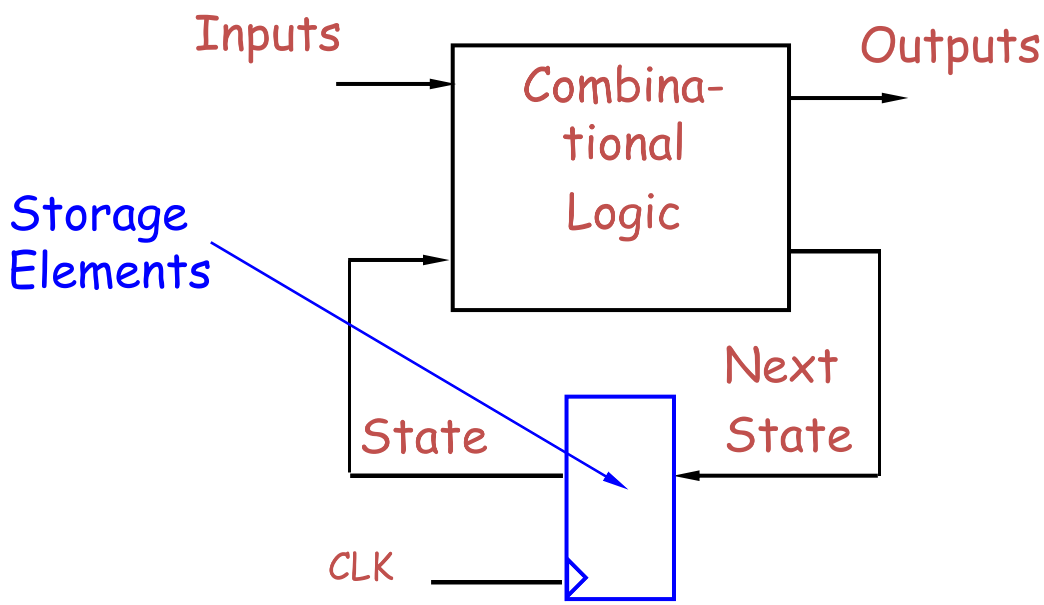
Sequential Synchronous Circuit Analysis

✓ General Model

- Current State at time (t) is stored in an array of flip-flops.
- Next State at time (t+1) is a Boolean function of State and Inputs.
- Outputs at time (t) are a Boolean function of State(t) and (Mealy) of Inputs (t).



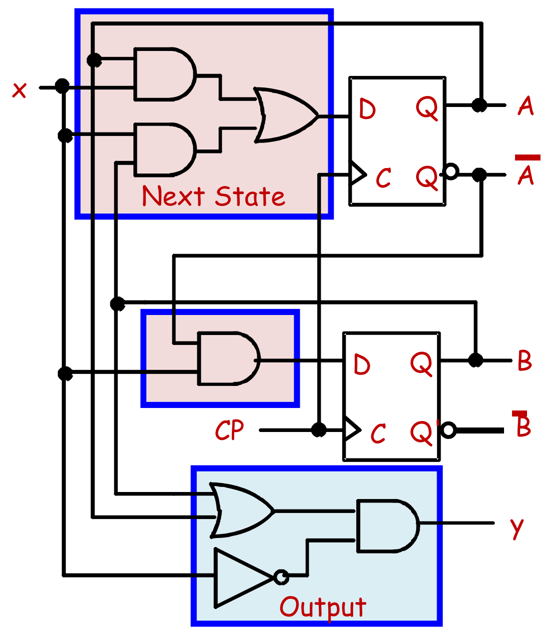
Example 1: Analysis

- Input: $x(t)$
- Output: $y(t)$
- State: $(A(t), B(t))$
- What is the Output Function?

$$y(t) = \overline{x}(t)(B(t) + A(t))$$

- What is the Next State Function?

$$A(t+1) = A(t)x(t) + B(t)x(t)$$
$$B(t+1) = \overline{A}(t)x(t)$$



State Table Characteristics

- ✓ State table – a multiple variable table with the following four sections:
  - Present State – the values of the state variables for each allowed state.
  - Input – the input combinations allowed.
  - Next-state – the value of the state at time (t+1) based on the present state and the input.
  - Output – the value of the output as a function of the present state and (Mealy) the input.
- ✓ From the viewpoint of a truth table:
  - the inputs are Input, Present State
  - and the outputs are Output, Next State

Example 1: Alternate State Table

$$A(t+1) = A(t)\,x(t) + B(t)\,x(t)$$
$$B(t+1) = \bar{A}(t)\,x(t)$$
$$y(t) = \bar{x}(t)\,(B(t) + A(t))$$

The time sequence of inputs, outputs, and flip-flop states can be enumerated in a **state table** (sometimes called **transition table**).

$$A+ = A\,x + B\,x$$
$$B+ = \bar{A}\,x$$
$$y = \bar{x}\,(B + A)$$

Table 5-2
State Table for the Circuit

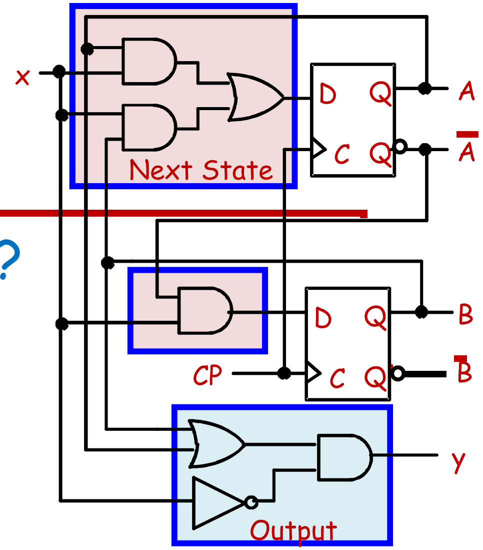
Present State		Input	Next State		Output
A	*B*	*x*	*A*	*B*	*y*
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Table 5-3
Second Form of the State Table

Present State	Next State		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
AB	*AB*	*AB*	*y*	*y*
00	00	01	0	0
01	00	11	1	0
10	00	10	1	0
11	00	10	1	0

Example 1

- ✓ Where in time are inputs, outputs and states defined?



