A HDL Based FSM Example for Education

Assistant Prof.Tirupati M.Goskula [1], Razik Khan [2], Mohammed Tauseef [2], Nafis Ansari [2]

[1]Assistant Professor, Electronics & Telecommunication, RTMNU/Anjuman College of Engineering & Tech, Nagpur, Maharashtra, India

[2]Scholars, Electronics & Telecommunication, RTMNU/Anjuman College of Engineering & Tech, Nagpur, Maharashtra, India

***Abstract*—With recent technological advancements, modern societies are apt more and more needful on the automated machines. It is in order to cope with their fast-going life. Modern mechanized machinery adapt their series of actions depending on their environment and events. The FSM (Finite state machine) is used to precisely express those orders of actions or instructions. In this article two FSM machines kinds, Moore and Mealy, are deliberated. Viewing dissimilar results in order to exhibit the status of FSM modeling. An code detector circuit is designed\ by engaging both Moore and Mealy machines. It is a FSM design example, can be used for students concepts building and display. These designs are executed in HDL. A comparison is also made based on both implementations**.

Index Terms*—*FSM; automation; VHDL; Xilinx-ISE; timing diagram; computer aided design.(keywords)

I. INTRODUCTION

In any programming languages, there will always be more than one method to code the same problem. Hardware portrayal languages like Verilog and VHDL are no dissimilar. It also delivers numerous replacements to the designer as to how to accomplish a similar task. So It is up to the designer to code in an resourceful way that results is finest act and minimum source use. In this paper, earliest the assessment between the Mealy and Moore state machine is delivered. Both has its own merit and dismertis so it is up to limited to pick proper design based on application. Same way assessment is done among different State programming patterns like binary and one hot programming. The effect on area and performance by using these schemes is also deliberated in the paper.

II. TYPE OF STATE MACHINES

There are generally two types of state machine.

1. Mealy state machine
2. Moore state machine

In Mealy state machine output depends on existing input and present state while in Moore state machine output only depend upon present state. Figure 1 shows the general flow chart of mealy state machine while figure 2 shows the general flow chart of Moore state machine.

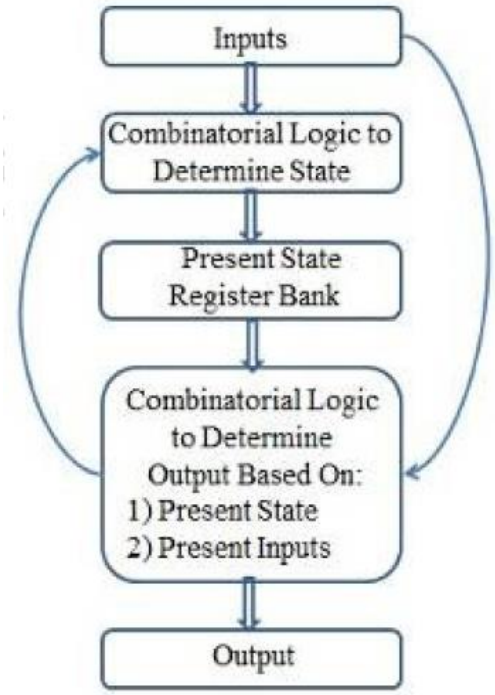

Fig.1 Mealy State Machine

As can be got from Figure 1 in Mealy state machine combinatorial logic is there to establish the value of next state liable upon the value of input. Output is also determined by combinatorial logic based on input and current state value. in order logic is only used for loading states value. Same is the case in Figure 2 for Moore machine but here output only depend upon the current state value.

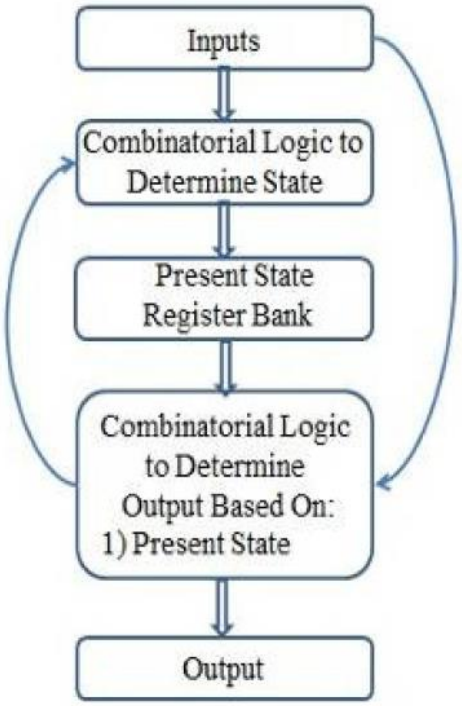

Fig. 2 Moore State Machine

Now to relate Mealy and Moore machine one diagram is taken. Study the case of a circuit to detect a pair of 1's or 0's in the single bit input. If two one's or two zero's comes one later added, output should go high. Or else output must be low. Here is a Moore type state evolution diagram for the circuit:

- When reset, state goes to 00
- State will be 01, If input is 1
- State goes to 10, If input is 0
- State will be 11 if input repeats
- After state 11, goes to 10 states or 01 depending on the input. Once the state reaches the state 11 then assign 1 to out since it is an asynchronous we have to check the reset before making output high.

