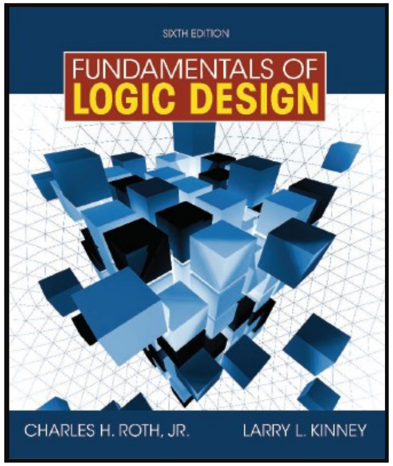
CHAPTER 13

ANALYSIS OF CLOCKED SEQUENTIAL CIRCUITS



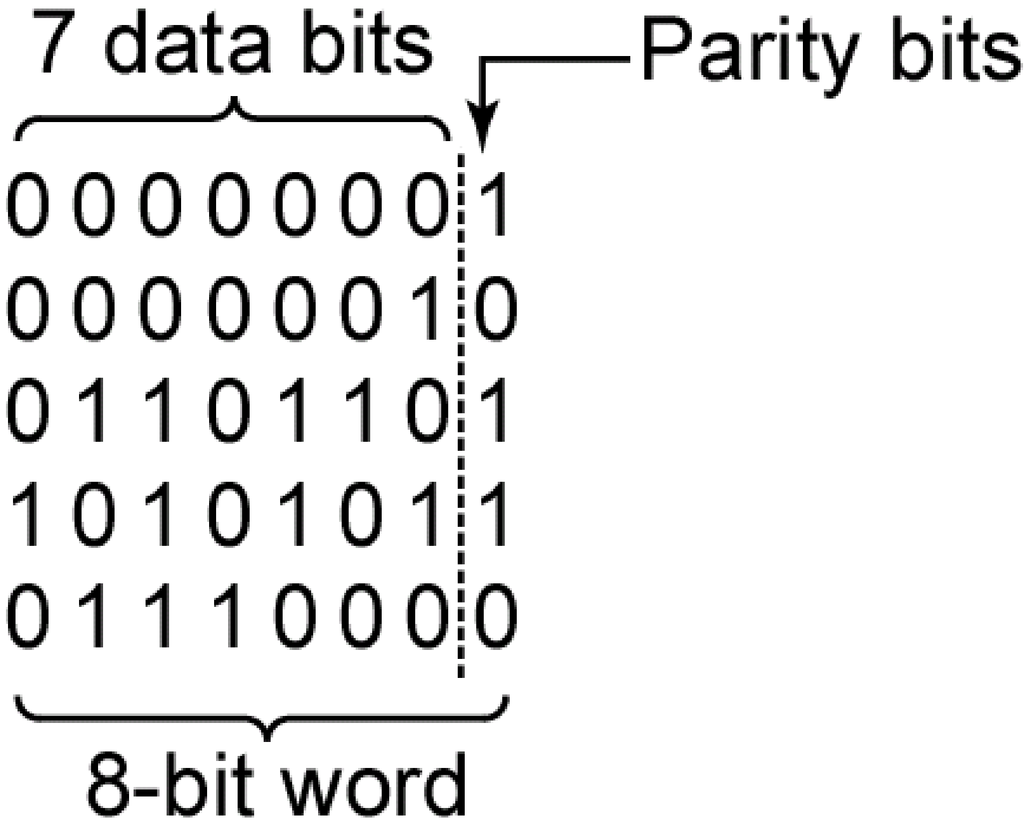
This chapter in the book includes:

A Sequential Parity Checker

When binary data is transmitted or stored, an extra bit (called a parity bit) is frequently added for purposes of error detection. For example, if data is being transmitted in groups of 7 bits, an eighth bit can be added to each group of 7 bits to make the total number of 1′s in each block of 8 bits an odd number.

When the total number of 1 bits in the block (including the parity bit) is odd, we say that the parity is odd.

Section 13.1 (p. 394)



7 data bits

Parity bits

0 0 0 0 0 0 0 1

0 0 0 0 0 0 1 0

0 1 1 0 1 1 0 1

1 0 1 0 1 0 1 1

0 1 1 1 0 0 0 0

8-bit word

Section 13.1, p. 395

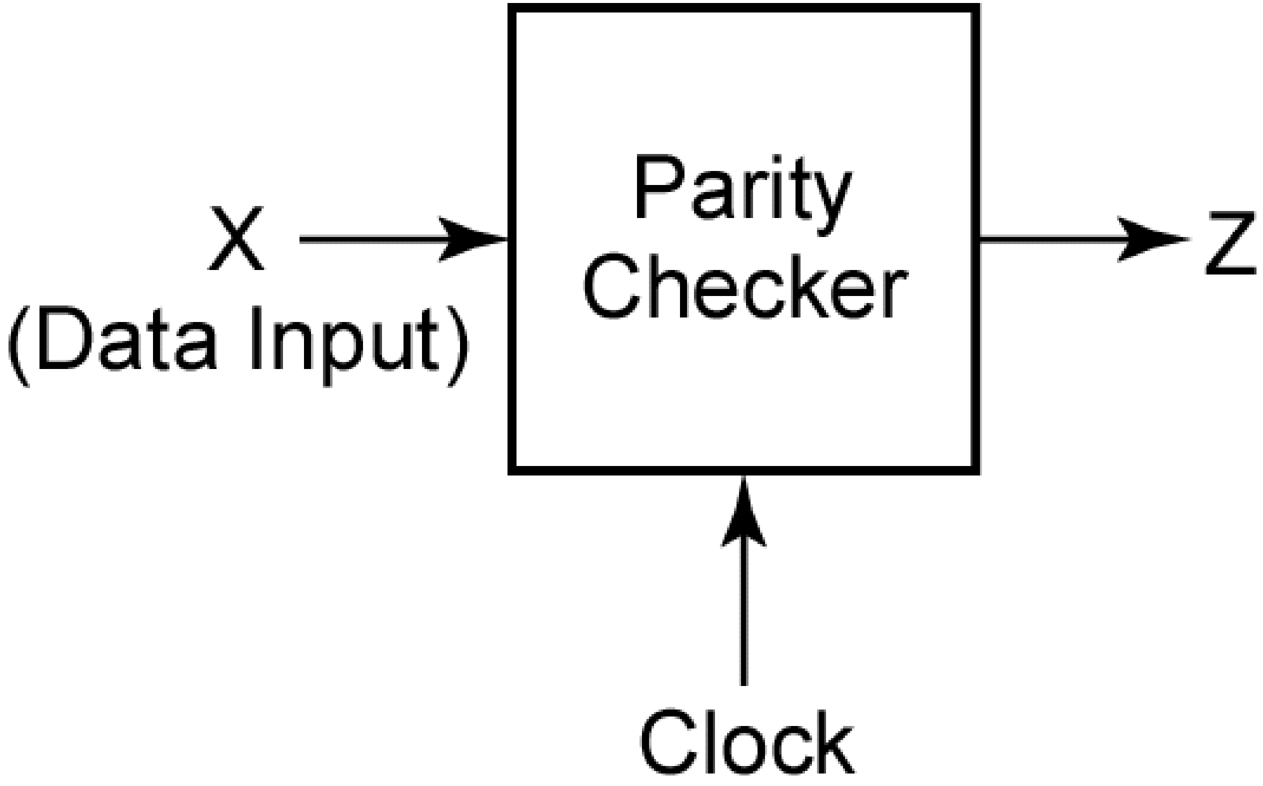


Figure 13-1: **Block Diagram for Parity Checker**

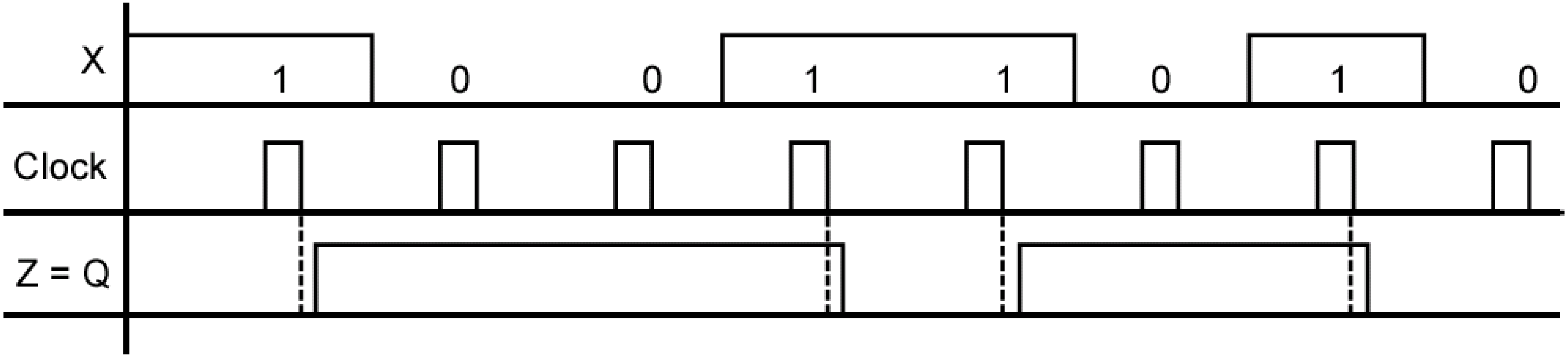


Figure 13-2: **Waveforms for Parity Checker**

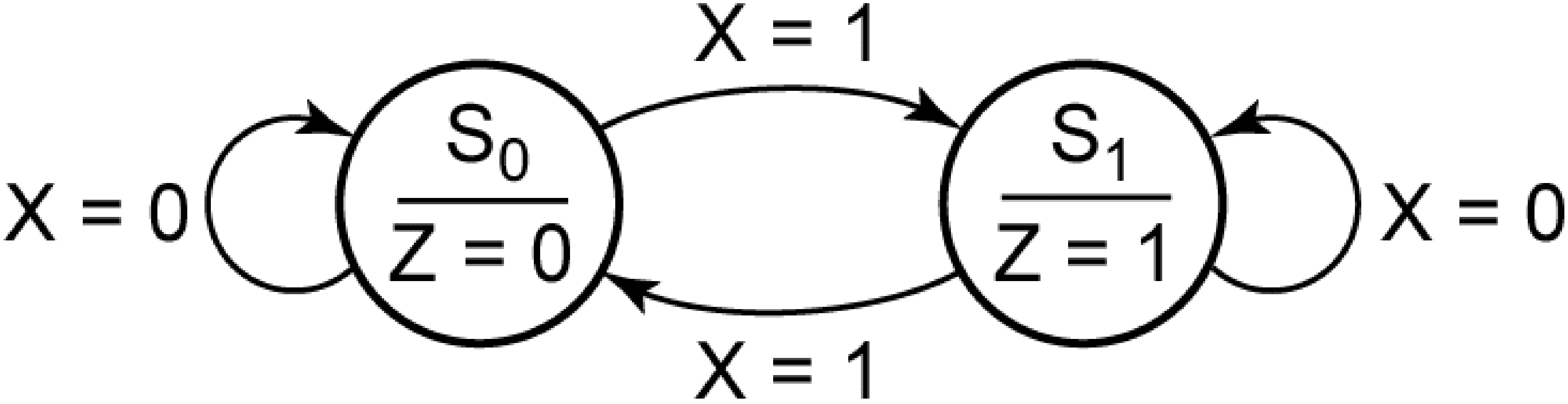


Figure 13-3: **State Graph for Parity Checker**

Table 13-1: State Table for Parity Checker

(a)

