
Logic and Computer Design Fundamentals

Chapter 5 – Sequential Circuits

Part 2 - Sequential Circuit Analysis

Charles Kime & Thomas Kaminski

© 2008 Pearson Education, Inc.
(Hyperlinks are active in View Show mode)

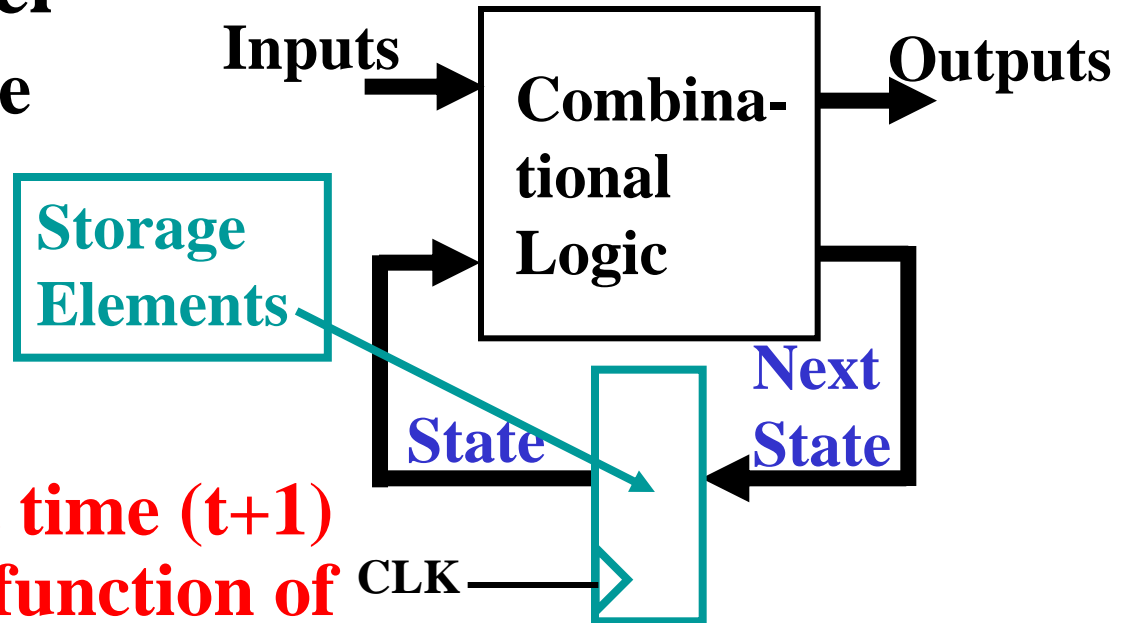
Overview

- **Part 1 - Storage Elements**
- **Part 2 - Sequential Circuit Analysis**
 - **State tables**
 - **State diagrams**
 - **Equivalent states**
- **Part 3 - Sequential Circuit Design**
- **Part 4 – State Machine Design**

Sequential Circuit Analysis

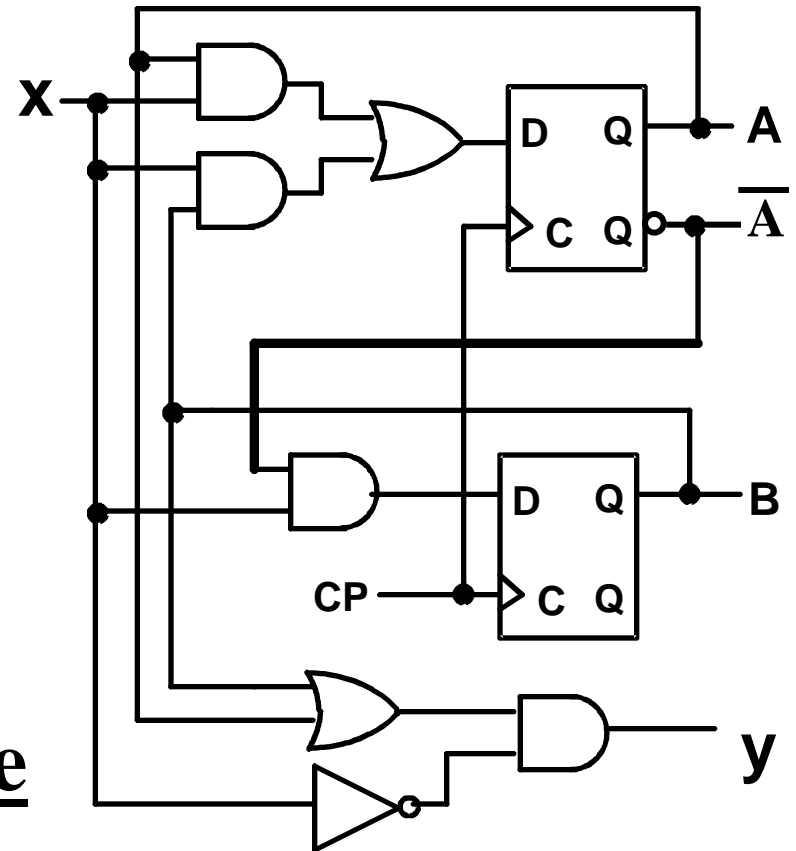
■ General Model

- **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 (sometimes) Inputs (t).**



