DATE OF PUBLICATION 26 NOV 2013 ISSN: 2348-4098

Nidhi Gopal

Volume: 1 Issue: 1

VHDL IMPLEMENTATION FOR FSM BASED APPROACH OF TRAFFIC LIGHT CONTROLLER

Nidhi Gopal[1], Madhuri Panwar[2], Pinky Gupta[3]

[1]M. Tech Scholar, Department of ECE, Jayoti Vidyapeeth Women's University, Rajasthan, India, nidhigopal8@gmail.com

[2]M. Tech Scholar, Department of ECE, Jayoti Vidyapeeth Women's University, Rajasthan, India, bhumipanwar01@gmail.com

[3]Assistant Professor, Department of ECE, Jayoti Vidyapeeth Women's University, Rajasthan, India, er.pinkygupta@gmail.com

Abstract

Finite state model is the best method now-a-days for describing control systems, since it represents temporal behavior of systems, in the form of states and transitions between them. This modeling technique for sequential logic circuits is very helpful in designing circuits which have well defined inputs, containing all possible system states necessary conditions of transitions, and all possible outputs from each state. This paper deals with VHDL implementation for FSM based approach for traffic light control systems. Traffic light controller follows a timed FSM architecture. Since VHDL is a very powerful and flexible language, code for this FSM is made, and results are observed.

Index terms: *Control systems, FSM, VHDL*

1. INTRODUCTION TO FINITE STATE MACHINES

A Finite State Machine (FSM) is an example of a state-oriented model. A state oriented model is that which represents the system as a set of states and a set of transitions between them, which are triggered by external events. **A state-oriented model** is most suitable for control systems, such as real time reactive systems, where the system's temporal behavior is the most important aspect of the design.

Basically, the FSM model consists of a set of states, a set of transitions between them, and the set of actions associated with these states, or, transitions. More formally, a FSM is a quintuple:-

$$< S, I, O, f: S\times I \rightarrow S, h: S\times I \rightarrow O >$$

where $S= \{s_1, s2,\ldots s_l\}$ is a set of states, $I = \{i_1, i2,\ldots i_{m}\}$ is a set of inputs and $O= \{o_1, o_2,\ldots o_n\}$ is a set of outputs, 'f' is a next state function, which determines the next state from current state and input, and 'h' is an output function, which determines the outputs, also from the current states and input. Note that each FSM has a state that is distinguished as a start state and a set of states distinguished as final states.

1.1 Example

The figure below shows FSM that models an elevator controller in a building with three floors. In this model, the set of inputs, $I= \{r_1, r2, r3\}$ represents the floor requested. For example r_2 means that floor 2 is requested. The set of outputs $O= \{d_2, d1, n, u1, u2\}$ represents the direction and number of floor the elevator should go. For example, d_2 means that the elevator should go down 2 floors, u_2 means that the elevator should go up 2 floors, and n means that the elevator should stay idle. From the figure, we can see that if the current floor is 2 (i.e. the current state is S_2), and floor 1 is requested, then the output will be d_1.

Nidhi Gopal

Volume: 1 Issue: 1

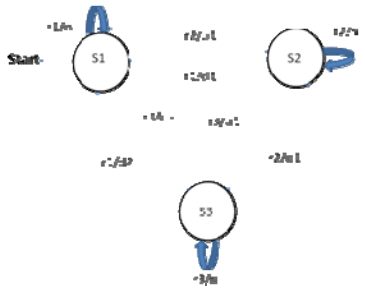

Figure 1: Finite State Model for an elevator Controller

1.2 Types of FSM :

There are two types of FSM models: Mealy Model and Moore Model

1. **Mealy Model :**
   a) It is a transition based model.
   b) In this, the output value depends on the state and input values (h:S × I → O)
   c) The example discussed above is an example of Mealy model.
   d) Practically, it requires less state than Moore model, because there may be multiple arcs pointing to a single state, each arc having different output value.

2. **Moore Model:**
   a) It is a state based model.
   b) In this, the output values depends only on the states of FSM (h: S → O)
   c) Practically, it requires more states, as all the output values would require its own state.

