4.11 SEQUENCE DETECTOR

The "sequence detector" is a clocked sequential circuit or machine which is used to detect the desired binary sequence. Once we know the sequence which is to be detected, we can draw the state diagram for it. From the state diagram, we can design the circuit (see Fig. 4.91)

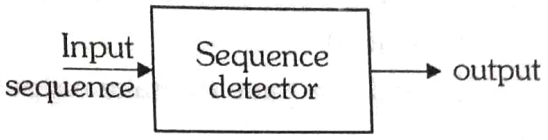

Fig. 4.91 Block diagram.

Generally it produces an output = 1, whenever it detects the desired input sequence and '0' in other cases. There are two types of detectors :

(1) A detector which detects over-lapping input sequence and

(2) A detector which detects non-overlapping input sequence.

The following example illustrates how to determine the state diagram from a given sequence.

Example 4.17. *Design a sequence detector which detects the sequence 100011.*

Solution. Generally the number of states in the state diagram is equal to the number of bits in the sequence. In this example we have six bits in the sequence, so we have six states in the state diagram.

Once the number of states is known, we have to draw the directed lines with inputs and outputs as weights between the two states. Let us start the drawing of directed lines, assuming (arbitrarily) initial state is A.

State A : In this state, the machine can receive either an input-0 or 1. For each of these, inputs, an arc is drawn originating in state A and terminating in the appropriate next state, as shown in Fig. 4.92(*a*).

When input is 1, we have detected first bit in the sequence, hence we have to go to next-state to detect the next bit in the sequence.

When input is 0, we have to remain in state A, because bit '0' is not the first bit in the sequence. In both cases the output is '0' since we have not yet detected all the bits in the sequence.

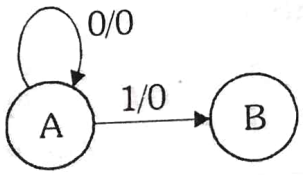

Fig. 4.92 (*a*)

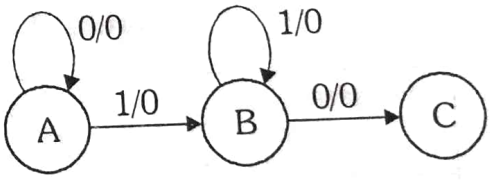

Fig. 4.92 (*b*)

The arc labelled 1/0 indicates input bit is '1' and output is '0'.

State B : When input is '0', we have detected the second, bit in the sequence. Hence, we have to go to next state (C) to detect the next bit in the sequence. (see Fig 4.92(*b*))

When input is 1, we have to remain in the state B, because 1 which we have detected may start the sequence. Output is still zero for both cases.

State C : When input is 0, we have detected the third bit in the sequence, hence we have to go next state (D) to detect the next bit in the sequence.

When input is 1, we have to go to state B, because 1 which we have detected may not be the bit in the sequence to be detected but it may be the start bit of the sequence, hence we are moving to state B. The output is still zero.

State D to state F : As explained for state A, state B and state C, if the desired bit is detected, we have to go for the next state, otherwise we have to go to the previous state from where we can continue the desired sequence.

When complete sequence is detected, we have to make output 1 and go to the initial state. This is shown in Fig. 4.92 (*c*).

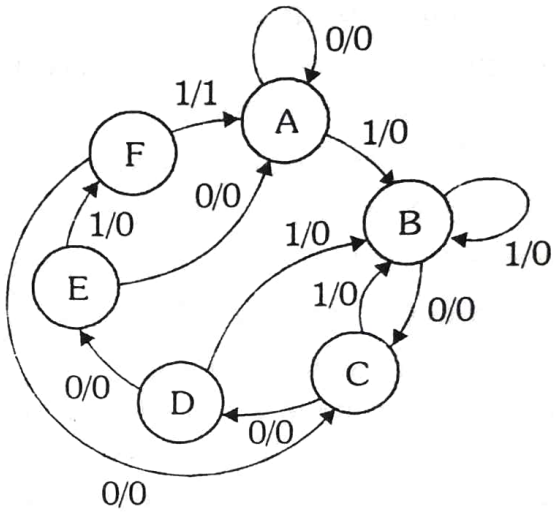

Fig. 4.92 (c)

To determine the logic diagram, let us assume state assignments as

A = 000, B = 001 C = 010

D = 011, E = 100 and F = 101

To design the logic diagram, we have 6 states, so we need three flip-flops. Assume that flip-flops are named as A, B and C and one output y and input x. The excitation table is shown in Table 4.38. The flip-flop input equations can be obtained by using K-maps as shown in Fig. 4.93.