Example 1 (from Fig. 5-15)

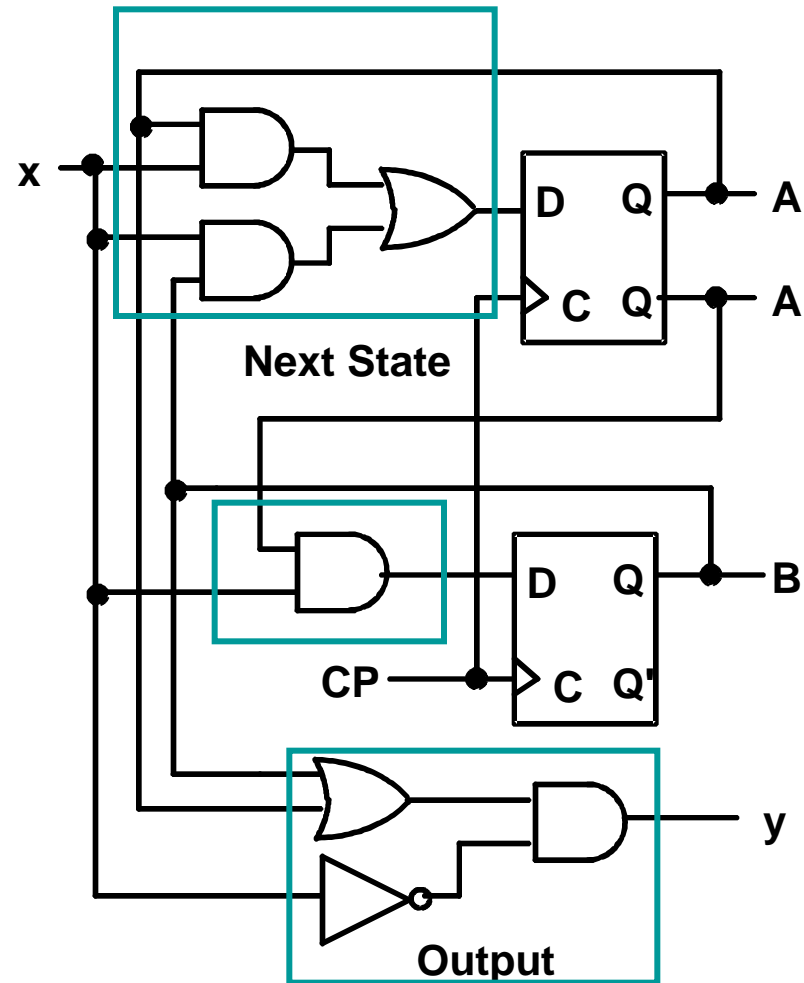
- Input: $x(t)$
- Output: $y(t)$
- State: $(A(t), B(t))$
- What is the Output Function?
- What is the Next State Function?



Example 1 (from Fig. 5-15) (continued)

■ Boolean equations for the functions:

