UNIT 14

Derivation of State Graphs and Tables

This chapter includes:

14.1 Design of a Sequence Detector
14.2 More Complex Design Problems
14.3 Guidelines for Construction of State Graphs
14.4 Serial Data Code Conversion
14.5 Alphanumeric State Graph Notation
14.6 Incompletely Specified State Tables

Learning Objectives

1. Given a problem statement for the design of a Mealy or Moore sequential circuit, find the corresponding state graph and table.
2. Explain the significance of each state in your graph or table in terms of the input sequences required to reach that state.
3. Check your state graph using appropriate input sequences.

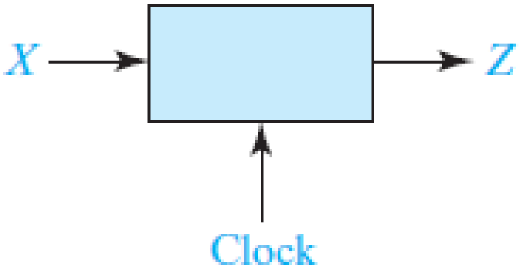
Design of a Sequence Detector

Sequence Detector:

- The circuit will examine a string of 0's and 1's applied to the X input and generate an output Z = 1 only when a prescribed input sequence occurs.
- It will be assumed that the input X can only change between clock pulses.

FIGURE 14-1 Sequence Detector to Be Designed

© Cengage Learning 2014



- The following slides use the example of a Mealy machine. See pages 460 to 463 for an example using a Moore machine.

Design of a Sequence Detector

State graphs:

FIGURE 14-2

© Cengage Learning 2014

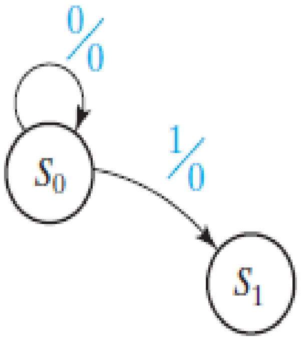


FIGURE 14-3

© Cengage Learning 2014

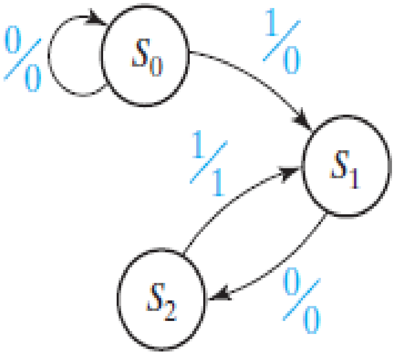
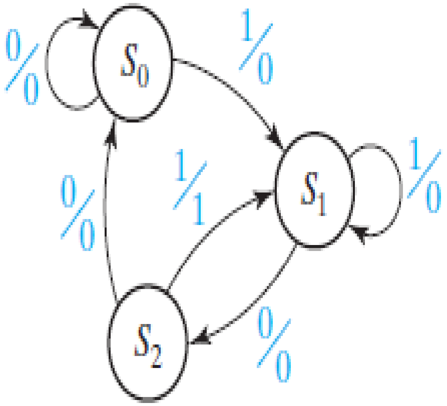


FIGURE 14-4
Mealy State Graph for Sequence Detector

© Cengage Learning 2014

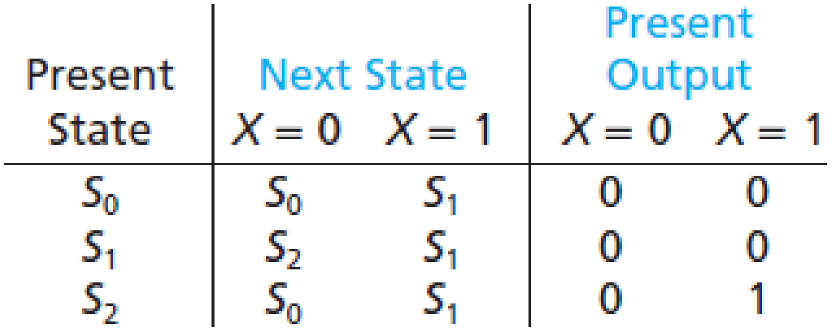


Design of a Sequence Detector

State Tables:

TABLE 14-1

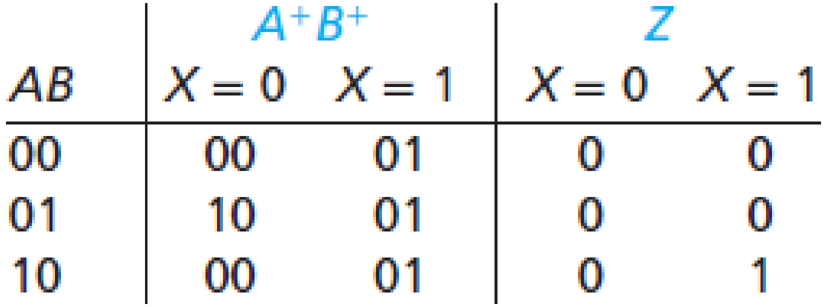
© Cengage Learning 2014



Present State	Next State		Present Output	
	$X = 0$	$X = 1$	$X = 0$	$X = 1$
S_0	S_0	S_1	0	0
S_1	S_2	S_1	0	0
S_2	S_0	S_1	0	1

TABLE 14-2

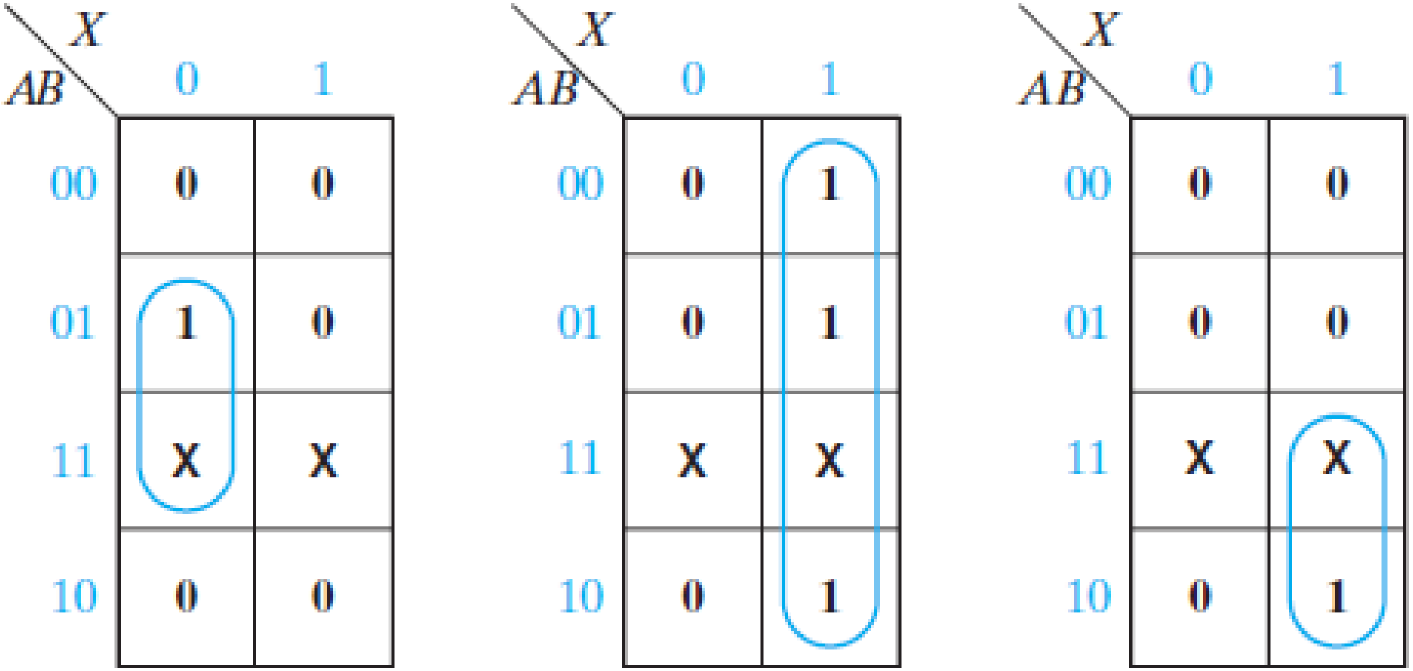
© Cengage Learning 2014



	A^+B^+		Z	
AB	$X = 0$	$X = 1$	$X = 0$	$X = 1$
00	00	01	0	0
01	10	01	0	0
10	00	01	0	1

Design of a Sequence Detector

Flip- Flop Next-State Maps and Map for output Z:



AB \ X	0	1
00	0	0
01	1	0
11	X	X
10	0	0

$A^+ = X'B$

AB \ X	0	1
00	0	1
01	0	1
11	X	X
10	0	1

$B^+ = X$

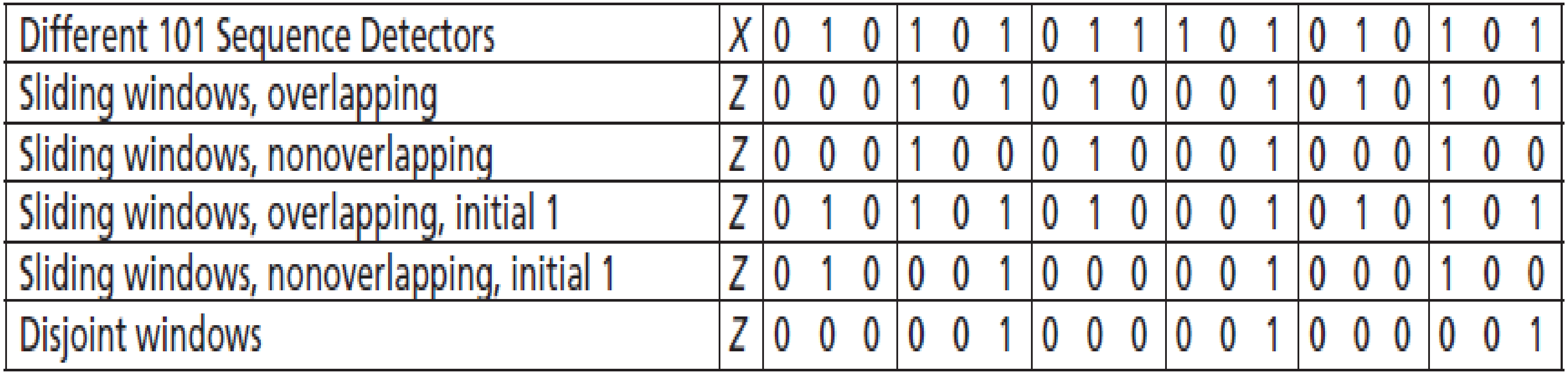
AB \ X	0	1
00	0	0
01	0	0
11	X	X
10	0	1

$Z = XA$

Design of a Sequence Detector

Different Types of Sequence Detectors:

FIGURE 14-7 Different Types of Sequence Detector

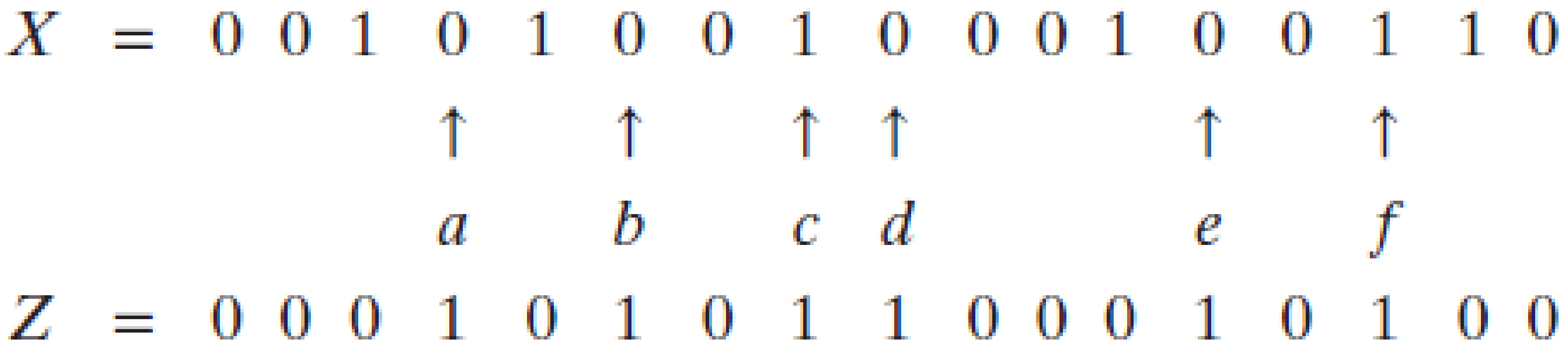


Different 101 Sequence Detectors	X	0	1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	0	1
Sliding windows, overlapping	Z	0	0	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	1
Sliding windows, nonoverlapping	Z	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
Sliding windows, overlapping, initial 1	Z	0	1	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	1
Sliding windows, nonoverlapping, initial 1	Z	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0
Disjoint windows	Z	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1

© Cengage Learning 2014

More Complex Design Problems

Desired output sequence Z for input sequence X:



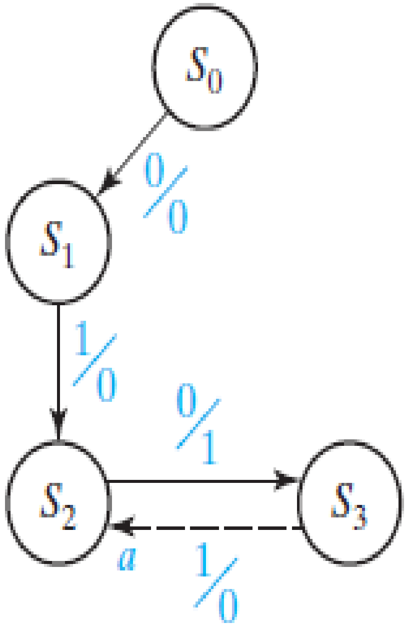
$$\begin{array}{ccccccccccccccccccc}
X & = & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
 & & & & & \uparrow & & \uparrow & & \uparrow & \uparrow & & & & \uparrow & & \uparrow & & \\
 & & & & & a & & b & & c & d & & & & e & & f & & \\
Z & = & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0
\end{array}$$

More Complex Design Problems

State Graph:

FIGURE 14-8

© Cengage Learning 2014

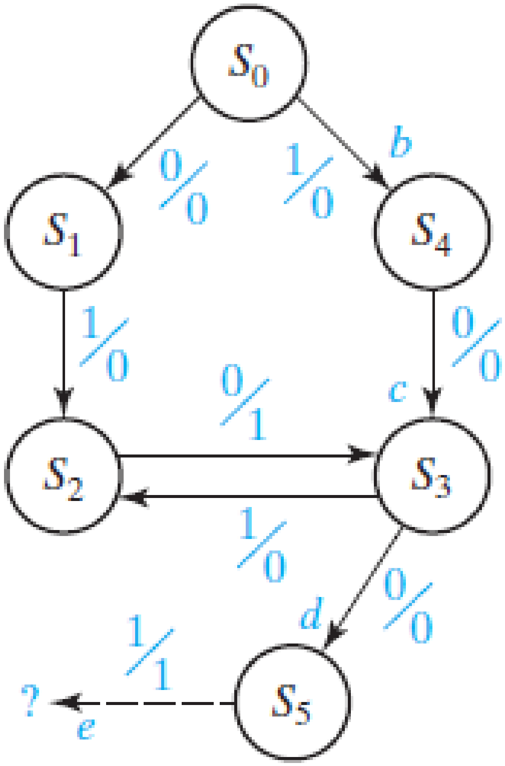


State	Sequence Received
S_0	Reset
S_1	0
S_2	01
S_3	010

More Complex Design Problems

FIGURE 14-9

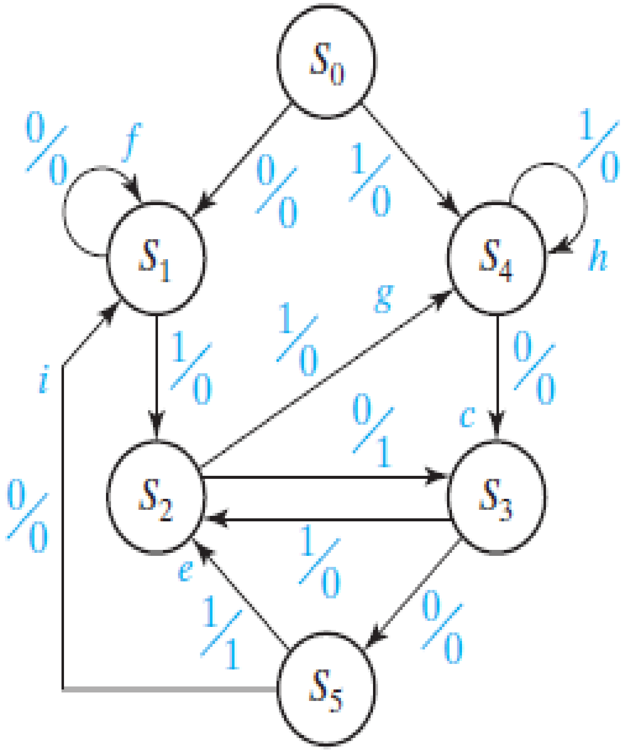
© Cengage Learning 2014



State	Sequence Ends in
S_0	Reset
S_1	0 (but not 10)
S_2	01
S_3	10
S_4	1 (but not 01)
S_5	100

More Complex Design Problems

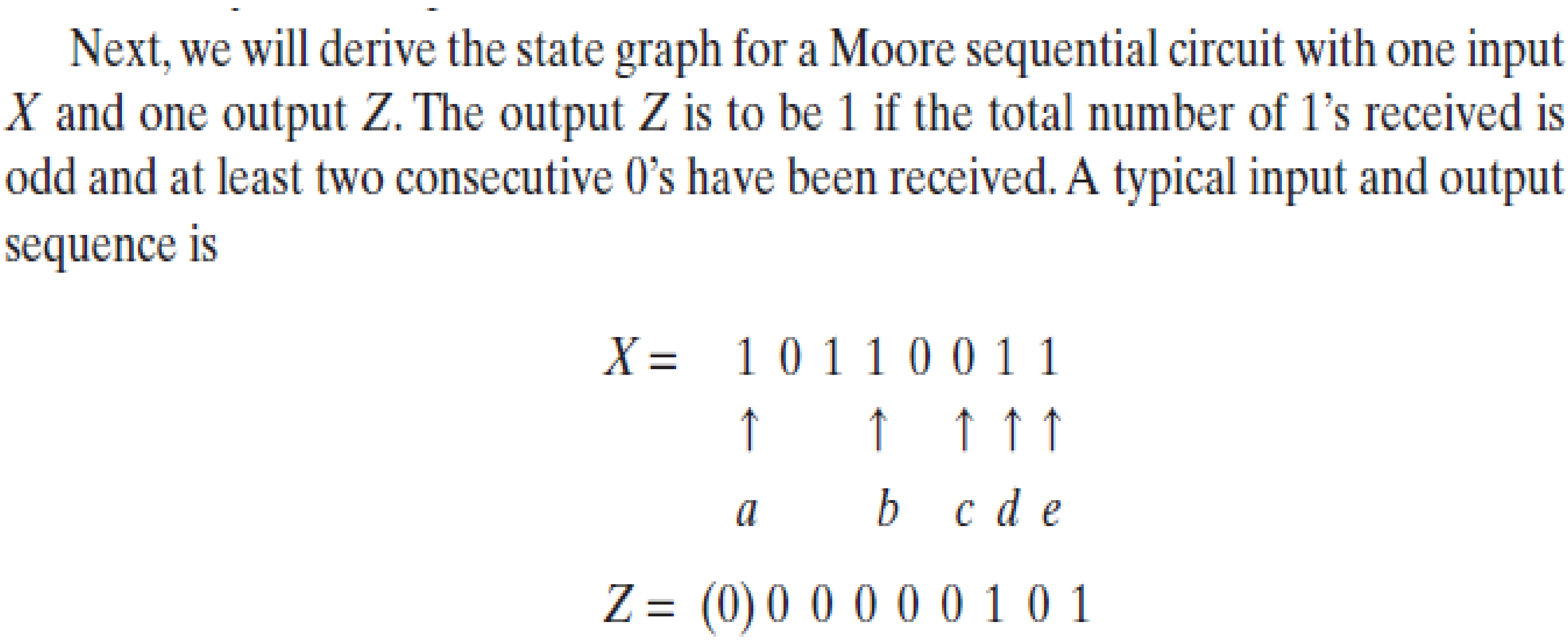
FIGURE 14-10



State	Sequence Ends in
S_0	Reset
S_1	0 (but not 10)
S_2	01
S_3	10
S_4	1 (but not 01)
S_5	100

More Complex Design Problems

Moore Sequential Circuit:



Next, we will derive the state graph for a Moore sequential circuit with one input X and one output Z. The output Z is to be 1 if the total number of 1's received is odd and at least two consecutive 0's have been received. A typical input and output sequence is

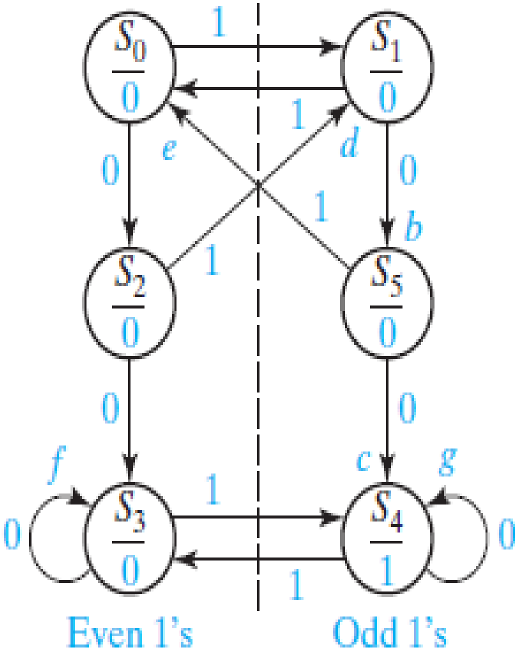
$$\begin{array}{rccccccccc} X = & & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ & & \uparrow & & & \uparrow & & \uparrow & \uparrow & \uparrow \\ & & a & & & b & & c & d & e \\ Z = & (0) & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{array}$$

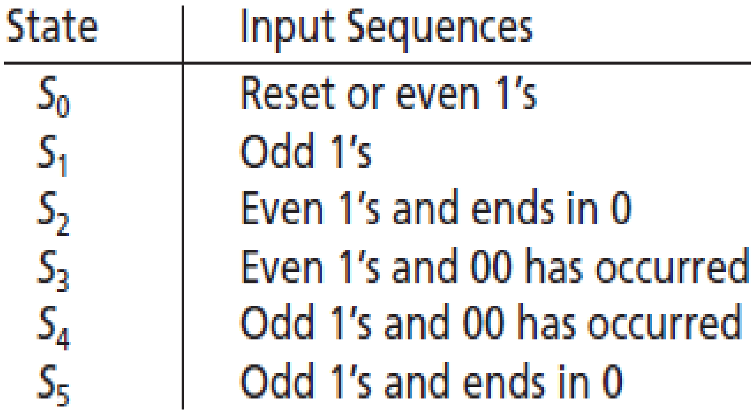
More Complex Design Problems

Complete Moore State Graph and Table:

FIGURE 14-13

© Cengage Learning 2014





State	Input Sequences
S_0	Reset or even 1's
S_1	Odd 1's
S_2	Even 1's and ends in 0
S_3	Even 1's and 00 has occurred
S_4	Odd 1's and 00 has occurred
S_5	Odd 1's and ends in 0

Guidelines for Construction of State Graphs

Guidelines:

1. First, construct some sample input and output sequences to make sure that you understand the problem statement.

2. Determine under what conditions, if any, the circuit should reset to its initial state.

3. If only one or two sequences lead to a nonzero output, a good way to start is to construct a partial state graph for those sequences.

Guidelines for Construction of State Graphs

Guidelines (continued):

4. Another way to get started is to determine what sequences or groups of sequences must be remembered by the circuit and set up states accordingly.

5. Each time you add an arrow to the state graph, determine whether it can go to one of the previously defined states or whether a new state must be added.

