sequential Circuit: Example 4

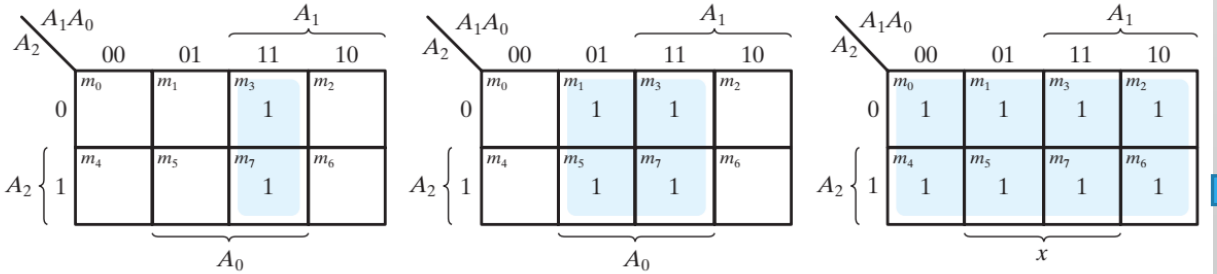
- Designing a binary counter using T flip-flops.
- An n-bit binary counter consists of n flip-flops that can count in binary from 0 to 2n - 1. The state diagram of a three-bit counter is shown

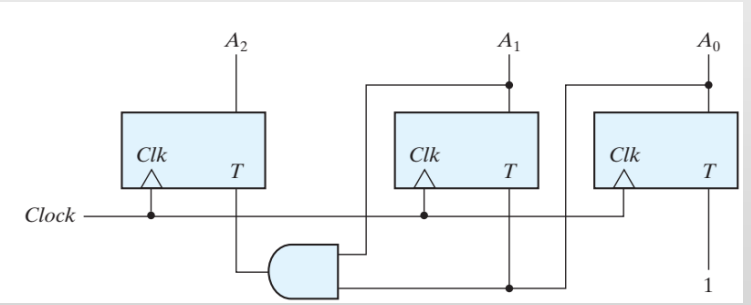


State diagram of three-bit binary counter

*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 |



Logic diagram of three-bit binary counter

# Practice

**5.6** A sequential circuit with two $D$ flip-flops $A$ and $B$, two inputs, $x$ and $y$; and one output $z$ is specified by the following next-state and output equations (HDL—see Problem 5.35):

$$A(t+1) = xy' + xB$$
$$B(t+1) = xA + xB'$$
$$z = A$$

(a) Draw the logic diagram of the circuit.
(b) List the state table for the sequential circuit.
(c) Draw the corresponding state diagram.

**5.9** A sequential circuit has two $JK$ flip-flops $A$ and $B$ and one input $x$. The circuit is described by the following flip-flop input equations:

$$J_A = x \quad K_A = B$$
$$J_B = x \quad K_B = A'$$

(a) Derive the state equations $A(t+1)$ and $B(t+1)$ by substituting the input equations for the $J$ and $K$ variables.
(b) Draw the state diagram of the circuit.

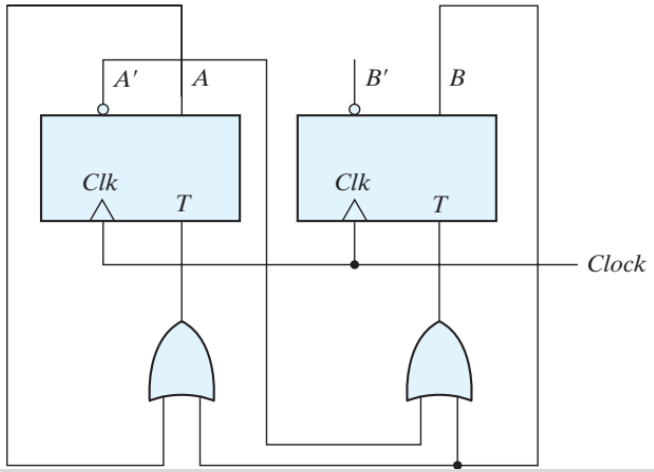
**5.10** A sequential circuit has two $JK$ flip-flops $A$ and $B$, two inputs $x$ and $y$, and one output $z$. The flip-flop input equations and circuit output equation are

$$J_A = Bx + B'y' \qquad K_A = B'xy'$$
$$J_B = A'x \qquad K_B = A + xy'$$
$$z = Ax'y' + Bx'y'$$

(a) Draw the logic diagram of the circuit.
(b) Tabulate the state table.
(c) Derive the state equations for $A$ and $B$.