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

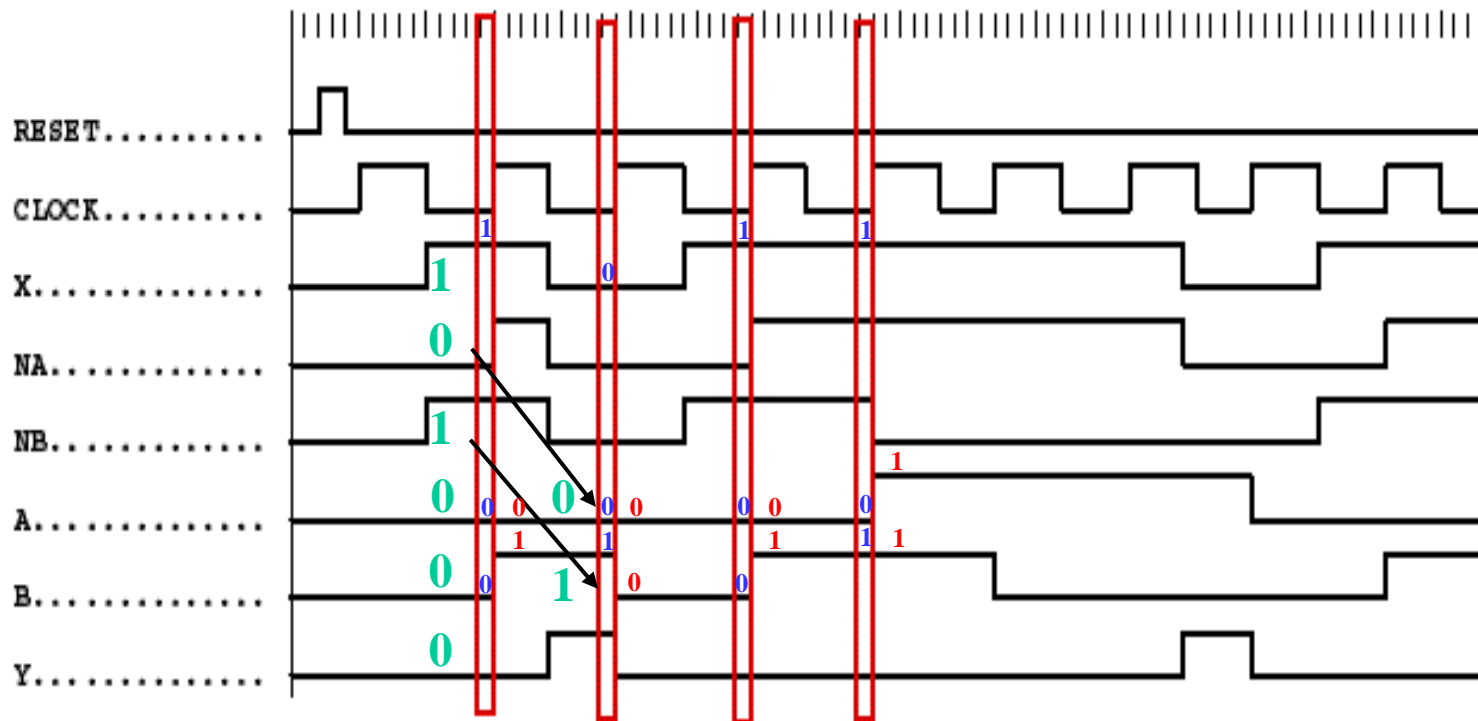


- **Where in time are inputs, outputs and states defined?**
 - $\mathbf{A}(t+1) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t)$

-
- The timing diagram illustrates the operation of a 4-bit ripple-carry adder. The signals shown are RESET, CLOCK, X, A, B, and Y. The CLOCK signal is a periodic square wave. The RESET signal is active-low, with a single pulse at the beginning. The inputs X and B are 4-bit numbers. The outputs A and Y are 4-bit numbers. The values of A and B at each clock edge are indicated by blue and red numbers respectively.
- | Clock Cycle | RESET | CLOCK | X | A | B | Y |
|-------------|-------|-------|------|------|------|------|
| 1 | 1 | 0 | 0000 | 0000 | 0000 | 0000 |
| 2 | 0 | 1 | 0000 | 0000 | 0000 | 0000 |
| 3 | 0 | 0 | 0000 | 0000 | 0000 | 0000 |
| 4 | 0 | 1 | 0000 | 0000 | 0000 | 0000 |
| 5 | 0 | 0 | 0000 | 0000 | 0000 | 0000 |
| 6 | 0 | 1 | 0000 | 0000 | 0000 | 0000 |
| 7 | 0 | 0 | 0000 | 0000 | 0000 | 0000 |
| 8 | 0 | 1 | 0000 | 0000 | 0000 | 0000 |
| 9 | 0 | 0 | 0000 | 0000 | 0000 | 0000 |
| 10 | 0 | 1 | 0000 | 0000 | 0000 | 0000 |
| 11 | 0 | 0 | 0000 | 0000 | 0000 | 0000 |
| 12 | 0 | 1 | 0000 | 0000 | 0000 | 0000 |
| 13 | 0 | 0 | 0000 | 0000 | 0000 | 0000 |
| 14 | 0 | 1 | 0000 | 0000 | 0000 | 0000 |
| 15 | 0 | 0 | 0000 | 0000 | 0000 | 0000 |
| 16 | 0 | 1 | 0000 | 0000 | 0000 | 0000 |

Example 1(from Fig. 5-15) (continued)

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



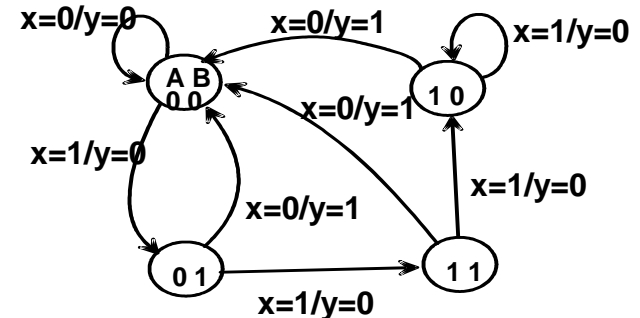
State Table Characteristics

- ***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 (sometimes) the input.
- **From the viewpoint of a truth table:**
 - the inputs are Input, Present State
 - and the outputs are Output, Next State

Example 1: State Table (from Fig. 5-15)

- The state table can be filled in using the next state and output equations:

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



Present State	Input	Next State	Output
A(t) B(t)	x(t)	A(t+1) B(t+1)	y(t)
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

Example 1: Alternate State Table

- 2-dimensional table that matches well to a K-map.
Present state rows and input columns in Gray code order.