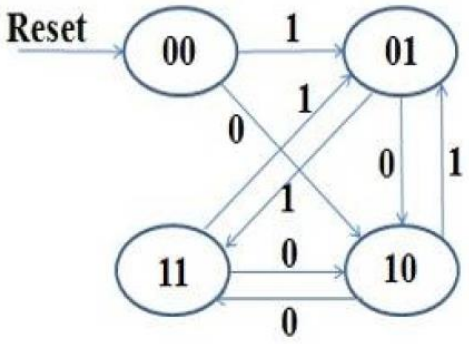

Fig. 3 State Diagram of Moore Machine

Now, let renovation the above circuit using Mealy style state machine. Output depends on together state and input State development diagram is as follows:

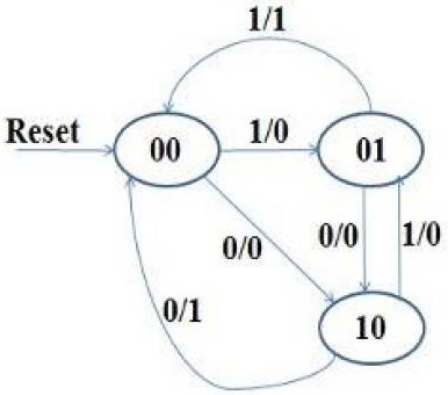

Fig. 4 State Diagram of Mealy Machine

III. Comparison Between Two State Machines

Moore state machine is easier to plan than Mealy. First plan the states liable on the earlier state and input. Then design output only depending on state. While in Mealy, you have to think nearby both state and input although scheming the output. In Moore four states are used to design the circuit and mealy three states are used so here are extra amount of states in mealy machine then Moore machine. So, occasionally number of flip flops will be extra in the Moore machine. In Mealy, output changes instantaneously when the input alterations. We can detect this point when you simulate the codes above. In Moore case, output develops high in the clock next to the clock in which state goes 11. So, Mealy is quicker than Moore. Mealy gives instant response to input and Moore gives response in the next clock. Now, most of the exclusive use only one always block to design state machine so combinational and chronological logic is mixed in this block so it might create a problem when you synthesize the code and then verify gate level netlist. So It is more efficient if you separate combinational and chronological logic in the Mealy and Moore machines.

Modelling Mealy state machine
always@(in or pres_state) //Next State Decoder
always@(posedge clock) //Memory
always@(in or pres_state) //Output Decoder

Modelling Moore state machine
always@(in or pres_state) //NS Decoder
always@(posedge clock) //Memory
always@(pres_state) //Output Decoder

IV. FSM State Encoding Schemes

Separate state in a finite state machine can be signify with a single pattern of one's and zeros and it is called state encoding. Two most popular programming schemes are binary and one hot encoding. In this paper, will briefly deliberate both of them and deliberate how to select the best programming scheme that suits your design, so proficient recital and resource usage can be certified. In binary encoding the linking between the amount of state bits and number of states is characterized by the following equation.

$$B= \log2(S)$$

So to implement the state machine by four states with a binary programming scheme, two flip flops or statebits can be used to uniquely encode four states as follows:

State1 = "00"
State2 = "01"
State3 = "10"
State4 = "11"

The Gray code binary indoctrination method can too be used where single bit transform on a point. Gray code binary proposal is helpful what time the outputs of the state bits are old asynchronously. For example, if state machine switches as of state '10' to '01' as it does in sequential binary programming and the registers perform not switch the outputs as accurately the similar time , temporary outputs of moreover '11' or '00' can exist. This sort of variation can grounds unpredictable results throughout the circuit. A one-hot programming scheme uses one register for each state. For instance four registers are used for a 4-state machine-by simply one state bit elevated at a time. State programming can be complete as follow in one hot programming:

State1 = "0001"
State2 = "0010"
State3 = "0100"
State4 = "1000"

V. CONCLUSION

In this paper dissimilar types of state machines and types of programming schemes are discussed. Comparison amongst them is as well made. When there is need for earlier state mechanism then mealy machine is used but it makes intend simple. In compound design, to construct state mechanism design simpler Moore machine is preferred. Encoding schemes also play a vital role in shaping the area and recital of the state machine. Binary programming scheme uses less area but can lead to impulsive performance if asynchronous outputs are used. One hot programming uses more area but has better technique recital.

REFERENCES

[1] Bhaumik Vaidy and Devani Anupam, "A Study of different Finite state machine design and efficient coding techniques,"Journal Of Information, Knowledge and Research in Electronics and Communication Engineering, Oct2013.

[2] IEEE standard Verilog Reference Manual , IEEE standard 2001

[3] Verilog FAQ by Shivkumar Chanod, Needamangalam Balachandar

[4] Design Compiler User Guide from Synopsys

[5] Verilog HDL by Samir Palnitkar