# Practice

**5.8*** Derive the state table and the state diagram of the sequential circuit shown in Fig
Explain the function that the circuit performs. (HDL—see Problem 5.36.)



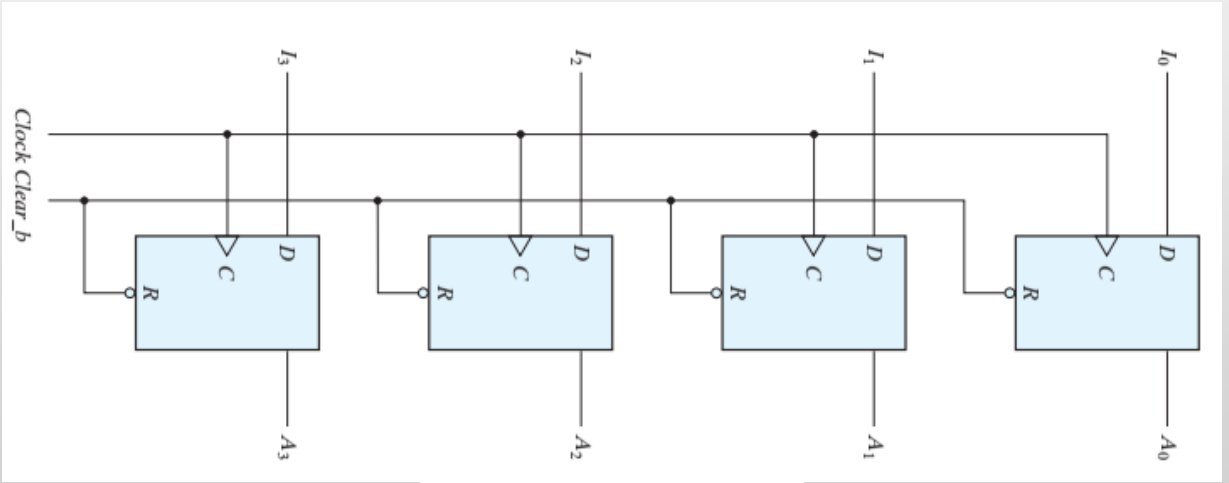

**5.12** For the following state table

| | Next State | | Output | |
|---|---|---|---|---|
| **Present State** | $x = 0$ | $x = 1$ | $x = 0$ | $x = 1$ |
| $a$ | $f$ | $b$ | 0 | 0 |
| $b$ | $d$ | $c$ | 0 | 0 |
| $c$ | $f$ | $e$ | 0 | 0 |
| $d$ | $g$ | $a$ | 1 | 0 |
| $e$ | $d$ | $c$ | 0 | 0 |
| $f$ | $f$ | $b$ | 1 | 1 |
| $g$ | $g$ | $h$ | 0 | 1 |
| $h$ | $g$ | $a$ | 1 | 0 |

(a) Draw the corresponding state diagram.
(b)* Tabulate the reduced state table.
(c) Draw the state diagram corresponding to the reduced state table.