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

Present State A(t) B(t)	Next State		Output	
	x(t)=0 A(t+1)B(t+1)	x(t)=1 A(t+1)B(t+1)	x(t)=0 y(t)	x(t)=1 y(t)
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 Diagrams

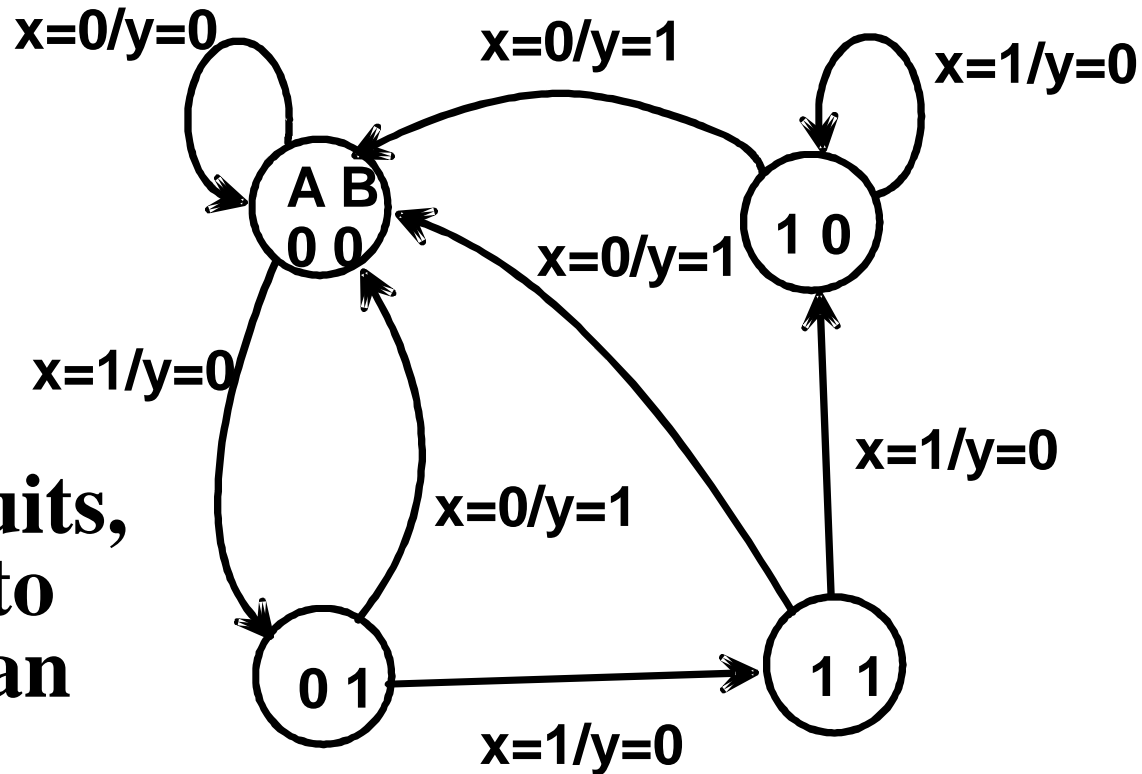
- The sequential circuit function can be represented in graphical form as a state diagram with the following components:
 - A circle with the state name in it for each state
 - A directed arc from the Present State to the Next State for each state transition
 - A label on each directed arc with the Input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

State Diagrams

- **Label form:**
 - **On circle with output included:**
 - **state/output**
 - **Moore type output depends only on state**
 - **On directed arc with the output included:**
 - **input/output**
 - **Mealy type output depends on state and input**