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

(b)

	Q^+		T		
Q	$X = 0$	$X = 1$	$X = 0$	$X = 1$	Z
0	0	1	0	1	0
1	1	0	0	1	1

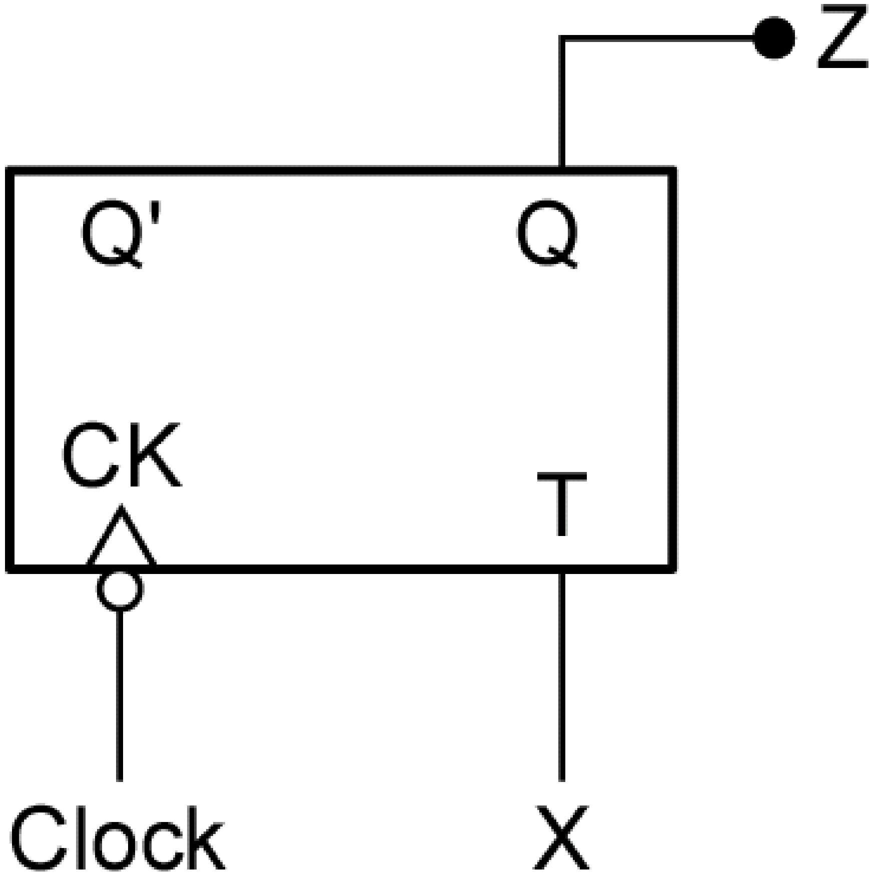


Figure 13-4: **Parity Checker**

Analysis by Signal Tracing and Timing Charts

We can analyze clocked sequential circuits to find the output sequence resulting from a given input sequence by tracing 0 and 1 signals through the circuit. The basic procedure is:

1. Assume an initial state of the flip-flops (all flip-flops reset to 0 unless otherwise specified).
2. For the first input in the given sequence, determine the circuit output(s) and flip-flop inputs.
3. Determine the new set of flip-flop states after the next active clock edge.
4. Determine the output(s) that corresponds to the new states.
5. Repeat 2, 3, and 4 for each input in the given sequence.

Section 13.2 (p. 397)

