Analysis of Clocked Sequential Circuit

- The behavior of a clocked sequential circuit is determined from its inputs, outputs and state of the flip-flops (i.e., the output of the flip-flops). • The analysis of a clocked sequential circuit consists of obtaining a table of a diagram of the time sequences of inputs, outputs and states. – E.g., given a current state and current inputs, how will the state and outputs change when the next active clock edge arrives???
- We have a basic procedure for analyzing a clocked sequential circuit:

  ✓Write down the equations for the outputs and the flip flop inputs.

  ✓Using these equations, derive a state table which describes the next state.

  ✓Obtain a state diagram from the state table.

- It is the state table and/or state diagram that specifies the behavior of the circuit. • The flip-flop input equations are sometimes called the excitation equations.

- The state table is sometimes called a transition table.

Analysis Example

- Is this a clocked sequential circuit?

  YES!

- What type of Memory?

  D Flip-Flops

- How many state variables?

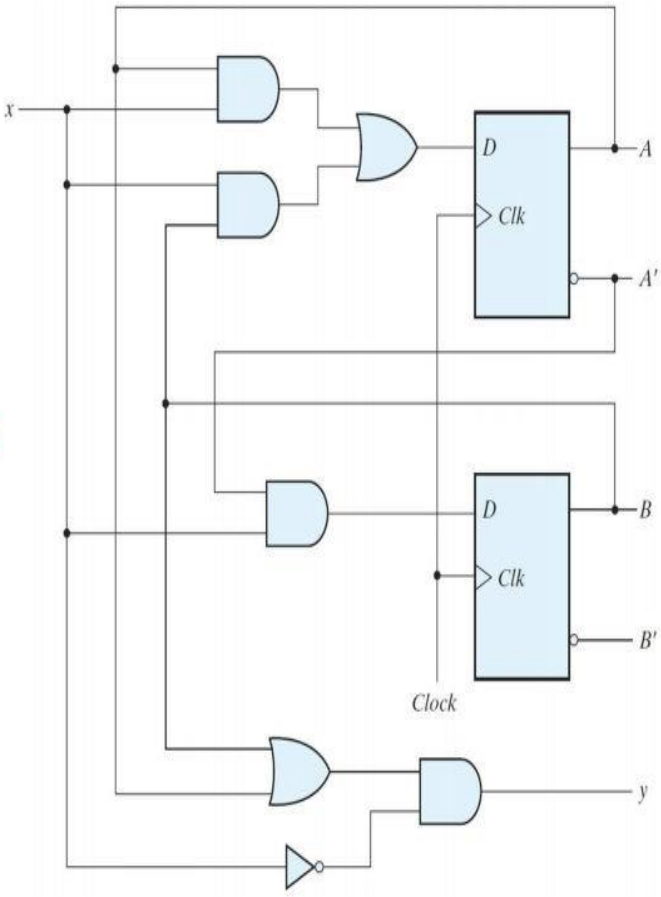
  Two state variables: *A* and *B*

- What are the Inputs?

  One Input: *x*

- What are the Outputs?

  One Output: *y*



State Equations

- A state equation is an algebraic expression that specifies the condition for a flip-flop state transition.
- The left side of the equation denotes the next state of the flip-flop and the right side of the equation is a Boolean expression that specifies the present state and input conditions that make the next state equal to 1.

- In above example, D inputs determine the flip-flop's next state, so it is possible to write a set of next-state equations for the circuit:

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

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

- In compact form:

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

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

- Similarly, the present-state value of the output y can be expressed algebraically as : $y(t) = [A(t) + B(t)]x'(t)$

- Removing the symbol (t) for the present state, the output Boolean function: $y = (A + B)x'$

State table

- The time sequence of inputs, outputs, and flip-flop states can be enumerated in a state table.

- The state table for the example circuit above is shown in the table below.

The table consists of four sections:

Present state: shows the states of flip flops A and B at any given time t

Input: gives a value of x for each possible present state

Next state: shows the states of the flip flops one clock period later at time t

Output: gives the value of y for each present state.

