Synchronous Sequential Circuits

10.1 Analysis of Synchronous Sequential Circuits

The behaviour of a sequential circuit is determined from the inputs, the outputs, and the states of its flip-flops. Both the outputs and the next state are function of the inputs and the present state. The analysis of sequential circuit consists of obtaining a table or a diagram for the time sequence of inputs, outputs and internal states. The success of analysis or design of sequential circuit depends largely on the aids and systematic techniques such as state tables, state diagrams and state equations used in these processes.

10.1.1 State Table

Fig. 10.1 shows a clocked sequential circuit. It has one input variable X, one output variable Y, and two clocked JK flip-flops. The flip-flops are labelled as A and B, and their outputs are labelled as A and $\overline{A}$, and B and $\overline{B}$, respectively. The output Y is determined by boolean expression $Y = XA\overline{B}$ which is derived using three input AND gate, as shown in the Fig. 10.1. As shown in the Fig. 10.1, outputs of flip-flop A and input X are used to derive inputs to flip-flop B and vice-versa. With this understanding of the given sequential circuit now we start analysing the circuit.

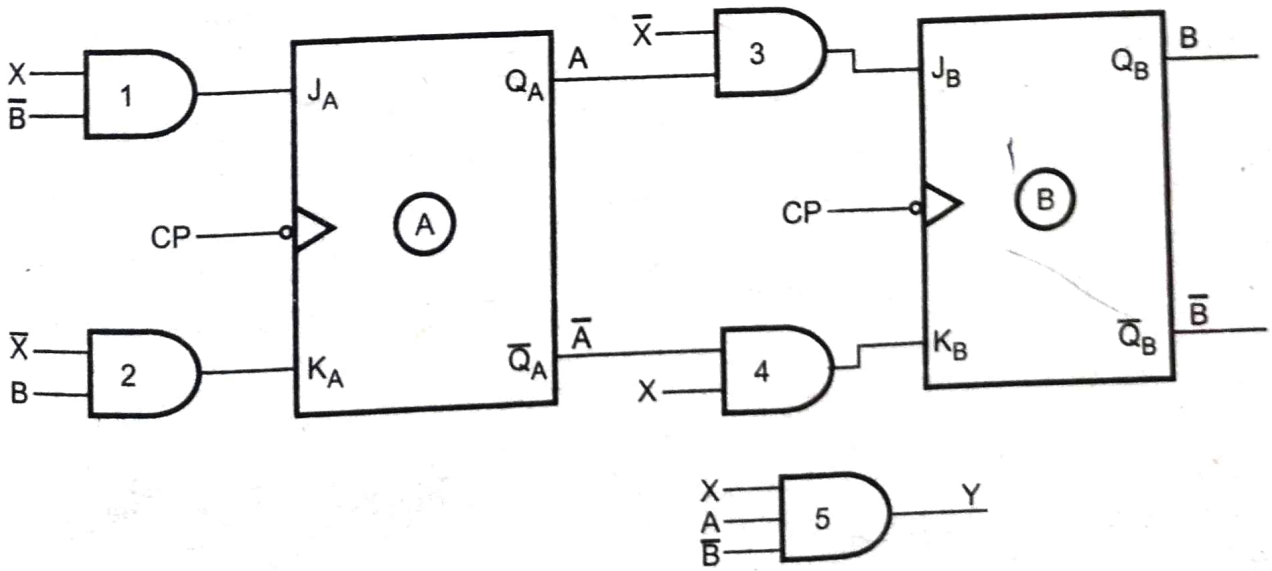


Table 10.1 shows the state table for the given sequential circuit. It represents relationship between input, output and flip-flop states. It consists of three sections labelled present state, next state, and output. The present state designates the state of flip-flops before the occurrence of a clock pulse. The next state is state of the flip-flops after the application of a clock pulse, and the output section gives the values of the output variables during the present state. Both the next state and output sections have two columns representing two possible input conditions: X = 0 and X = 1.

Present State	Next State		Output	
	X = 0	X = 1	X = 0	X = 1
AB	AB	AB	Y	Y
00	00	10	0	0
01	01	00	0	0
10	11	10	0	1
11	01	11	0	0

Table 10.1

We can derive the state table as follows :

Step 1 : Find the next state for all possible present states and input states.