$A+ = A\,x + B\,x$

$B+ = \bar{A}\,x$

$y = \bar{x}\,(B + A)$

Functional Simulation - Fig. 4-18 Mano & Kime

Reset

Clock

X

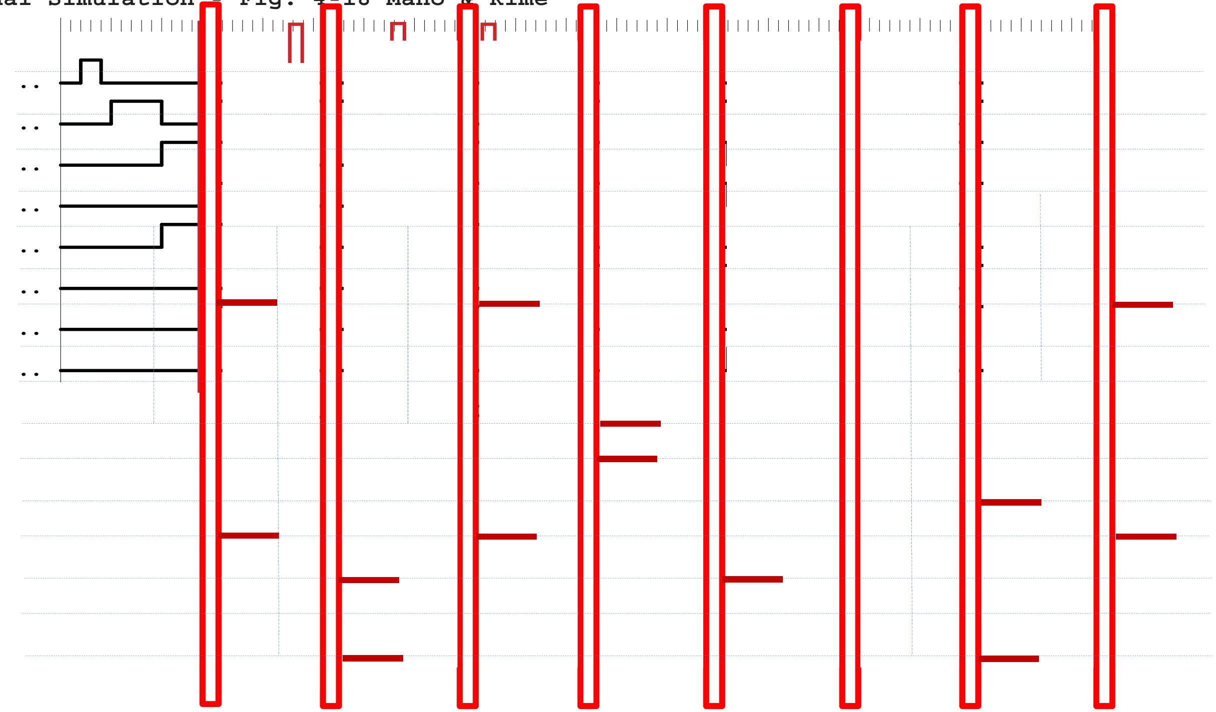
Next A

Next B

A

B

y



State Diagrams

- ✓ The sequential circuit function can be represented in *graphical form* as a state diagram with the following components:
  - A circle with the state name in it for each state
  - A directed arc from the Present State to the Next State for each state transition
  - A label on each directed arc with the Input values which causes the state transition, and
  - A label:
    - On each circle with the output value produced: **Moore** type

      or

    - On each directed arc with the output value produced: **Mealy** type.

Example 1: State Diagram

- ✓ Diagram gets confusing for large circuits
- ✓ For small circuits, usually easier to understand than the state table