Guidelines for Construction of State Graphs

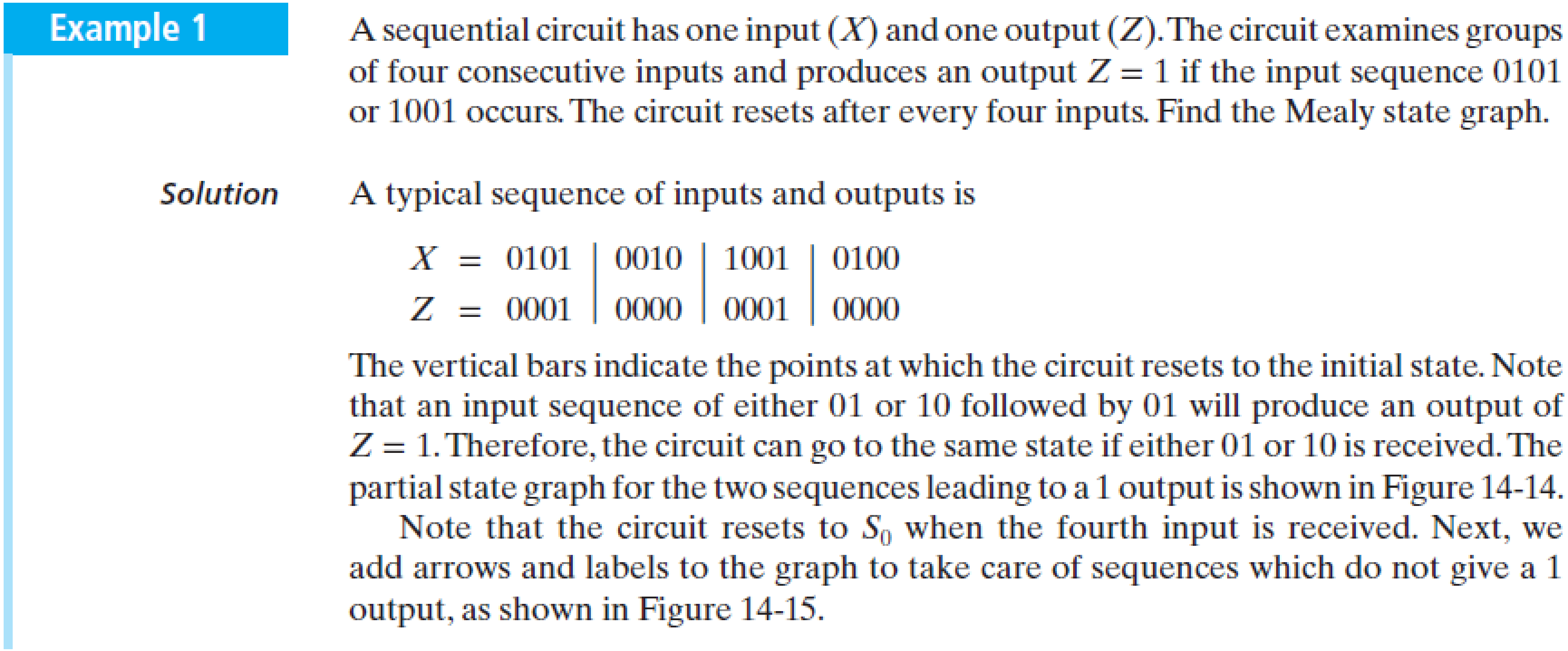
Guidelines (continued):

6. Check your graph to make sure there is one and only one path leaving each state for each combination of values of the input variables.

7. When your graph is complete, test it by applying the input sequences formulated in part 1 and making sure the output sequences are correct.

Guidelines for Construction of State Graphs

Example 1:



Example 1

A sequential circuit has one input (X) and one output (Z). The circuit examines groups of four consecutive inputs and produces an output $Z = 1$ if the input sequence 0101 or 1001 occurs. The circuit resets after every four inputs. Find the Mealy state graph.

Solution A typical sequence of inputs and outputs is

$$\begin{array}{rcl|l|l|l} X & = & 0101 & 0010 & 1001 & 0100 \\ Z & = & 0001 & 0000 & 0001 & 0000 \end{array}$$

The vertical bars indicate the points at which the circuit resets to the initial state. Note that an input sequence of either 01 or 10 followed by 01 will produce an output of $Z = 1$. Therefore, the circuit can go to the same state if either 01 or 10 is received. The partial state graph for the two sequences leading to a 1 output is shown in Figure 14-14.

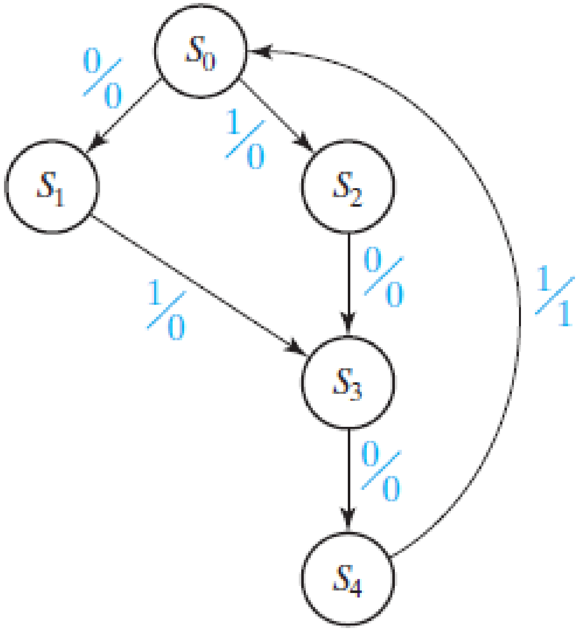
Note that the circuit resets to S_0 when the fourth input is received. Next, we add arrows and labels to the graph to take care of sequences which do not give a 1 output, as shown in Figure 14-15.

Guidelines for Construction of State Graphs

Example 1 (continued):

FIGURE 14-14 Partial State Graph for Example 1

© Cengage Learning 2014



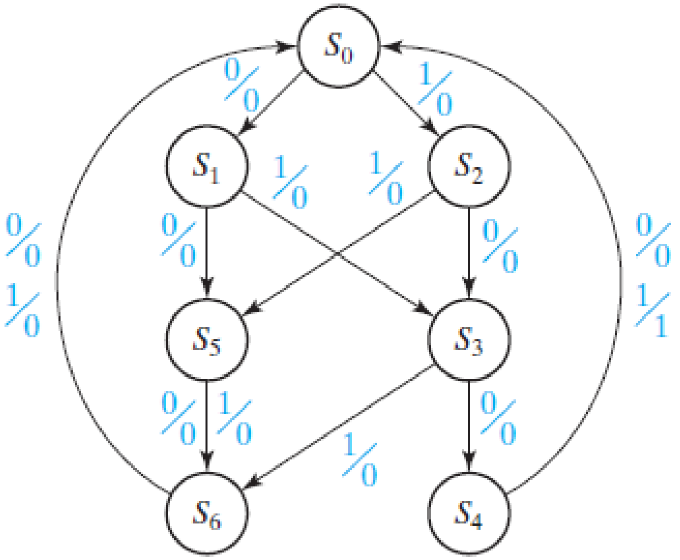
State	Sequence Received
S_0	Reset
S_1	0
S_2	1
S_3	01 or 10
S_4	010 or 100

Guidelines for Construction of State Graphs

Example 1 (continued):

FIGURE 14-15 Complete State Graph for Example 1

© Cengage Learning 2014



State	Sequence Received
S_0	Reset
S_1	0
S_2	1
S_3	01 or 10
S_4	010 or 100
S_5	Two inputs received, no 1 output is possible
S_6	Three inputs received, no 1 output is possible

- See pages 469-472 for more examples.

Serial Data Code Conversion

Serial Data Code Converter:

- Binary data is frequently transmitted between computers as a serial stream of bits.

