Finite State Machine – Analysis and Design Methodologies

Dr. Saibal Kumar Pradhan, Professor

1. **Introduction:** In digital electronics, we come across two types of circuits – combinational and sequential. Flip-flop makes the difference - combinational circuits haven't but sequential circuits have the flip-flops along with other logic gates. In undergraduate course, digital electronics is taught as a single paper covering logic gates, Boolean algebra, number system, combinational and sequential circuits. For obvious reasons, a lot of lectures have been devoted to logic gates, Boolean algebra, number system and combinational circuits for concept building and at the end we often found only a few classes left to cover sequential circuits leading to negligence in sequential circuit analysis and design. This topic is chosen aiming to bridge this gap so that students can have a better understanding in sequential circuit design.

   Sequential circuits have one or more flip-flops along with other logic gates. Each flip-flop has two outputs (0 or 1) or states. A sequential circuit having two flip-flops has four states, 00, 01, 10 and 11. Thus number of states of a sequential circuit depends upon the number of flip-flops in the circuit. A sequential circuit is often called a ***state machine*** or a ***finite state machine*** (**FSM**) or a ***finite automaton*** (plural automata) as it is in one of a finite number of states at any given time and it can change from one state to another in response to an external input.

   An FSM is of two types – Moore Machine and Mealy Machine. A Mealy Machine is an FSM whose output depends on the present state as well as the present inputs whereas the output of a Moore Machine depends only on the present state.

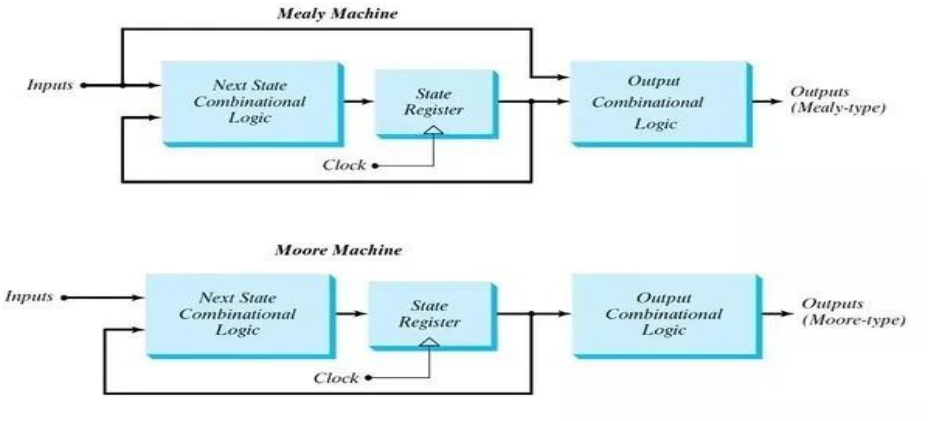

Fig.1: Block Diagram of a Mealy and a Moore Machine

Thus hardware implementation of an FSM requires a register to store state variables, a block of combinational logic that determines the state transition, and a second block of combinational logic that determines the output.

2. **Analysis of a Mealy Machine:** ***Analysis*** helps us to predict the functions of a given circuit. Fig.2 shows a Mealy Machine whose output y is a function of input x as well as present state (AB).

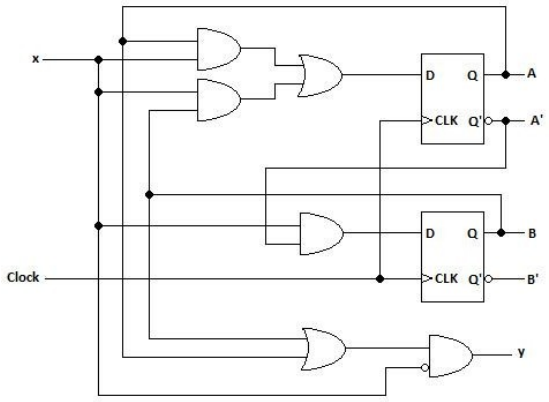

Fig.2: A Mealy Machine

2.1 State Equations: A ***state equation*** specifies the next state as a function of present state and inputs and describes the behavior of a clocked sequential circuit. Outputs (AB) of the D flip-flops represent the present state and its inputs represent the next state. Thus state equations for the above Mealy Machine are:

A(t+1) = A(t).x(t) + B(t).x(t) B(t+1) = A'(t).x(t)

where, terms with t+1 represents next state of the flip-flops (outputs after the clock edge). For convenience, let us omit the designation (t) from all present state variables and rewrite the state equations as:

A(t+1) = Ax + Bx B(t+1) = A'x

Similarly, the present value of the output is: y(t) = (A+B)x'

2.2 State Table: A ***state table*** records the sequence of inputs, outputs and flip-flop states. The state table of the circuit in Fig.2 is as follows:

Present State		Input	Next State		Output
A	**B**	**x**	**A(t+1)=Ax+Bx**	**B(t+1)=A'x**	**y=(A+B)x'**
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

Table-1: State Table for the Sequential Circuit in Fig.2

State table in Table-1 clearly indicates the transition from present state to the next state for different input on arrival of a clock edge.