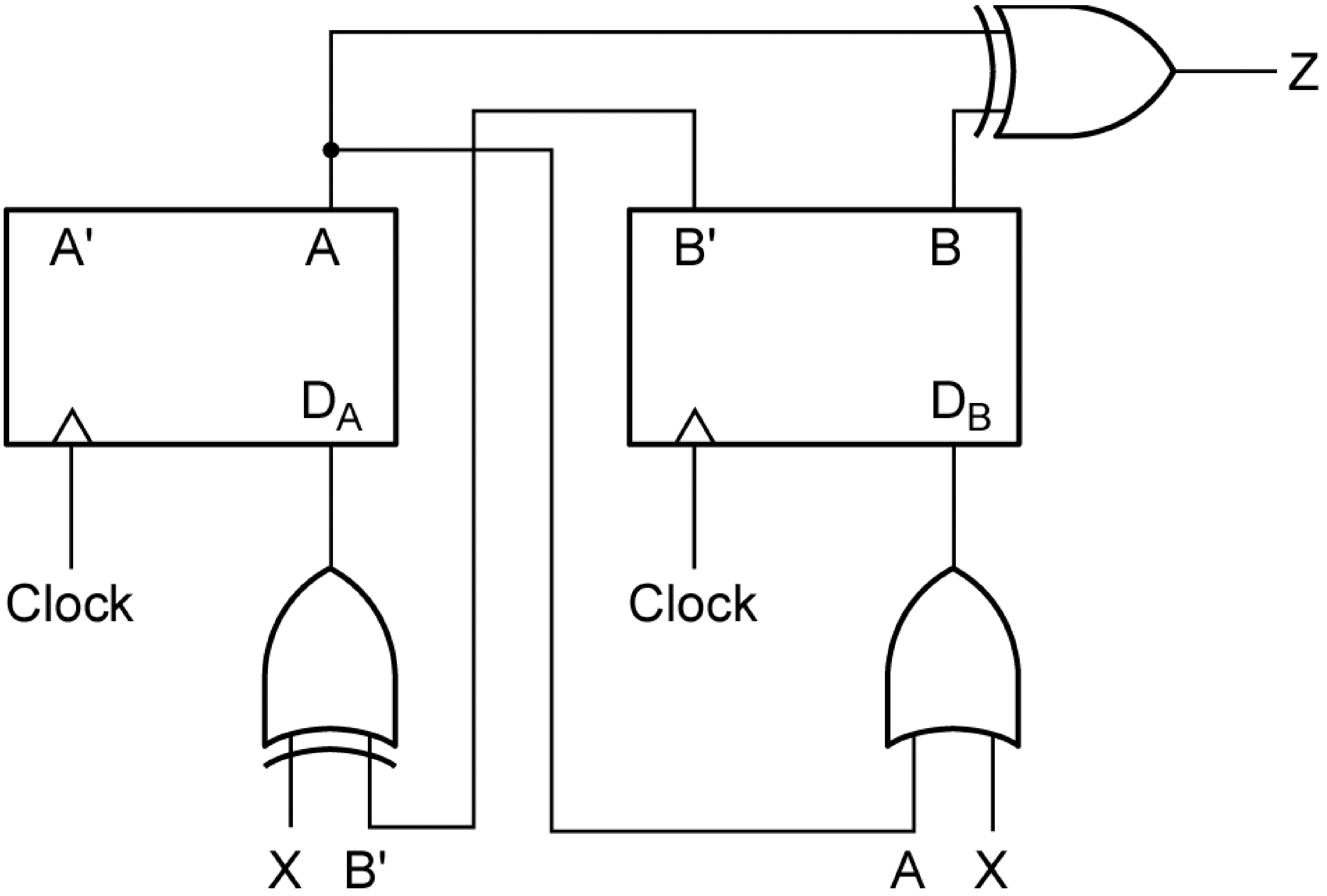


Figure 13-5: **Moore Sequential Circuit to be Analyzed**

Analysis of previous circuit for input sequence *X* = 01101

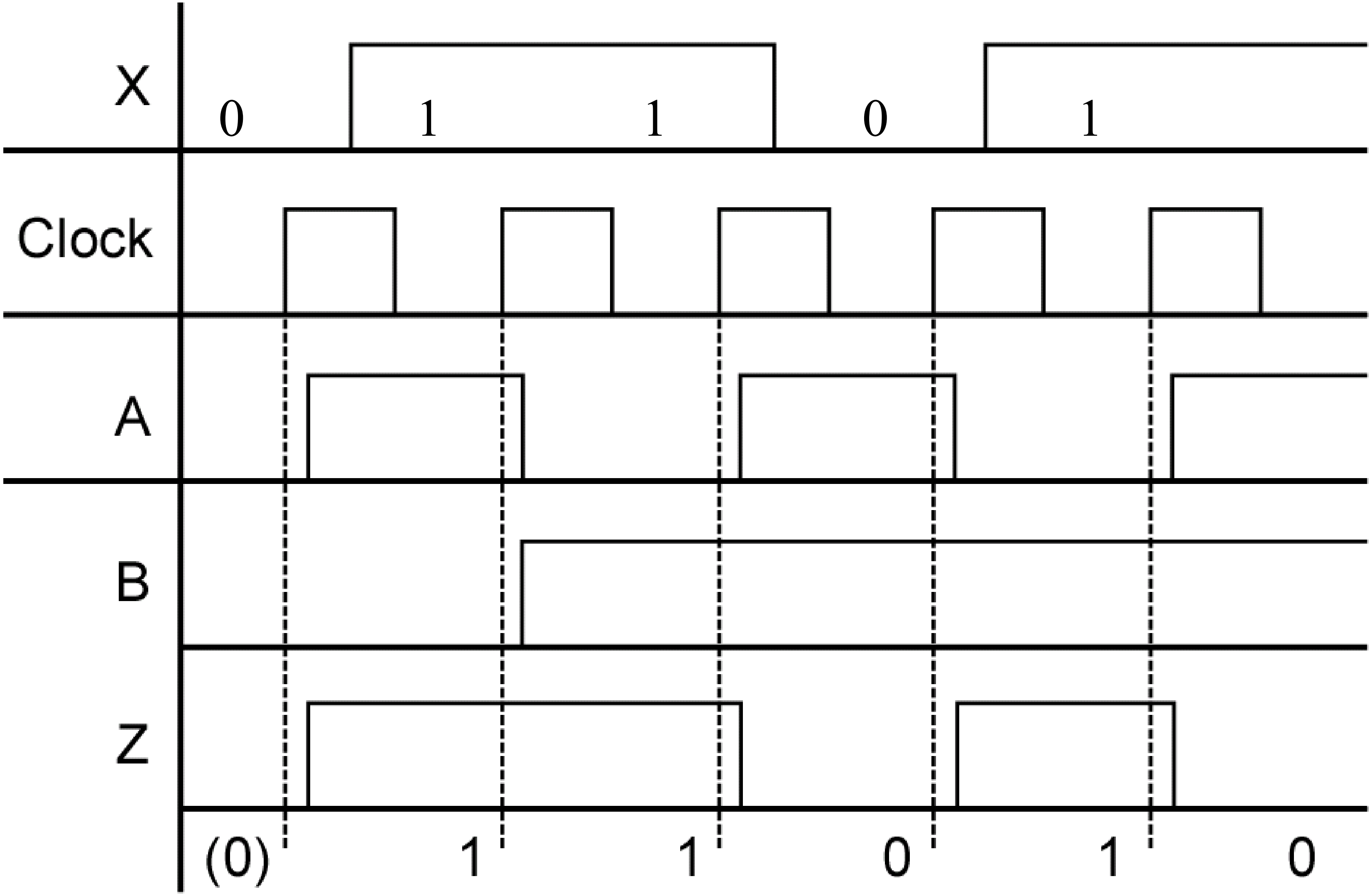


Figure 13-6: **Timing Chart for Figure 13-5**

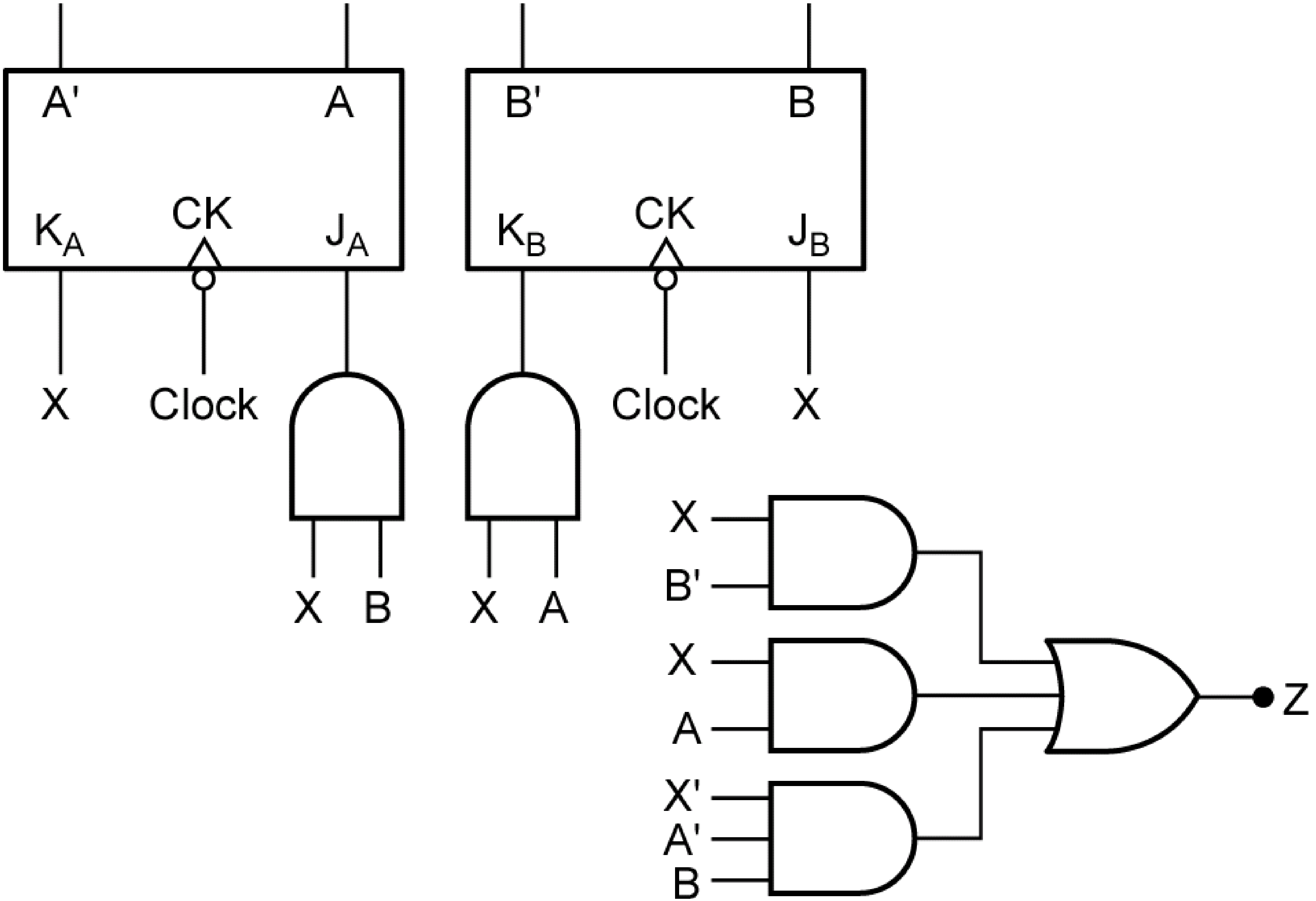


Figure 13-7: **Mealy Sequential Circuit to be Analyzed**

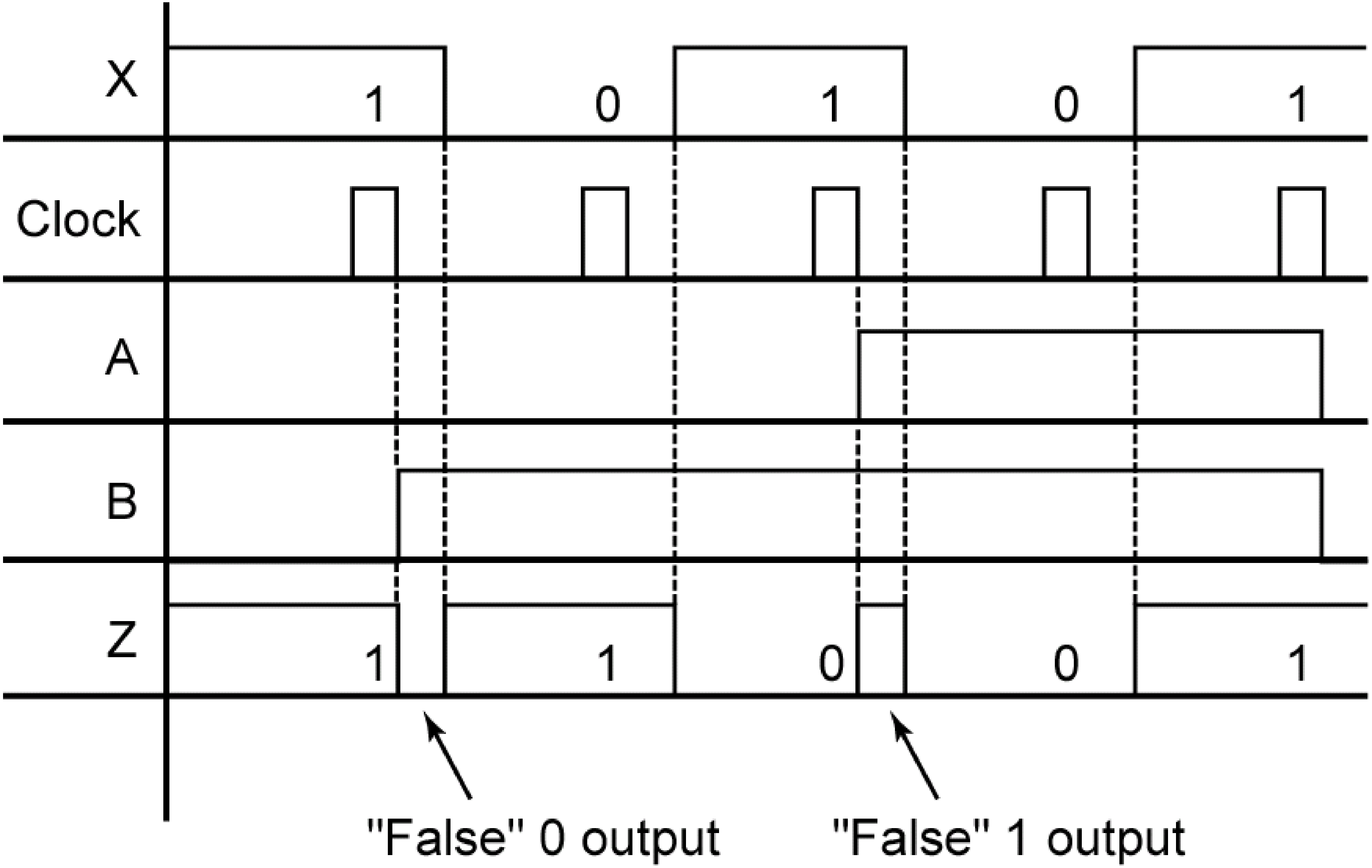


Figure 13-8: **Timing Chart for Circuit of Figure 13-7**

State Tables and Graphs

Although constructing timing charts is satisfactory for small circuits and short input sequences, the construction of state tables and graphs provides a more systematic approach which is useful for the analysis of larger circuits and which leads to a general synthesis procedure for sequential circuits.

The state table specifies the next state and output of a sequential circuit in terms of its present state and input.

Section 13.3 (p. 401)

The following method can be used to construct the state table:

1. Determine the flip-flop input equations and the output equations from the circuit.
2. Derive the next-state equation for each flip-flop from its input equations, using one of the following relations:

D flip-flop $Q^+ = D$ (13-1)

D-CE flip-flop $Q^+ = D \bullet CE + Q \bullet CE'$ (13-2)

T flip-flop $Q^+ = T \oplus Q$ (13-3)

S-R flip-flop $Q^+ = S + R'Q$ (13-4)

J-K flip-flop $Q^+ = JQ' + K'Q$ (13-5)

3. Plot a next-state map for each flip-flop.
4. Combine these maps to form the state table. Such a state table, which gives the next state of the flip-flops as a function of their present state and the circuit inputs, is frequently referred to as a transition table.

As an example of this procedure, we will derive the state table for the circuit of Figure 13-5:

1. The flip-flop input equations and output equation are

$D_A = X \oplus B' \qquad D_B = X + A \qquad Z = A \oplus B$

2. The next-state equations for the flip-flops are

$A^+ = X \oplus B' \qquad B^+ = X + A$

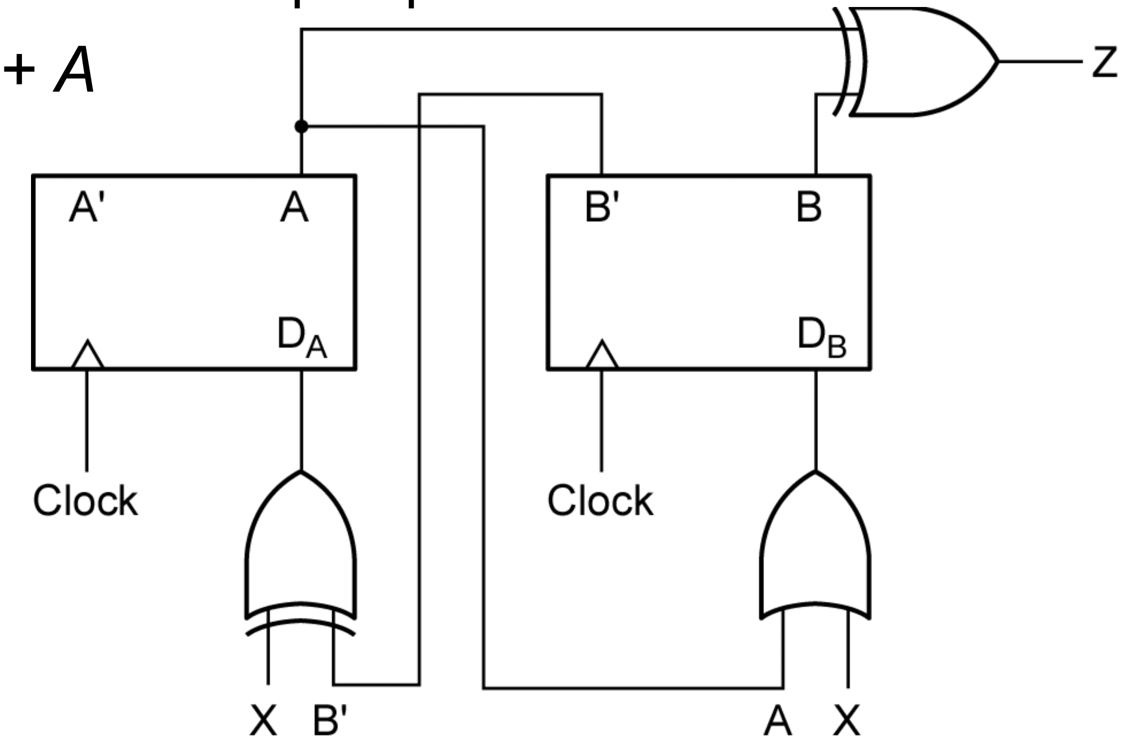


Figure 13-5

3. The corresponding maps are

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

A^+

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

B^+

4. Combining these maps yields the transition table in Table 13-2(*a*), which gives the next state of both flip-flops (A^+B^+) as a function of the present state and input. The output function *Z* is then added to the table. In this example, the output depends only on the present state of the flip-flops and not on the input, so only a single output column is required.