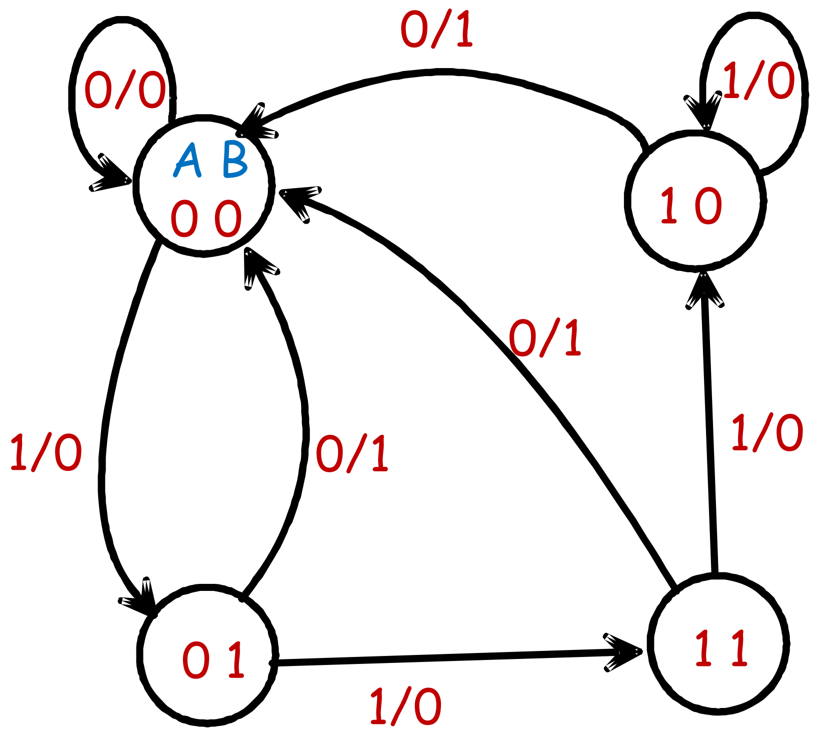
$$A+ = A\,x + B\,x$$

$$B+ = \overline{A}\,x$$

$$y = \overline{x}\,(B + A)$$

Table 5-3
Second Form of the State Table

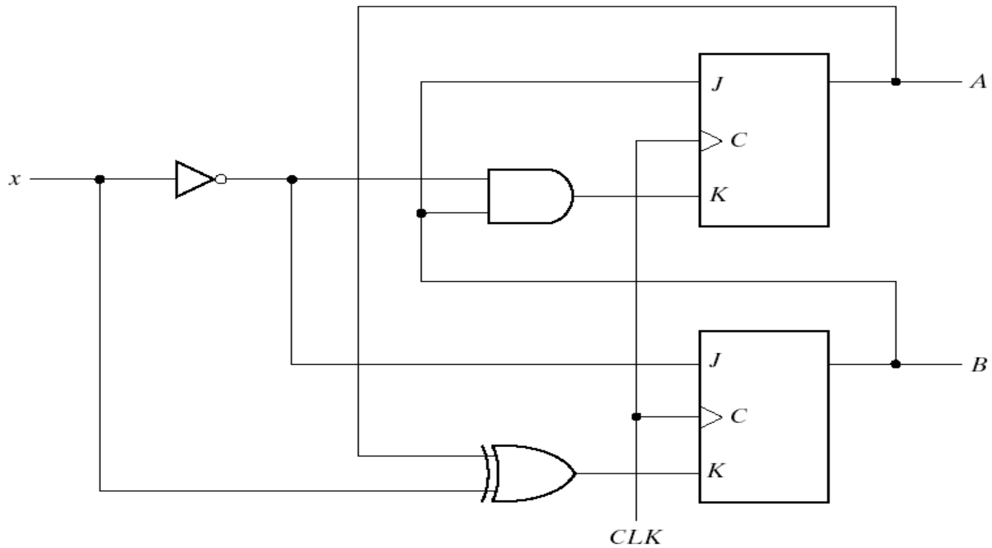
Present State	Next State		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
AB	*AB*	*AB*	*y*	*y*
00	00	01	0	0
01	00	11	1	0
10	00	10	1	0
11	00	10	1	0



1/0 : means input x = 1
output y = 0

Mealy

Analysis with JK Flip-Flops



$J_A = B \qquad K_A = B\,\overline{x}$

$J_B = \overline{x} \qquad K_B = \overline{A}x + A\overline{x} = A \oplus x$

Analysis with JK Flip-Flop

✓ The circuit can be specified by the flip-flop input equations:

$$J_A = B \qquad K_A = B\bar{x}$$

$$J_B = \bar{x} \qquad K_B = \bar{A}x + A\bar{x} = A \oplus x$$

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

Analysis with JK Flip-Flops

$$A+ = J\overline{A} + \overline{K}A$$
$$B+ = J\overline{B} + \overline{K}B$$

$$J_A = B \qquad K_A = B\,\overline{x}$$
$$J_B = \overline{x} \qquad K_B = \overline{A}x + A\overline{x} = A \oplus x$$

- ✓ Substituting the values of J_A and K_A from the input equations, we obtain the state equation for A:

$$A = B\overline{A} + (\overline{B\overline{x}})A = \overline{A}B + A\overline{B} + Ax$$

- ✓ The state equation provides the bit values for the column under next state of A in the state table. Similarly, the state equation for flip-flop B can be derived from the characteristic equation by substituting the values of J_B and K_B:

$$B = \overline{x}\overline{B} + (\overline{A \oplus x})B = \overline{B}\overline{x} + ABx + \overline{A}B\overline{x}$$

Analysis with JK Flip-Flops

✓ The state diagram of the sequential circuit is:

$$A = B\bar{A} + (\overline{B\bar{x}})A = \bar{A}B + A\bar{B} + Ax$$

$$B = \bar{x}\bar{B} + (\overline{A \oplus x})B = \bar{B}\bar{x} + ABx + \bar{A}B\bar{x}$$

Table 5-4
State Table for Sequential Circuit with JK

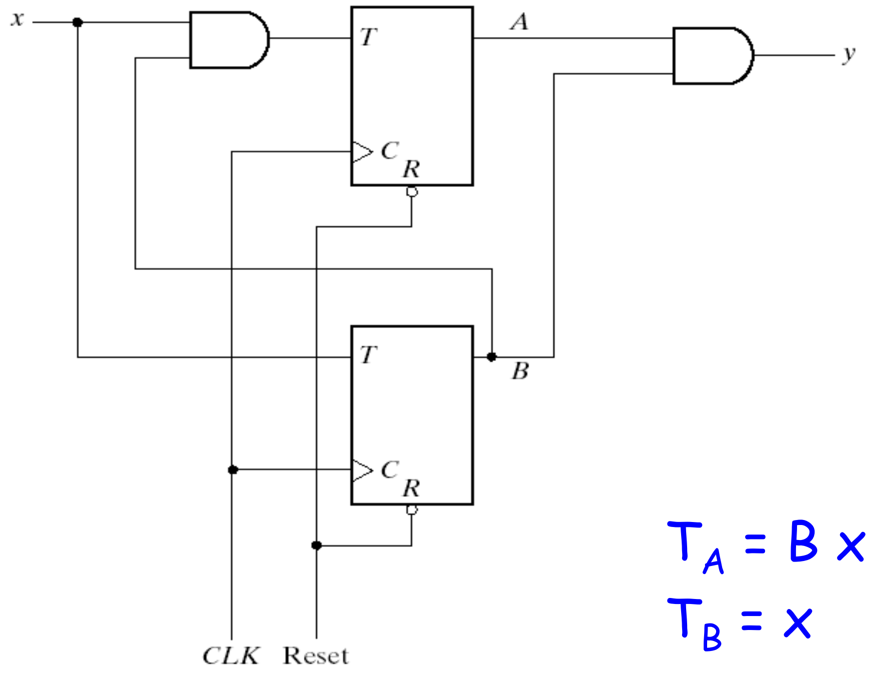
Present State		Input	Next State	
A	*B*	*x*	*A*	*B*
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Moore