1.3 Controller architecture :

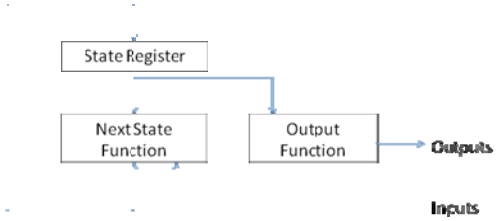

Figure 2: A generic FSM block Diagram

The controller architecture of FSM is application specific architecture. It is the simplest form, and a straight forward implementation of FSM defined by < S, I, O, f, h>.

A controller consists of a register, and two combinational blocks (As shown in figure). The register is usually called the "state register", is designed to store the states in S, while the two combinational blocks, referred to as the "Next state function" and the "Output Function", implements the function f and h. "Inputs" and "Outputs" are representations of Boolean signals, that are defined by sets I and O.

There are two types of Controllers:

1. **Transition Based:** Output function is dependent upon two parameters, namely, state registers and inputs.
2. **State based:** Output function is dependent only on State registers.

Note: Since the inputs and output are Boolean signals, in either case, this architecture is well suited to implementing controllers that do not require complex data manipulation. The controller synthesis consists of state minimization and encoding, Boolean minimization and technology mapping for the nest state and output functions.

2. FSM FOR TRAFFIC LIGHT CONTROLLER

Traffic light system is a best way to manage traffic in cities. It helps in avoiding accidents and manage rules and regulation for the vehicles as well as humans. Traffic light controller is a main part of traffic control system.

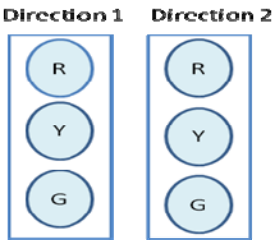

Figure 3: 2-way Traffic Light Controller showing Red, Yellow and Green colors

1) Three modes of operation: regular test, and standby.
2) ***In regular mode*** : Four states of operation, called RG (red in direction 1 and green in direction 2 ON), RY (Red in direction 1 and yellow in direction 2 ON), GR(Green in direction 1 and red in direction 2 ON), and YR(Yellow in direction 1 and red in direction 2 ON), each with an independent time duration.
3) ***In test mode:*** Allow all preprogrammed times to be overwritten (by a manual switch) with a small value, such that the system can be easily tested during maintenance (1 second per state).
4) ***In standby mode:*** If set (by a sensor accusing malfunctioning, for example, or by a manual switch) the system should activate the yellow lights in both directions, remaining so while the standby signal is active.
5) High precision is not required in this kind of application, so assume that the clock is a 60 Hz square wave derived from the power line itself (otherwise, a regular crystal oscillator should be employed).

2.1 Simulation:

The state diagram for traffic light controller is shown in the figure. Also, in this, the time values change with the state and with the operating mode (regular or test). Note that all transitions are timed only. The inputs are ***clk, stby, and, test,*** while the output are ***r1, y1, g1, r2, y2 and g2*** (red, yellow, and green lights in directions 1 and 2).

A VHDL code for this FSM obeying the modified template introduced in this section. The time values were specified using GENERIC declaration.

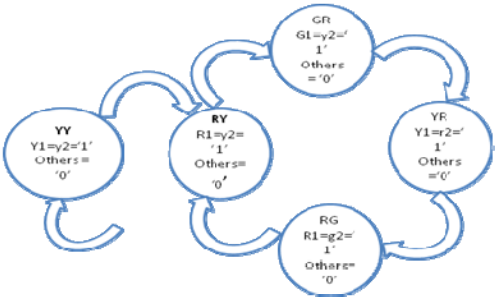

Figure 4 : State- Transition Diagram for Traffic Light Controller

STATE	OPERATING MODES		
	Regular	Test	Standby
	Timer :	Timer :	Timer :
RG	timeRG(30s)	timeTEST (1s)	----
RY	timeRY(5s)	timeTEST (1s)	----
GR	timeGR(45s)	timeTEST (1s)	-----
YR	timeYR(5s)	timeTEST (1s)	----

Table 1: Table depicting Operating Modes of TLC

3. SIMULATION RESULTS

After implementation of a VHDL code for two way traffic Light controller, we've got the following simulation results.