Present state : AB = 00

Input X= 0 When present state is 00, i.e. A = 0 and B = 0 and X input is 0, outputs of all AND gates are logic 0. Therefore, J and K inputs for both the flip-flops are 0, giving next state same as present state.

Input X = 1 With AB = 00 and X = 1, AND gate 1 and 4 produce a logic 1 signal at the J_A input of flip-flop A and K_B input of flip-flop B, respectively. * Therefore, in the next state A is set and B is cleared.

*

J	K	Q
0	0	Nc
0	1	0
1	0	1
1	1	Toggle

Present state : AB = 01

Input X = 0 When present state is 01, i.e. A = 0 and B = 1 and X input is 0. The output of AND gate 2 is logic 1, making K_A input of flip-flop A high. Therefore in the next state A is reset and B remains unchanged, i.e. in set state.

Input X = 1 With AB = 01 and X = 1, the output of AND gate 4 is logic 1, making K_B input of flip-flop B high. Therefore, in the next state A remains unchanged and B resets.

Present State 10

Input X = 0 When present state is 10, i.e. A = 1 and B = 0 and input X is 0, the output of AND gate 3 is logic 1, making J_B input of flip-flop B high. Therefore, in the next state A remains unchanged and B sets.

Input X = 1 With AB = 10 and X = 1, the output of AND gate 1 is logic 1, making J_A input of flip-flop A high. Therefore, in the next state A sets and B remains unchanged.

Present state 11

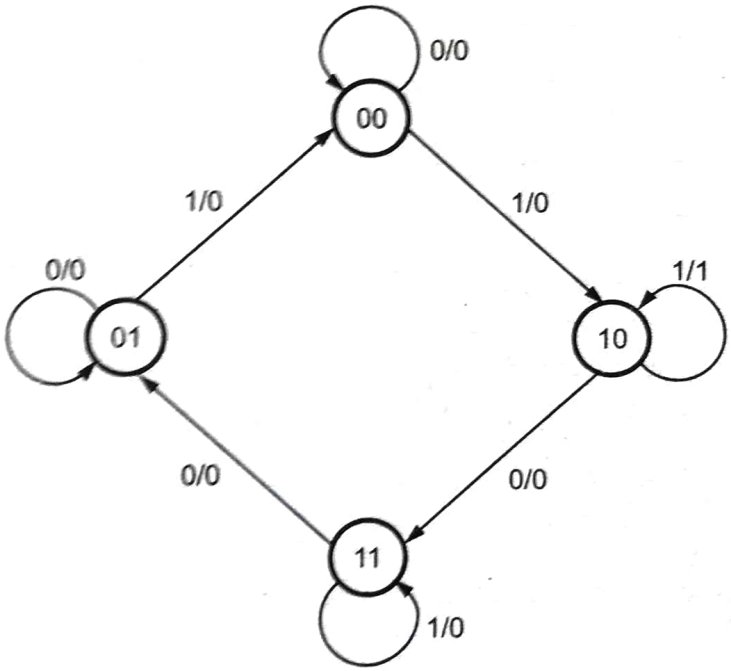
Input X = 0 When present state is 11, i.e. A = 1 and B = 1 and input X is 0, the outputs of AND gate 2 and AND gate 3 are logic 1, making K_A and J_B inputs high. Therefore, in the next state A resets and B sets.

Input X = 1 With AB = 11 and X = 1, the outputs of all AND gates are logic 0. Therefore in the next state A and B both remain unchanged.

To determine the entries in the output section, we have to examine the AND gate 5 for all possible present and input states. For X = 0, regardless of any present state output Y is logic 0. For X = 1, the output Y is equal to 1 only when A = 1 and B = 0.

The state table of any sequential circuit is obtained by the same procedure used in the above example. In the above example, sequential circuit has 2 flip-flops and 1 input variable, producing 4 rows, and 2 columns in the next state and output sections. In general, a sequential circuit with m flip-flops and n input variables produce 2^m rows, one for each state and 2^n columns, one for each input combination in the next state and output sections of the state table.