# Registers

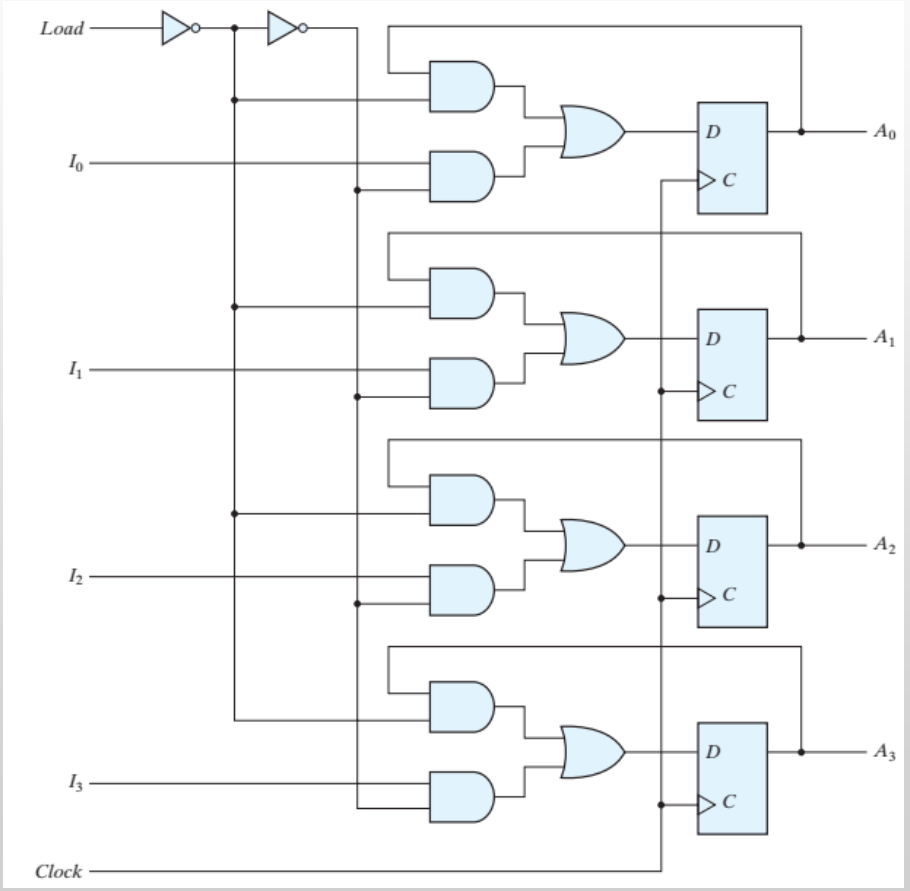
- A register is a group of flip-flops, each one of which shares a common clock and is capable of storing one bit of information. An n-bit register consists of a group of n flip-flops capable of storing n bits of binary information.