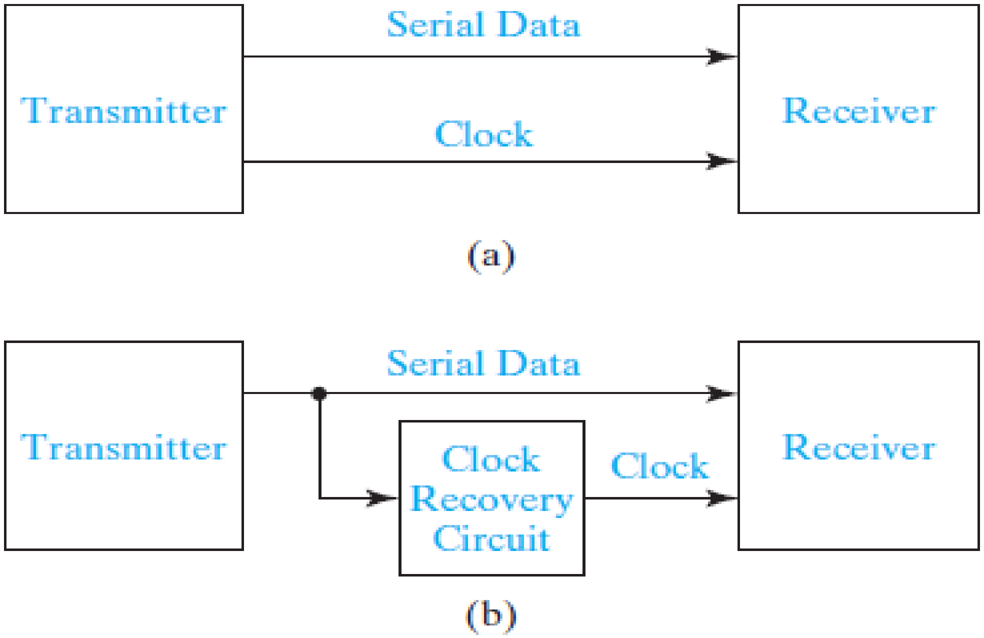


FIGURE 14-19
Serial Data Transmission

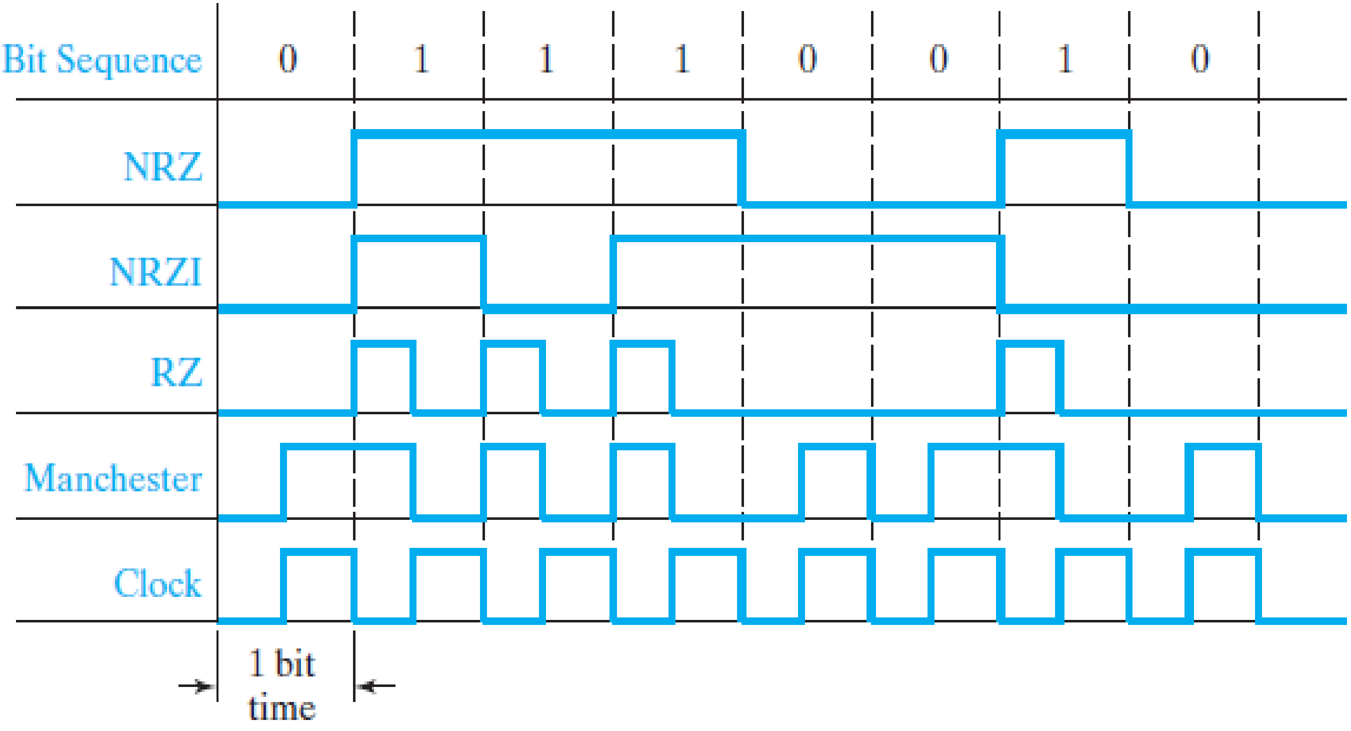
© Cengage Learning 2014

Serial Data Code Conversion

Coding Schemes for Serial Data Transmission:

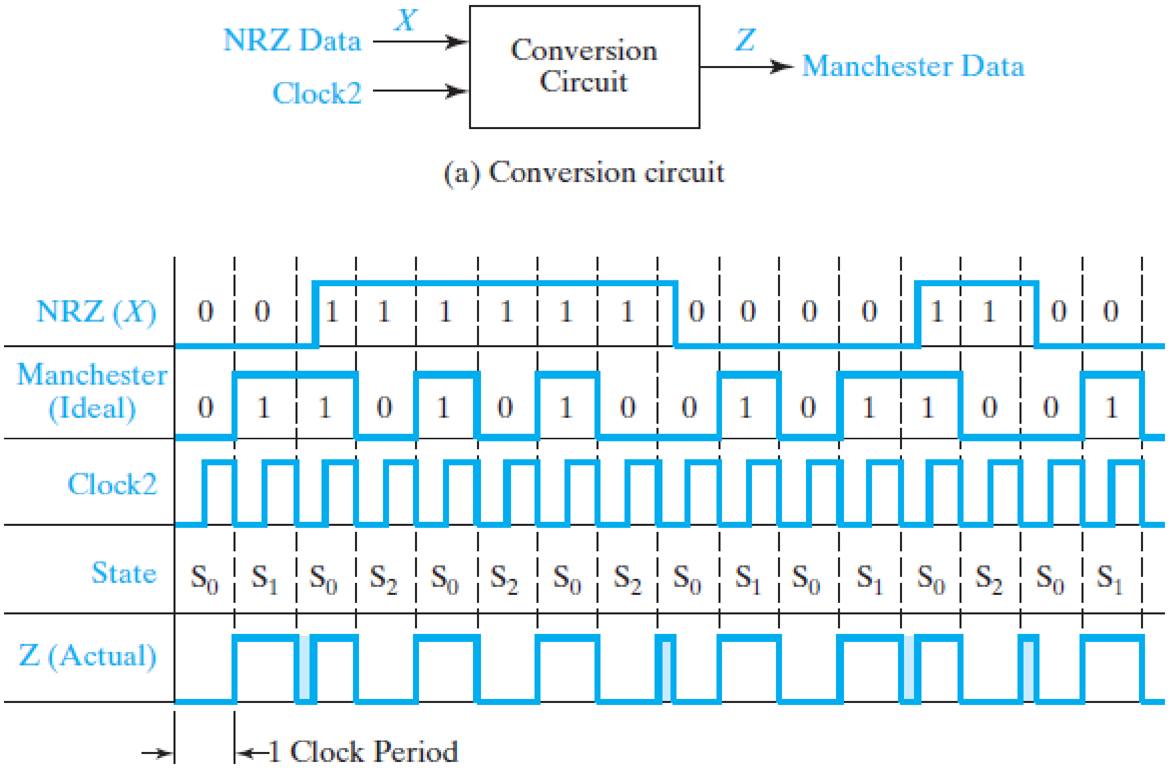
FIGURE 14-20
Coding Schemes for Serial Data Transmission