10.1.2 State Diagram



mealy

Fig. 10.2 State diagram circuit shown in Fig. 10.1

State diagram is a graphical representation of a state table. Fig. 10.2 shows a state diagram for sequential circuit shown in Fig. 10.1. As shown in the Fig. 10.2, state is represented by a circle, and the transition between states is indicated by directed lines connecting the circles. A directed line connecting a circle with itself indicates that next state is same as present state. The binary number inside each circle identifies the state represented by the circle. The directed lines are labelled with two binary numbers separated by a symbol '/'. The input value that causes the state transition is labelled first and the output value during the present state is labelled after the symbol '/'.

This state equation is the same as the state equation obtained from state table.

Ex. 10.1 : *Derive the state table and state diagram for sequential circuit shown in Fig. 10.4.*

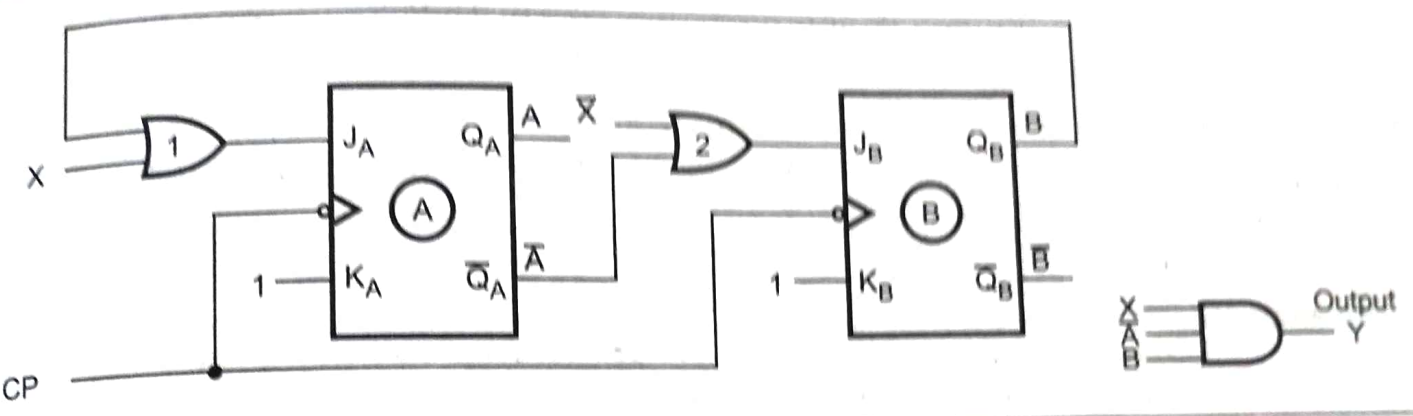


Fig. 10.4

Sol. : The given circuit has two JK flip-flops, so they produce 4 rows in the state table. As next state and the output depends on present state as well as on input they produce two column each. Let us derive the next states and output for all possible present states and inputs.

Present state 00 (A = 0 and B = 0) with input X = 0

With this condition, output of OR gate 1 is logic 0 and output of OR gate 2 is logic 1 which resets flip-flop A and toggles flip-flop B. Therefore, in the next state A = 0 and B = 1. The output for this present state is $X\ A\ B = 0\ \overline{0}\ 0 = 0$.

Present state 01 (A = 0 and B = 1) with input X = 0

With this condition, output of both OR gates is logic 1, which toggles both the flip-flops. Therefore, in the next state A = 1 and B = 0. The output for this present state is $X\overline{A}\ B = 0\ \overline{0}\ 1 = 0$.

Present state 10 (A = 1 and B = 0) with input X = 0

With this condition, output of OR gate 1 is logic 0 and output of OR gate 2 is logic 1, which resets flip-flop A and toggles flip-flop B. Therefore, in the next state A = 0 and B = 1. The output for this present state is $X\ \overline{A}\ B = 0\ \overline{1}\ 0 = 0$.

Present state 11 (A = 1 and B = 1) with input X = 0

With this condition, output of both OR gates is logic 1, which toggles both the flip-flops. Therefore, in the next state A = 0 and B = 0. The output for this present state is $X\ \overline{A}B = 0\ \overline{1}1 = 0$.

