Design & Simulation of Dual Elevator Controller using FPGA

[1]Miss Harshita Soni, [2]Mr. Ashish Purani, [3]Mr. Vipul Dabhi

[1]Electronics & Communication Engineering, M.B.I.C.T College, New V.V. Nagar, Gujarat, India, harshitasoni18@gmail.com
[2]Technical Associate, e-info chips Training & research Academy, Ahmedabad, Gujarat, India,ashish.purani@eitra.org
[3]Assistant Professor, M.B.I.C.T College New V.V.Nagar, Gujarat, India, vmdabhi@mbict.ac.in

Abstract

This paper presents for Dual Elevator Controller. Dual Elevator controller "moves people or goods up & down in the buildings or mines, which controls two elevators". Elevator Controller controls the entire operation of the Dual Elevator system. The elevator controller also reads the status, if any, from any of the request positions through the flip-flops. If the door of any elevator is open, the timer signals from the elevator keep controller informed of being busy. The control state machine receives all these signals. It is programmed by the algorithm by which it should control the system. In this work, the real time Elevator units & Controller units are modeled with verilog HDL code using Finite State Machine (FSM) model to achieve the logic in optimized way.

Keywords*— Verilog HDL, Xilinx 14.2 tool*

I. INTRODUCTION

An elevator is a device designed as a convenience appliance that has evolved to become an unavoidable feature of modern day urban life. This elevator can be used for vertical movements of goods and people. While a standalone elevator is a simple electro-mechanical device, an elevator system may consist of multiple standalone elevator units whose operations are controlled and coordinated by a master controller. Such controllers are designed to operate with maximum efficiency [1] in terms of service as well as resource utilization.

II BASIC PRINCIPLE

Dual elevator system is an elementary system consisting two elevator cars serving two floors. Each of the elevator car has a pair of control buttons (up/down) for moving the elevator up and down. Each of the floors having call buttons to call for the service of the elevator system. The following principles have been applied during the design of the dual elevator controller:

1. The floors are defined as ground floor (gndflr) and first floor (frstflr).
2. The elevator cars are designated as elevator-A (elevtrA) and elevator-B (elevtrB).
3. Elevator-A gives the priority to ground floor and elevator-B gives the priority to first floor.
4. A floor is serviced using the nearest available elevator.
5. Upon arrival at a floor, the doors open immediately.
6. Doors remain open approximately 20 seconds, before closure.
7. If an obstruction is detected when door is about to close, it remains open.
8. Each elevator car is treated as a sub-system controlled by the controller.
9. Elevator up/down buttons are connected to individual elevator units.
10. Elevator units depend on floor sensor inputs to determine the current status inputs.
11. Each door is treated as a subsystem controlled by a elevator car.
12. Timing and obstruction detection systems act as slave systems to the door units.
13. Floor call buttons are connected to the elevator controller.

This is all about the dual elevator controller and how it works. Now the next part is the introduction to verilog on which whole this system works.

III.INTRODUCTION TO VERILOG

1. Gate-Level Modeling:

Gate level modeling[4] is a design at a low level abstraction. At gate level circuit is described in terms of gates (e.g. and, nand).Hardware design at this level is intuitive for a user with a basic knowledge of digital logic design because it is possible to see one-to-one correspondence between the logic circuit diagram and the verilog description. Actually very basic level of design a hardware is switch (transistor) level modeling[3] . But it would be very complex for designing point of view so the gate level modeling is to be used.

2. Dataflow modeling[4]:

For small circuits, the gate level design approach works very well because the number of gates are limited. But in complex design it is very complicated to use gate modeling because the number of gates are in large amount. So to overcome that problem data flow modeling is used. Currently, automated tools are used to create a gate-level circuit from a dataflow design description. This process is called logic synthesis[4].

3. Behavioral modeling[4]:

With the increasing complexity of the digital design, it has become vitally important to make wise design decisions early in a project, Designers need to be able to evaluate the trade off of various architectures and algorithms before they decide on the optimum architecture and algorithm to implement in hardware. Thus, architectural evaluation takes place at an algorithmic level where the designers do not necessarily think in terms of logic gates or data flow[4] but in terms of the algorithm they wish to implement in hardware.

Verilog provides designers the ability to describe functionality in an algorithmic manner. In other words designer describes the behavior of the circuit. Thus, behavioral modeling represents the circuit at a very high level abstraction. Design at this level resembles C programming more than it resembles digital circuit design. Behavioral verilog constructs are similar to C language constructs in many ways. Verilog is rich in behavioral constructs that provide the designer with a great amount of flexibility.

IV.DIFFERENT UNITS OF DUAL ELEVATOR

The three different units are Controller unit, Elevator unit and Door unit. So for developing purpose, FSM(finite state machine)is required. That is why, algorithms for the individual units have been done. The state diagram, simulation result and RTL[Register Transfer Level] hardware of controller units & elevator units are shown in following figures.