2.3 State Diagram: The information available in state table can be represented graphically in the form of a ***state diagram*** where the state changes are more prominent. In a state diagram, a state is represented by a circle and the sate transition between states is indicated by a directed line connected the circles. Fig.3 below depicts the state diagram for the circuit in Fig.2.

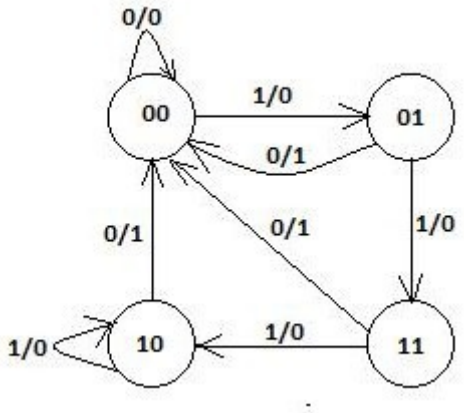

Fig.3: State Diagram for the Circuit in Fig.2

The binary numbers inside the circles represent the states of the flip-flops, the directed lines are labeled with the values of input and output separated by a slash. State diagram and state table carry the same information but sometimes state diagram being pictorial representation is more suitable for human interpretation. For example, the above state diagram clearly shows that the output is 0 as long as the input is 1.

3. **Design Methodology:** In circuit analysis, starting point is the circuit diagram and we have to determine its behavior. Design is the reverse process where the behavior of the circuit is given and we have to design the circuit. Designing synchronous sequential circuits follow a systematic approach and are summarized below:
   1. From the word description and specifications of the desired operations, derive a state diagram.
   2. Assign the binary values to the states.
   3. Obtain the binary coded state table.
   4. Choose the type of flip-flop.
   5. Expand the state table by adding inputs of each flip-flop.
   6. Derive the simplified flip-flop input equations and output equations.
   7. Draw the logic diagram.

3.1 Problem Description: *Design a MOD -5, 3-bit synchronous counter to count in the following sequence: 2, 3, 5, 1 and 7. The counter must be self-starting with the count states of 0, 4, and 6 leading directly to state 2.*

3.2 State Diagram: State diagram of the counter is drawn in Fig.4. The counter as such has no input and output other than flip-flop output states.

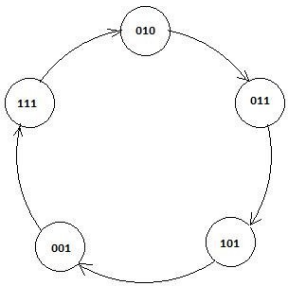

Fig.4: State Diagram of the Mod-5 counter stated in the problem

3.3 Assignment binary values to the states: Circuit to be designed is a mod-5 counter, so it has 5 states which will require 3 flip-flops. Thus the states of the counter will be a 3- bit binary number which are written inside the states in the state diagram.

3.4 Binary Coded State Table: State table can just be prepared following the state diagram. As the circuit has no input and output, the state table will contain only present states and the next states of the flip-flops.

Present State	Next State
ABC	ABC
010	011
011	101
101	001
001	111
111	010

Table-2: State Table of the Counter

3.5. Choose the Type of Flip-Flop: Any of the four flip-flops viz. D, SR, JK and T can be used to design the circuit. Let us use JK flip-flop for the said design. Before proceeding further, let us recall the excitation table of JK flip-flop which will be needed to expand the state table for finding flip-flop input equations.

PS	NS	Input	
Q	Q+	J	K
0	0	0	X
0	1	1	X

1	0	X	1
1	1	X	0

Table-3: Excitation Table for JK Flip-Flop

Present State	Next State	JK Inputs for the said state transition					
ABC	ABC	J_A	K_A	J_B	K_B	J_C	K_C
010	011	0	X	X	0	1	X
011	101	1	X	X	1	X	0
101	001	X	1	0	X	X	0
001	111	1	X	1	X	X	0
111	010	X	1	X	0	X	1

Table-4: Extension of the State Table Including the Inputs of the Flip-Flops

3.5. Derivation of Simplified Flip-Flop Input Equations: All six inputs of 3 JK flip- flops A, B and C are obtained in Table-4 and now we have to find simplified expressions for all the six inputs viz. J_A, K_A, J_B, K_B, J_C and K_C with the help of K-maps.

