# FPGA Implementation of Elevator Controller

Jillsmon Varghese
Student ,ECE Department
Saintgits College
Kottayam

Linju Achu Philip
Student ,ECE department
Saintgits College
Kottayam

Shyni Susan Mammen
Student ,ECE Department
Saintgits College
Kottayam

Yasha MiriamThomas
Student, ECE Department
Saintgits College
Kottayam

Riboy Cheriyan
Assoc. Professor
Saintgits College
Kottayam

## ABSTRACT
In this modern era, elevators have become an integral part of any commercial or public complex. It facilitates the faster movement of people and luggage between floors. The elevator control system is one of the most important aspects in electronics control module that are used in automotive application. Usually elevators are designed for a specific building taking into account the main factors such as the height of the building, the number of people traveling to each floor and the expected periods of high usage. The elevator system is designed with different control strategies. This implementation is based on FPGA, which can be used for a building with any number of floors, with the specified inputs and outputs. This controller can be implemented for an elevator with the required number of floors by simply changing a control variable in the HDL code. This approach is based on an algorithm which reduces the amount of computation required, by focusing only on the relevant rules that improves the performance of the group of elevator system. The elevator controller was developed using Verilog HDL and was successfully implemented on a Xilinx Artix7 FPGA.

## General Terms
FPGA, Mealy, Verilog

## Keywords
Reconfigurable, Elevator

## 1. INTRODUCTION
An elevator is vertical transport equipment used to take luggage or passengers between floors of a building. Elevators plays vital role in both commercial as well as residential locations. Traditional elevator control systems are mainly based on relay logic, PLC and microcontroller but the major disadvantage of these systems is that they have reduced number of inputs and outputs.An elevator can be considered as a complex reactive system which requires parallel event processing with a number of inputs and outputs. Therefore FPGAs make a better solution for implementing an elevator controller with added pros of re-configurability, less power consumption, low response times and flexibility in expansion of designs [2]. The elevator control system is basically a finite state machine (FSM).

FSM is a digital sequential circuit that consists of different defined states that are controlled by inputs. The given elevator control system is based on Mealy machine in which the output values are determined both by its current state and the current inputs. We developed an elevator algorithm for an elevator with any number of floors. Verilog HDL was used due to its syntactical familiarity. Verilog is a hardware description language (HDL) which is commonly used to describe digital circuits in a textual manner in designing [1].

## 2. ELEVATOR CONTROLLER AND ITSWORKING
The complete operation of the elevator system is controlled by the elevator controller. The proximity sensors are located to sense the positions of the cars providing the current state storing it in register [3]. The elevator controller also reads the requests from any of the request positions through the flip-flops. The block diagram for an elevator controller is shown in Fig 1. When the door of any elevator is open, the timer signals from the elevator inform the controller of being busy. The control state machine (CSM) of the controller receives all these signals. It is programmed according to the algorithm which controls the entire operation of the system. The CSM then generates control signals for movement of the elevators and for its next position [9]. The elevator system faces some conflict in the operation that which car should process the request when both are at the same positions. Therefore some assumptions are implemented in the elevator algorithm. The assumptions considered are:

A. Elevator Priority: Elevators are prioritized for the requests. The elevator1 has a priority for the request from first floor the elevator2 for the second floor request.

B. Default State: The default states are elevator1 on first floor with closed door and elevator2 at second floor with closed door. These default positions provide faster response to the request coming at any of the two floors.

C. Closing the Elevator Door: Door of the elevator closes after some time as defined by the timer present in the module. By default, the timer should be 0 and after closing it must be 1.