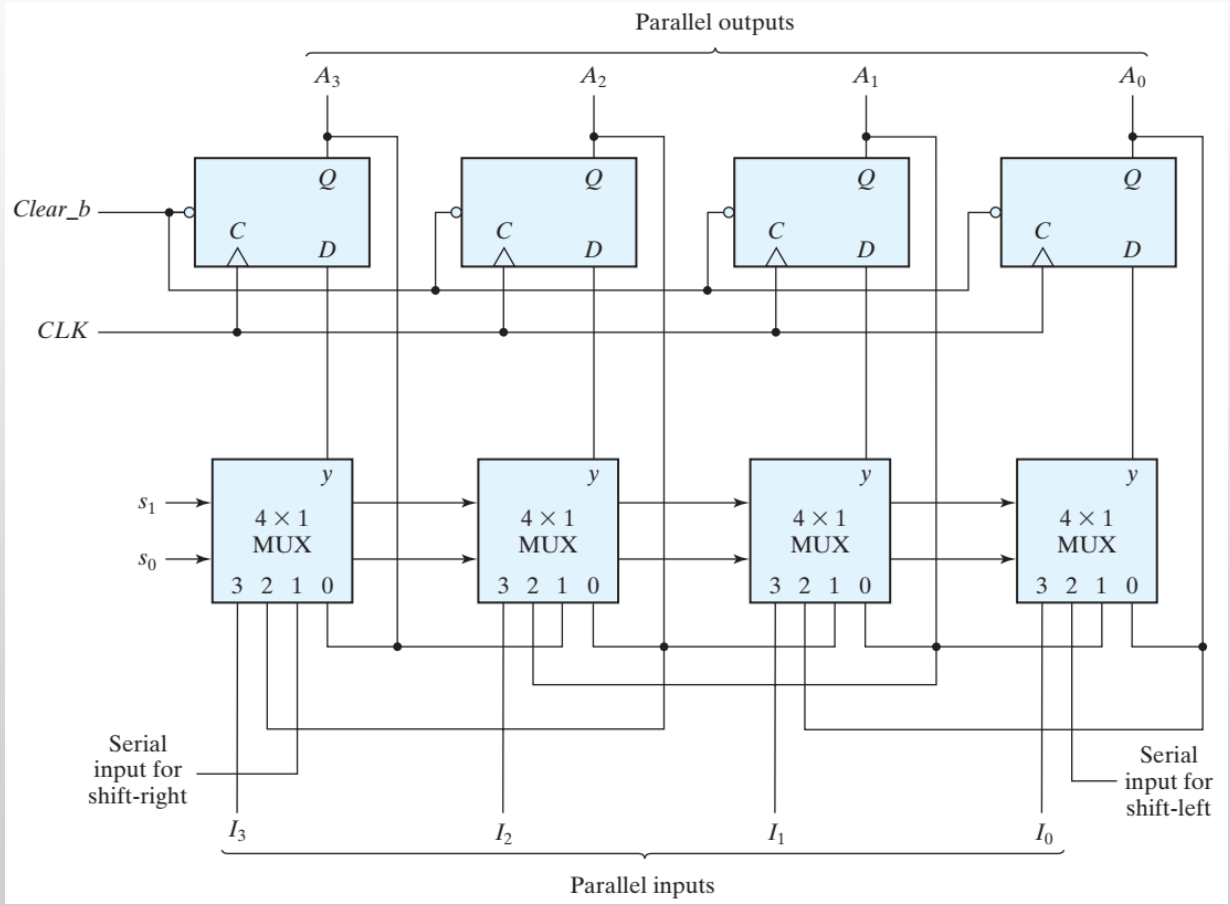

Four-bit register

# Register with Parallel Load

- Registers with parallel load are a fundamental building block in digital systems.