Table 4.38 Excitation table

Input x	*Present state* $A_n\ B_n\ C_n$	*Next state* $A_{n+1}\ B_{n+1}\ C_{n+1}$	*Output* z	*Flip-flop outputs* $J_AK_A\ J_BK_B\ J_CW_C$
0	0 0 0	0 0 0	0	0 × 0 × 0 ×
0	0 0 1	0 0 1	0	0 × 1 × × 1
0	0 1 0	0 1 1	0	0 × × 0 1 ×
0	0 1 1	1 0 0	0	1 × × 1 × 1
0	1 0 0	0 0 0	0	× 1 0 × 0 ×
0	1 0 1	0 1 0	0	× 1 1 × × 1

1	0 0 0	0 0 1	0	0 × 0 × × ×
1	0 0 1	0 0 1	0	0 × 0 × 1 0
1	0 1 0	0 0 1	0	0 × × 1 1 ×
1	0 1 1	0 0 1	0	0 × × 1 × 0
1	1 0 0	1 0 1	0	× 0 0 × 1 ×
1	1 0 1	0 0 0	1	× 1 0 × × 1

K-map simplification

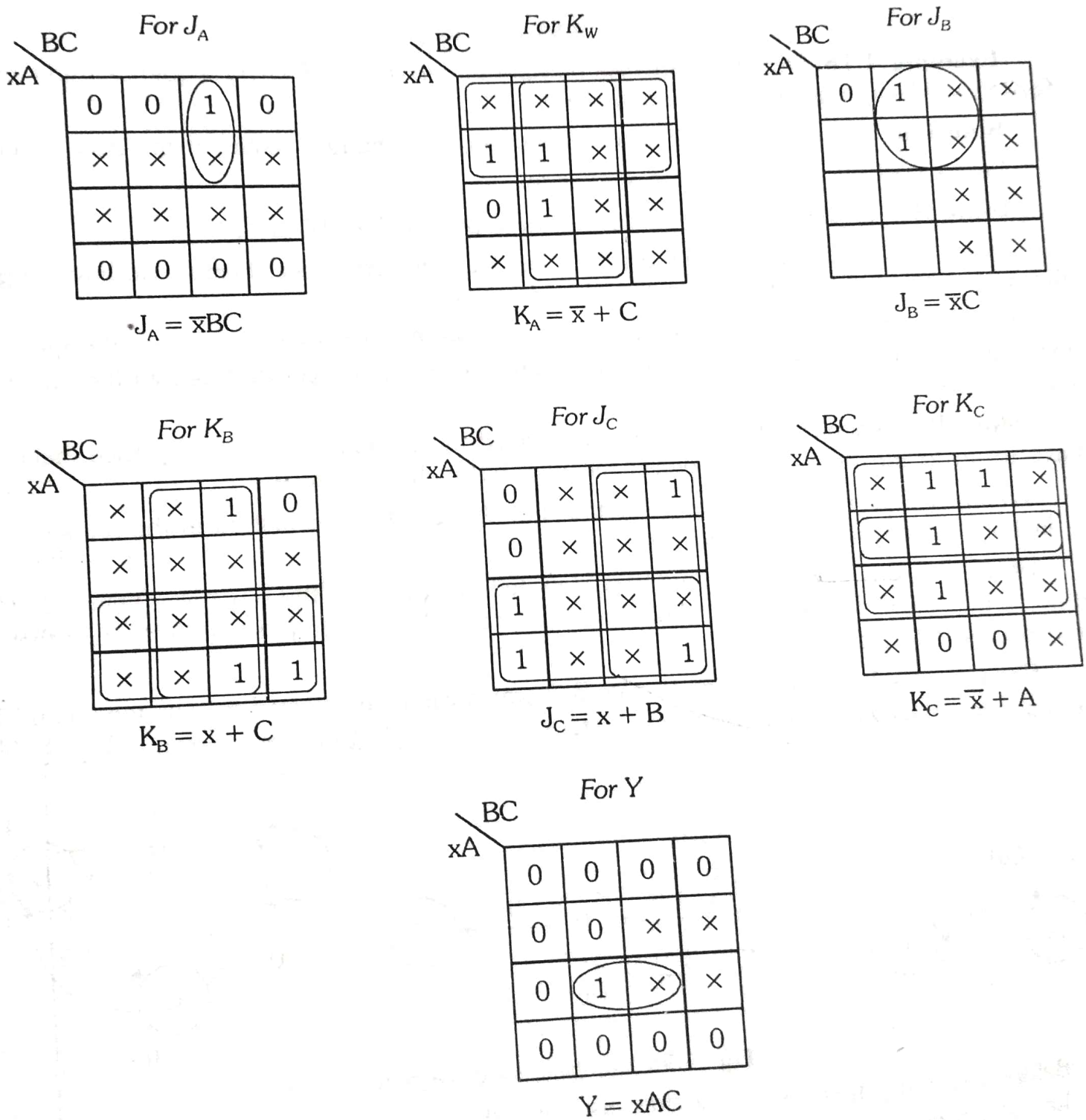

Fig. 4.93 K-maps.

Logic diagram. The logic diagram shown in Fig 4.94.

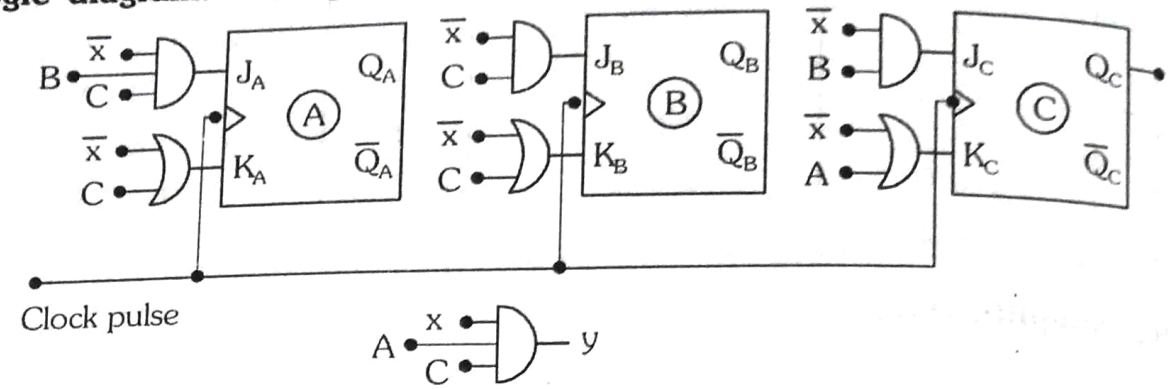

Fig. 4.94 Logic diagram.

Example 4.18. *Design a sequence detector to detect the following sequence using JK flip-flops (use Mealy machine). The sequence is '110'.*

Solution. The given sequence has 3-bits. So, we require 3 states in the state diagram. Let us start by assuming initial state is 'A'.

State A : In this state, the machine may receive either '0' or '1'.

When input is 1, we have detected the first bit in the sequence. Hence, we have to go to the next state to detect the next bit in the sequence.

When input is 0, we have to remain in the state 'A' because bit '0' is not the first bit in the sequence. In both cases, the output is 0, since we have not yet detected all the bits in the given sequence.

State B : When input is 1, we have detected the second bit in the sequence, hence we have to go to the next state to detect the next bit in the sequence.

When input is '0', we have to go to the state 'A', to detect the first bit in the sequence *i.e.*, 1. In both cases, the output is still zero.

State C : When input is '0', we have detected the last bit in the sequence, hence we have to go to the initial state to detect the next sequence and make output high indicating that the sequence is detected.

When input is 1, we have to go to state B, 1 which we have detected may be the second 1 in the sequence (thus we are not moving to state A). The state diagrams are shown in Fig. 4.95.