[1] CONTROLLER UNIT:

The function of controller unit is to control the elevator units & door unit. The inputs & outputs of controller units are up button(upbtn),downbutton(dwnbtn),clock(clk),reset(rst)&firstfloor(frstflr),groundfloor(gndflr)respectively.

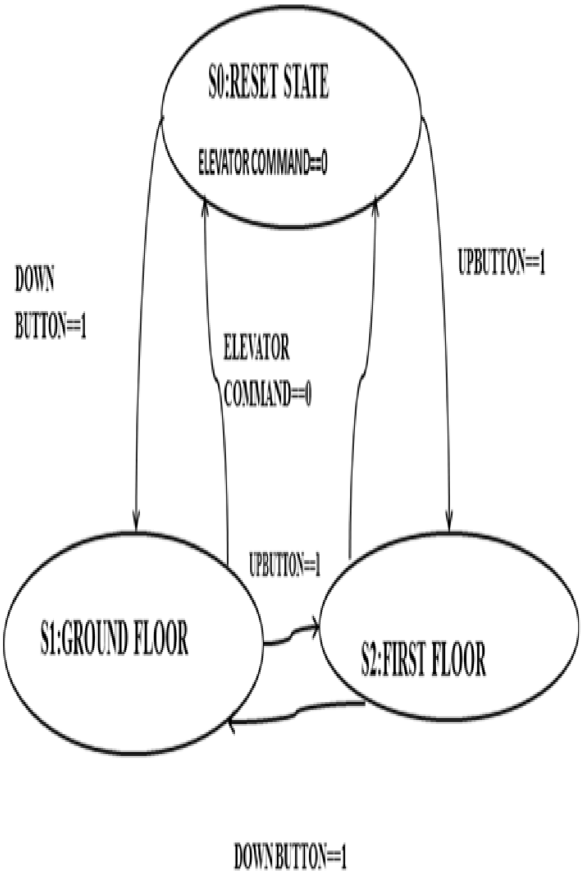

Fig.1 State diagram of controller unit

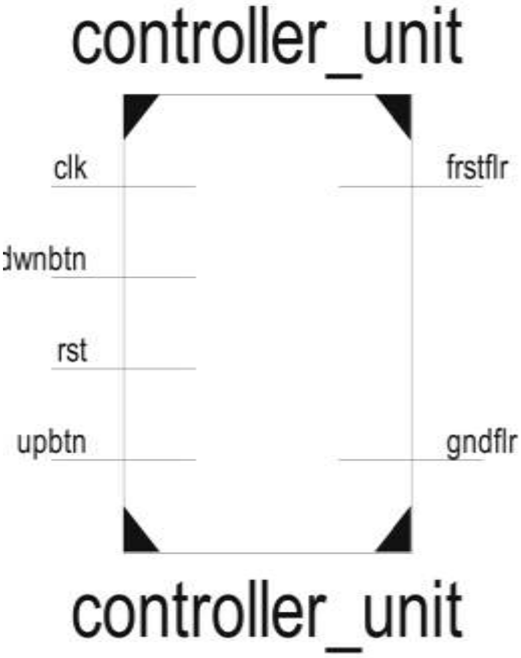

Fig.2 RTL schematic of controller unit.

.

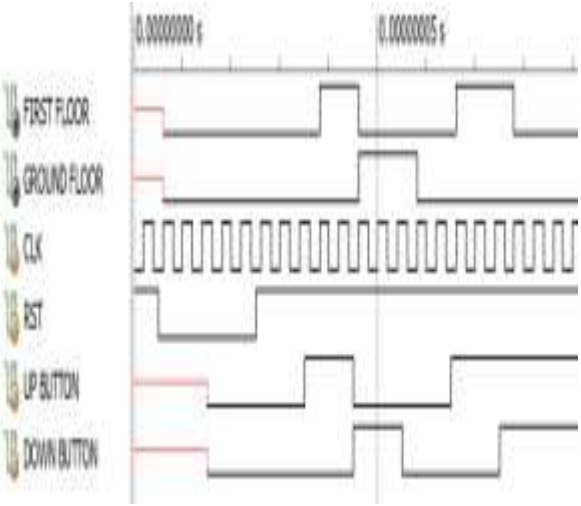

Fig.3 Simulation result of controller unit.

[2]ELEVATOR UNIT:
The following synthesis & simulation results for the elevator unit in which inputs are btn_frstflr_call(button on first floor for elevator call), btn_gndflr_call(button on ground floor for elevator call),clk(clock),dwnbtn(downbutton),upbtn(upbutton),rst(reset) and outputs are door_frstflr(door of first floor),door_gndflr(door of ground floor),frstflr(first floor),gndflr(ground floor).