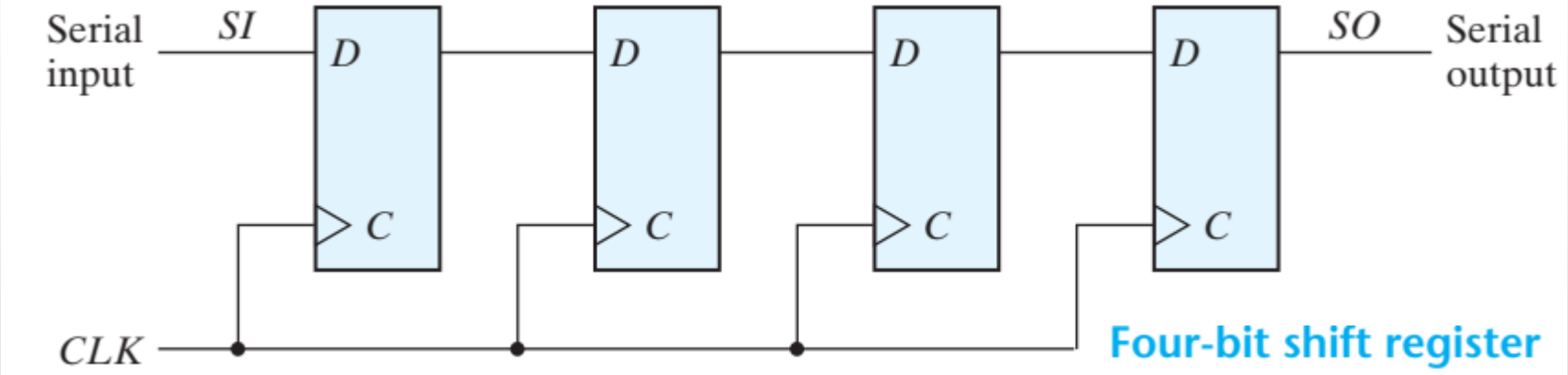
# Four-bit universal shift register



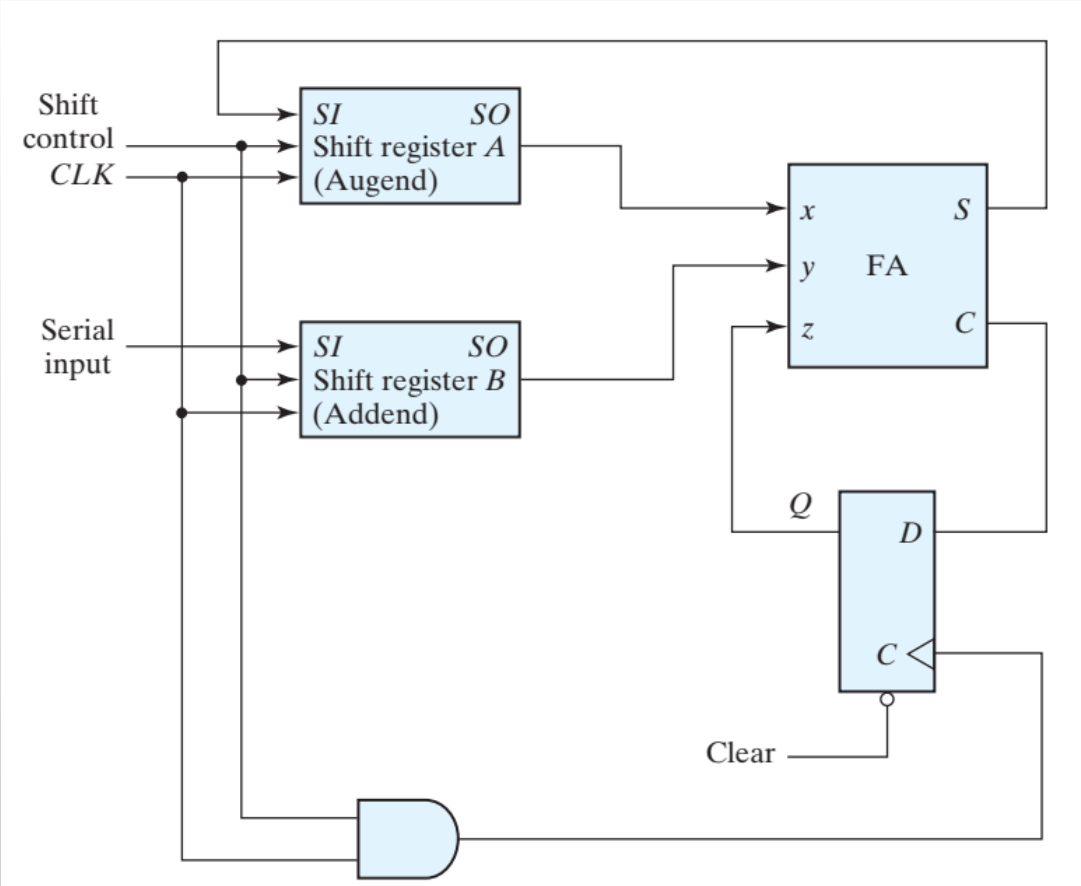

*Function Table for the Register of Fig. 6.7*

| Mode Control | | |
|---|---|---|
| $s_1$ | $s_0$ | Register Operation |
| 0 | 0 | No change |
| 0 | 1 | Shift right |
| 1 | 0 | Shift left |
| 1 | 1 | Parallel load |

# Serial/Shift Register



# Serial Addition



# Counters

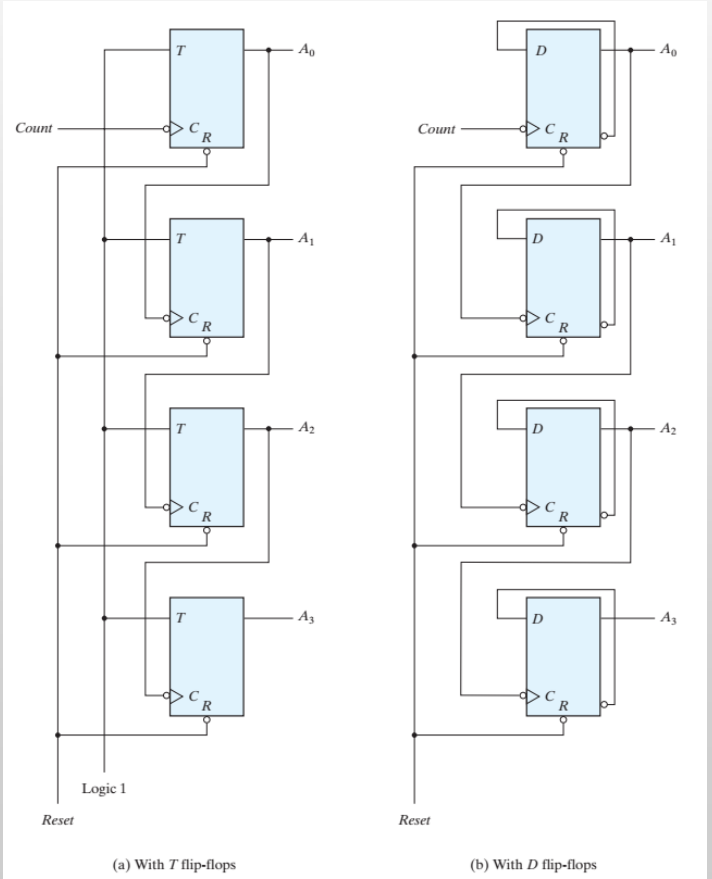
- A register that goes through a prescribed sequence of states upon the application of input pulses is called a counter.
- The input pulses may be clock pulses, or they may originate from some external source and may occur at a fixed interval of time or at random.
- Counters are available in two categories:
  - ripple counters
  - synchronous counters.