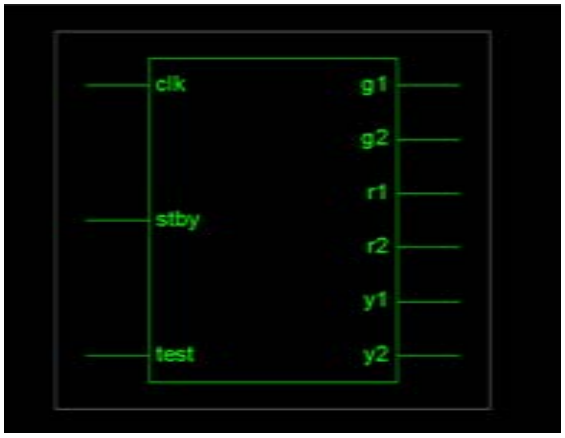

Figure 5: I/P- O/P Block Diagram

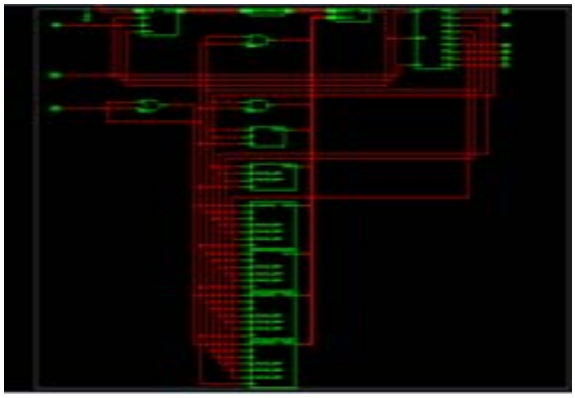

Figure 6: RTL Schematic View

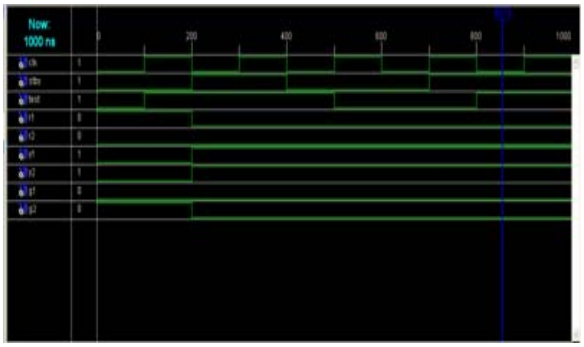

Figure 7: Output Test bench Waveform

4. ACKNOWLEDGEMENT

The authors would like to thanks to Head of Departmant, Mr. Koushik Chakraborty, Jayoti Vidyapeeth Womens University, Jaipur and Dr. S. Lal, Dean- Engineering, Jayoti Vidyapeeth Womens University, Jaipur for his continuous support and encouragement.

5. REFERENCES

[1]. Circuit Design and Simulation with VHDL, by Volnei A. Pedroni.
[2]. Specification and design of Embedded systems, by Daniel D. Gajski, Frank Vahid.
[3]. WM El-Medany, MR Hussain, "FPGA Based Advanced Real Traffic Light Controller". 4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 6-8 Sept. 2007 Page(s):100 – 105.

6. BIOGRAPHIES:

	Nidhi Gopal did her B.tech in ECE from Jayoti Vidyapeeth Womens University, Jaipur, Rajasthan. She is pursuing M.tech in VLSI from JVWU. Her area of interest are Digital Image processing, IC designing etc.
	Madhuri Panwar did her B.Tech in ECE from Jayoti Vidyapeeth Women's University, Jaipur, Rajasthan. She is pursuing M.Tech in VLSI from JVWU. Her area of interest are Electronics devices and circuits, Microelectronics etc.
	Pinky Gupta did her M.tech in VLSI from Banasthali Vidyapeeth, Jaipur, Rajasthan. She is currently working as assistant professor in JVWU, Jaipur, Rajasthan. Her areas of interests are CAD of IC's etc.