Analysis With T Flip-Flops

- ✓ Characteristic equation:

$$Q = T \oplus Q$$



$$T_A = B\,x$$
$$T_B = x$$
$$y = A\,B$$

Moore

Analysis With T Flip-Flops

- ✓ Consider the previous sequential circuit. It has two flip-flops *A* and B, one input x, and one output y. It can be described algebraically by two input equations and an output equation:

$$T_A = B\,x$$
$$T_B = x$$
$$y = A\,B$$

$$A = (\overline{Bx})A + (Bx)\overline{A}$$
$$= A\overline{B} + A\overline{x} + \overline{A}Bx$$
$$B = x \oplus B$$

Table 5-5
State Table for Sequential Circuit with T Flip-Flops

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

Analysis With T Flip-Flops

- Characteristic equation:

$$Q(t + 1) = T \oplus Q$$

Table 5-5
State Table for Sequential Circuit with T Flip-Flops

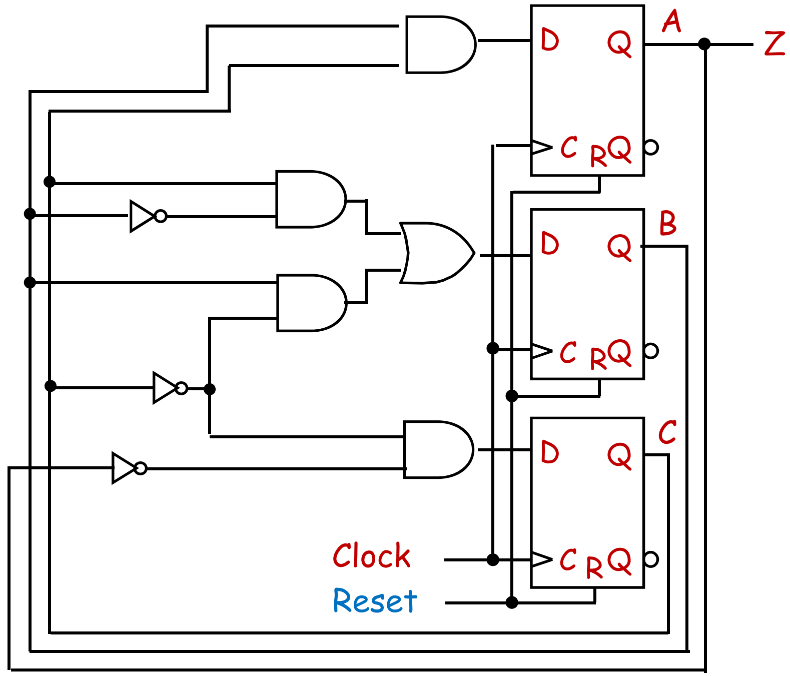
Present State		Input	Next State		Output
A	*B*	*x*	*A*	*B*	*y*
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	1	0	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	1

Sequential Circuit Analysis

- ✓ Initialization: reset to (0, 0, 0)
- ✓ Equations:

$$A = B\,C$$
$$B = \overline{B}\,C + B\,\overline{C}$$
$$C = \overline{A}\,\overline{C}$$