# Counter: Ripple

- In a ripple counter, a flip-flop output transition serves as a source for triggering other flip-flops.

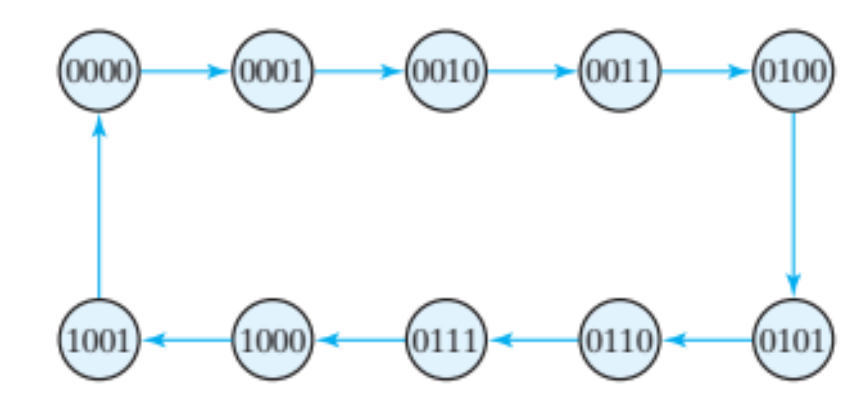
**Binary Count Sequence**

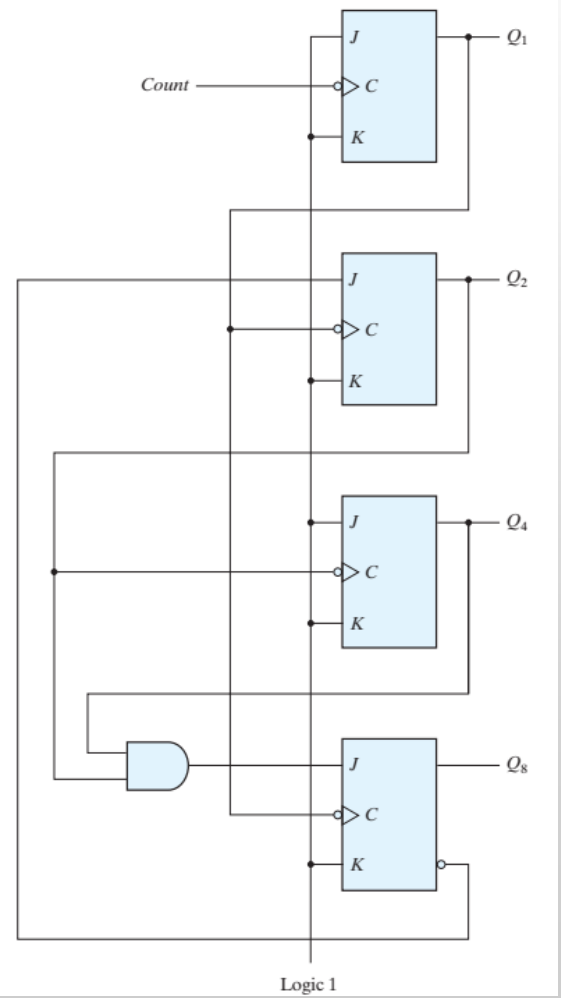
| $A_3$ | $A_2$ | $A_1$ | $A_0$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |



# Counter: BCD Ripple Counter

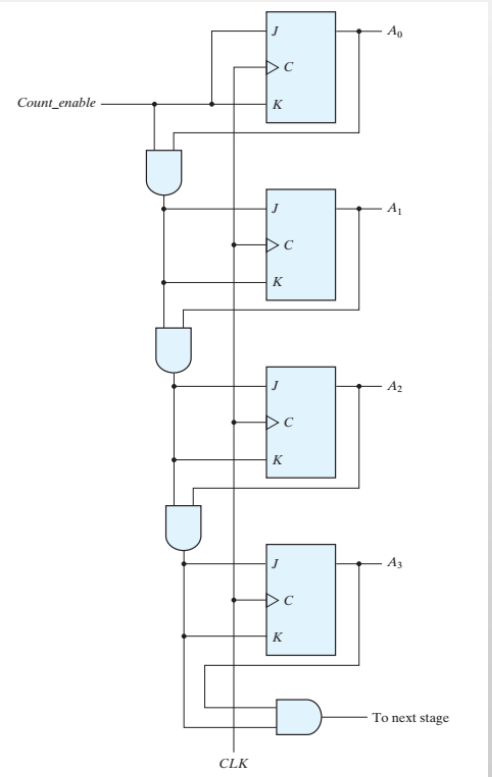
- The logic diagram of a BCD ripple counter using JK flip-flops is shown:



# Counter: synchronous counters

- In a synchronous counter, the C inputs of all flip-flops receive the common clock.
- Synchronous counters are presented in the next two sections.




Four-bit synchronous binary counter

# Counter: synchronous up-down counters

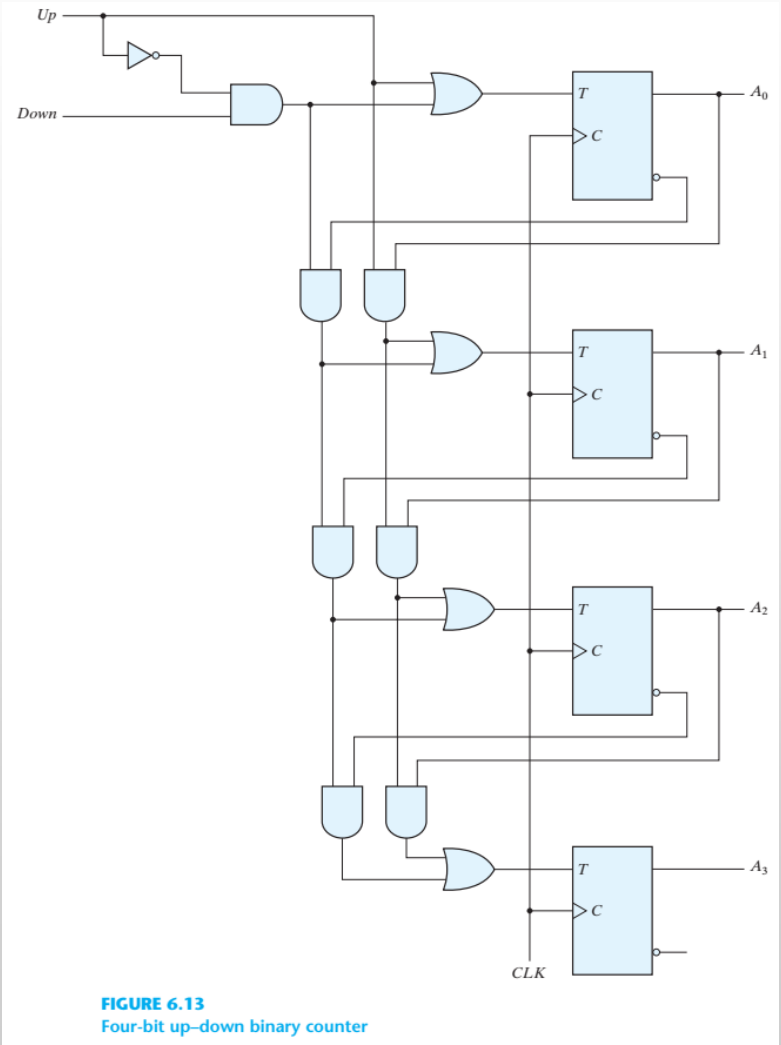
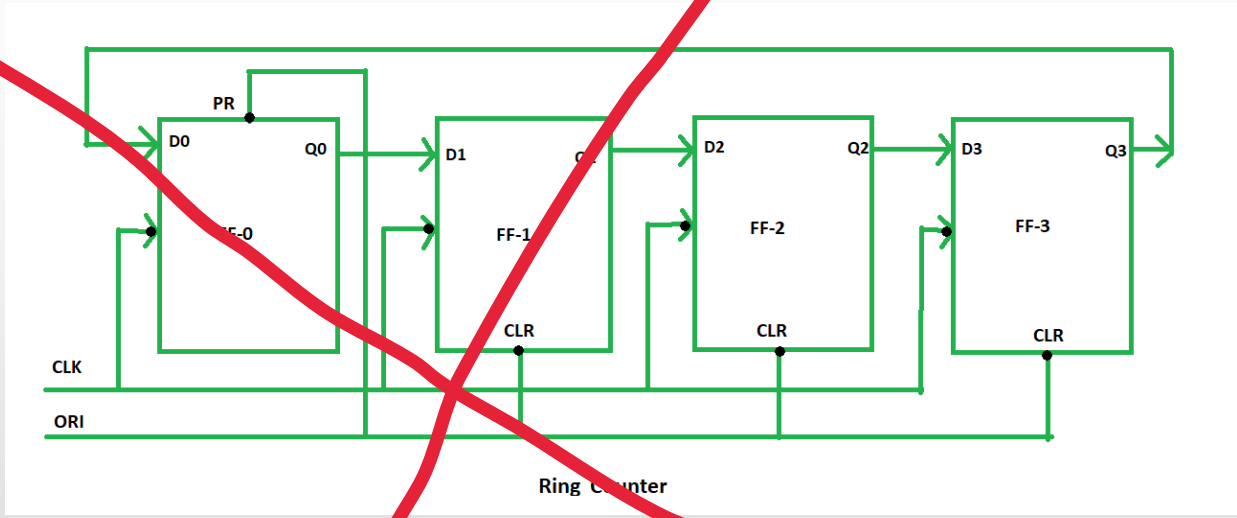



