



**DIGITAL SYSTEM DESIGN AND LABORATORY FINAL PROJECT REPORT
DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITAS INDONESIA**

**DESIGN AND SIMULATION OF A MODULAR FPGA-BASED ELEVATOR
CONTROLLER SYSTEM USING VHDL**

GROUP 6-KKI

| | |
|----------------------------------|-------------------|
| Evandra Rasya Fadhillah | 2406450352 |
| Muhammad Arfassya Setyadi | 2406368914 |
| Derryl Liandryo Putra | 2406368920 |
| Muhammad Risjad Ali | 2406368984 |

PREFACE

We begin by expressing our gratitude to God Almighty for His guidance and blessings, which have enabled us to complete this final project. This work represents the development of a VHDL-based digital elevator controller system designed to simulate realistic elevator operations on an FPGA platform. Its completion is the result of collaborative effort, discipline, and dedication from the 06 Group, consisting of Evandra Rasya Fadhillah, Muhammad Arfassya Setyadi, Derryl Liandryo Putra, and Muhammad Risyard Ali.

The growing adoption of digital systems and automation in modern infrastructure highlights the need for reliable and efficient control mechanisms. Elevators, in particular, require precise logic to manage floor requests, control door operations, ensure safety, and optimize movement. FPGA-based controllers offer significant advantages in flexibility and accuracy, allowing complex behaviors such as emergency stopping, overload detection, and optimized floor scheduling to be modeled effectively through VHDL.

This project addresses these needs by implementing a modular elevator controller architecture that integrates key functionalities including floor request processing, door double-click handling, emergency logic, overweight detection with LED and alarm indicators, and movement optimization. With the support of a comprehensive testbench, the system successfully demonstrates realistic operational behavior while showcasing the practical application of digital design principles.

We extend our sincere appreciation to our lecturers, mentors, and peers who provided continuous support, constructive feedback, and valuable insights throughout the development of this project. All comments and suggestions received will serve as a foundation for future improvements and development within the field of digital control and FPGA-based system design.

Depok, December 07, 2025

Group 6-KKI

TABLE OF CONTENTS

CHAPTER 1: INRODUCTION

- 1.1 Background
- 1.2 Project Description
- 1.3 Objectives
- 1.4 Roles and Responsibilities

CHAPTER 2: IMPLEMENTATION

- 2.1 Equipment
- 2.2 Implementation

CHAPTER 3: TESTING AND ANALYSIS

- 3.1 Testing
- 3.2 Result
- 3.3 Analysis

CHAPTER 4: CONCLUSION

REFERENCES

APPENDICES

- Appendix A: Project Schematic
- Appendix B: Documentation

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Elevator control systems can be utilized for residential and commercial purposes including multi-story buildings and warehouses; industrial use also includes vertical lifting devices. Digital elevators are equipped with an advanced controller which uses software to interpret input from buttons, indicators and/or sensors in order to produce output that is reflective of user requests for carriage movements. Reliability and efficiency for carriage transportation is state-of-the-art today; safety is also essential for maximizing customer satisfaction and minimizing wait times between the time when they push a button to request a carriage and when they reach their destination. Historically, relay-based and mechanical-based control methods were used for managing elevator movements. However, advancements in digital design have made it possible to create modularized, flexible, and extensible design architectures for elevators using hardware description languages such as VHDL.

A full elevator controller developed as a digital systems educational experience is a valuable way to apply theoretical concepts to a practical, multi-module engineering problem. The system outlined in this report models a fully operational elevator system implemented in VHDL on an FPGA. The system incorporates many of the key characteristics of an elevator system, including a floor request system for floors 1-8, a call system that simulates actual passenger behaviour, an emergency stop switch, a weight detection system with visual (LED) and audible (alarm) outputs, and a door control system that includes double-click functionality. By developing the system based on digital design principles, the project provides a platform for studying concurrent operation, state machines, signal synchronization, modular design, and testbench verification of all components.