Next state and output column is derived from the state equations.

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

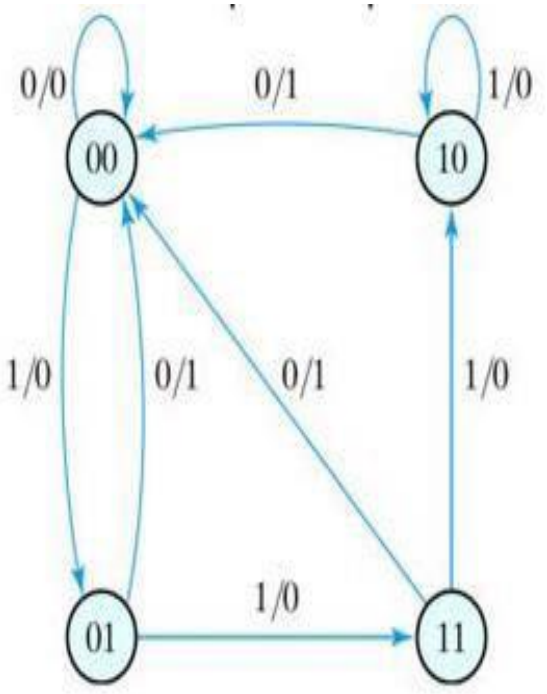
Second Form of the State Table

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 a state diagram.
- In this diagram, a state is represented by a circle, and the transition between states is indicated by directed lines connecting the circles.
- Each directed line is labeled

▪ a logic diagram ⇔ a state table ⇔ a state diagram



State Reduction

State Reduction

- The reduction of the number of flip-flops in a sequential circuit is referred to as the state-reduction problem.
- State-reduction, reducing the number of states in a state table, while keeping the external input–output requirements unchanged, can reduce the number of flip-flops used in a sequential circuit.
- Since m flip-flops produce 2^m states, a reduction in the number of states may (or may not) result in a reduction in the number of flip flops.
- Reducing the number of flip-flops sometimes results the equivalent circuit with fewer flip-flops but more combinational gates to realize its next state and output fewer flip flops but more combinational gates to realize its next state and output logic.

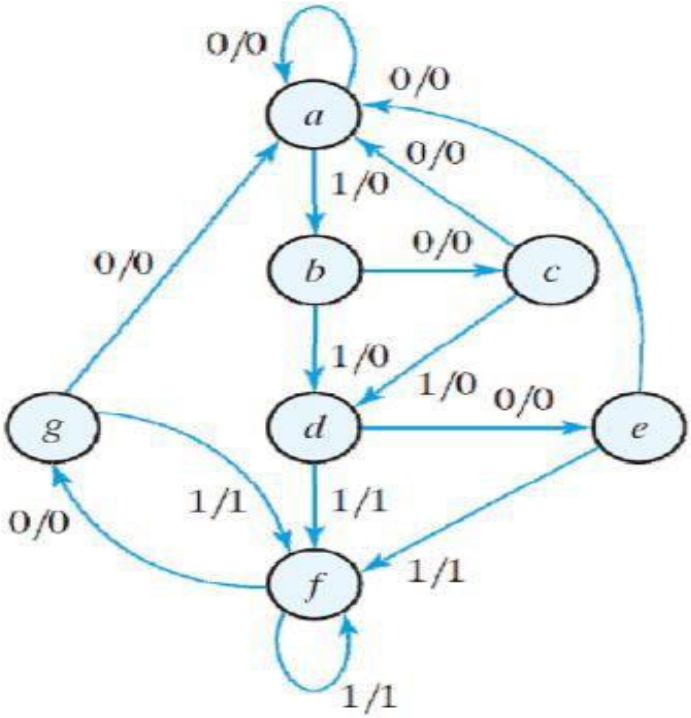
State Reduction Example

- Two circuits are equivalent if identical input sequences are applied to the two circuits and identical outputs occur for all input sequences, then one may be replaced by the other.