Table 13-2. Moore State Tables for Figure 13-5

	A^+B^+		
AB	X=0	X=1	Z
00	10	01	0
01	00	11	1
11	01	11	0
10	11	01	1

(a)

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

(b)

Table 13-2

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

(b)

Each node in the graph represents a state in the circuit.

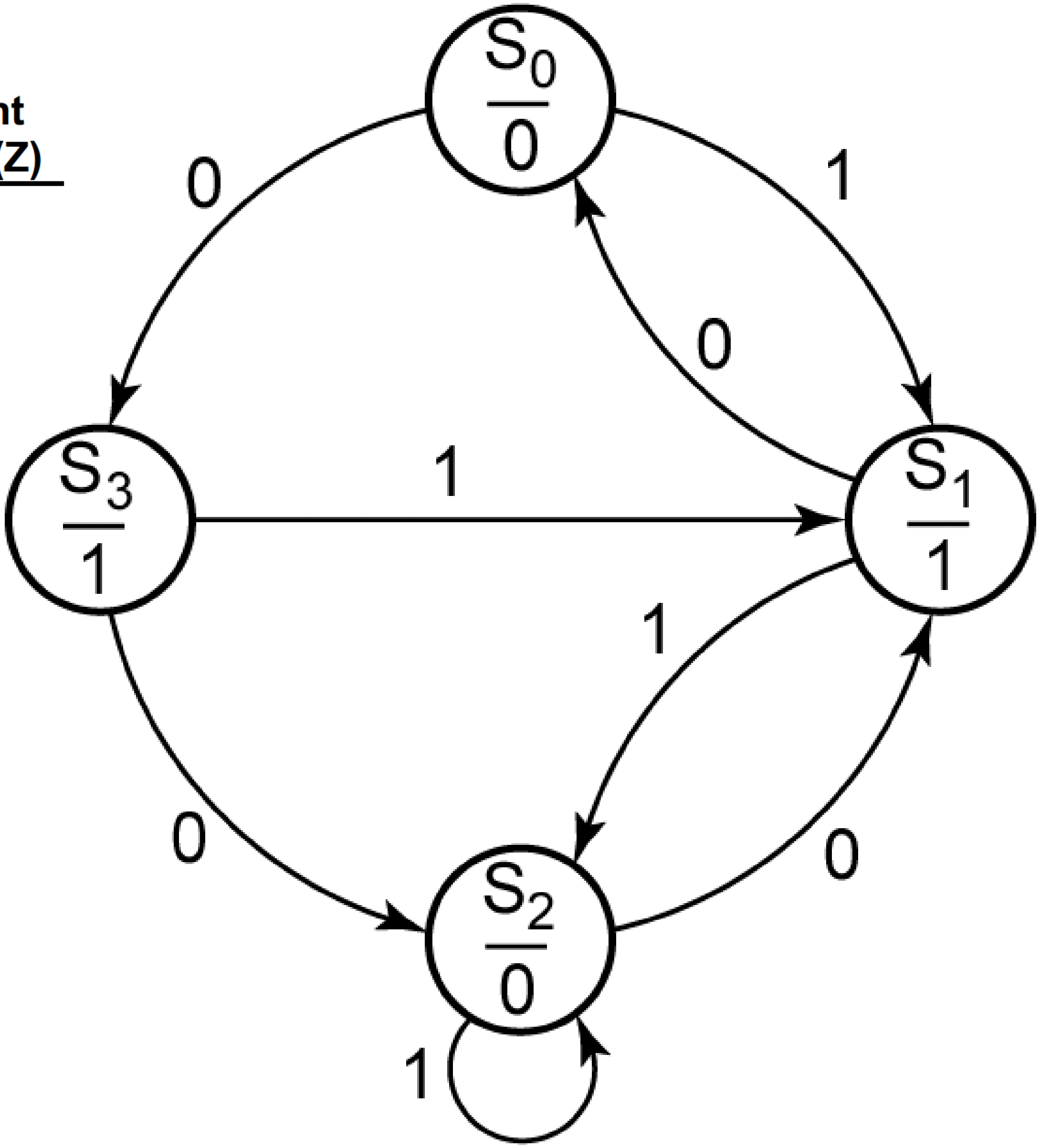


Figure 13-9: **Moore State Graph for Figure 13-5**

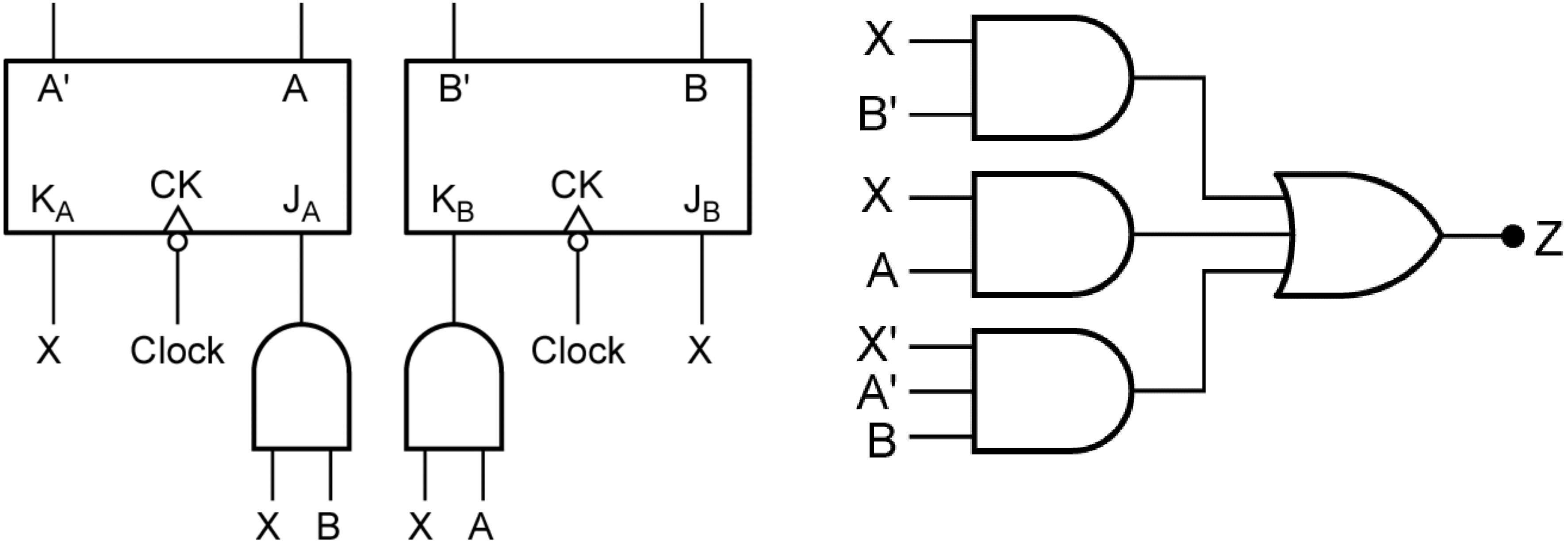


Figure 13-7

We can construct the next-state and output equations from the circuit diagram:

$$A^+ = J_A A' + K'_A A = XBA' + X'A$$

$$B^+ = J_B B' + K'_B B = XB' + (AX)'B = XB' + X'B + A'B$$

$$Z = X'A'B + XB' + XA$$

Next, we can plot Karnaugh maps for *A*$^+$, *B*$^+$, and *Z*. We can then use these maps to derive the transition table.

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

A^+

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

B^+

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

Z

Figure 13-10

Table 13-3. Mealy State Tables for Figure 13-7

(a)

	A^+B^+		Z	
AB	$X = 0$	1	$X = 0$	1
00	00	01	0	1
01	01	11	1	0
11	11	00	0	1
10	10	01	0	1

(b)

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

State tables derived from Karnaugh maps.

Finally, we can draw the state graph from the tables we derived:

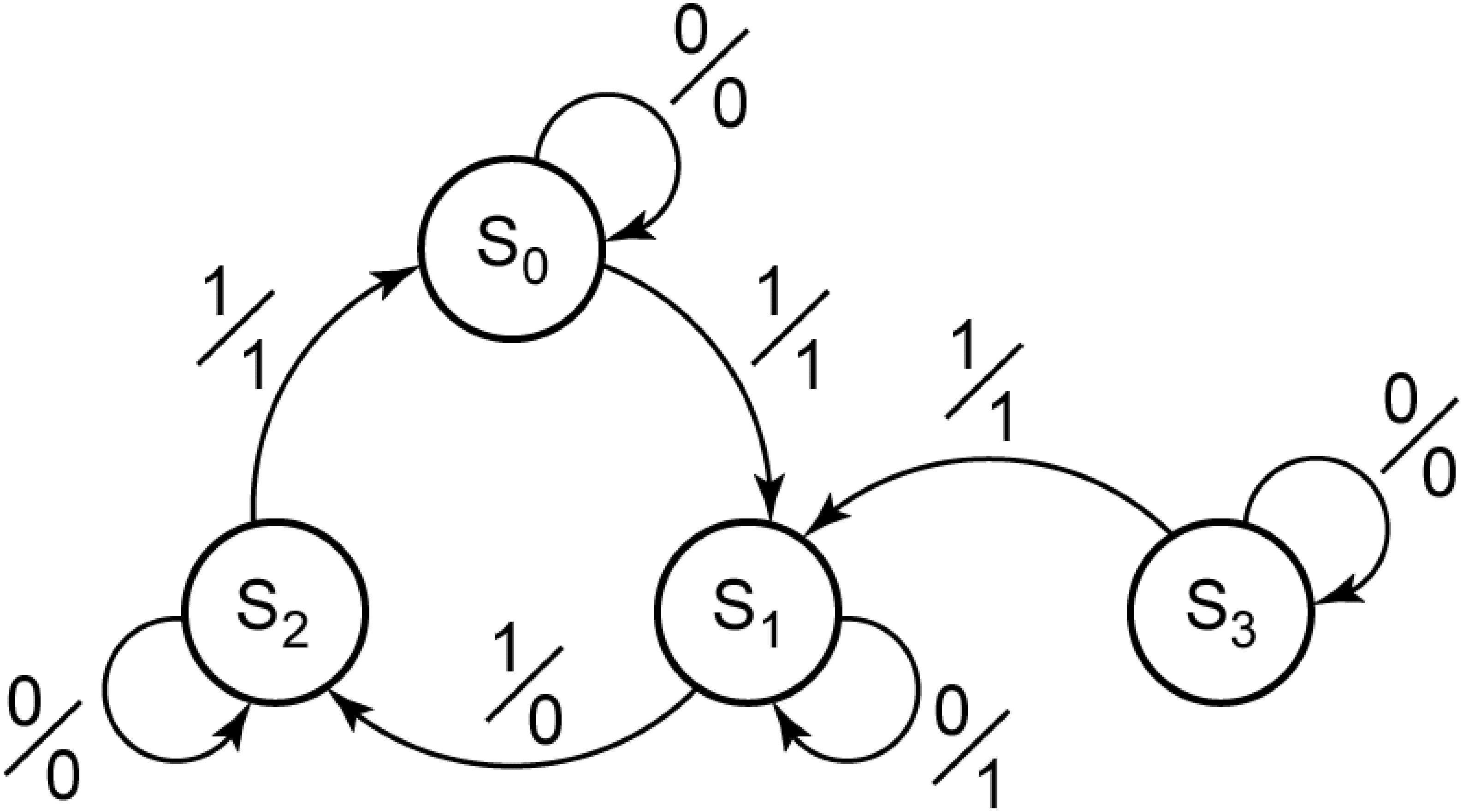
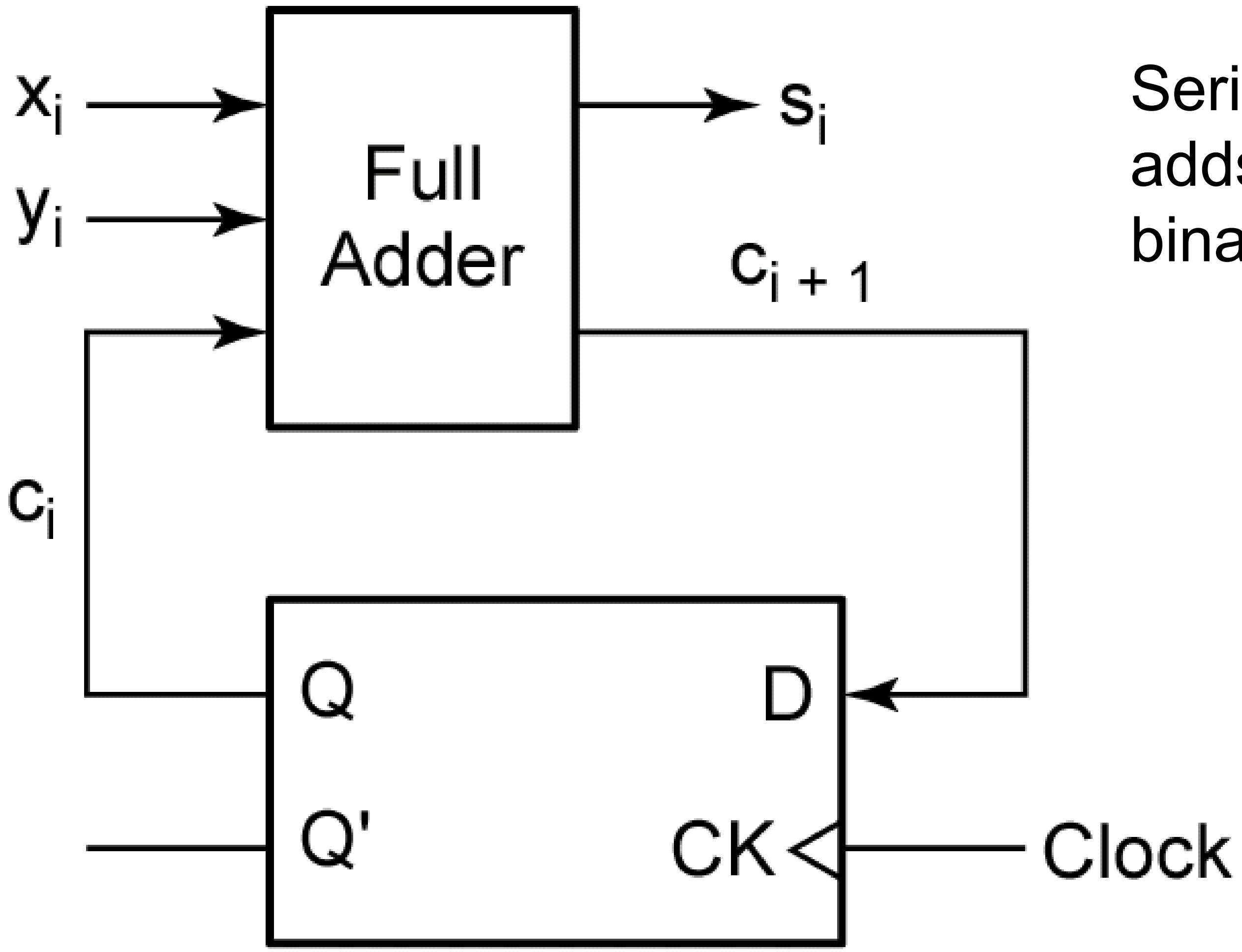


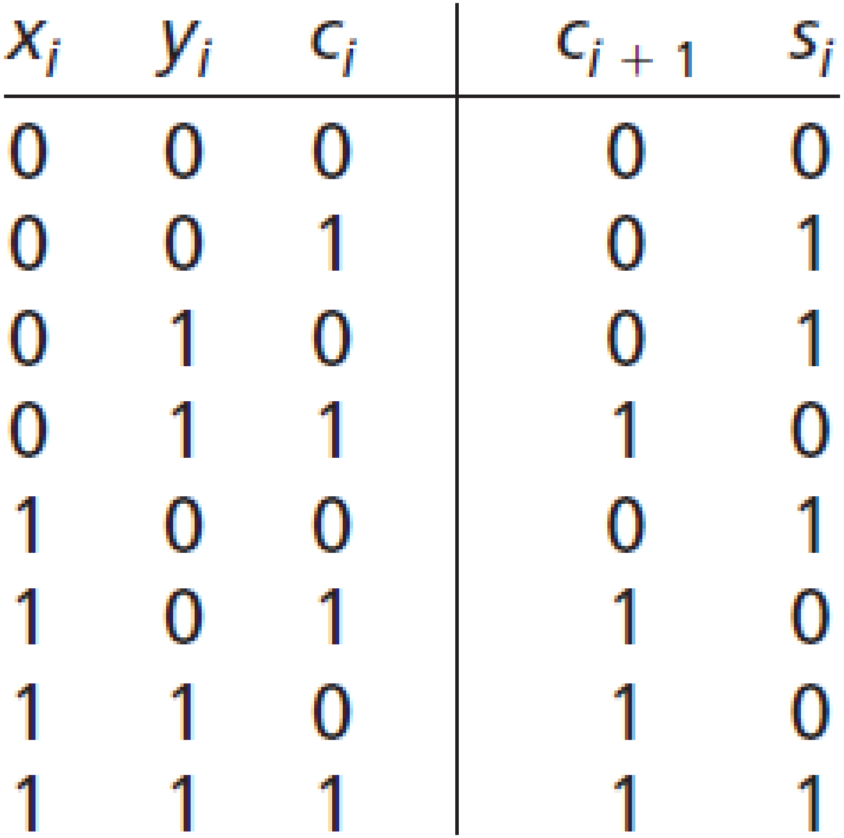
Figure 13-11: **Mealy State Graph for Figure 13-7**