In addition to the academic relevance and importance of this and related projects, many aspects are also of practical significance. Therefore, from a practical standpoint, most modern elevators are designed using modular controllers that can react appropriately to unexpected and non-predictable events. Along with these types of events, the simulation of passenger arrival times, weight changes and dynamic floor requests provide insight into the real-life operational challenges that an actual elevator must overcome. Therefore, the implementation of this project

also illustrates how digital logic discipline, accurate timing, and solid-state design should be considered not only as academic concepts, but as critical engineering assets for a successful elevator system.

1.2 PROJECT DESCRIPTION

Elevator_controller is the central element that combines several sub-blocks of elevator_controller, such as the floor-request handling, the door control logic, the emergency handler, the weight monitor, the motion controller, and the optimization unit. In addition, a Passenger Generator is available for testing purposes. All these sub-blocks work together to simulate an elevator that responds to user requests, while ensuring safety and optimized operation.

The Floor System enables the elevator to process and respond to floor requests from push buttons for floors 1 through 8. Floor requests remain in effect until passengers either exit from the elevator or cancel them. The Call System, which is implemented by the passenger_generator module, models dynamic passenger behaviour by simulating the creation of floor requests, entering the elevator, and exiting the elevator upon arrival at the requested location. In addition, the Door Control System processes door open and door close commands, including an added feature for double-click detection, to closely resemble the real-life behaviour of elevators when passengers quickly hit the open button twice in order to extend the time for the doors to be open.

Safety-oriented subsystems are integrated into the elevator controller. The Emergency Feature allows the elevator to immediately stop, cease accepting new requests, and force the doors into a safe condition. The Weight System continuously monitors the combined weight of all the passengers currently in the elevator. If the Weight System exceeds its safe weight limit, it activates an overload warning using an LED and alarm. The Motion Controller is responsible for providing control over the physical movement of the elevator, as well as controlling how the elevator travels from floor to floor, and stopping at each floor when required. The Optimization Algorithm determines the next most efficient floor to serve based on the pending requests made to the system, which reduces unnecessary movement and provides customers with the shortest possible wait times.

1.3 OBJECTIVES

The objective of this project is to design and validate a complete digital elevator controller system using VHDL. The system is intended to demonstrate modular hardware design principles, accurate signal processing, and real-time decision-making within the constraints of synchronous digital logic. Specific objectives of the project include:

1. To implement an FPGA-based elevator control system capable of handling floor requests, door control, emergency responses, and overload detection.
2. To develop modular VHDL subsystems, each responsible for a specific aspect of system functionality, ensuring clarity, scalability, and ease of debugging.
3. To integrate safety-critical features such as the emergency stop mechanism and weight overload detection system with LED and alarm indicators.
4. To incorporate realistic user interaction, including the call system and door double-click functionality, to replicate real elevator behaviour.
5. To construct a comprehensive testbench simulating multiple passengers, floor requests, emergency events, and timing sequences over a long virtual simulation period.
6. To analyze system performance, focusing on metrics such as total energy consumption and average passenger waiting time, ensuring operational efficiency.
7. To demonstrate practical application of VHDL design, including state machine implementation, signal synchronization, and modular architecture construction.

1.4 ROLES AND RESPONSIBILITIES

The roles and responsibilities assigned to the group members are as follows:

| Roles | Responsibilities | Person |
|----------------------|--|-------------------------|
| Making the VHDL code | Base Code, Floor System 1-8, Emergency Feature, Double click Feature | Evandra Rasya Fadhillah |

| | | |
|-------------------------|---|---------------------------|
| Making the VHDL code | Weight System, Call System, LED and Alarm for Weight system | Muhammad Arfassya Setyadi |
| Making the Report | Making the whole report | Derryl Liandryo Putra |
| Making the Presentation | Making the whole presentation | Muhammad Risyad Ali |

Table 1. Roles and Responsibilities

CHAPTER 2

IMPLEMENTATION

2.1 EQUIPMENT

The tools that are going to be used in this project are as follows:

- Vivado
- GitHub

2.2 IMPLEMENTATION

The implementation of the digital elevator controller system is centered on a modular VHDL architecture, where each subsystem is constructed as an independent entity that cooperates with others through well-defined signal interfaces. The overall design is executed within the Elevator_Controller top-level module, which serves as the integration point for modules handling floor requests, button interactions, emergency logic, overload detection, passenger simulation, motion control, and optimization scheduling. This modular approach ensures that complex elevator functionality can be broken down into simpler, manageable components, thereby increasing clarity, reliability, and maintainability. All modules operate synchronously using a 5-second system clock, allowing deterministic control of state transitions and predictable behavior throughout the entire elevator operation cycle.

The floor request handling module "floor_button_handler" represents the first major subsystem within this application. This module will provide an interface into the floor button system and will also include a register for storing floor requests from users. The primary purpose of this module will be to allow the continued storage of user-requested floors until they are cancelled. When a user presses a button to request a floor, the floor button handler will latch onto that request until an elevator arrives at the requested floor and the user exits the elevator. This will provide the user experience of pressing a button to request a floor as it appears on a standard elevator call panel. This module will also include a cancel button that is contained in a Cancel_Request vector, which is used to clear a user's selected floor request as soon as the user exits the elevator. By providing a mechanism to both store and cancel user requests, the Floor Request Handling Module provides the best representation of real-world

elevator functionality and the simplest means of accomplishing this functionality in the digital domain.

In addition to the base system, there are also various safety interaction subsystems integrated into the elevator system. These subsystems include, emergency button handler, overweight handler, and door control logic. The emergency button handler sets a continuous emergency state when the button is pressed, which causes the elevator to immediately stop, ignore all additional requests and hold the door open. The overweight handler is specified to calculate and assess total weight per passenger prior to entering the car, and if it passes the maximum allowable limit, the handler activates an over load light to indicate that it cannot move, and sounds a visual or audible alarm. The door control subsystem is designed to receive and process both commands to open and close the doors, and includes a double-click feature which is also common with most elevators, which adds more additional time when holding the doors open.

In order to develop a comprehensive test dummy for realistic testing of an evolving elevator control system, the implementation includes a full system-level "passenger_generator" interface. The interface functions as a model of actual human behaviour in the elevator system; each user has randomised arrival times, user weights, and destinations. Each user transits through the states of waiting, entering, travelling, and exiting the elevator. The passenger_generator also includes all of the logic needed to compute metrics, such as energy consumption and average user wait times, for the elevator control system simulation, and these metrics will serve as valuable feedback regarding the performance of the elevator control system overall. Together, through the integration of these subsystems into the elevator control system, the elevator control system produces an accurate and functional representation of a real-world elevator control system within a fully simulated digital environment.