Present state 00 (A = 0 and B = 0) with input X = 1

With this condition, output of both OR gates is logic 1, which toggles both the flip-flops. Therefore, in the next state A = 1 and B = 1. The output for this present state is $X\bar{A}B = 1\,\bar{0}\,0 = 0$.

Present state 01 (A = 0 and B = 1) with input X = 1

With this condition, output of both OR gates is logic 1, which toggles both the flip-flops. Therefore, in the next state A = 1 and B = 0. The output for this present state is $X\bar{A}B = 1\,\bar{0}\,1 = 1$.

Present state 10 (A = 1 and B = 0) with input X = 1

With this condition, output of OR gate 1 is logic 1 and output of OR gate 2 is logic 0, which toggles flip-flop A and resets flip-flop B. Therefore, in the next state A = 0 and B = 0. The output for this present state is $X\bar{A}B = 1\,\bar{1}\,0 = 0$.

Present state 11 (A = 1 and B = 1) with input X = 1

With this condition, output of OR gate 1 is logic 1, and output of OR gate 2 is logic 0, which toggles flip-flop A and resets flip-flop B.Therefore, in the next state A = 0 and B = 0. The output for this present state is $X\bar{A}B = 1\,\bar{1}\,1 = 0$.

Derived next states and outputs can be tabulated in the state table shown in the 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	1	1	1	0	0
0	1	1	0	1	0	0	1
1	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0

Table 10.2 State table

From state table, the state diagram can be drawn as shown in the Fig. 10.5.

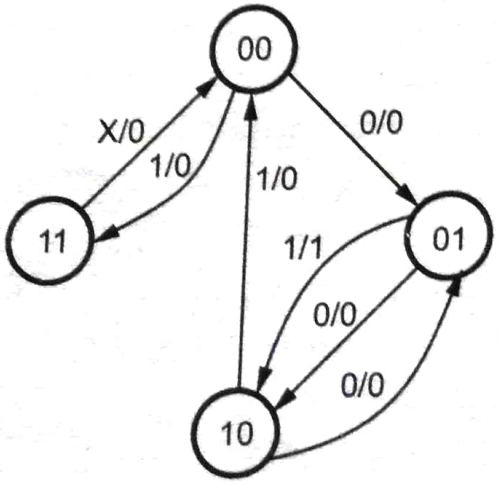


Fig. 10.5 State diagram

Note : The state diagram shown in Fig. 10.5 has a directed line labelled with X/0. This means that for input, X = 0 or X = 1, the next state is 00 and output is logic 0.

Ex. 10.2 : *Derive the state table and state diagram for sequential circuit shown in Fig. 10.6.*