Example 1: State Diagram

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

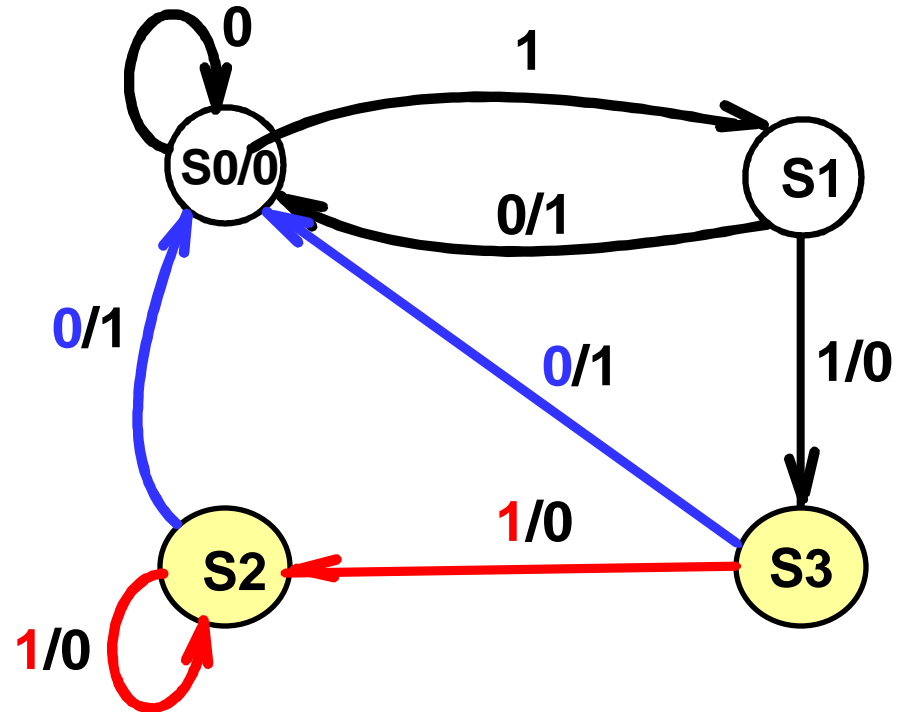


Equivalent State Definitions

- 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.**

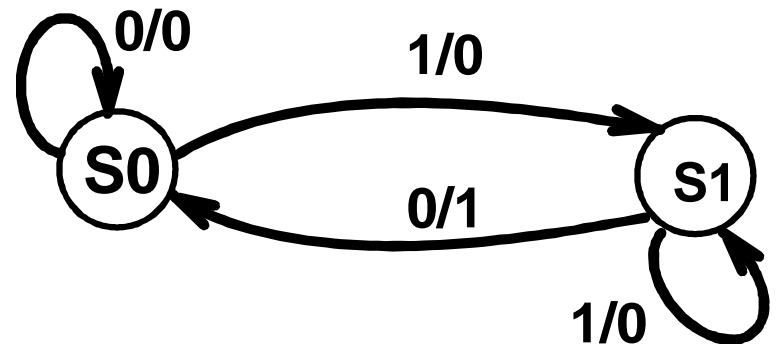
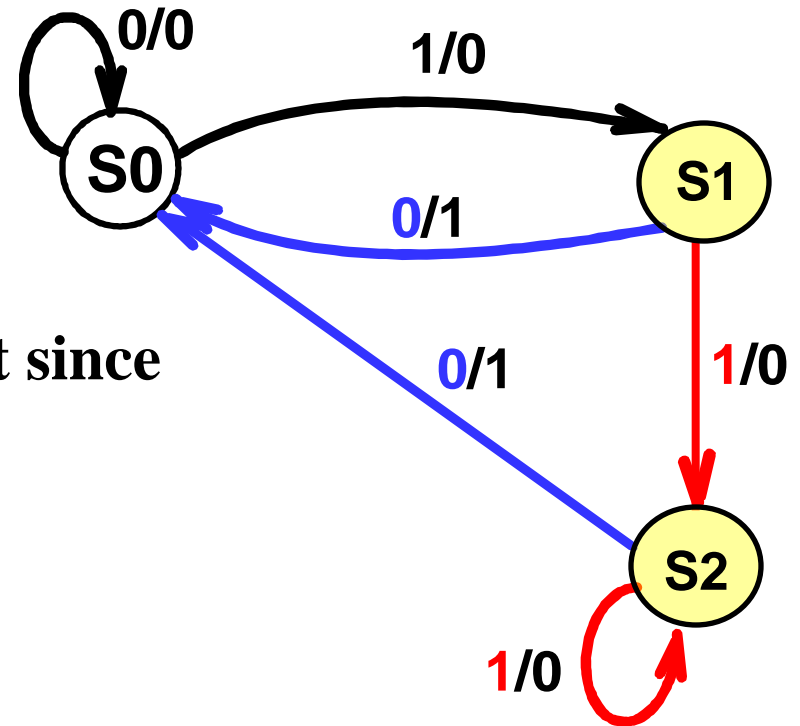
Equivalent State Example

- Text Figure 5-17(a):
- 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.
- **By the alternative definition, 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