$$Z = A$$



Moore

Example 2: State Table

$A = B\,C$

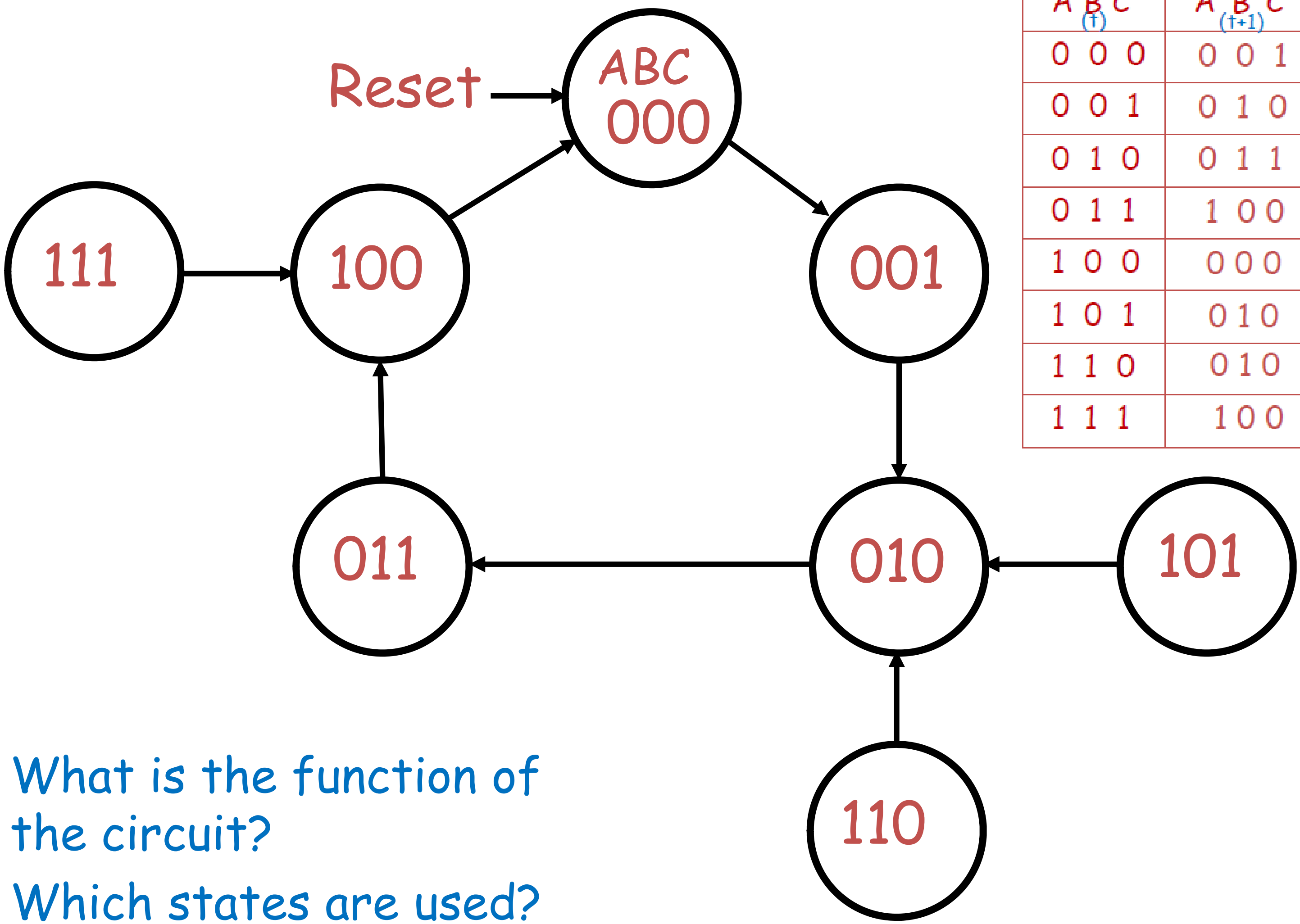
$B = \overline{B}\,C + B\,\overline{C}$

$C = \overline{A}\,\overline{C}$

$Z = A$

A B C (t)	A' B' C' (t+1)	Z
0 0 0	0 0 1	0
0 0 1	0 1 0	0
0 1 0	0 1 1	0
0 1 1	1 0 0	0
1 0 0	0 0 0	1
1 0 1	0 1 0	1
1 1 0	0 1 0	1
1 1 1	1 0 0	1

Example 2: State Diagram

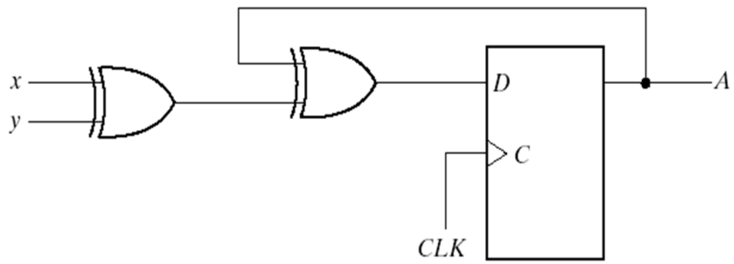


A B C (t)	A' B' C' (t+1)	Z
0 0 0	0 0 1	0
0 0 1	0 1 0	0
0 1 0	0 1 1	0
0 1 1	1 0 0	0
1 0 0	0 0 0	1
1 0 1	0 1 0	1
1 1 0	0 1 0	1
1 1 1	1 0 0	1

- ✓ What is the function of the circuit?
- ✓ Which states are used?

Moore

Analysis with D Flip-Flop



- ✓ The circuit we want to analyze is described by the input equation

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

- ✓ The D_A symbol implies a D flip-flop with output A. The x and y variables are the inputs to the circuit. No output equations are given, so the output is implied to come from the output of the flip-flop.

Moore

Analysis with D Flip-Flop

- ✓ The binary numbers under A x y are listed from 000 through 111. The next state values are obtained from the state equation

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

- ✓ The state diagram consists of two circles-one for each state

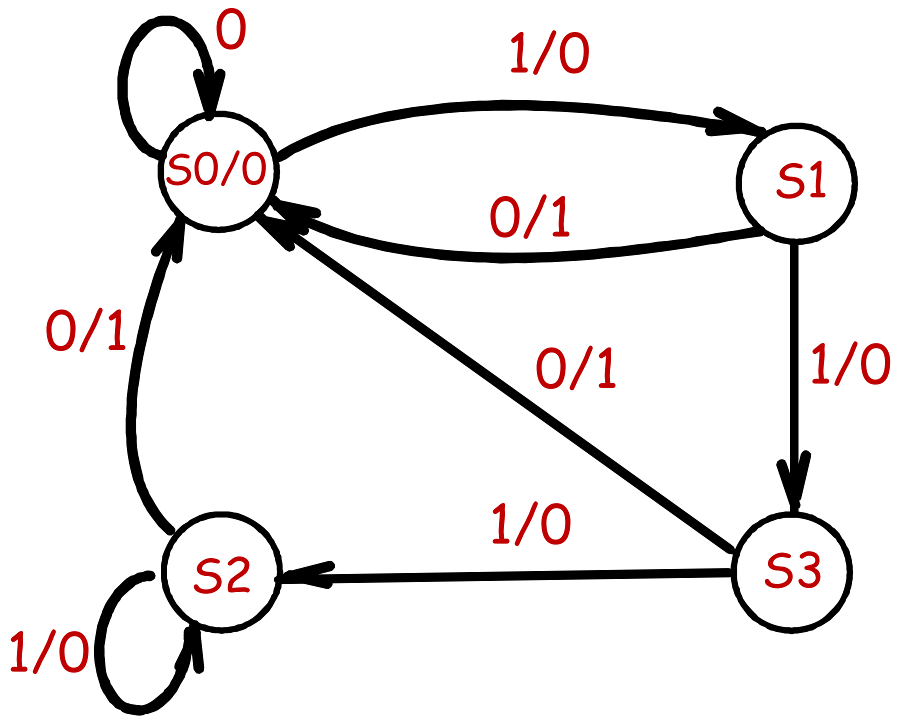
Present state	Inputs		Next state
A	x	y	A
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Equivalent State Definitions

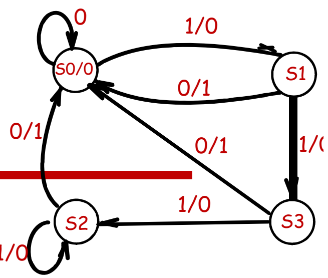
- ✓ Two states are equivalent if their response for each possible **input sequence** is an identical output sequence.
- ✓ Alternatively, two states are equivalent if their outputs produced for each **input symbol** is identical and their next states for each **input symbol** are the same or equivalent.
- ✓ Two states that are not equivalent are distinguishable