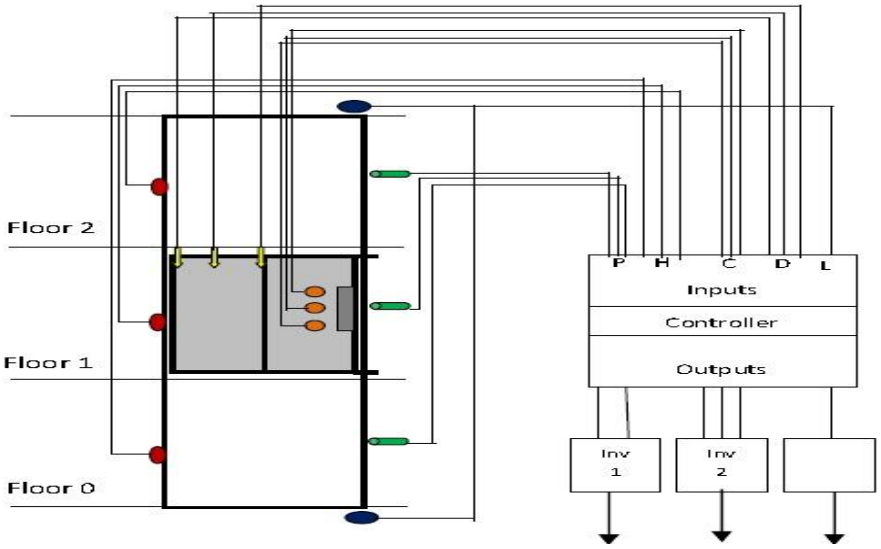


**Fig 1: Elevator Controller**

## 3. FPGA

FPGA or Field Programmable Gate Array consists of the following components: (i) Programmable Logic blocks (ii) Interconnection Resources (iii) Input output blocks FPGA consists of programmable logic components called "logic blocks" and a hierarchy of reconfigurable interconnects that interconnects the logic blocks. Logic blocks can be configured to perform simple logic gates like AND and XOR or even complex combinational functions. In most FPGAs, the logic blocks also include memory elements, which can be either simple flip-flops or complete blocks of memory. Contemporary FPGAs have large number of logic gates and RAM blocks to implement complex digital functions. It becomes a challenge to verify correct timing of valid data within setup time and hold time as FPGA designs employ very fast I/Os and bidirectional data buses. Floor planning enables resources allocation within FPGA to meet these time constraints.

FPGAs can be used to implement logical functions that an ASIC could perform. The advantages of FPGA over ASIC include the ability to update its functionality after shipping, partial reconfiguration of a portion of the design and the low non-recurring engineering costs [8].Some FPGAs have analog functions in addition to the digital features. The most common analog feature is programmable drive strength and slew rate on each of the output pin, allowing to set slow rates on lightly loaded pins that would otherwise ring or couple unacceptably, to set stronger, faster rates on heavily loaded pins on high-speed channels that would otherwise run too slowly. Another common analog feature is differential comparators. FPGAs can be reprogrammed for desired functionality requirements after manufacturing. This feature distinguishes FPGAs from Application Specific Integrated Circuits (ASICs), which are custom designed for specific design tasks [7].

## 4. 4. STATE DIAGRAM

The state diagram of the elevator controller for a building is shown in Fig 2 [4]. In this, Mealy model is implemented. A control signal is used to select the number of floors for which the elevator should work [6]. An example of a four floored building is also shown.

F1_OPEN, F2_OPEN, F3_OPEN, F4_OPEN: These state variables are used to represent that the elevator door is open at the corresponding floor. For example, F1_OPEN indicates that the elevator is open at the first floor. F1_OPEN is the default state.

F1_MOVING, F2_MOVING, F3_MOVING, F4_MOVING: These variables are used to indicate the floor to which the elevator is moving. F1_MOVING indicates that the elevator is moving to the first floor as a result of the request it has obtained.

F_1, F_2, F_3, F_4 : These are the buttons inside the car which are pressed by the user to indicate the floor to which the car should move. When F_2 is pressed from a floor, the elevator enters the F2_MOVING state [5].

U_1, U_2, U_3: These buttons are outside the car and the user presses these buttons for moving in the upward direction from the current floor. For example, U_2 shows that the person is in the second floor and needs to move to an upper floor (third or fourth floor).

D_2, D_3, D_4 : These buttons are used to move in the downward direction from the current floor and are seen outside the car. For example, D_3 represents that a person in the third floor wants to move to a lower floor (first or second floor).