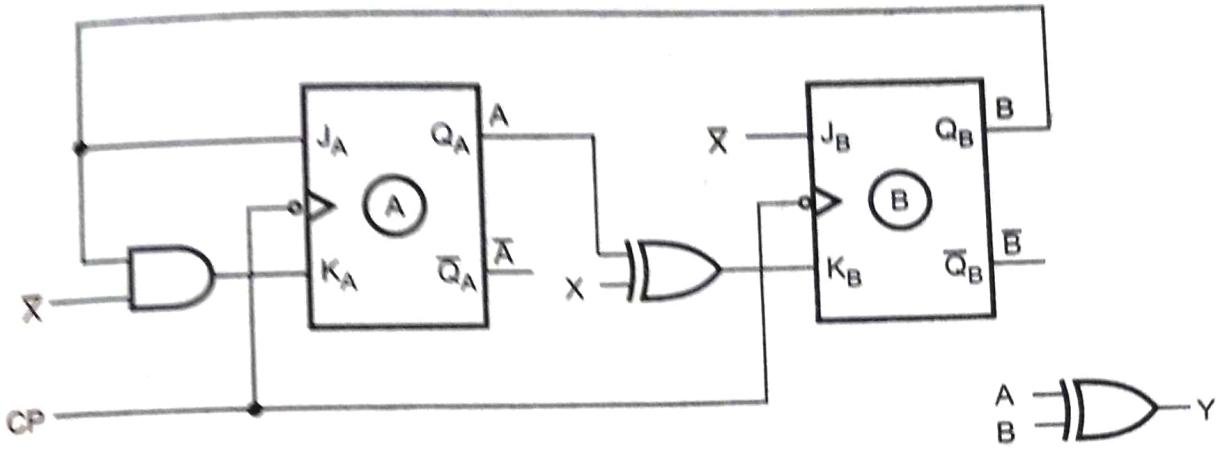


Fig. 10.6

Sol. : The given sequential circuit has two JK flip-flops, so they produce 4 rows in the state table. The circuit has 1 input which produces 2 columns in the next state. The output of the circuit depends only on present state hence it produces only one column.

The state table for the sequential circuit can be derived by defining flip-flop input functions. For the given sequential circuit flip-flop input functions are :

$J_A = B$ $\qquad$ $J_B = \overline{X}$

$K_A = B\overline{X}$ $\qquad$ $K_B = \overline{A}X + A\overline{X} = A \oplus X$

Now from the present states and input values we can derive the flip-flop inputs and hence the next state of the flip-flops. The output can be derived from the present states of the flip-flop. Therefore, the state table for the circuit shown in Fig. 10.6 is as shown in Table 10.3.

Present state		Next state				Output
		X = 0		X = 1		
A	B	A	B	A	B	Y
0	0	0	1	0	0	0
0	1	1	1	1	0	1
1	0	1	1	1	0	1
1	1	0	0	1	1	0

Table 10.3

From the state table, the state diagram can be drawn as shown in the Fig. 10.7.

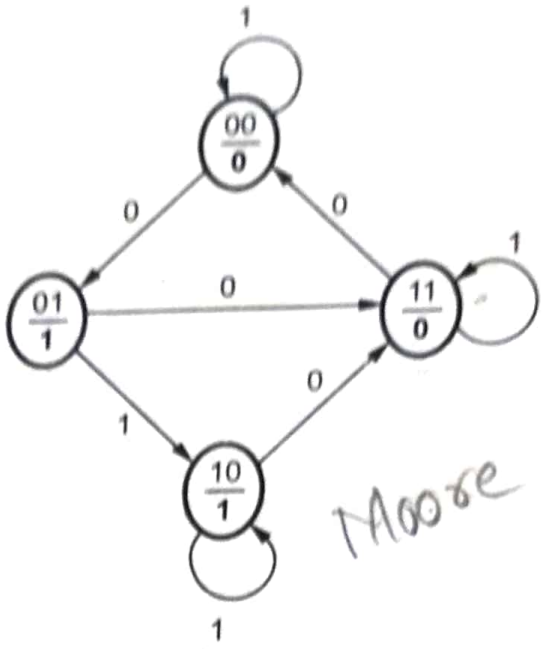


Fig. 10.7 State diagram

10.2 State Reduction

The state reduction technique basically avoids the introduction of redundant states. The reduction in redundant states reduce the number of required flip-flops and logic gates, reducing the cost of the final circuit. The two states are said to be redundant or equivalent, if every possible set of inputs generate exactly same output and same next state. When two states are equivalent, one of them can be removed without altering the input-output relationship.

Let us illustrate the state reduction technique with an example. We start with a sequential circuit whose specification is given in the state diagram of Fig. 10.8. As shown in the diagram, the states are denoted by letter symbols instead of their binary values, because in state reduction technique internal states are not important; but only input-output sequences are important.