**FIGURE 6.13**
Four-bit up–down binary counter

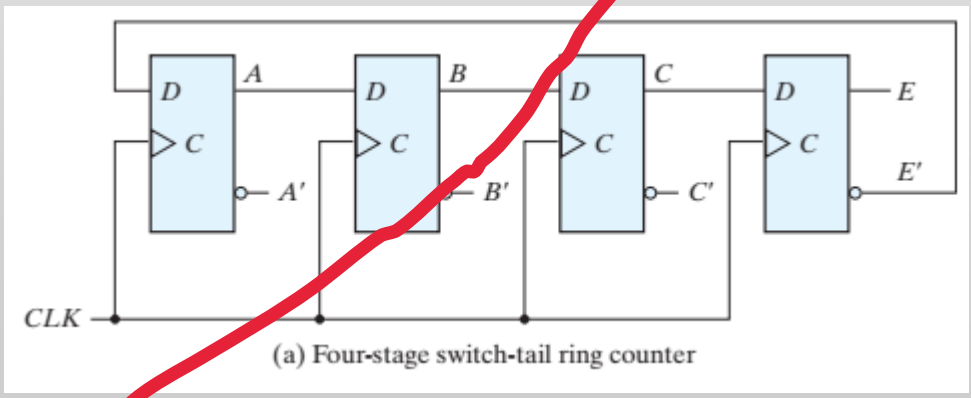
# Counter: Ring Counter

- Ring Counter:



Ring Counter

- Johnson Counter:



(a) Four-stage switch-tail ring counter

| Sequence number | Flip-flop outputs A | B | C | E | AND gate required for output |
|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | $A'E'$ |
| 2 | 1 | 0 | 0 | 0 | $AB'$ |
| 3 | 1 | 1 | 0 | 0 | $BC'$ |
| 4 | 1 | 1 | 1 | 0 | $CE'$ |
| 5 | 1 | 1 | 1 | 1 | $AE$ |
| 6 | 0 | 1 | 1 | 1 | $A'B$ |
| 7 | 0 | 0 | 1 | 1 | $B'C$ |
| 8 | 0 | 0 | 0 | 1 | $C'E$ |

(b) Count sequence and required decoding