AF_1, AF_2, AF_3, AF_4 : These are the position sensors which, when high, indicates that the corresponding floor has been reached. AF_2 shows that the car has reached the second floor. When AF_2 makes a transition from 0 to 1, the F2_MOVING state will be changed to F2_OPEN state.The programming is done using Mealy model. S0 and S1 are the control lines that are used to select the number of floors. This concept is advantageous as the same controller can be used for a building if more floors are built later; only the control signal is changed [7]. The program is also developed in such a way that it can handle multiple requests at a time and the directions are also taken into consideration. If the upward requests are in process, the requests to go

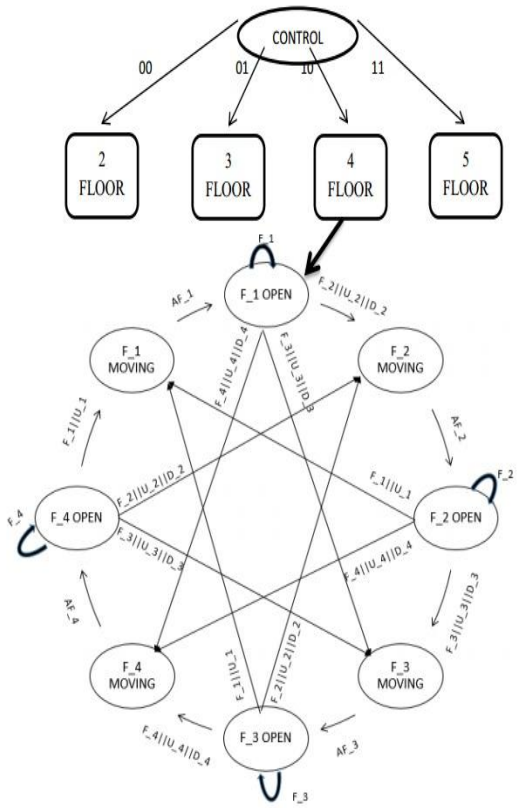


**Fig 2: State Diagram**

downward will be processed only after all the upward requests are executed. There are buttons inside the car (F_1,F_2,F_3 etc.) which allows the user to go to a particular floor. The buttons outside the car (U_1, U_2, D_2, U_3, D_3 etc.) are pressed when a user needs to enter the car and move to an upper or a lower floor. The car moves to a floor (MOVING state) when buttons inside the car or the outside buttons are being pressed. The position sensors (AF_1,AF_2,AF_3 etc.) at each floor are used to know the elevators position. Once the requested floors position sensor is high, the door of the elevator is opened, that is, the state changes from MOVING to OPEN. The OPEN state of first floor (F1_OPEN) is set as the default state. Upon reset, all the requests and the control signal will be cleared and the elevator will be in the default state.

## 5. RESULTS AND REPORTS

The simulation of the verilog code was done in Xilinx ISE (Integrated Software Environment) version 14.2, which is a software tool produced by Xilinx. The simulation process observes the transformations and translations of signals as they propagate through the FPGA from the input pins and provides responses that eventually reach an output pin. It performs the following type of simulations:

- Logical verification, to ensure the module produces only expected results
- Behavioral verification, to verify logical and timing issues
- Post-place & route simulation, to verify the behaviour after placement of the module within the reconfigurable logic of the FPGA [4]. The simulation result for the elevator controller is shown in Fig 3