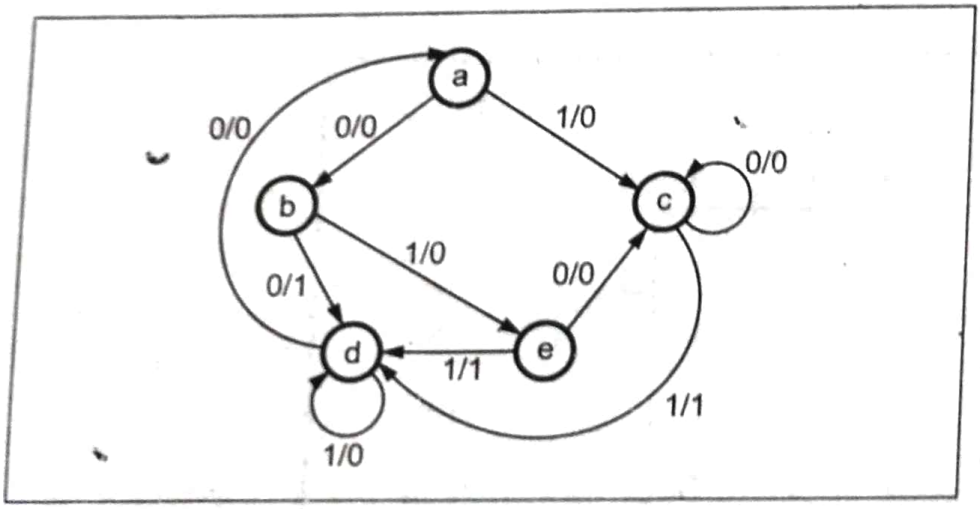


Fig. 10.8 State diagram

Step 1 : Determine the state table for given state diagram

Table 10.4 shows the state table for given state diagram

Present state	Next state		Output	
	X = 0	X = 1	X = 0	X = 1
a	b	c	0	0
b	d	e	1	0
c	c	d	0	1
d	a	d	0	0
e	c	d	0	1

Table 10.4 State table

Step 2 : Find equivalent states

As mentioned earlier, in equivalent states every possible set of inputs generate exactly same output and same next state. In the given circuit there are two input combinations : X = 0 and X = 1. Looking at the state table for two present states that go to the same next state and have the same output for both input combinations, we can easily find that states c and e are equivalent. This is because, c and e both states go to states c and d and have outputs of 0 and 1 for X = 0 and X = 1, respectively. Therefore, state e can be removed and replaced by c. The final reduced table is shown in Table 10.5. The state diagram for the reduced table consists of only four states and is shown in Fig. 10.9.

Present state	Next state		Output	
	X = 0	X = 1	X = 0	X = 1
a	b	c	0	0
b	d	c	1	0
c	c	d	0	1
d	a	d	0	0

Table 10.5 Reduced state table

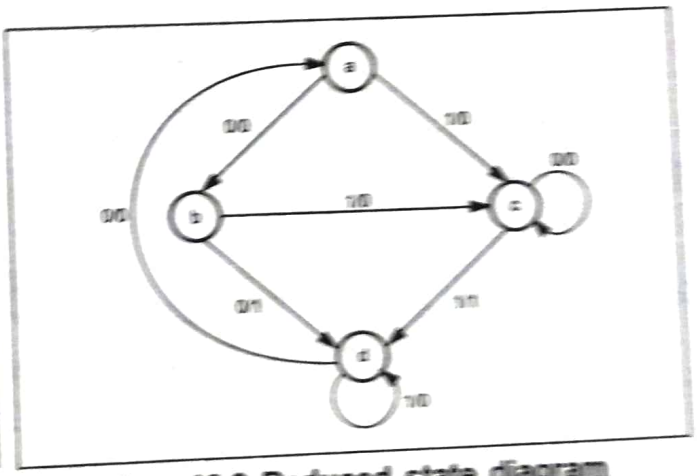


Fig. 10.9 Reduced state diagram

10.3 State Assignment

In sequential circuits we know that the behaviour of the circuit is defined in terms of its inputs, present states, next states and outputs. To generate desired next state at particular present state and inputs, it is necessary to have specific flip-flop inputs. These flip-flop inputs are described by a set of Boolean functions called flip-flop input functions. To determine the flip-flop input functions, it is necessary to represent states in the state diagram using binary values instead of alphabets. This procedure is known as state assignment. We must assign binary values to the states in such a way

that it is possible to implement flip-flop input functions using minimum logic gates. Fig. 10.10 shows the state diagram shown in Fig. 10.9 after state assignments.