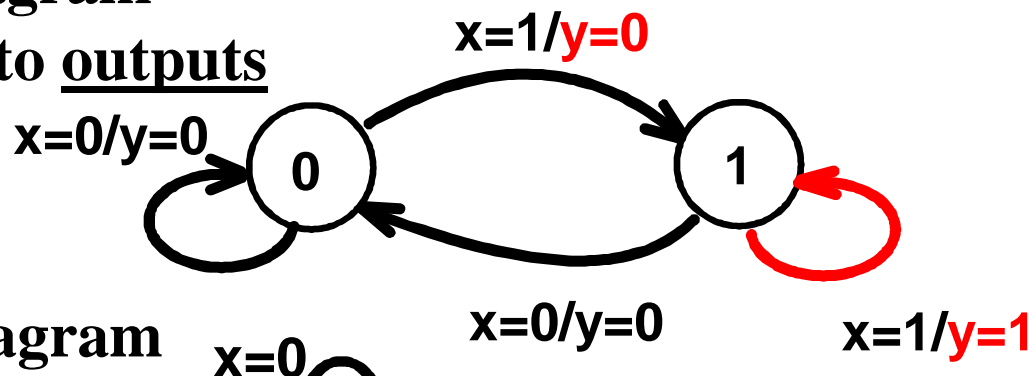
- Named after E.F. Moore
- **Outputs are a function ONLY of states**
- Usually specified on the states.

- Mealy Model

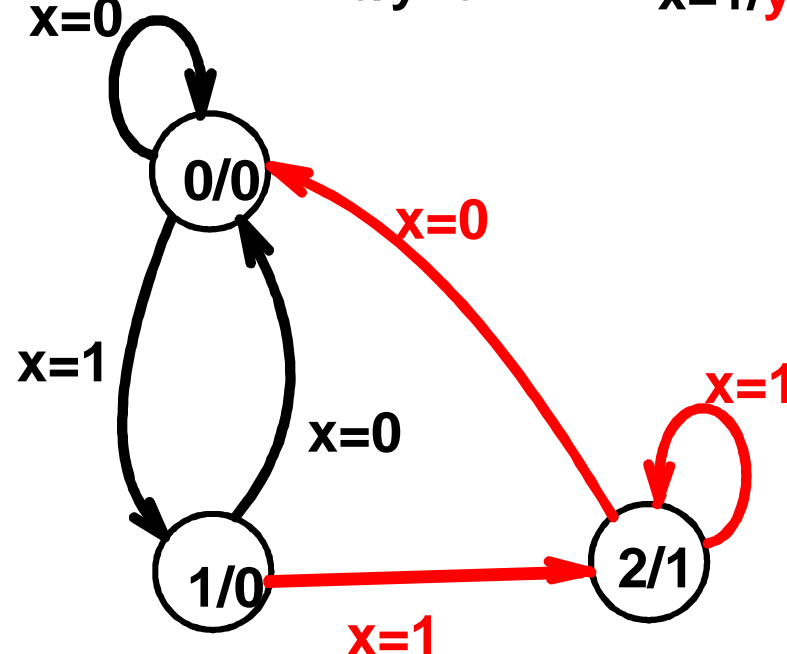
- Named after G. Mealy
- **Outputs are a function of inputs AND states**
- Usually specified on the state transition arcs.

Moore and Mealy Example Diagrams

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



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



Moore and Mealy Example Tables

- **Moore Model** state table maps state to outputs

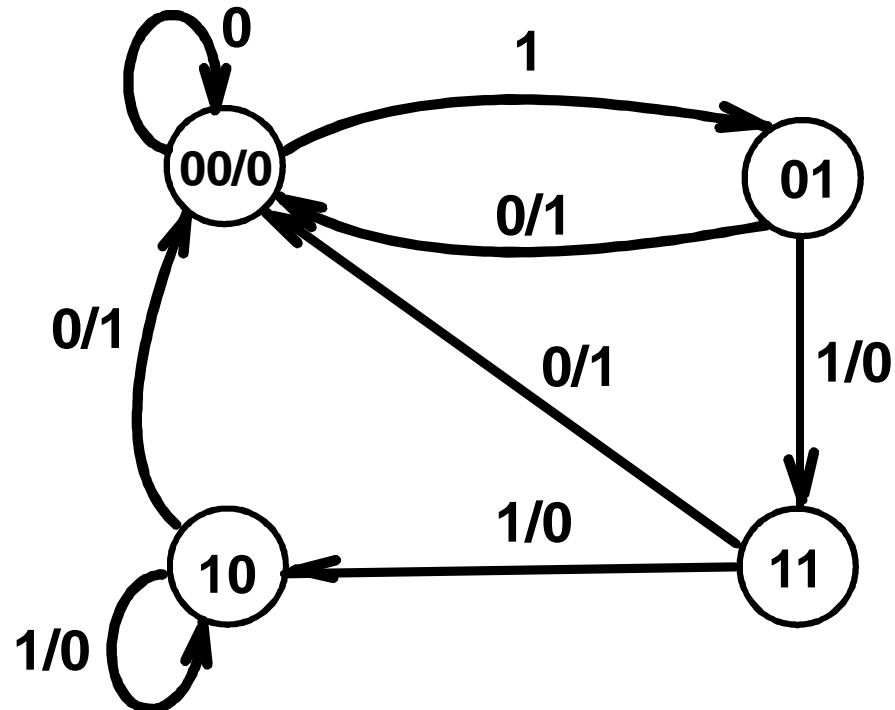
Present State	Next State		Output
	x=0	x=1	
0	0	1	0
1	0	2	0
2	0	2	1