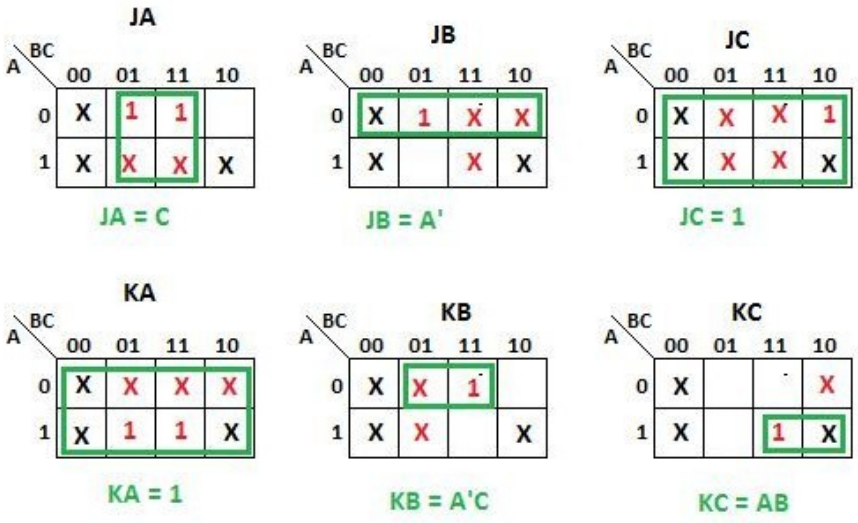

Fig.5: K-map Simplification for Flip-Flop Input Equations

3.6. Circuit Diagram: Logic diagram of mod-5 counter is shown in Fig.6 using three JK flip-flops and two 2-input AND gates.

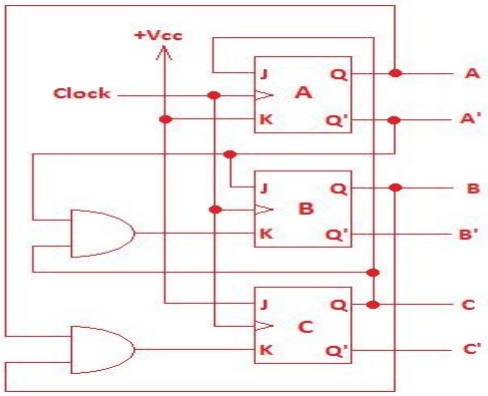

Fig.6: Circuit Diagram

4. **Unused States and Lock-out Conditions:** The counter circuit shown in Fig.6 cycles through the states 2, 3, 5, 1, 7 when on power on the counter launches in one of these states. However, if the counter launches in any of the unused states viz. 0, 4 and 6 then the counter may either come back to the desired counting cycle after a few clock or it may cycles through the unused states and never comes back to the desired loop. If the counter comes back to the desired loop it is called ***self starting*** otherwise the counter is said to be lock out condition. In case of lock out conditions can be overcome in two ways – a) consider all the unused states in the state diagram right at the beginning such that the transitions from all of them take place to one of the valid states and the circuit so designed will never be in lock out condition b) after the design is over if it found that the counter is in lock out condition then develop a circuit such that the next states from all the unused states will be one of the valid states. Let us check our design for lock out condition.

Present State	JK Inputs for the said state transition						Next State
ABC	$J_A = C$	$K_A = 1$	$J_B = A'$	$K_B = A'C$	$J_C = 1$	$K_C = AB$	ABC
000	0	1	1	0	1	0	011
100	0	1	0	0	1	0	011
110	0	1	0	0	1	1	011

Table-5: Verification of the Counter Circuit

It is evident from the Table-5 that if the counter launches in any one of the unused states, it comes back to 011 which is a valid state in the very next clock and then cycles through the valid states only. So the counter is self starting. Modified state diagram of the circuit is shown in Fig.7.

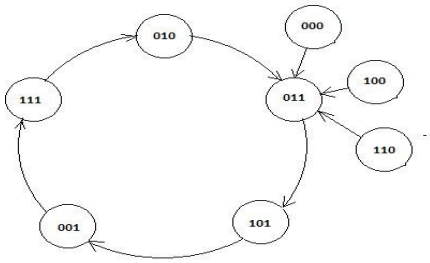

Fig.7: Modified State Diagram

5. **Conclusion:** A sequential circuit can be designed using any of the four flip-flop types but it is always advantageous to design with D flip-flop as the inputs will be the outputs after the clock edge and input equations can be obtained directly from the state table. Implementation using T flip-flop is also straight forward. However, the design using SR and JK flip-flops has more input equations and requires more involved combinational circuit design.

Reference:

1. Digital Design by Morris Mano and Michael D Ciletti, Pearson
2. https://en.wikipedia.org/wiki/Moore_machine
3. https://en.wikipedia.org/wiki/Mealy_machine