• State reduction reduces the number of states in a sequential circuit without altering the input–output relationships.

• Only the input-output sequences are important in this example.

- Consider the input sequence 01010110100 starting from the initial state a



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	
output	0	0	0	0	0	1	1	0	1	0	0	

- State table is more convenient for state reduction than a diagram.
- State reduction algorithm: “Two states are said to be equivalent if, for each member of the set of inputs, they give exactly the same output and send the circuit either to the same state or to an equivalent state.”
- When two states are equivalent, one of them can be removed without altering the input–output relationships.

Original State Table

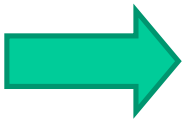
Present State	Next State		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1

e=g

If next state and output of two present states are same then we can eliminate one state

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



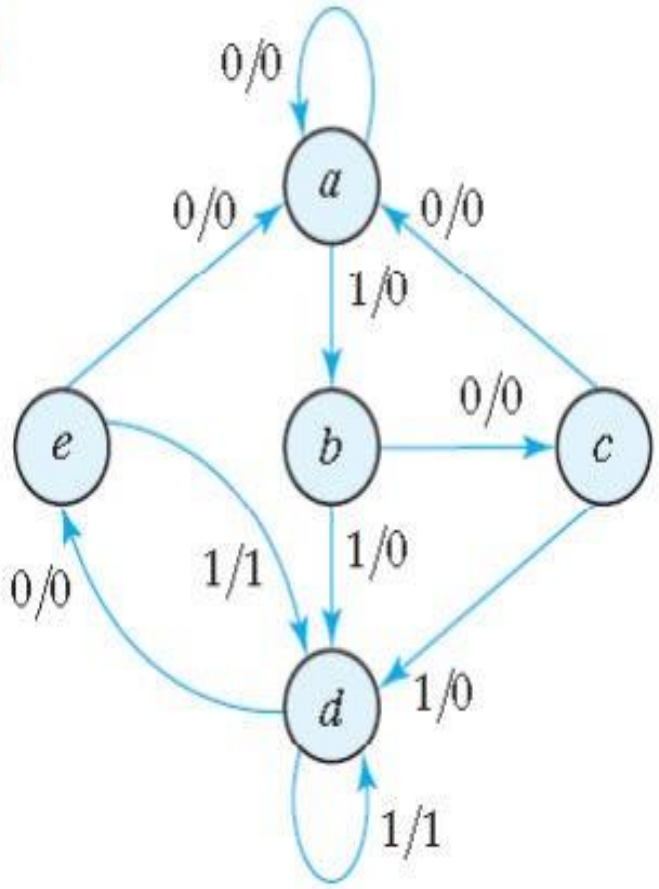
d=f

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

State Reduction Example

Reduced State Diagram



Design Procedure of Sequential Circuit

- A synchronous sequential circuit is made up of flip-flops and combinational gates.
- The design of the circuit consists of choosing the flip-flops and then finding a combinational gate structure that, together with the flip-flops, produces a circuit which fulfills the stated specifications.
- The design steps for synchronous sequential circuits can be summarized as:

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

Design Example

- The state diagram consists of four states with binary values already assigned.

- Directed lines contain single binary digit without a slash, we conclude that there is one input variable and no output variables. (The state of the flip-flops may be considered the outputs of the circuit).

  • The two flip-flops needed to represent the four states are designated A and B.

- The input variable is designated x.