- **Mealy Model** state table 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

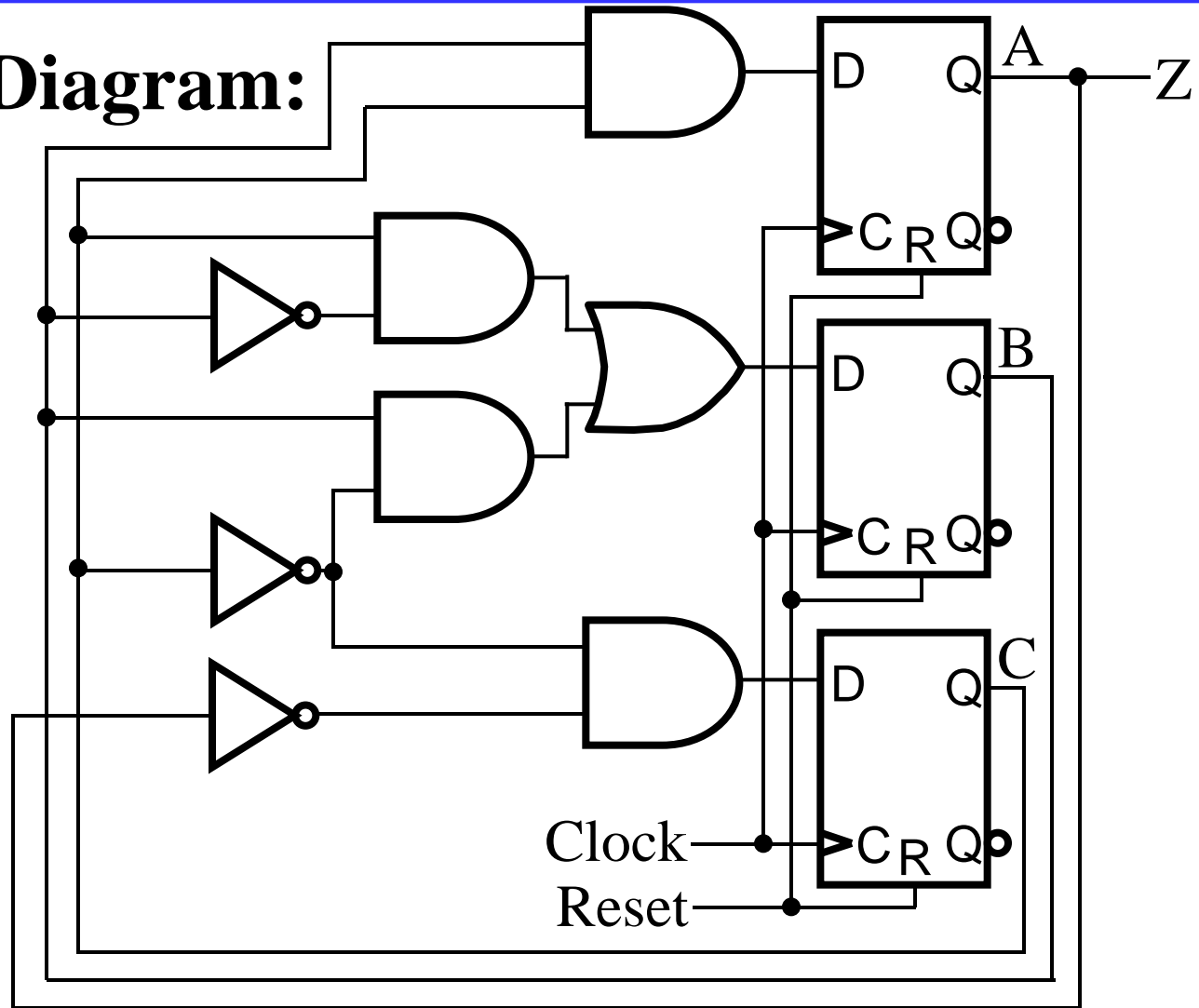
Mixed Moore and Mealy Outputs

- In real designs, some outputs may be Moore type and other outputs may be Mealy type.
- Example: Figure 5-17(a) can be modified to illustrate this
 - State 00: Moore
 - States 01, 10, and 11: Mealy
- Simplifies output specification