Serial Adder that adds two *n*-bit binary numbers

(a) With D flip-flop

Figure 13-12a: **Serial Adder**



x_i	y_i	c_i	c_{i+1}	s_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

(b) Truth table

Figure 13-12b: **Serial Adder Truth Table**

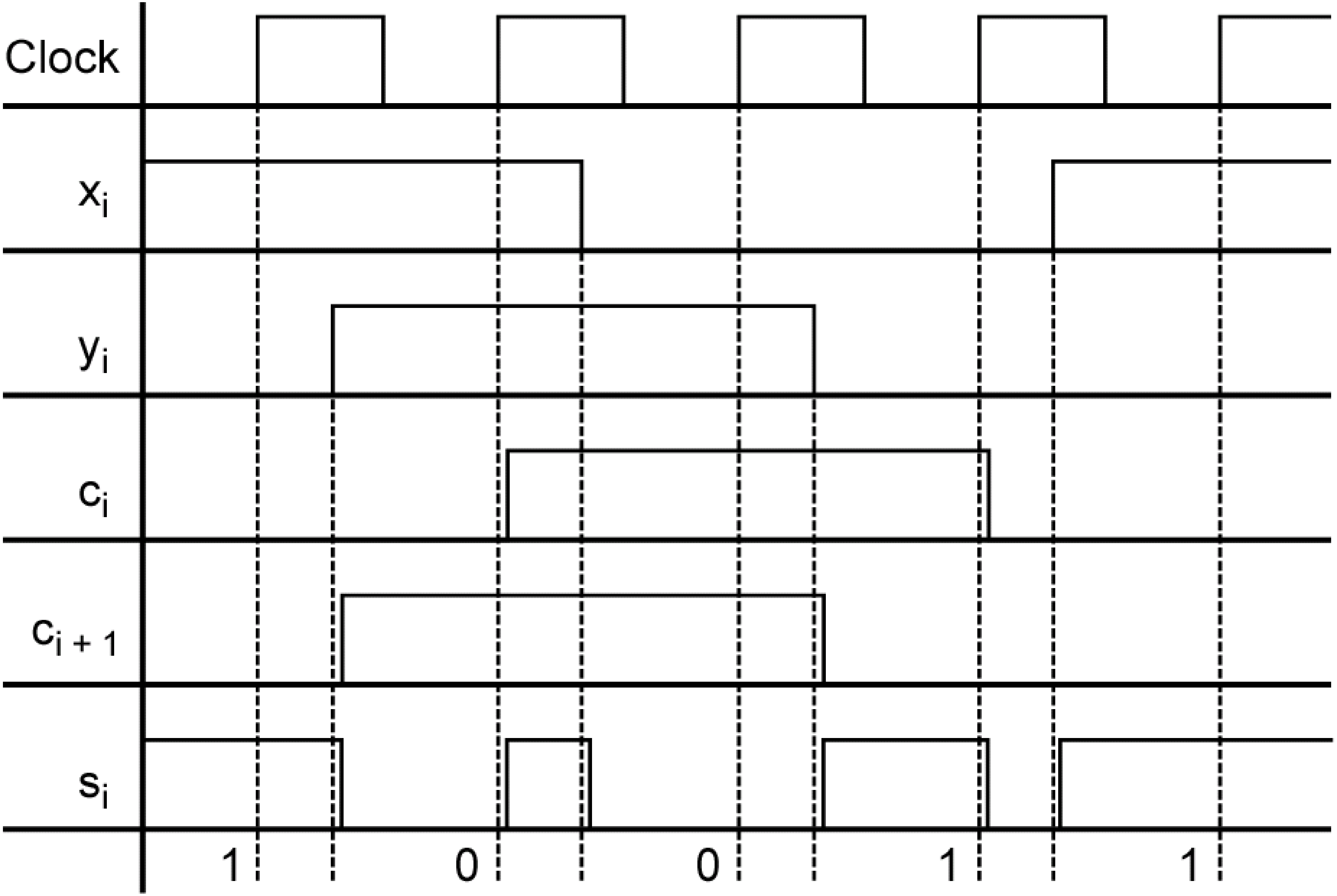


Figure 13-13: **Timing Diagram for Serial Adder**

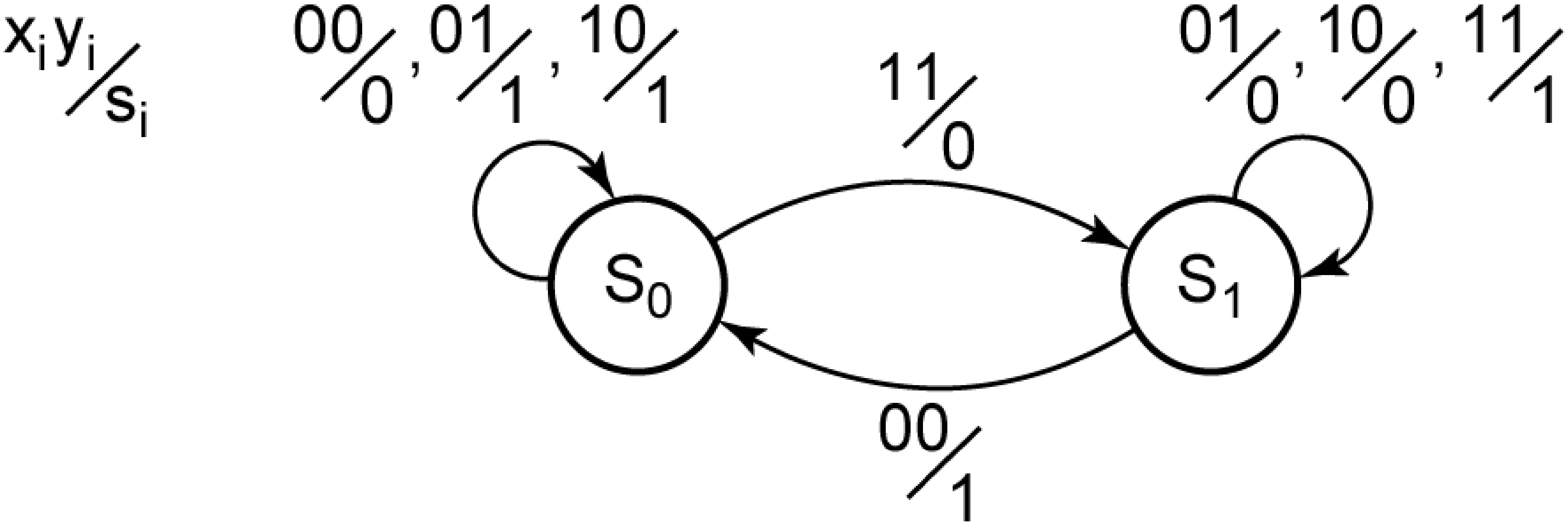
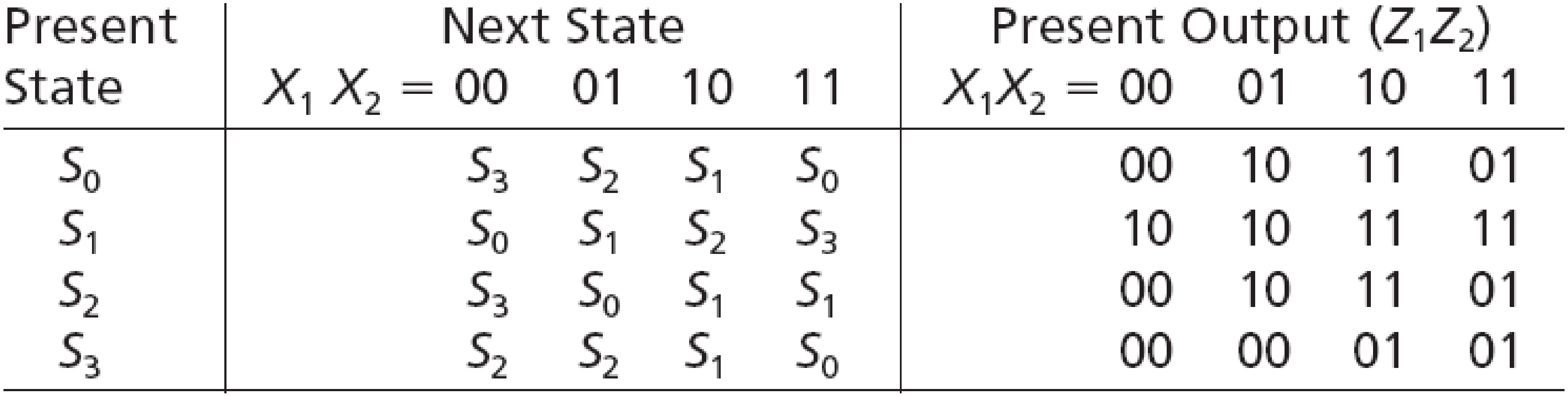


Figure 13-14: **State Graph for Serial Adder**

Table 13-4. A State Table with Multiple Inputs and Outputs



Present	Next State				Present Output (Z_1Z_2)			
State	$X_1 X_2 = 00$	01	10	11	$X_1X_2 = 00$	01	10	11
S_0	S_3	S_2	S_1	S_0	00	10	11	01
S_1	S_0	S_1	S_2	S_3	10	10	11	11
S_2	S_3	S_0	S_1	S_1	00	10	11	01
S_3	S_2	S_2	S_1	S_0	00	00	01	01

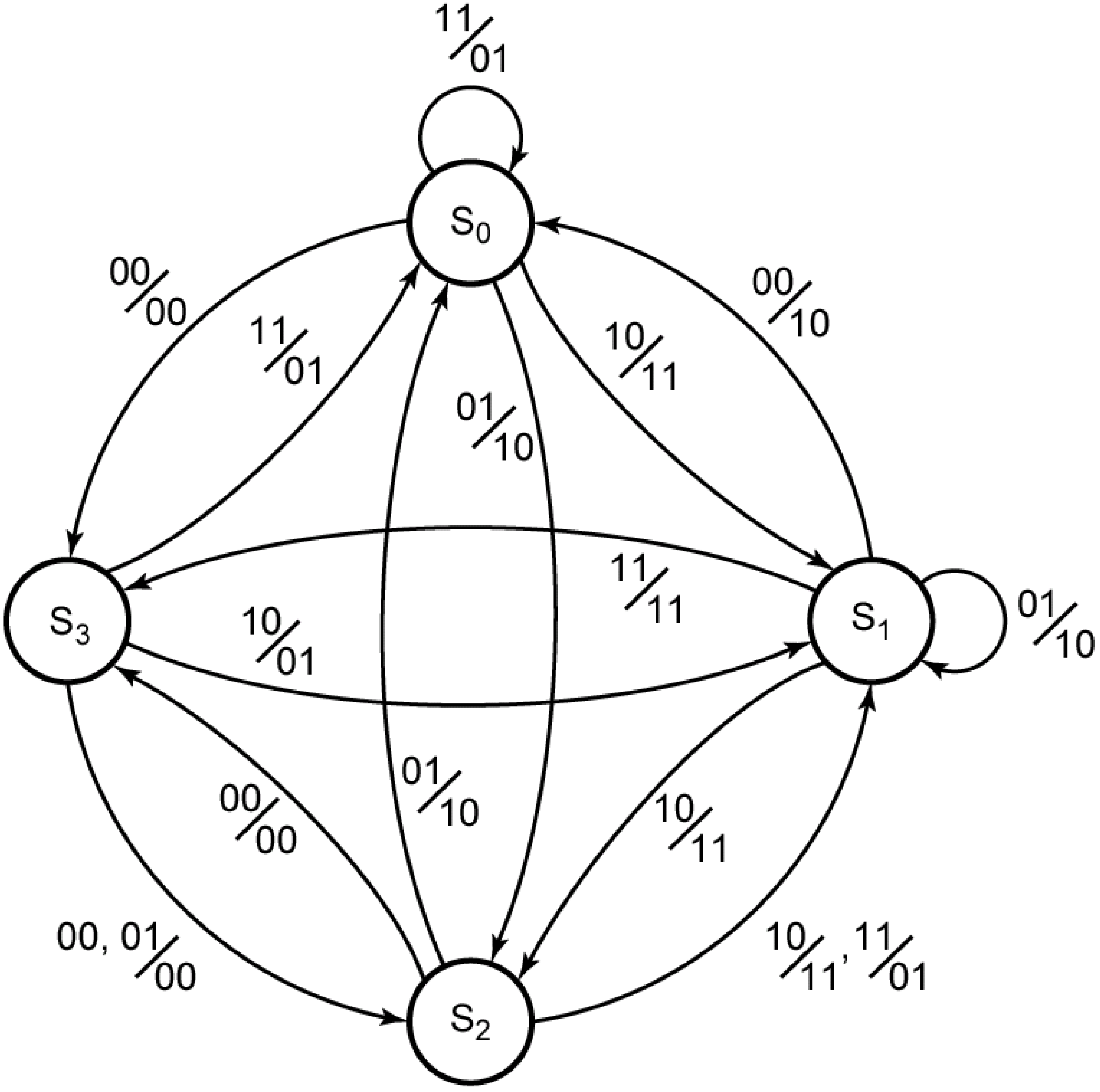


Figure 13-15: **State Graph for Table 13-4**

Construction and Interpretation of Timing Charts

Several important points concerning the construction and interpretation of timing charts are summarized as follows:

1. When constructing timing charts, note that a state change can only occur after the rising (or falling) edge of the clock.
2. The input will normally be stable immediately before and after the active clock edge.
3. For a Moore circuit, the output can change only when the state changes, but for a Mealy circuit, the output can change when the input changes as well as when the state changes. A false output may occur between the time the state changes and the time the input is changes to its new value.

Section 13.3 (p. 406)

4. False outputs are difficult to determine from the state graph, so use either signal tracing through the circuit or use the state table when constructing timing charts for Mealy circuits.

5. When using a Mealy state table for constructing timing charts, the procedure is as follows:

   (a) For the first input, read the present output and plot it.

   (b) Read the next state and plot it (following the active edge of the clock pulse).

   (c) Go to the row in the table which corresponds to the next state and read the output under the old input column and plot it (This may be a false output.)

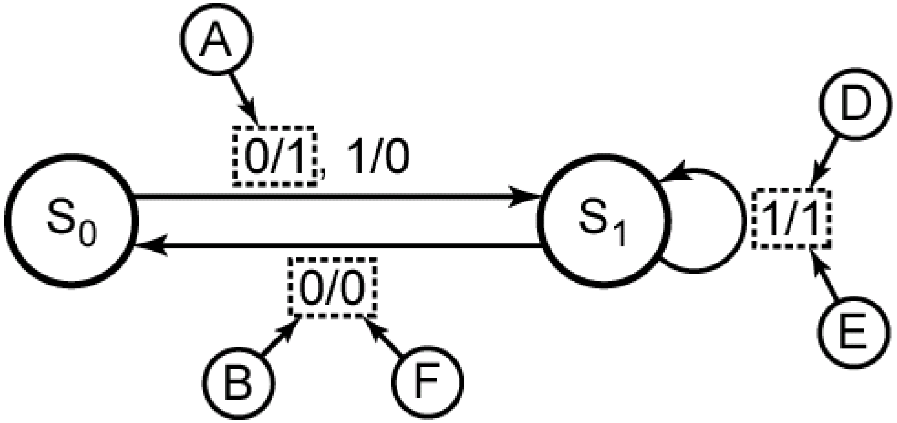
   (d) Change to the next input and repeat steps (a), (b), and (c).