CHAPTER 3

TESTING AND ANALYSIS

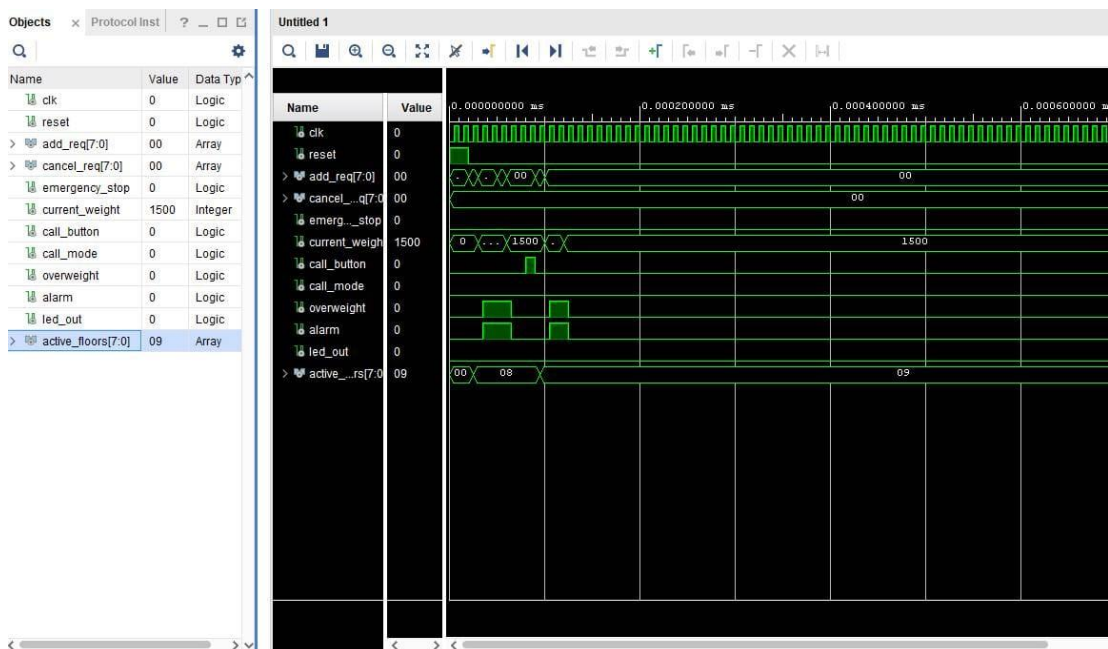
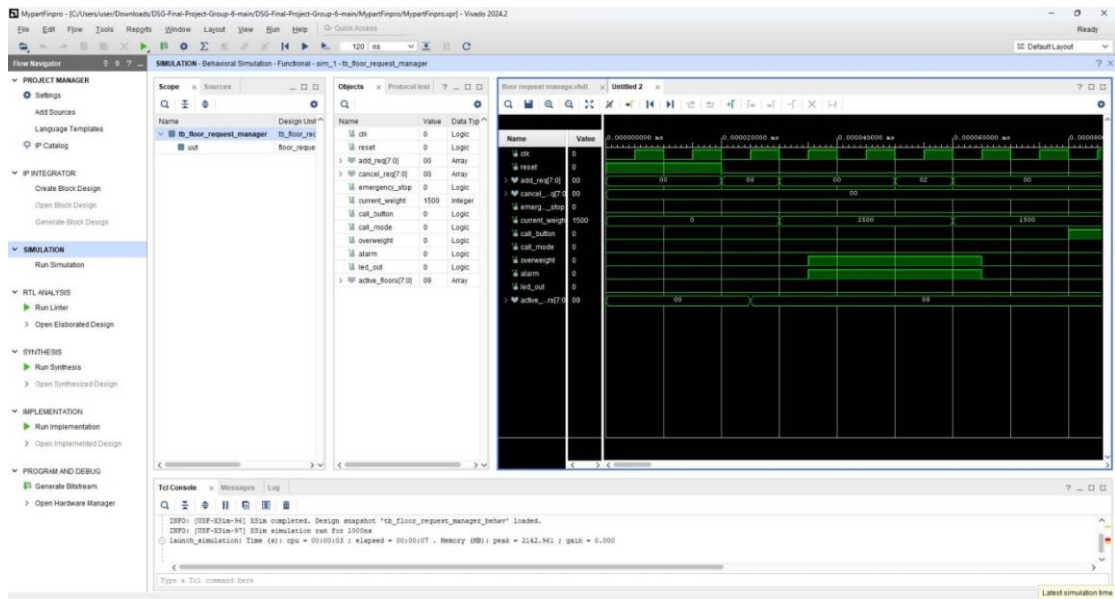
3.1 TESTING

The elevator controller system will be tested in VHDL test bench which is able to replicate real-world elevator controller systems over extended periods of time. The elevator controller's behaviour can be tested using this test bench because it generates an accurate 5-sec clock signal during its operation. Additionally, the testbench will simulate 20 hours of operational time by creating a continuous functioning set of stimuli within a simulated environment. Each time the elevator controller gets a signal from the test electrician's facility to send an elevator car to a particular floor, the passenger_generator module creates an infinite-set of passengers with random amounts of time, destination floors, and passenger weighs, creating a constantly moving load for the elevator controller to handle.