Example 2: Sequential Circuit Analysis

- **Logic Diagram:**



Example 2: Flip-Flop Input Equations

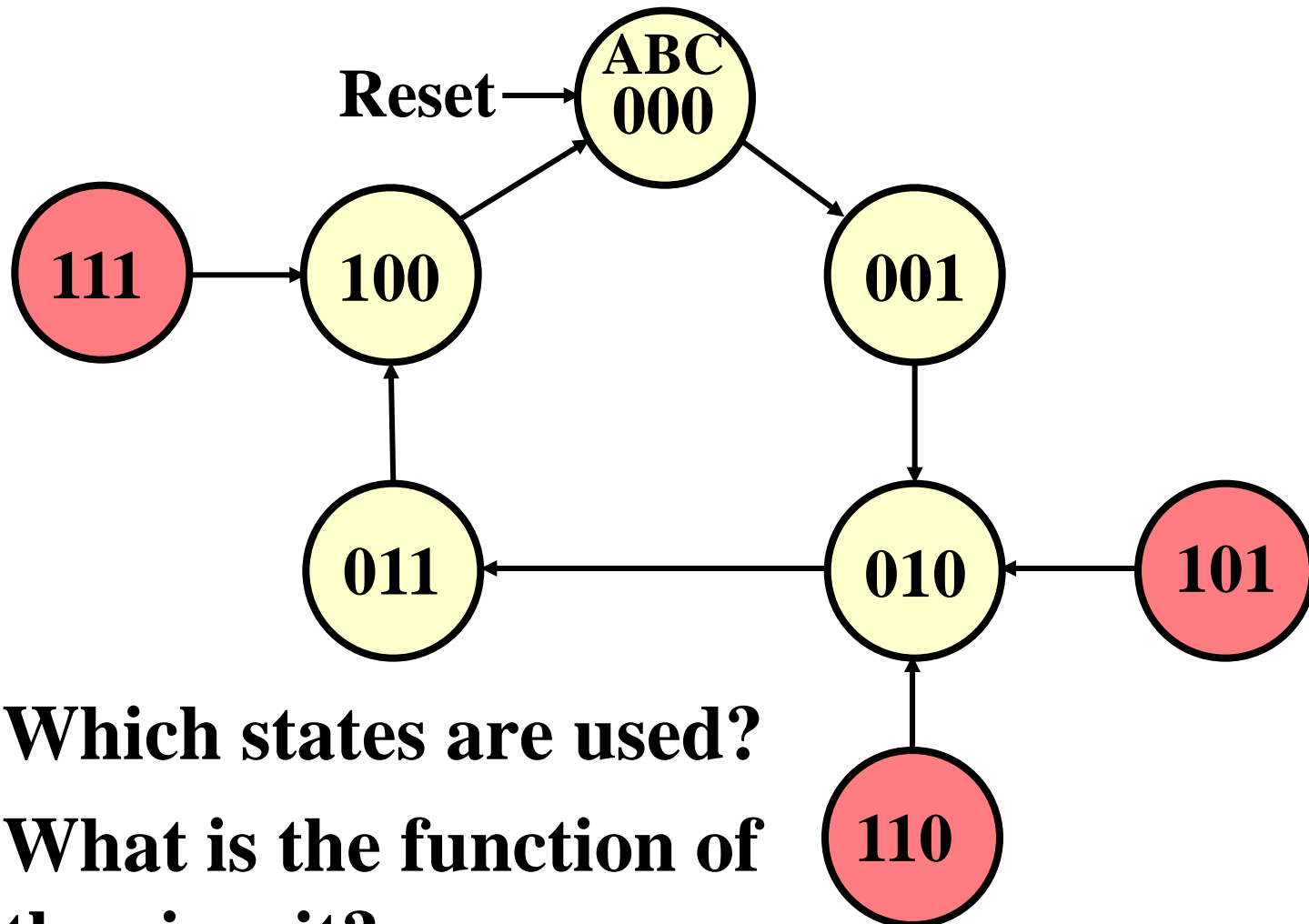
- **Variables**
 - **Inputs: None**
 - **Outputs: Z**
 - **State Variables: A, B, C**
- **Initialization: Reset to (0,0,0)**
- **Equations**
 - $A(t+1) = BC$
 - $B(t+1) = B'C + BC'$
 - $C(t+1) = A' C'$
 - $Z = A$

Example 2: State Table

$X' = X(t+1)$

A B C	A'B'C'	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



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

Terms of Use

- **All (or portions) of this material © 2008 by Pearson Education, Inc.**
- **Permission is given to incorporate this material or adaptations thereof into classroom presentations and handouts to instructors in courses adopting the latest edition of Logic and Computer Design Fundamentals as the course textbook.**
- **These materials or adaptations thereof are not to be sold or otherwise offered for consideration.**
- **This Terms of Use slide or page is to be included within the original materials or any adaptations thereof.**