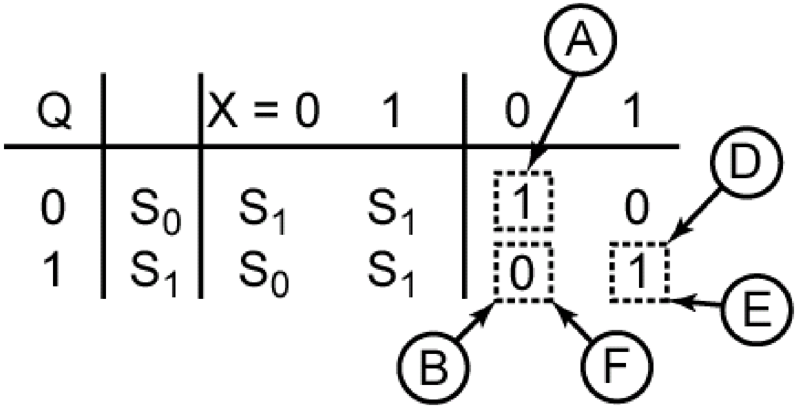
6. For Mealy circuits, the best time to read the output is just before the active edge of the clock, because the output should always be correct at that time. A “false” output may occur after the state has changed and before the input has changed.

The following example shows the relationships among the state graph, state table, circuits, and timing chart. The input sequence is:

$$X = 0\ 1\ 0$$

Note that the correct output occurs ***before*** the rising edge of the clock!





Q		X = 0	1	0	1
0	S_0	S_1	S_1	1	0
1	S_1	S_0	S_1	0	1

Inital values are shown

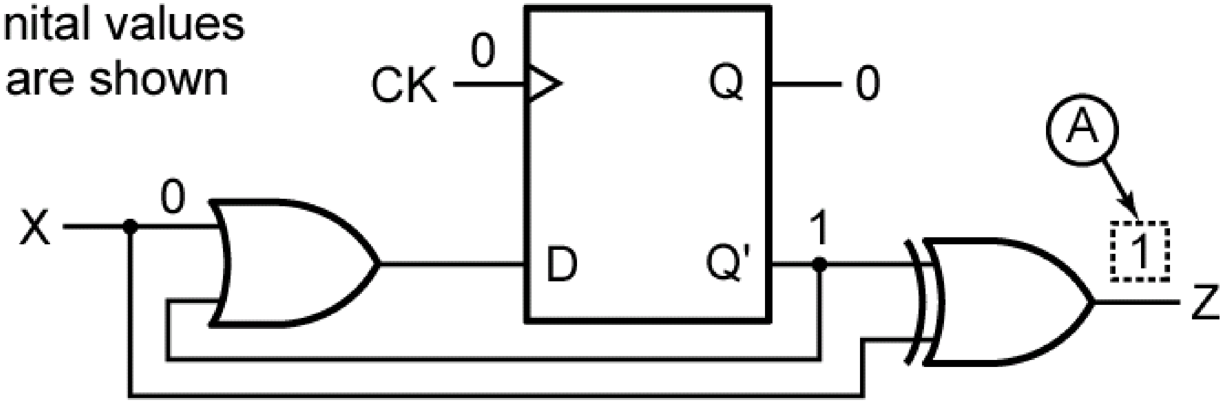
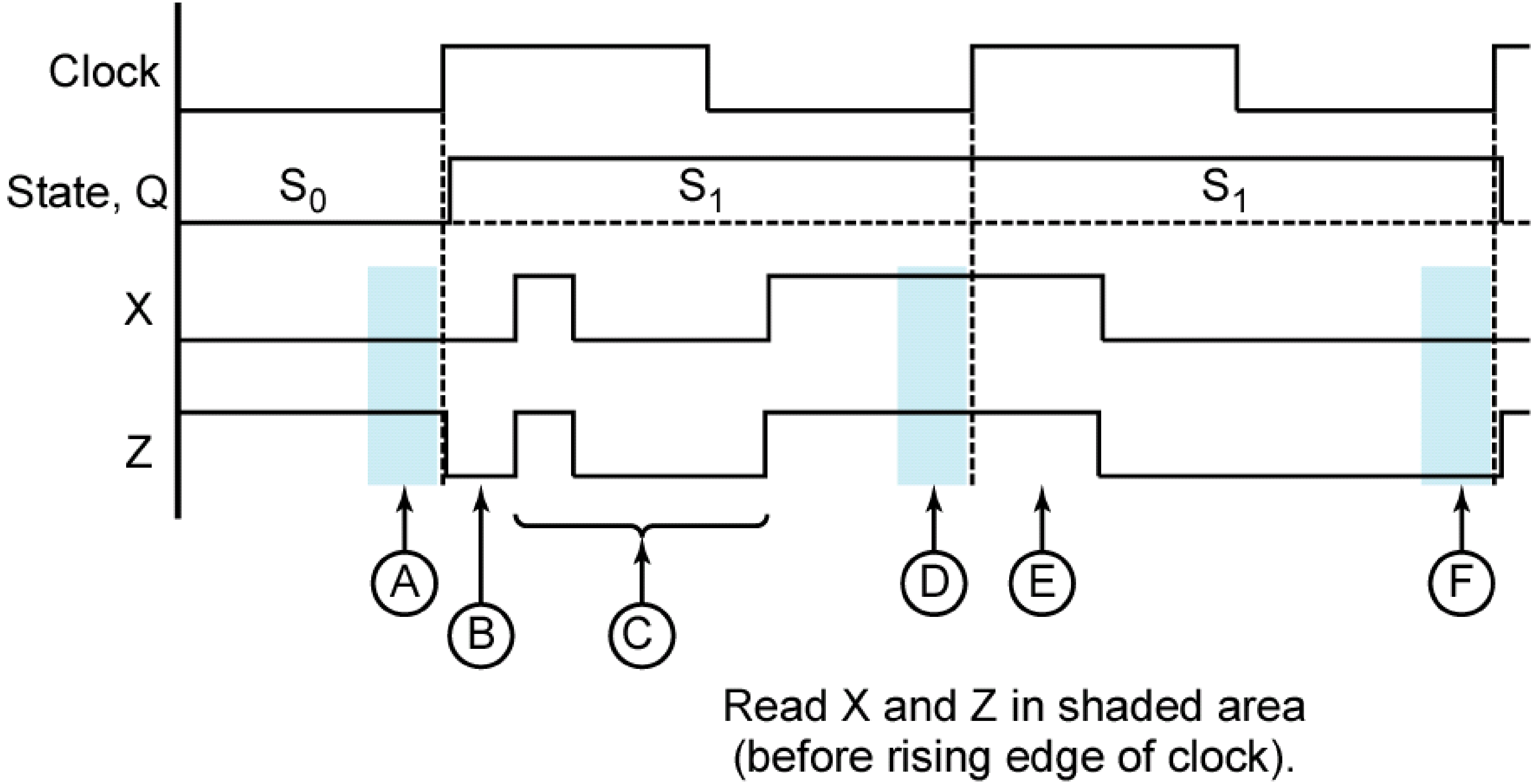


Figure 13-16



Clock
State, Q
S_0
S_1
S_1
X
Z
A
B
C
D
E
F
Read X and Z in shaded area
(before rising edge of clock).

Figure 13-16

General Models for Sequential Circuits

A sequential circuit can be divided conveniently into two parts -- the flip-flops which serve as memory for the circuit and the combinational logic which realizes the input functions for the flip-flops and the output functions.

The combinational logic may be implemented with gates, with a ROM, or with a PLA.

Section 13.4 (p. 408)

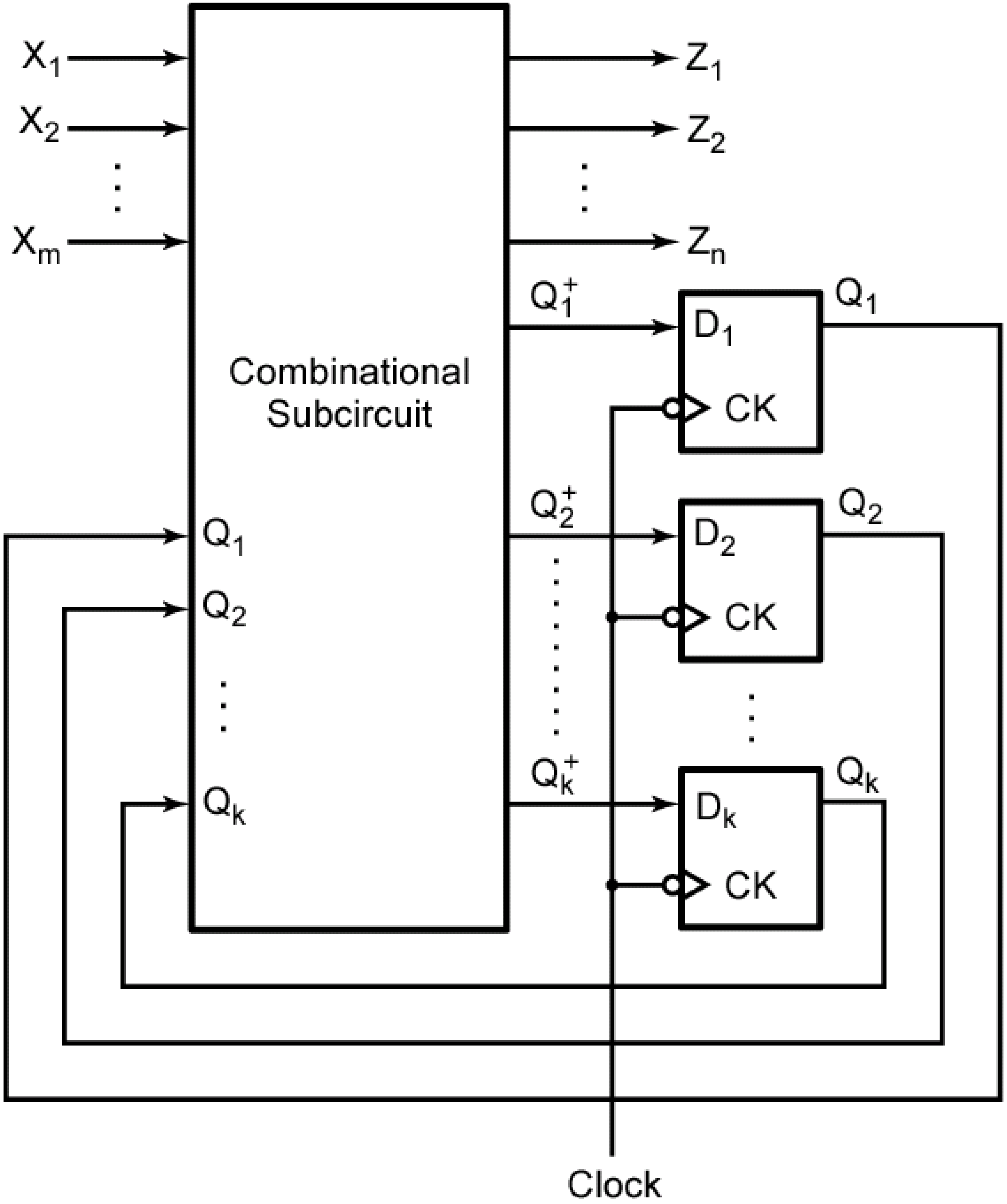


Figure 13-17:

General Model for Mealy Circuit Using Clocked D Flip-Flops

Minimum Clock Period

We can determine the fastest clock speed (the minimum clock period) from the general model of the Mealy circuit.