Equivalent State Example

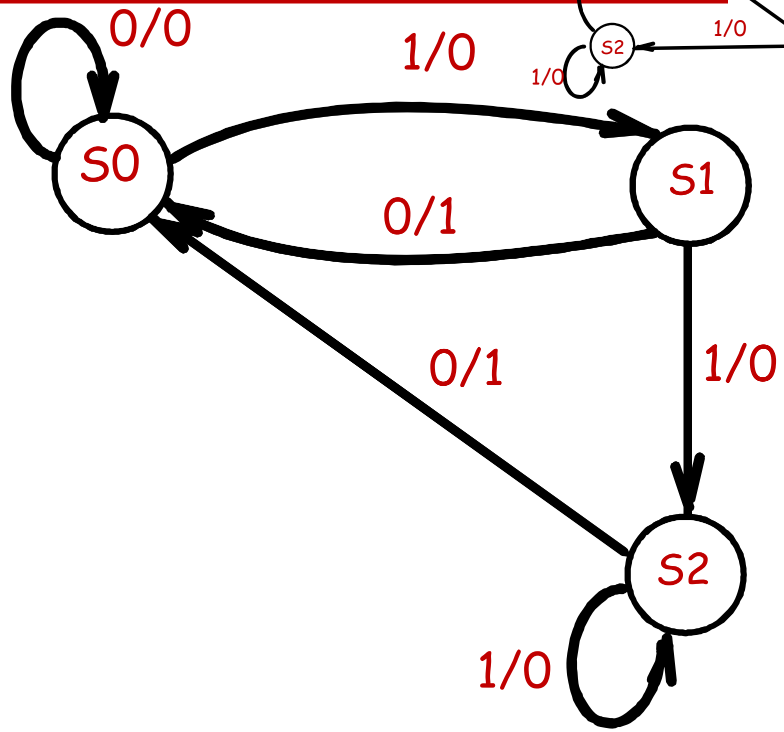
- For states S3 and S2,
  - the output for input 0 is 1 and input 1 is 0, and
  - the next state for input 0 is S0 and for input 1 is S2.
  - states S3 and S2 are equivalent.



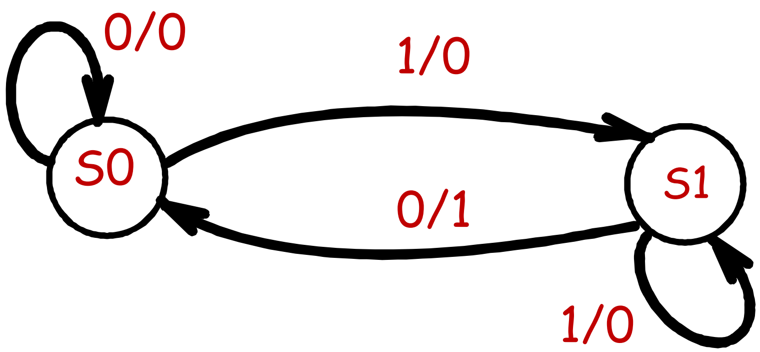
Equivalent State Example



- Replacing S3 and S2 by a single state gives state diagram:



- Examining the new diagram, states S1 and S2 are equivalent since
  - their outputs for input 0 is 1 and input 1 is 0, and
  - their next state for input 0 is S0 and for input 1 is S2,
- Replacing S1 and S2 by a single state gives state diagram:



Moore and Mealy Models

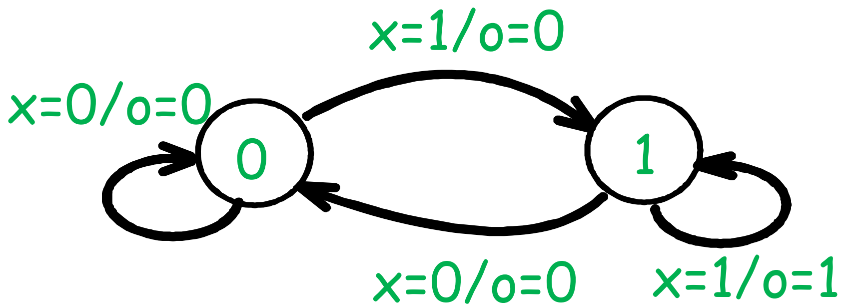
- ✓ Sequential Circuits or Sequential Machines are also called Finite State Machines (FSMs).
- ✓ Two formal models exist:

- **Moore Model**
  - Named after E.F. Moore
  - Outputs are a function **ONLY of states**
  - Usually specified on the states.
- **Mealy Model**
  - Named after G. H. Mealy
  - Outputs are a function of **inputs AND states**
  - Usually specified on the state transition arcs.

Moore and Mealy Example Diagrams & Tables

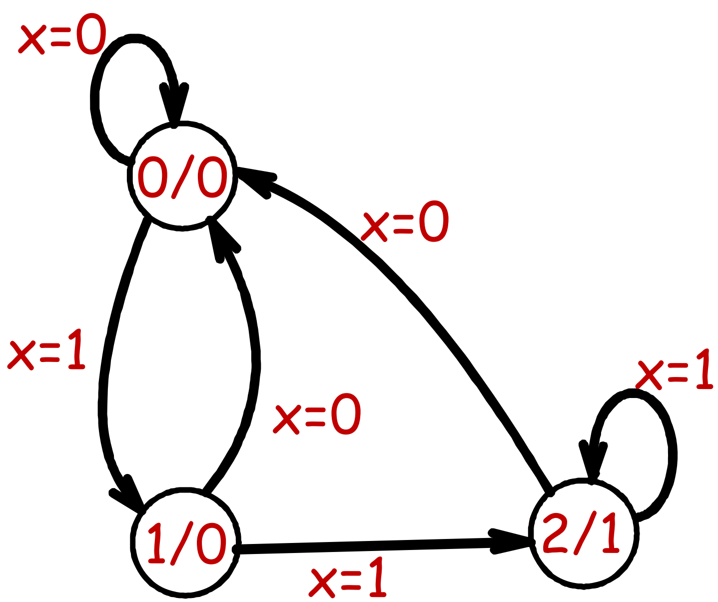
- ✓ Mealy Model State Diagram
  maps **inputs and state** to **outputs**

