11 Capabilities and Minimisation of Sequential Machines

11.1 Introduction

In synchronous or clocked sequential circuits, clocked flip-flops are used as memory elements, which change their individual states in synchronism with the periodic clock signal. Therefore, the change in states of flip-flops and change in state of the entire circuit occurs at the transition of the clock signal.

The synchronous or clocked sequential circuits are represented by two models.

Moore circuit : The output depends only on the present state of the flip-flops.

Mealy circuit : The output depends on both the present state of the flip-flop(s) and on the input (s).

The another name of sequential circuit is **sequential machine**. Therefore, Mealy and Moore circuit are also referred to as **Mealy machine** and **Moore machine**, respectively.

11.2 Moore Circuit

As mentioned earlier, when the output of the sequential circuit depends only on the present state of the flip-flop, the sequential circuit is referred to as **Moore Circuit or Moore Machine.** Let us see one example of Moore circuit. Fig. 11.1 shows a sequential circuit which consists of two JK flip-flops and AND gate. The circuit has one input X and one output Y.

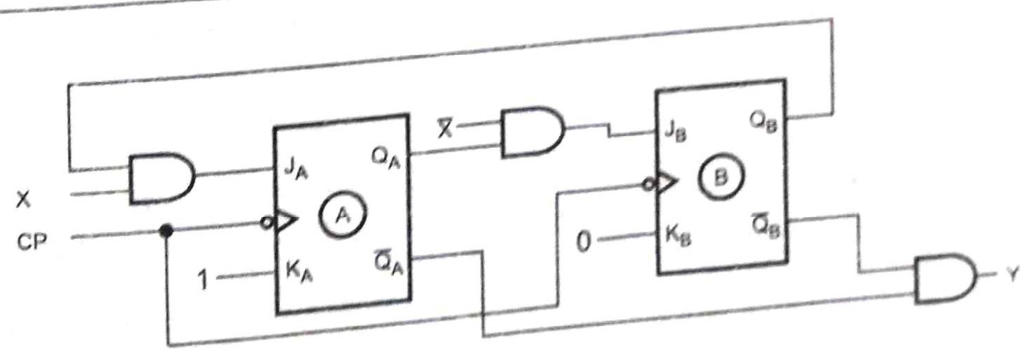

Fig. 11.1 Example of Moore circuit

11.4.1 Moore Vs Mealy Machines

Moore machine	Mealy machine
1) Its output is a function of present state only.	1) Its output is a function of present state as well as present input.
2) Input changes does not affect the output.	2) Input changes may affect the output of the circuit.
3) Moore circuit requires more number of states for implementing same function.	3) It requires less number of states for implementing same function.
4) $z(t) = \lambda\{ s(t) \}$	4) $z(t) = \lambda\{ s(t), x(t) \}$

Table 11.1

11.5 Capabilities and Limitations of Finite-State Machine

1. **Periodic sequence of finite states** : With n-state machine we can generate periodic sequence of n states or smaller than n states. For example, in 5 state machine we can have maximum periodic sequence as 0, 1, 2, 3, 4, 0, 1 .....
2. **No infinite sequence** : Consider an infinite sequence such that an output is 1 when and only when the number of inputs received so far is equal to $K(K+1)/2$, for K = 1, 2, 3, ... i.e., the desired input output sequence has the form.

Input : X X X X X X X X X X X X X X X X X X X X X

Output : 1 0 1 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 .....

Such infinite sequence can not be produced by finite state machine.

3. **Limited memory** : The finite-state machine has a limited memory and due to limited memory it can not produce certain outputs. Consider a binary multiplier circuit for multiplying two arbitrarily large binary numbers. If we implement this with a finite - state machine capable of performing serial multiplication, we can find that it is not possible to multiply certain numbers. Such limitation does occur due to the limited "memory " available to the machine. This memory is not sufficient to store arbitrarily large partial products resulted during multiplication.