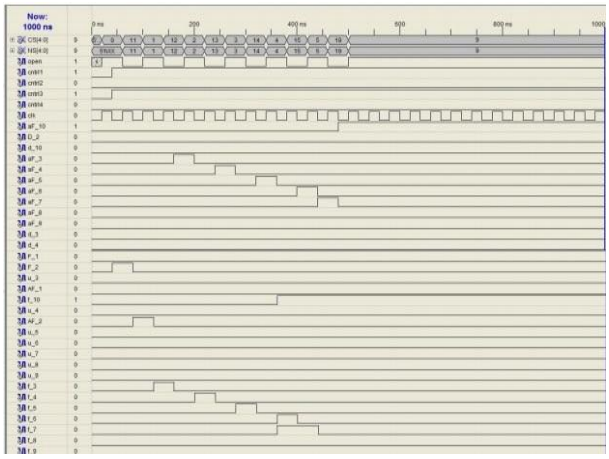

**Fig 3: Simulation Result**

Here, the control signal is 1010 and the system works as a 9 floor elevator controller. Thereafter different conditions for the elevator controller were checked. When F_7 button is pressed by the user, the car moves to the seventh floor and the current state will be F7_MOVING. The state of the door at any instant is indicated by the state variable ‘open’; the door remains closed during the moving state. When AF_ 7 goes high, it indicates that the elevator has now reached the seventh floor and the door will be open.

## 6. CONCLUSION

We focused on implementation and verification of a controller for an elevator with basic functions in a reconfigurable structure and proposing a methodology to improve the reliability of the controller. Since the simulations could be done not only for the prototype elevator with three floors, but for an elevator with “N” number of floors by simply changing a variable in the code, it was easier to develop the controller in a general concept [11]. The verification process which was carried out on a Spartan 3AN FPGA to a prototype elevator with three floors revealed modifications. In addition, further research can be done on developing the elevator algorithm for more functions with increased efficiency and desirable aspects. Thus this research creates many fresh ideas in a broad area to develop and modify a reliable FPGA based reconfigurable elevator controller.

## 7. ACKNOWLEDGMENT

We are greatly indebted to our project guide Er. Riboy Cherian, Associate Professor, Dept. of Electronics and Communication, Saintgits College of Engineering for his invaluable suggestions and ideas which enabled us in the finalization of this project. We also thank the management and all staff members of the department for providing us with the best facilities and guidance.

## 8. REFERENCES

[1] Michael D.Ciletti, “Advanced Digital Design with theVERILOG HDL”, Prentice-hall of India Ltd, India.

[2] SithuminiEkanayake,RuwanEkanayake,SomasundaramSanjayan, Sunil G.Abeyratne, S.D. Dewasurendra“FPGA Based Elevator Controller with Improved Reliability” 2013 UKSim 15th International Conference on Computer Modelling and Simulation IEEE.

[3] PatchalaKiranBabu, H. RaghunathRao “A VLSI Implementation of Three-Lift Controller Based on

Verilog" International Journal of EngineeringTrends.and Technology (IJETT) – Volume 6 Number 1- Dec 2013

[4] H.P.A.P. Jayawardana, Amarasekara, P.T.S. Peelikumbura "Design and Implementation of a Statechart Based Reconfigurable Elevator Controller" 2011 6th International Conference on Industrial and Information Systems, ICIIS 2011, Aug. 16-19, 2011

[5] D. Harel, "Statecharts : A Visual Formalism for Complex Systems", Science of Computer Programming, Vol.8, (3), pp.231-274, 1987

[6] Y – S. Huang, S – L. Chung, and M – D. Jeng, Modelling and Control of elevators by Statecharts", Asian Journal of Control, Vol. 6, No. 2, pp. 242-252, June 2004

[7] Brian Pratt, Michael Caffrey,Paul Graham, Keith Morgan, Michael Wirthlin, " Improving FPGA Design Robustness with partial TMR", Reliability Physics Symposium Proceedings, 2006 IEEE

[8] Karl Schabas,Stephen D. Brown, "Using Logic Duplication to Improve Performance in FPGAs", University of Toronto, Toronto, Canada

[9] ZHAO Yuhang, MA Muyan, "Implementation of six – layer automatic elevator controller based on FPGA", Wireless Communications Networking and Mobile Computing(WiCOM), 2010 IEEE

[10] XILINX Triple Module Redundancy Design Techniques for Virtex FPGAs", Application Note

[11] Monzurul Islam Dewan, Md. Arafat Mahmud, Md. Tashfiq Bin Kashem, MushfikaBaishakhiUpama, "A Simulation Study of Elevator Control of a Building using Digital Logic Circuit" IOSR Journal of Engineering Vol. 3, Issue 12 (December. 2013)