As soon as the elevator controller receives a signal from a test electrician's facility that there is a new passenger waiting to be picked up, the testbench sends all of the necessary signals to enable the floor_button inputs of the elevator and record the new passenger's request and schedule the elevator car to go to that passenger's destination floor and pick them up. In addition to normal operations-testing procedures, the testbench will also create and use several special event scenarios to help determine if the elevator system's safety features properly override the normal operation features of an elevator.

In order to evaluate the timing behaviour, state transition behaviour and inter-module communication of the elevator controller during testing, the testbench provides a set of all input and output signals for visual representation of testing on a waveform viewer by using Vivado.

3.2 RESULT



Based on the results of the simulation, it has been determined that the elevator controller system will work properly and consistently, and respond as expected to all test cases and input for each of the subsystems. Under normal conditions of operation, the elevator will be able to accept floor requests from passengers, deliver passengers to each floor via the route indicated by the optimization handler, and open the doors upon arrival at the requested floor. The cancel_request functionality will remove requests from the elevator when the last passenger leaves or exits the elevator, which prevents it from stopping unnecessarily at any floor after that point in time. In addition, when an emergency event is activated, the system will immediately discontinue all motion of the elevator, let the passengers exit by holding the door open for them, and stop processing all additional requests until the state of emergency has been

cleared from the system. Also, the overweight detection subsystem will provide a very reliable method of detecting whether the total weight of the passengers exceeds the predetermined safety limit by providing no service to the elevator when it is overloaded, and providing an indicator that indicates an overload condition exists. Finally, the motion controller will consist of an incremental counter, and the motion controller will provide an exact real-time output of the elevator's carrier floor, as well as accurate door status based upon the various door command inputs and the emergency and overload condition flags set in the system.

3.3 ANALYSIS

The evaluation of the performance and character of elevator controller system behavior during testing has concluded that the modular nature of the elevator controller's architecture has a major influence on the stability, ability to expand the size of the elevator system, and correctness of the system. The modular design separates each sub-system into a separate VHDL entity, isolating any functional errors which makes finding functional errors quicker and prevents unintendedly modifying the interface between the various components of the elevator controller. The use of the testbench in conjunction with the simulated environment has shown true and consistent communication between the various modules, indicating the optimization handler is properly influencing the motion controller, the floor handler has the ability to consistently hold request and clear functions, and the safety modules will always override the operation commands when necessary. The results of the testing conducted at the stress level of experience demonstrates that the elevator controller is designed and built in such a way that it will always return to a safe operating condition prior to resuming normal operation of the elevator system. The waveform analysis of the simulation data shows that the elevator controller maintains proper synchronization of all synchronous signals throughout the entire testing period. While areas of elevator controller design such as the movement logic and door timing could be improved to more accurately represent the actual dynamics of commercial elevators in a more real-world fashion, there is ample evidence to prove that this implementation is correctly modeling the basic elements of an elevator system.

CHAPTER 4

CONCLUSION

The system's successful creation of a digital elevator controller illustrates VHDL-based Hardware Design Principles used in creating a high-performance, real-world controller of many complex control systems. With a modular structure for separating floor request processing from motion control, Door Management, Emergency Response, Overload Detection and that which simulates passenger activity, the project has provided the community with a high-quality product. By bringing together these different components in an integrated manner, the Top-Level Integration of Components allows the elevator to perform its job of responding to requests, transporting passengers to their desired location and providing for safe operation when operating outside normal conditions. The simulation results using a custom built testbench, demonstrated that the elevator operated deterministically under multiple operational scenarios, including standard passenger loads, emergency stop situations, and overload circumstances. Performance metrics such as energy consumption and average time waiting for the elevator to arrive further substantiate that this system has the capability to provide accurate performance indicators and therefore can be used for education and foundation for more complex designs.