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



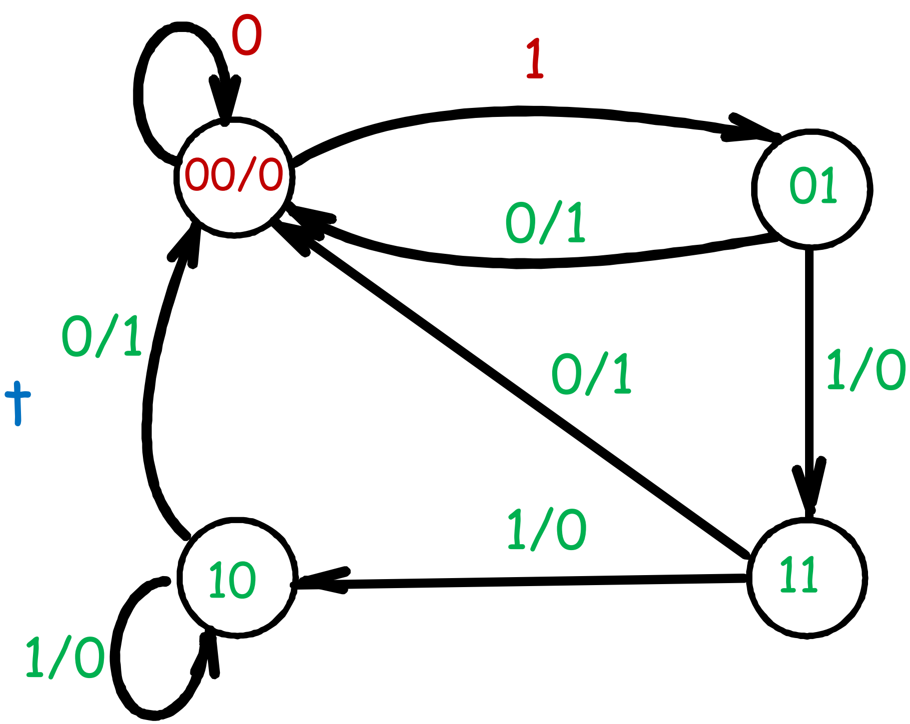
- ✓ Moore Model State Diagram
  maps **states** to **outputs**

Present State	Next State		Output
	x=0	x=1	
0	0	1	0
1	0	2	0
2	0	2	1



Mixed Moore and Mealy Outputs

- In real designs, some outputs may be Moore type and other outputs may be Mealy type.
- Example: Figure can be modified to illustrate this
  - State 00: Moore
  - States 01, 10, and 11: Mealy



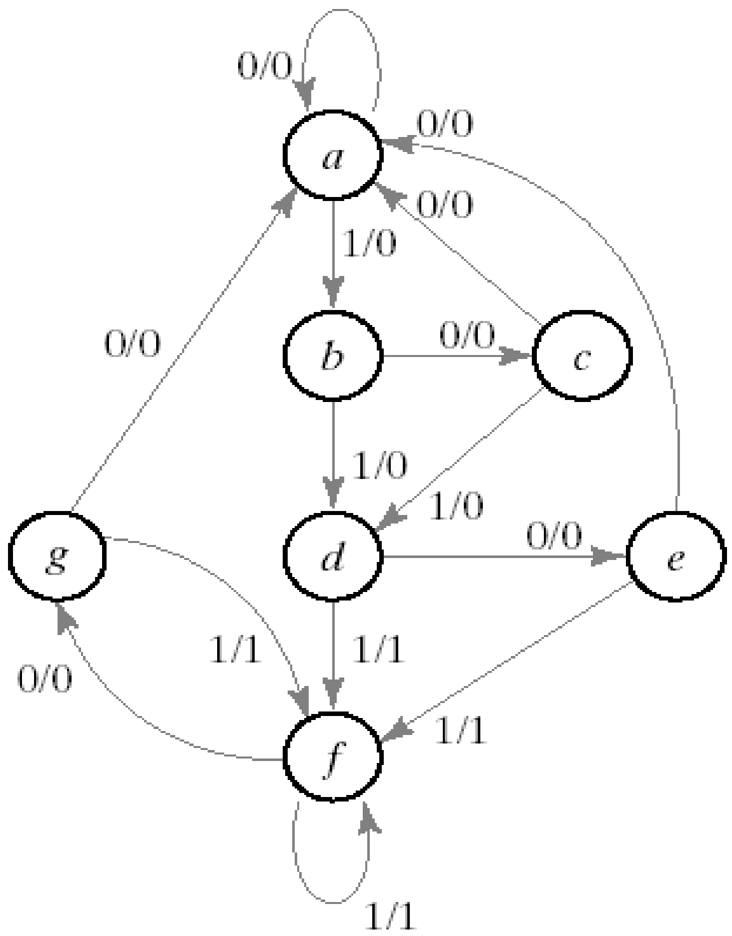
- This simplifies output specification

State Reduction

Example : next state →

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	

t_0 t_1 t_2 t_3 ...



State Reduction

✓ We now proceed to reduce the number of states for this example. First, we need the state table; it is more convenient to apply procedures for state reduction using a table rather than a diagram. The state table of the circuit is listed in Table 5-6 and is obtained directly from the state diagram.