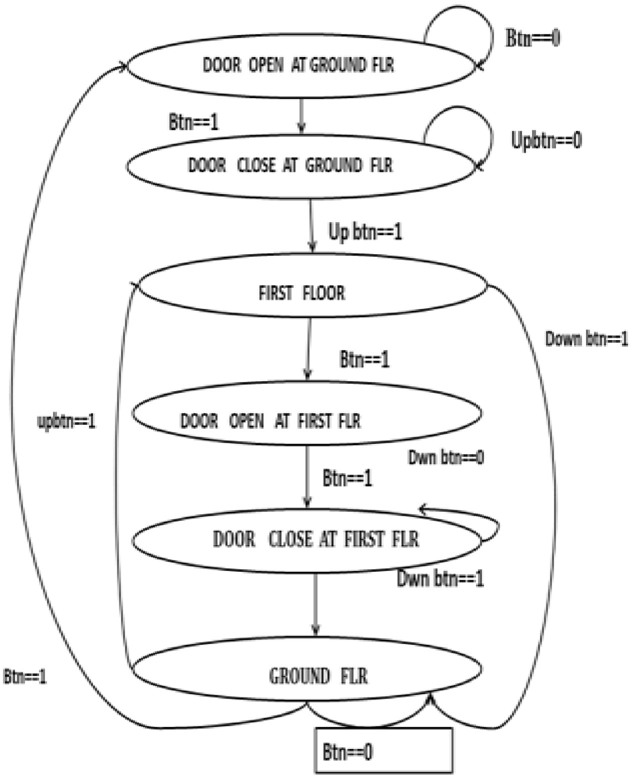

Fig.4 state diagram of elevator units

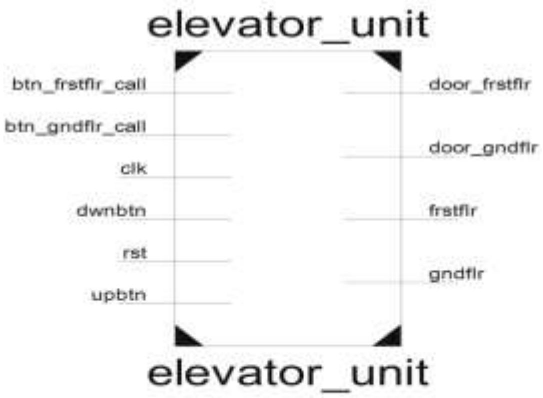

Fig.5 RTL schematic of elevator unit

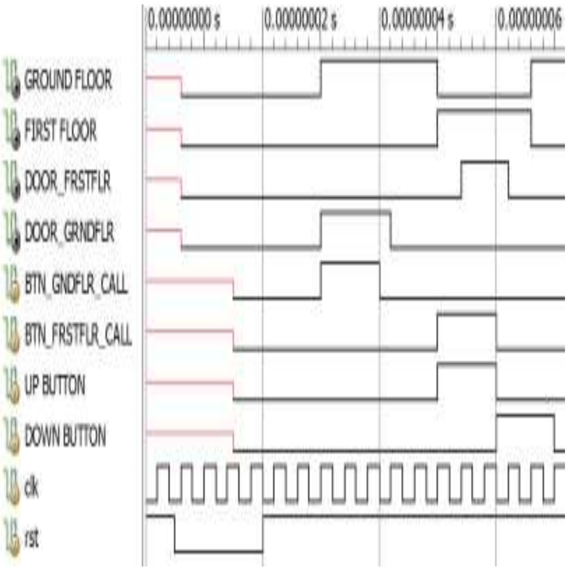

Fig. 6 Simulation Result of elevator unit

The next thing to be done was to make a timer unit which would create a time delay of 20 seconds. System can be implemented using Spartan kit. According to datasheet of spartan 6 kit, global frequency of a kit would be 100MHz. so the next task was to make timer of 20seconds. In the end of calculations, 41 bits are needed for the said task.

V. CONCLUSION

The controller unit & elevator unit are done as shown in simulation results which are the individual results for the Dual Elevator Controller.

In the Dual Elevator Controller there are main units i.e. Door Units, Controller Units, Elevator Units. All the units have been made individually and tested using directed test benches.

REFERENCES

1. Hun-Mo-Kim, “DUAL ELEVATOR ENABLING SYSTEM”, IEEE international journal, August 2008.
2. Digital Logic and Computer Design By M.Morris Mano.
3. Fundamentals of Digital Circuits By A.Anand Kumar, Director of Sasi Institute of Technology and Engineering Tadepalligudem, West Godavari District, Andhra Pradesh.
4. Verilog HDL: A Guide to Digital Design & Synthesis, By Samir Palnitkar.