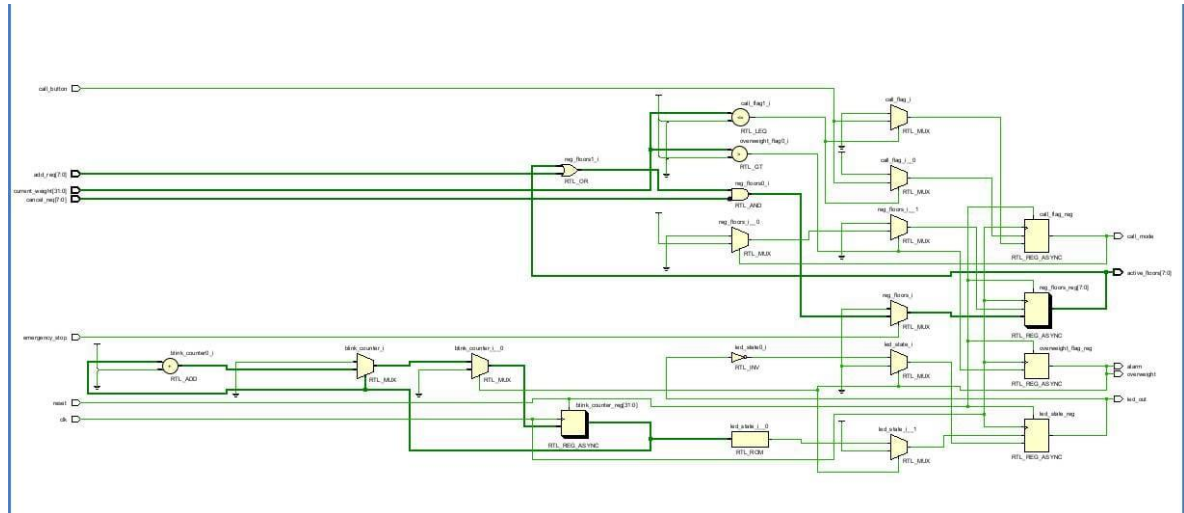
While the project's functionality is already satisfactory, there are also opportunities for continued development within some behavioral elements of the project that may be optimally performed through improved door timer logic, more advanced optimization algorithm selection, advanced scheduling techniques, etc. The extended simulations, when viewed in their entirety, demonstrate a stable logical system that consistently prioritizes safety while allowing multiple subsystems to operate together without error. Additionally, while the project is functional in its meeting of expected deliverables, this project also provided experience in the construction of synchronized digital systems, coordination of concurrent processes, and validation of a complete system through the use of VHDL testbenches. Thus, the knowledge and experience gained through this project are fundamental to continued growth and exploration within digital system design, embedded control, and the use of FPGA development.

REFERENCES

- [1] D. M. Harris and S. L. Harris, *Digital Design and Computer Architecture*. Morgan Kaufmann, 2016.
- [2] M. M. Mano and M. D. Ciletti, *Digital Design: With an Introduction to the Verilog HDL, VHDL, and SystemVerilog*. Pearson, 2017.
- [3] P. J. Ashenden, *The Designer's Guide to VHDL*. Morgan Kaufmann, 2010.
- [4] IEEE Standard 1076-2008, *IEEE Standard VHDL Language Reference Manual*, IEEE, 2008.
- [5] S. Brown and Z. Vranesic, *Fundamentals of Digital Logic With VHDL Design*. McGraw-Hill, 2019.
- [6] Xilinx Inc., *Vivado Design Suite User Guide: Logic Simulation (UG900)*, Xilinx Documentation, 2021.
- [7] M. D. Ciletti, *Advanced Digital Design With the Verilog HDL and VHDL*. Pearson, 2020.

APPENDICES

Appendix A: Project Schematic



Appendix B: Documentation