© Cengage Learning 2014



Serial Data Code Conversion

Mealy Circuit for NRZ to Manchester Conversion:



NRZ (X)	0	0	1	1	1	1	1	1	0	0	0	0	1	1	0	0
Manchester (Ideal)	0	1	1	0	1	0	1	0	0	1	0	1	1	0	0	1
State	S_0	S_1	S_0	S_2	S_0	S_2	S_0	S_2	S_0	S_1	S_0	S_1	S_0	S_2	S_0	S_1

(a) Conversion circuit

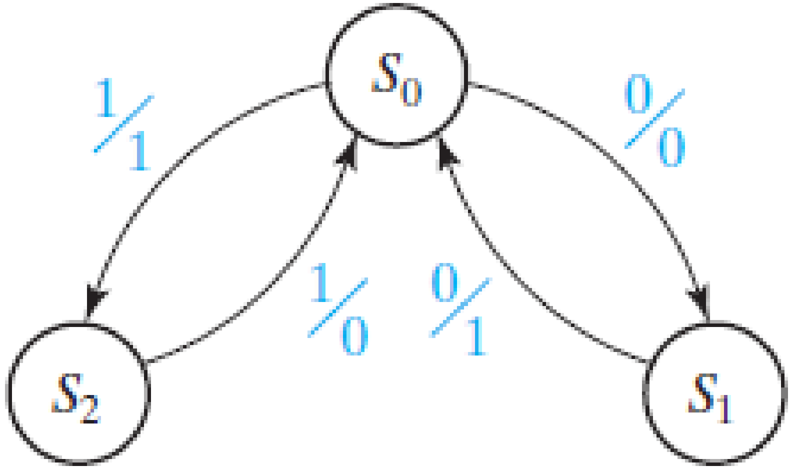
(b) Timing chart

FIGURE 14-21 Mealy Circuit for NRZ to Manchester Conversion

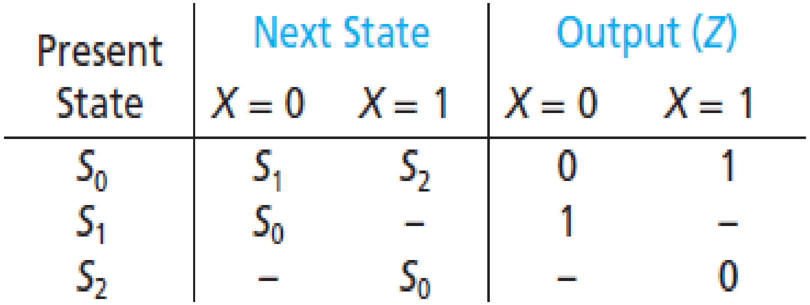
© Cengage Learning 2014

Serial Data Code Conversion

Mealy Circuit for NRZ to Manchester Conversion (continued):



(c) State graph

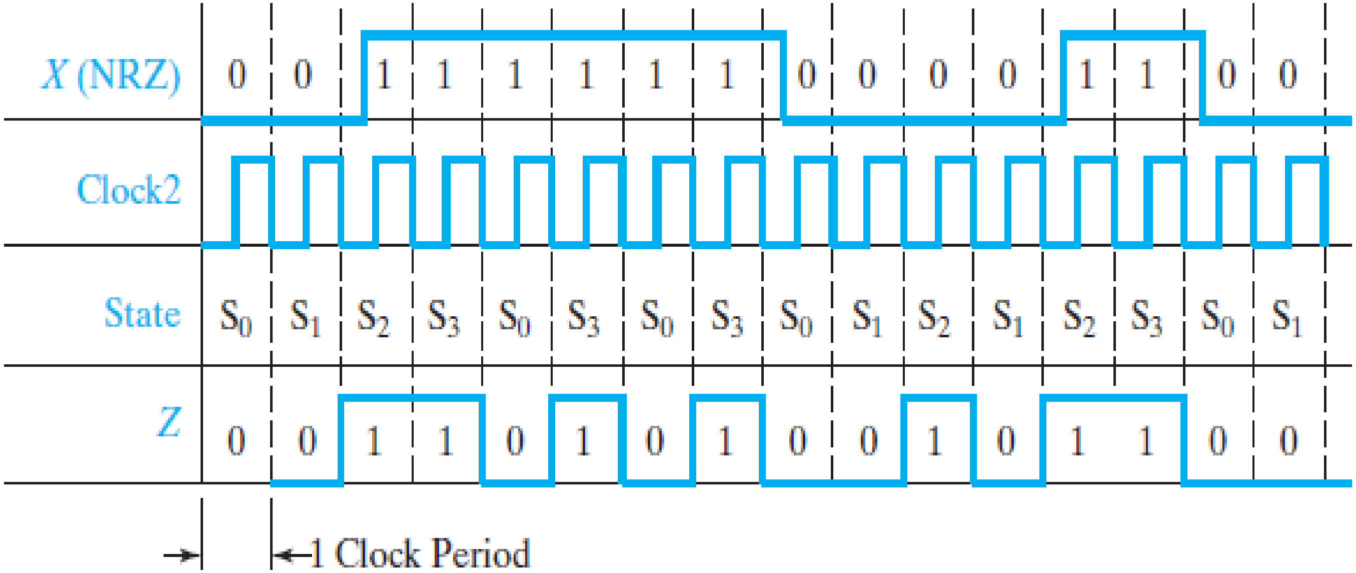


Present State	Next State		Output (Z)	
	$X = 0$	$X = 1$	$X = 0$	$X = 1$
S_0	S_1	S_2	0	1
S_1	S_0	–	1	–
S_2	–	S_0	–	0

(d) State table

Serial Data Code Conversion

Moore Circuit for NRZ to Manchester Conversion:



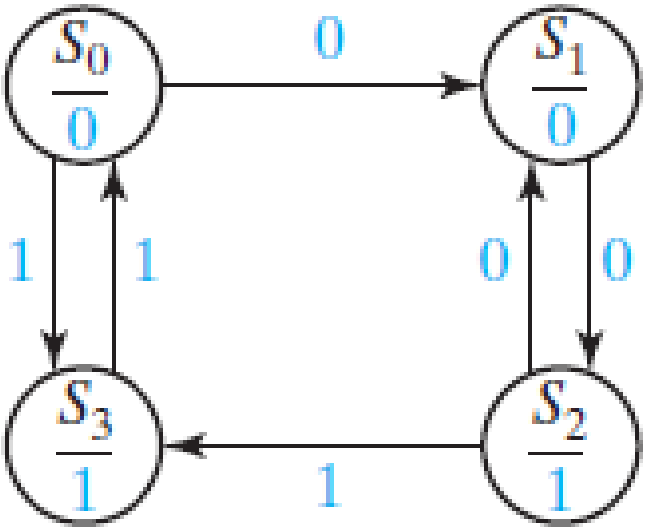
(a) Timing chart

FIGURE 14-22 Moore Circuit for NRZ to Manchester Conversion

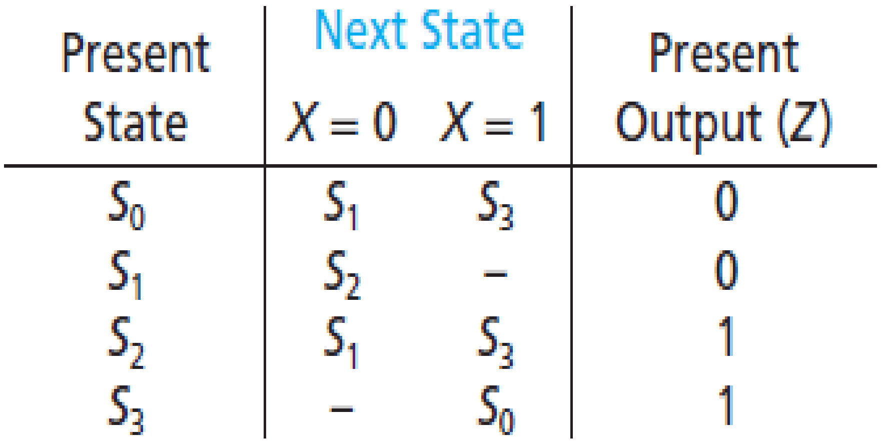
© Cengage Learning 2014

Serial Data Code Conversion

Moore Circuit for NRZ to Manchester Conversion (continued):