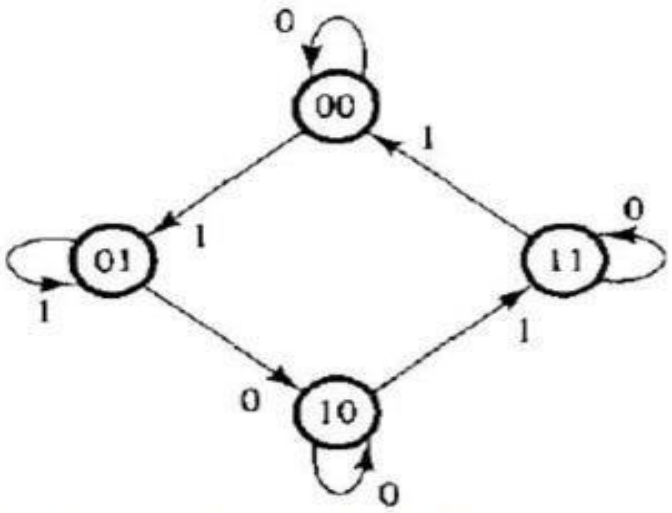


Fig: State-diagram for design example

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

Fig: State Table

Flip-Flop Excitation Tables

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

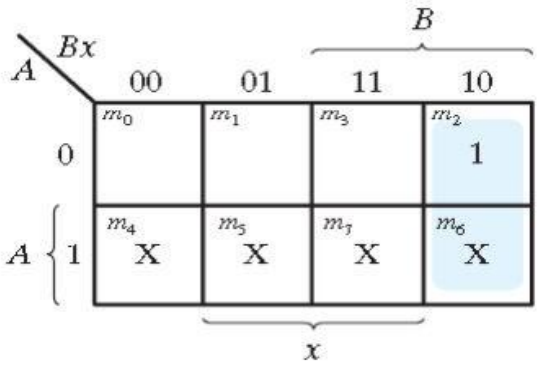
(a) *JK* Flip-Flop

Design Example Using JK Flip Flop

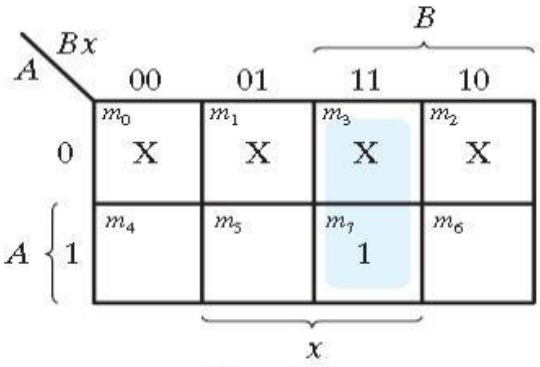
State Table and JK Flip-Flop Inputs

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

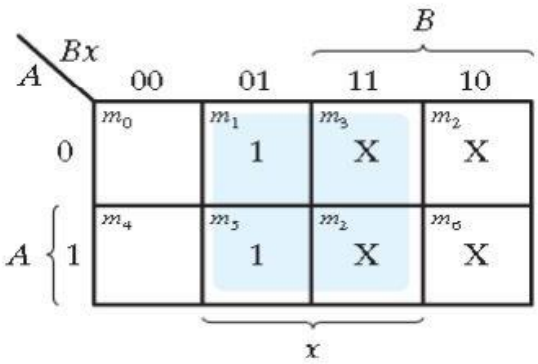
K-Maps for JK Input Equations



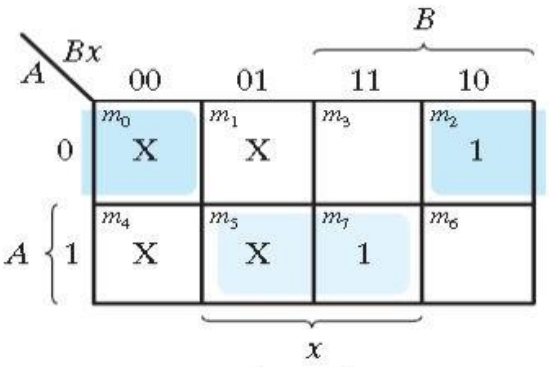
$J_A = Bx'$



$K_A = Bx$



$J_B = x$



$K_B = (A \oplus x)'$

Logic Diagram with JK Flip-flops

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