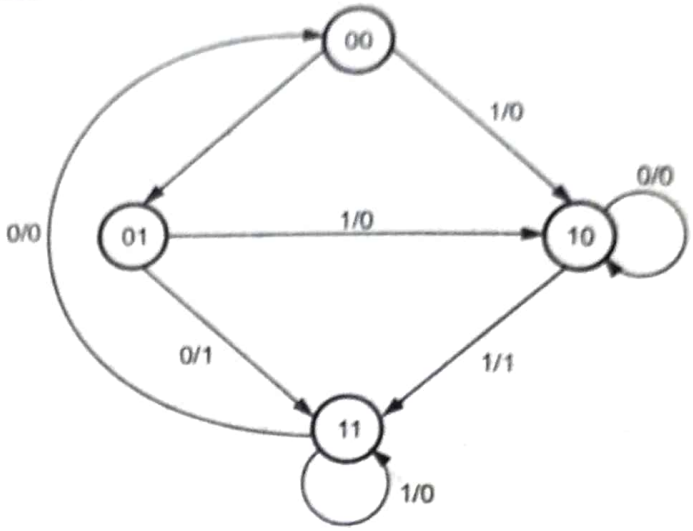


Fig. 10.10 State diagram with binary states

Rules for state assignments

There are two basic rules for making state assignments.

Rule 1 : *States having the **same** NEXT STATES for a given input condition should have assignments which can be grouped into logically adjacent cells in a K-map.*

Fig. 10.11 shows the example for Rule 1. As shown in the Fig. 10.11, there are four states whose next state is same. Thus states assignments for these states are 100, 101,110 and 111, which can be grouped into logically adjacent cells in a K-map.

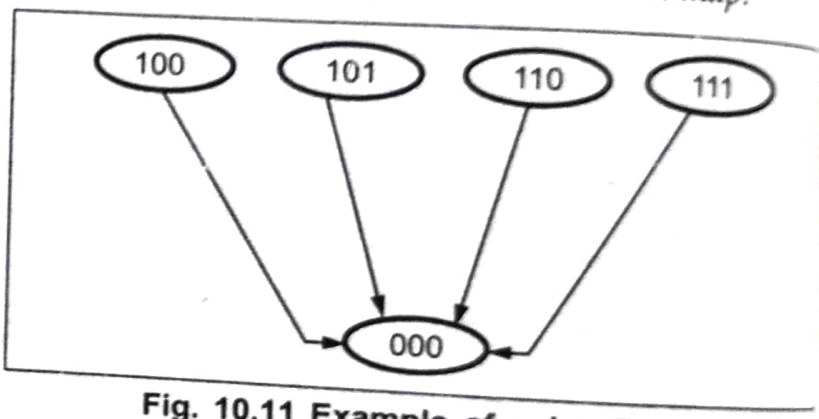


Fig. 10.11 Example of using Rule 1

Rule 2 : *States that are the NEXT STATES of a **single state** should have assignments which can be grouped into logically adjacent cells in a K-map.*

Fig. 10.12 shows the example for Rule 2. As shown in the Fig. 10.12, for state 000, there are four next states. These states are assigned as 100, 101, 110 and 111 so that they can be grouped into logically adjacent cells in a K-map.

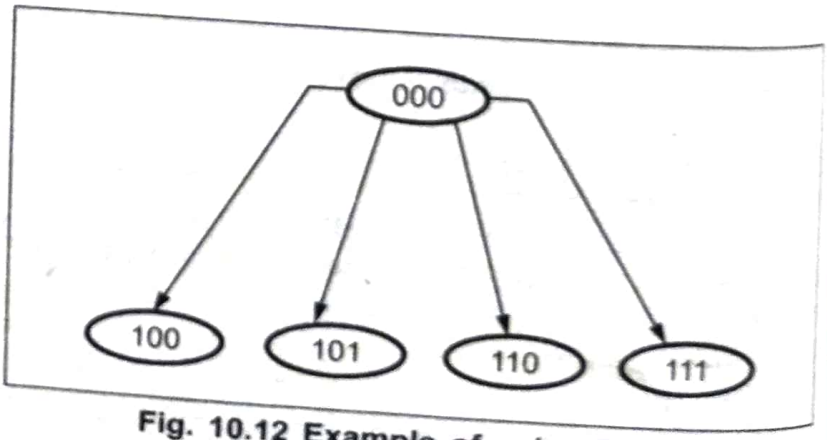


Fig. 10.12 Example of using Rule 2