(b) State graph



Present State	Next State X = 0	Next State X = 1	Present Output (Z)
S_0	S_1	S_3	0
S_1	S_2	–	0
S_2	S_1	S_3	1
S_3	–	S_0	1

(c) State table

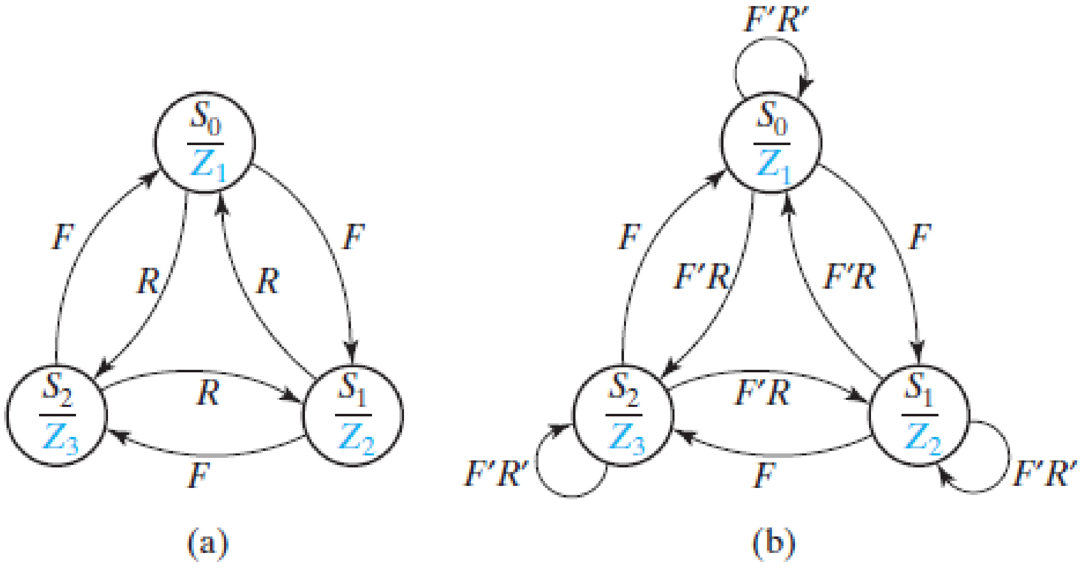
Alphanumeric State Graph Notation

Alphanumeric State Graph Notation:

- When a state sequential circuit has several inputs, it is often convenient to label the state graph arcs with alphanumeric input variable names instead of 0's and 1's.

FIGURE 14-23
State Graphs with Variable Names on Arc Labels

© Cengage Learning 2014



Alphanumeric State Graph Notation

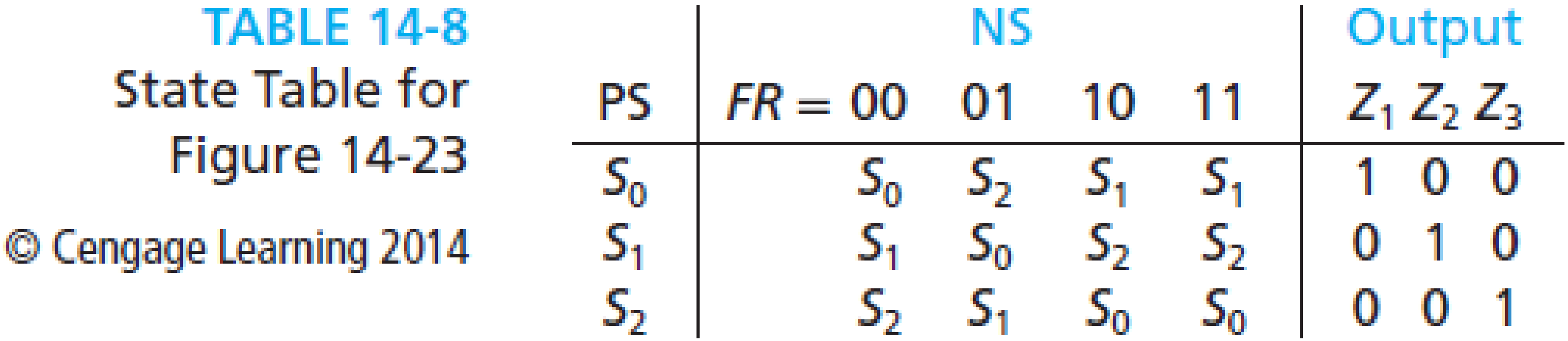


TABLE 14-8
State Table for
Figure 14-23

© Cengage Learning 2014

	NS				Output		
PS	$FR = 00$	01	10	11	Z_1	Z_2	Z_3
S_0	S_0	S_2	S_1	S_1	1	0	0
S_1	S_1	S_0	S_2	S_2	0	1	0
S_2	S_2	S_1	S_0	S_0	0	0	1

Alphanumeric State Graph Notation

Properties of Completely Specified State Graphs:

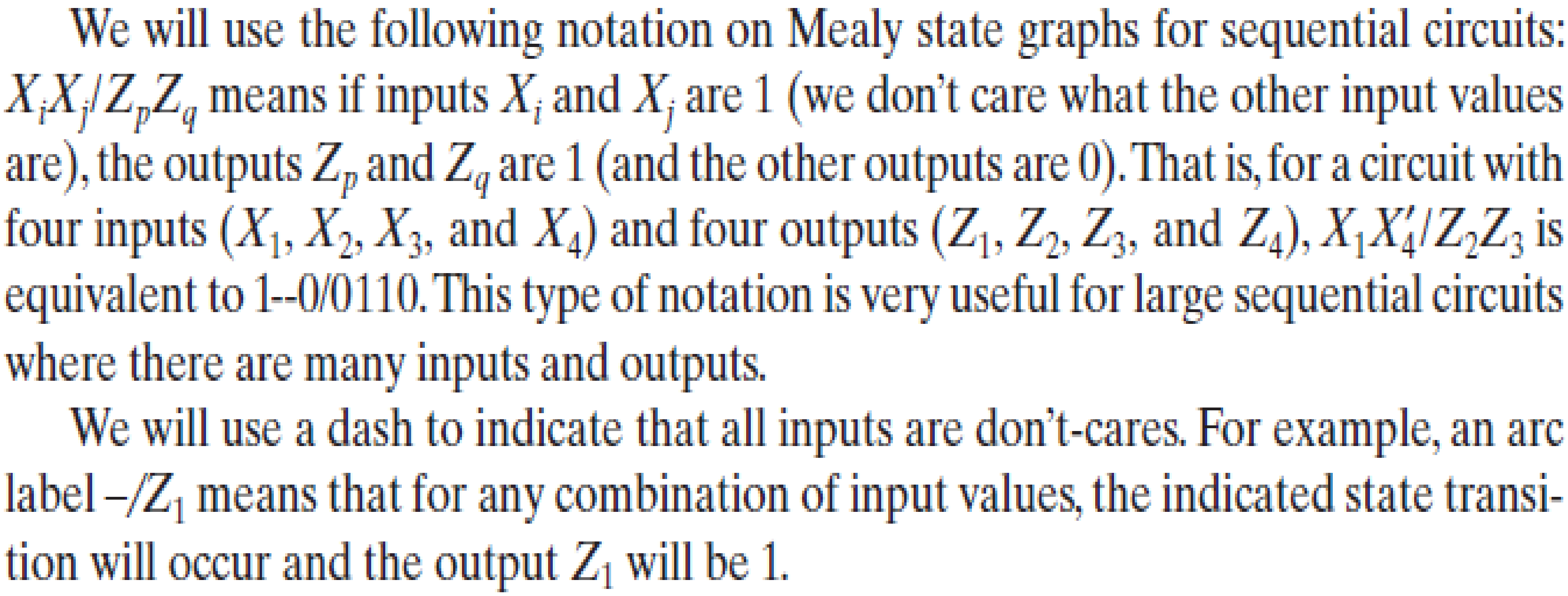
(1) When we OR together all input labels on arcs emanating from a state, the result reduces to 1.

(2) When we AND together any pair of input labels on arcs emanating from a state, the result is 0.