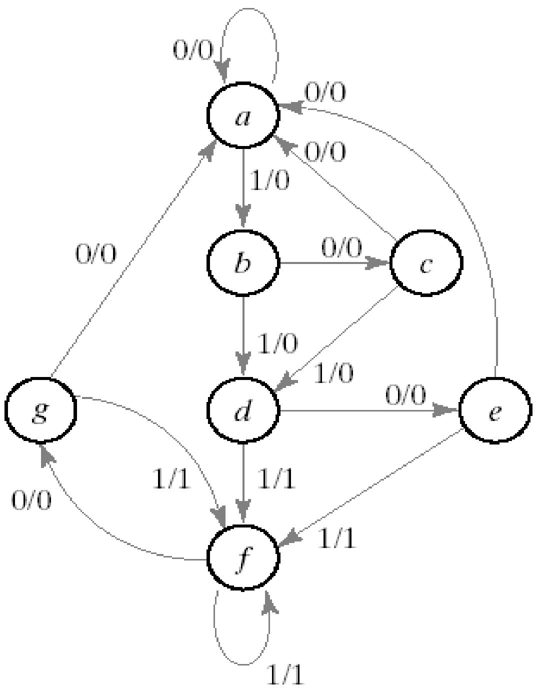


Table 5-6
State Table

	Next State		Output	
Present State	**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

- ✓ States g and e are two such states: they both go to states a and f and have outputs of 0 and 1 for x=0 and x=1, respectively. Therefore, states g and e are equivalent and one of these states can be removed. The procedure of removing a state and replacing it by its equivalent is demonstrated in Table 5-7. The row with present g is removed and state g is replaced by state e each time it occurs in the next-state columns.

Table 5-7
Reducing the State Table

	Next State		Output	
Present State	$x = 0$	$x = 1$	$x = 0$	$x = 1$
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

- ✓ Present state f now has next states e and f and outputs 0 and 1 for x=0 and x=1, respectively. The same next states and outputs appear in the row with present state d. Therefore, states f and d are equivalent and state f can be removed and replaced by d. The final reduced table is shown in Table 5-8. The state diagram for the reduced table consists of only five states.

Table 5-8
Reduced State Table

	Next State		Output	
Present State	$x = 0$	$x = 1$	$x = 0$	$x = 1$
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 :

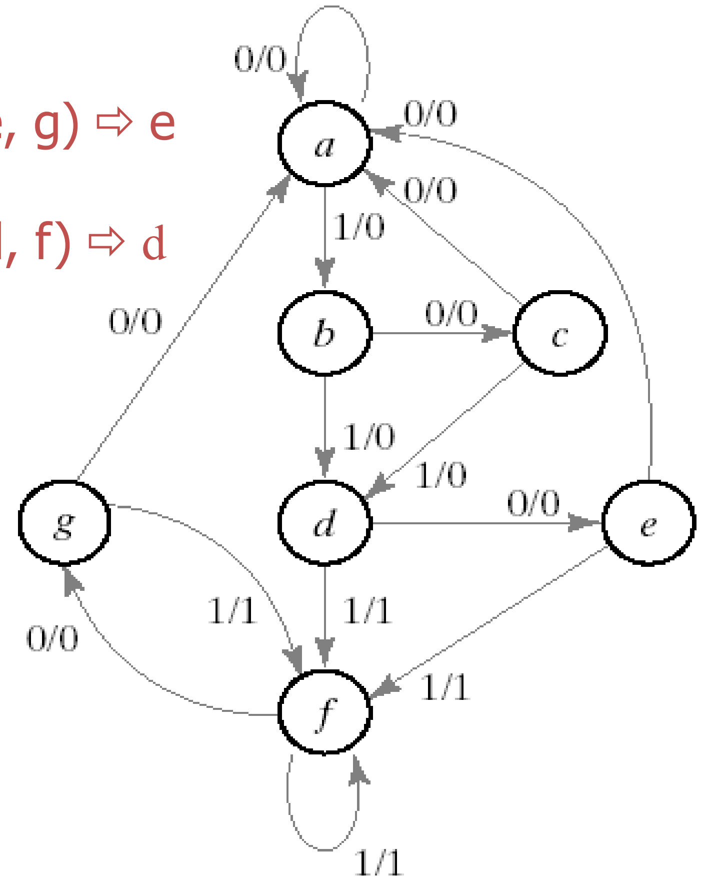
state $\boxed{a}$ a b c d e f f g f g a

input 0 1 0 1 0 1 1 0 1 0 0

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

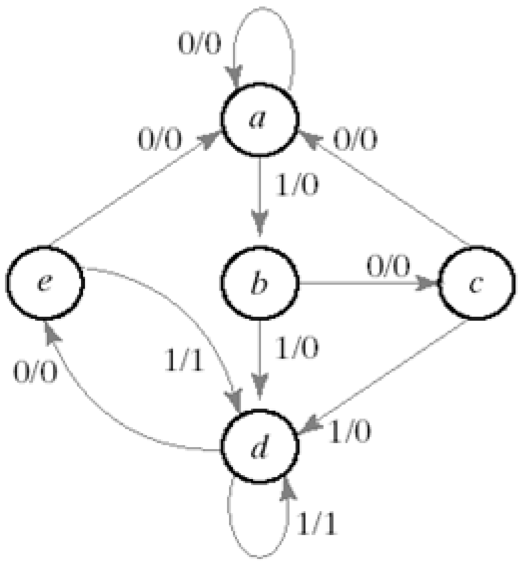
(e, g) ⇨ e

(d, f) ⇨ d



state $\boxed{a}$ a b c d e d d e d e a

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



State Assignment

Table 5-9
Three Possible Binary State Assignments

State	Assignment 1 Binary	Assignment 2 pseudo 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

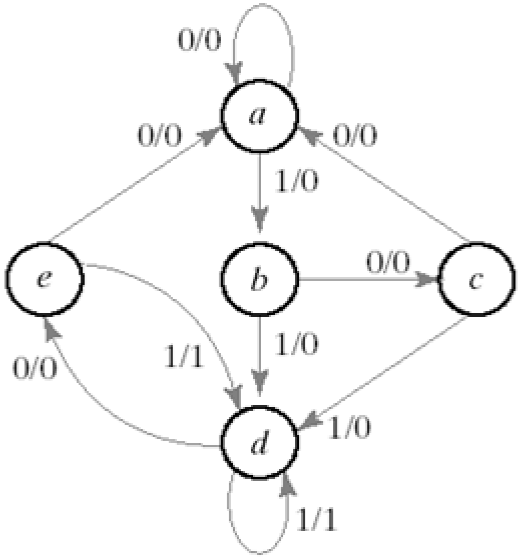


Table 5-10
Reduced State Table with Binary Assignment 1

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