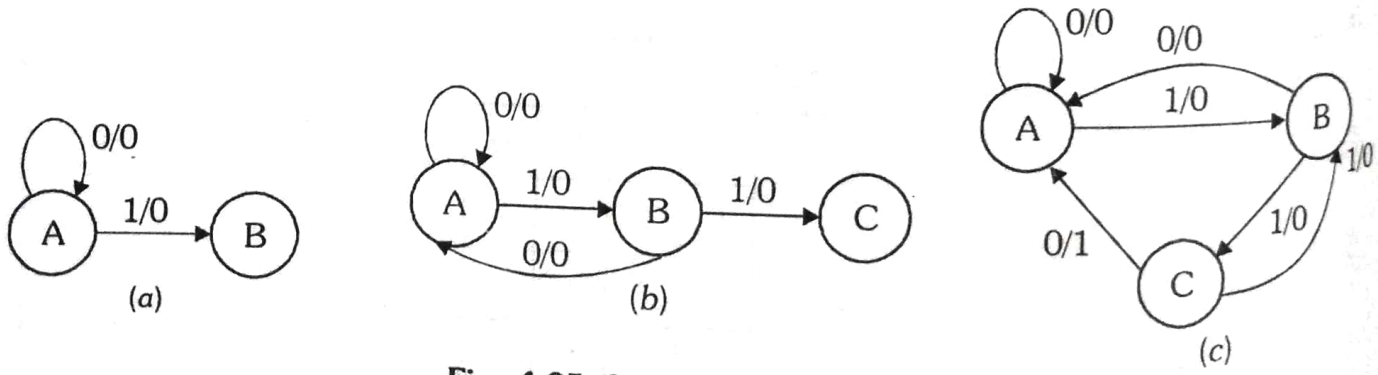

Fig. 4.95 Complete state diagram.

Before designing, first we go to state assignment.

Let us assume A = 00, B = 01 and C = 10.

The excitation table is shown in Table 4.39. Totally, we have three states. So we need B flip-flops named as A and B, output y and input is x.

Table 4.39 Excitation table

Input	*Present state*	*Next state*	*Output*	*Flip-flop inputs*
x	$A_n\ B_n$	$A_{n+1}\ B_{n+1}$	y	$J_AK_A\ \ J_BK_B$
0	0 0	0 0	0	0 × 0 ×
0	0 1	0 0	0	0 × × 1
0	1 0	0 0	1	× 1 0 ×
1	0 0	0 1	0	0 × 1 ×
1	0 1	1 0	0	1 × × 1
1	1 0	1 0	0	× 0 0 ×

K-map simplification. Flip-flop input functions are shown in Fig. 4.96.

For J_A

x \ AB	00	01	11	10
0	0	0	×	×
1	0	1	×	×

$J_A = xB$

For K_A

x \ AB	00	01	11	10
0	×	×	×	1
1	×	×	×	1

$K_A = \overline{X}$

For J_B

x \ AB	00	01	11	10
0	0	×	×	0
1	1	×	×	1

$J_B = X\overline{A}$

For K_B

x \ AB	00	01	11	10
0	×	1	×	×
1	×	1	×	×

$K_B = 1$

Fig. 4.96

Logic diagram. The logic diagram is shown in Fig. 4.97.

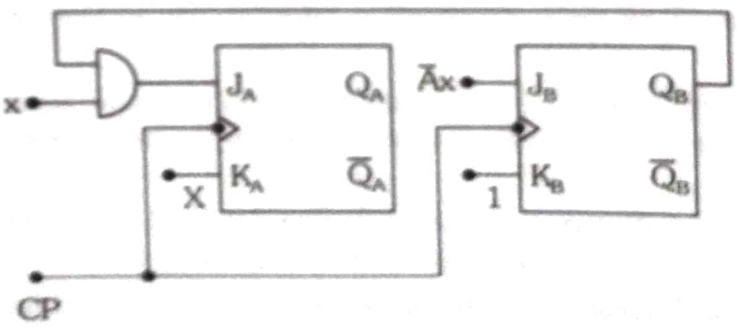

Fig. 4.97 Logic diagram.