- Property (1) ensures that for every input combination, at least one next state is defined. Property (2) ensures that for every input combination, no more than one next state is defined.
- If both properties are true, then exactly one next state is defined, and the graph is properly specified.

Alphanumeric State Graph Notation

Mealy State Graph Notation:



We will use the following notation on Mealy state graphs for sequential circuits: X_iX_j/Z_pZ_q means if inputs X_i and X_j are 1 (we don't care what the other input values are), the outputs Z_p and Z_q are 1 (and the other outputs are 0). That is, for a circuit with four inputs (X_1, X_2, X_3, and X_4) and four outputs (Z_1, Z_2, Z_3, and Z_4), X_1X_4'/Z_2Z_3 is equivalent to 1--0/0110. This type of notation is very useful for large sequential circuits where there are many inputs and outputs.

We will use a dash to indicate that all inputs are don't-cares. For example, an arc label $–/Z_1$ means that for any combination of input values, the indicated state transition will occur and the output Z_1 will be 1.

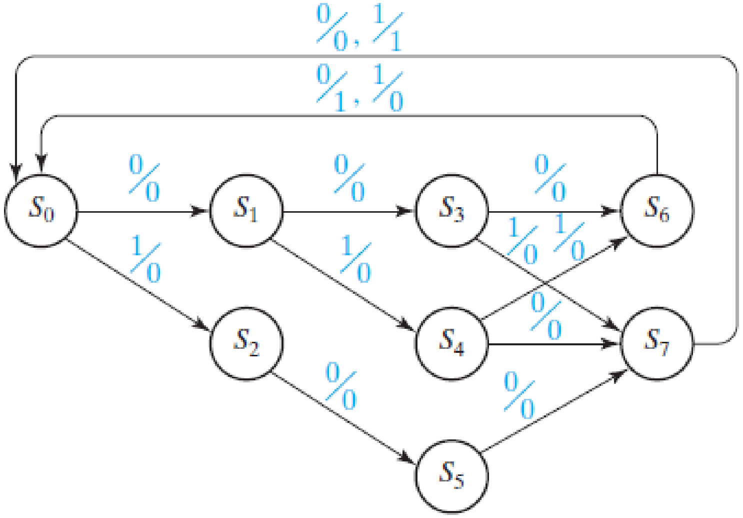
Incompletely Specified State Tables

BCD Parity Detector:

A Mealy model sequential circuit examines the input sequence and generates an output of 1 when the last bit of a BCD digit is received if the parity of the digit is even; otherwise, the sequential circuit output is 0.

FIGURE 14-24
State Graph for
BCD Parity Detector

© Cengage Learning 2014



Incompletely Specified State Tables

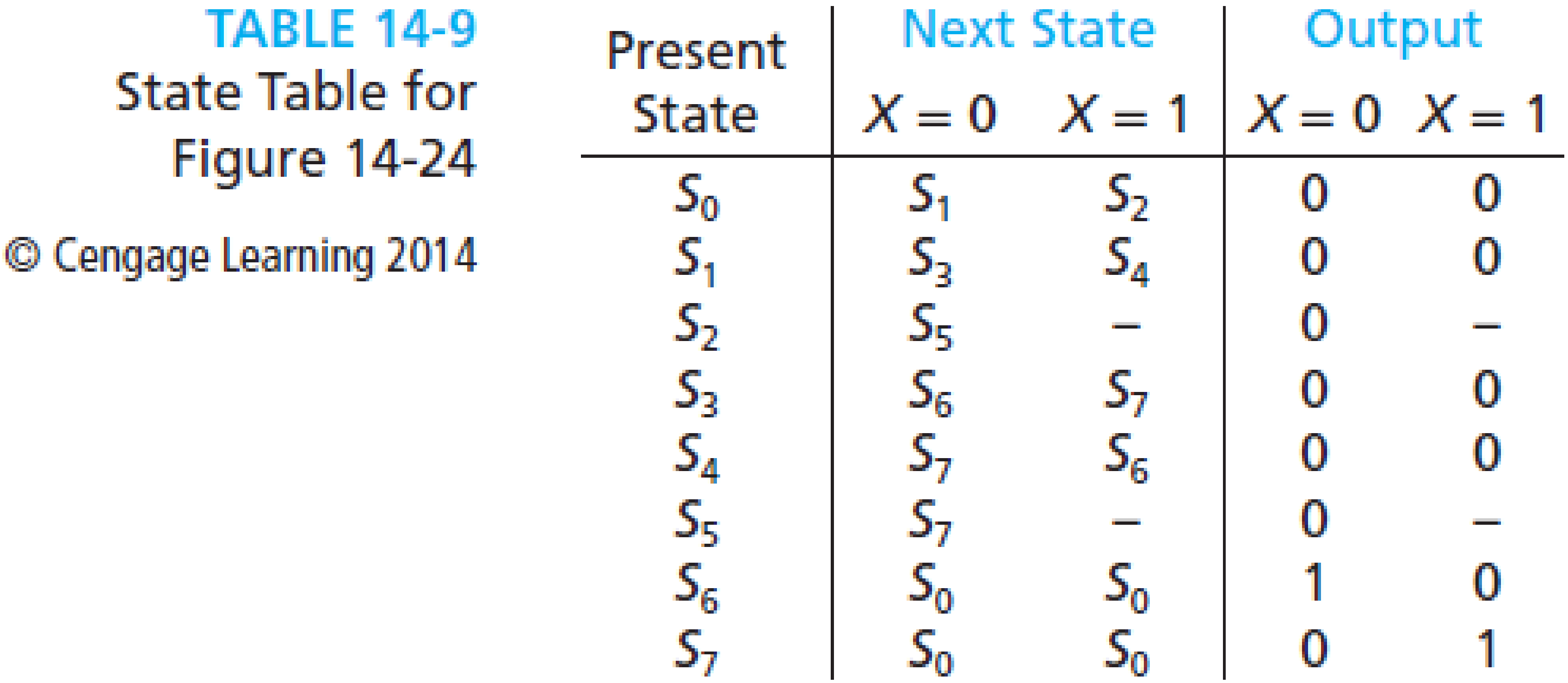


TABLE 14-9
State Table for Figure 14-24

© Cengage Learning 2014

Present State	Next State		Output	
	$X = 0$	$X = 1$	$X = 0$	$X = 1$
S_0	S_1	S_2	0	0
S_1	S_3	S_4	0	0
S_2	S_5	–	0	–
S_3	S_6	S_7	0	0
S_4	S_7	S_6	0	0
S_5	S_7	–	0	–
S_6	S_0	S_0	1	0
S_7	S_0	S_0	0	1

Incompletely Specified State Tables

FIGURE 14-25
Disjoint Window
101 Detector

© Cengage Learning 2014

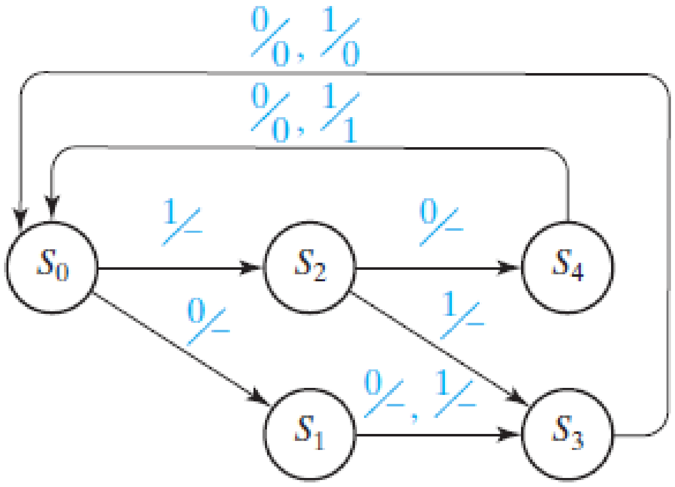


TABLE 14-10
State Table for
Figure 14-25

© Cengage Learning 2014

Present State	Next State		Output	
	$X = 0$	$X = 1$	$X = 0$	$X = 1$
S_0	S_1	S_2	–	–
S_1	S_3	S_3	–	–
S_2	S_4	S_3	–	–
S_3	S_0	S_0	0	0
S_4	S_0	S_0	0	1