Following the active edge of the clock the flip-flops change state, and the flip-flop output is stable after the propagation delay (t_p).

The new values of *Q* then propagate through the combinational circuit so that the *D* values are stable after the combinational circuit delay (t_c).

Then, the flip-flop setup time (t_{su}) must elapse before the next active clock edge.

Section 13.4 (p. 410)

Thus, the propagation delay in the flip-flops, the propagation delay in the combinational subcircuit, and the setup time for the flip-flops determine how fast the sequential circuit can operate, and the minimum clock period is:

$$t_{\text{clk}}\,(\text{min}) = t_{\text{p}} + t_{\text{c}} + t_{\text{su}}$$

This assumes that the *X* inputs are stable no later than $t_{\text{c}} + t_{\text{su}}$ before the next active clock edge. If this is not the case, then we must calculate the minimum clock period by:

$$t_{\text{clk}}\,(\text{min}) = t_{\text{x}} + t_{\text{c}} + t_{\text{su}}$$

Where t_{x} is the time after the active clock edge at which the *X* inputs are stable.

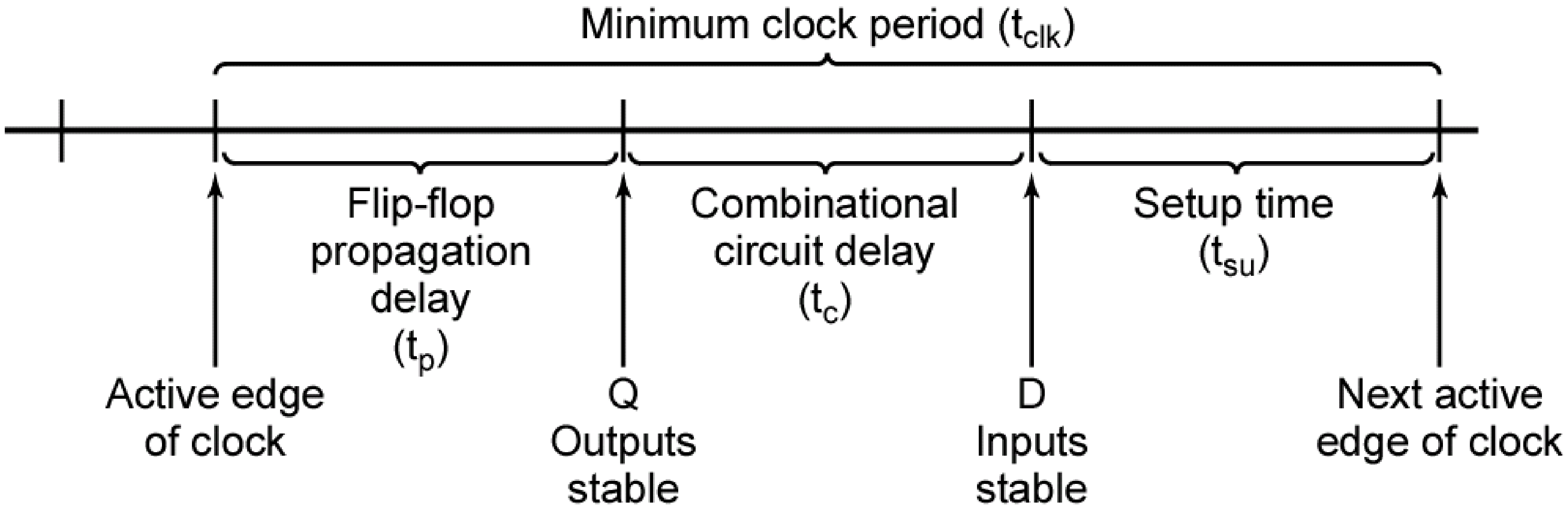

Minimum clock period (t_{clk})
Flip-flop propagation delay (t_p)
Combinational circuit delay (t_c)
Setup time (t_{su})
Active edge of clock
Q
Outputs stable
D
Inputs stable
Next active edge of clock

Figure 13-18: **Minimum Clock Period for a Sequential Circuit**

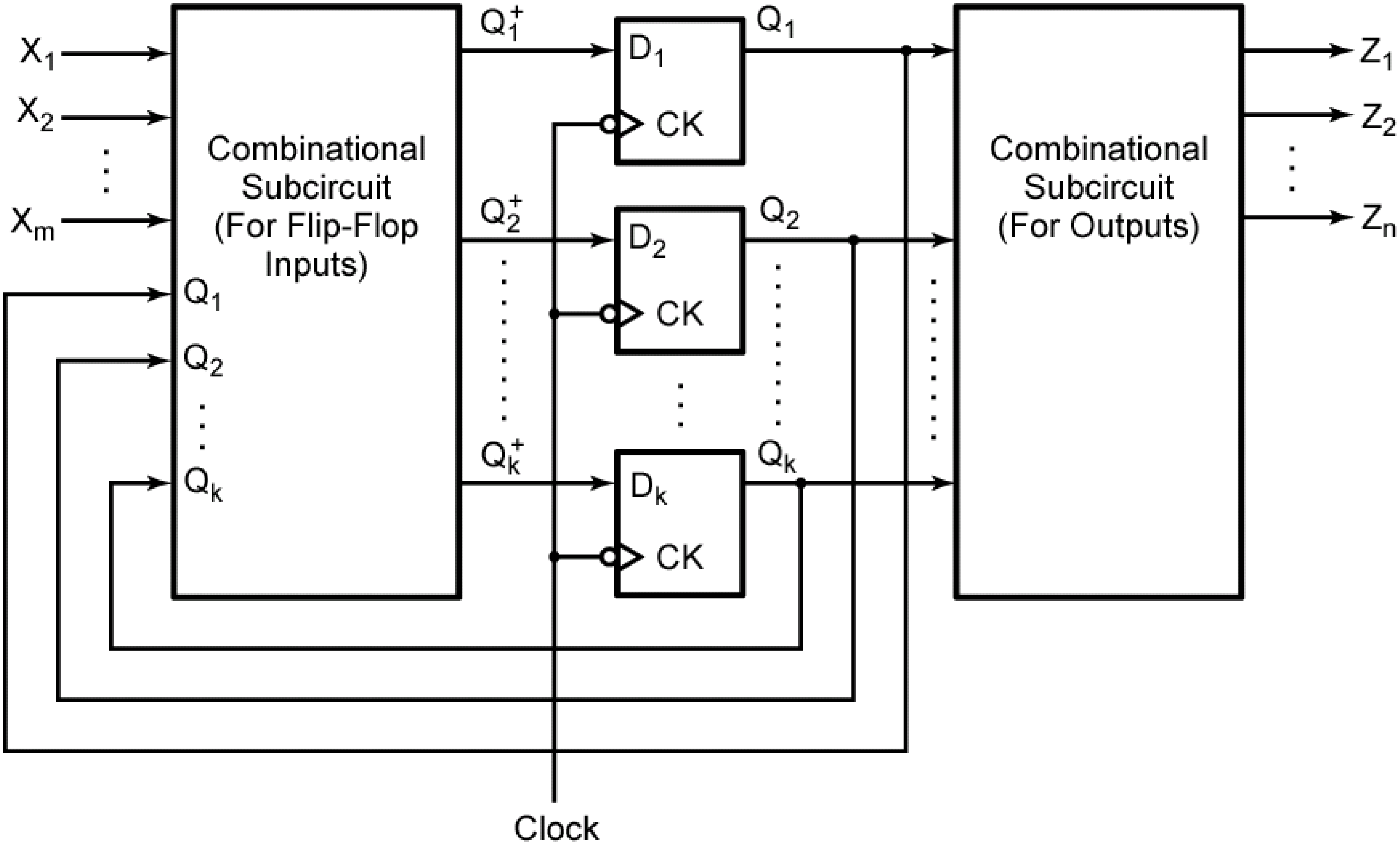
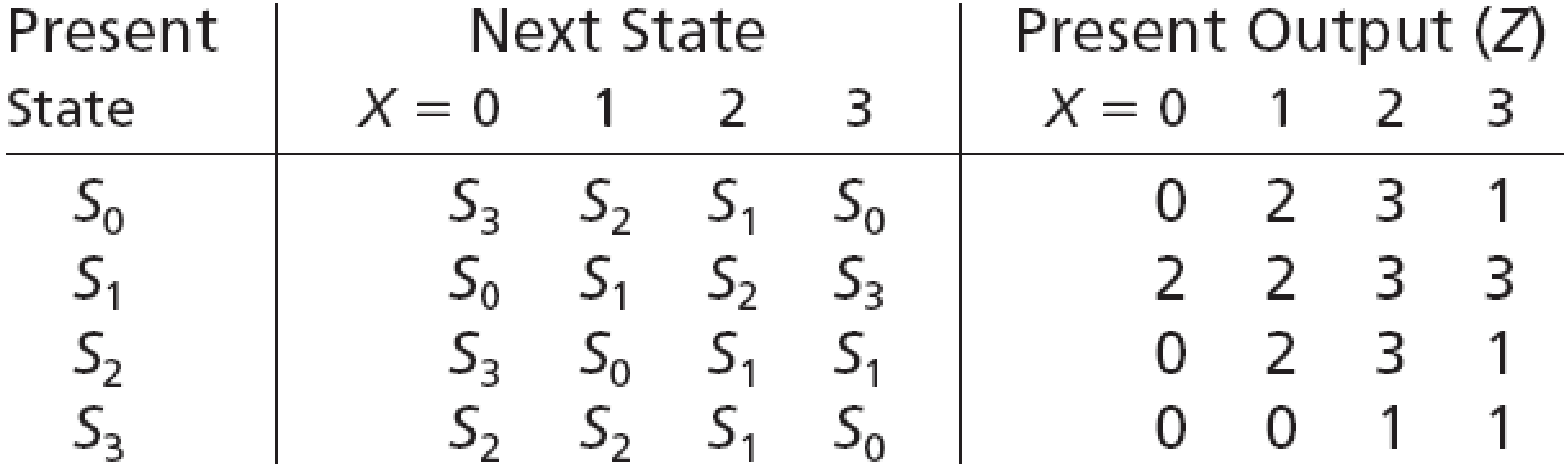


Figure 13-19: **General Model for Moore Circuit Using Clocked D Flip-Flops**

Table 13-5. Meely State Table with Multiple Inputs and Outputs



Present	Next State				Present Output (Z)			
State	$X = 0$	1	2	3	$X = 0$	1	2	3
S_0	S_3	S_2	S_1	S_0	0	2	3	1
S_1	S_0	S_1	S_2	S_3	2	2	3	3
S_2	S_3	S_0	S_1	S_1	0	2	3	1
S_3	S_2	S_2	S_1	S_0	0	0	1	1

Next state: $\delta\,(S_0, 1) = S_2$ $\qquad$ $\delta\,(S_2, 3) = S_1$

Output: $\lambda\,(S_0, 1) = 2$ $\qquad$ $\lambda\,(S_2, 3) = 1$