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

WASHING MACHINE CONTROLLER IN VHDL

GROUP 3

HAFIZH AKBAR GHIFARIE RAMADHAN	**2406450384**
MUHAMMAD AGIB ANUGRAH PRATAMA	**2406450415**
RYAN GAZENDRA IRAWAN	**2406368952**
SYIFA SARAH NURAINI	**2406368883**

PREFACE

This laboratory was conducted as part of Digital System and VHDL Design coursework, with the objective of applying theoretical concepts to the development of a functional hardware based system. Our project focuses on the design, implementation and verification of a “Washing Machine Controller” using the VHDL hardware description language and Vivado design. In this final project the main principles of digital logic, finite state machine, microprogrammed control and modular hardware design are translated into a practical and integrated

Throughout this project, we learned many things that go beyond basic classroom theory. We had to deal with real issues such as syntax errors, incompatible libraries, missing files, and fixing module connections. These challenges helped us understand how hardware design actually works when building a full system. We also became more confident in handling simulations, debugging errors, and organizing multiple VHDL files even though we are still learning.

This final project report explains how the system was built, how each module works, and what the simulation results showed. It also reflects the knowledge and skills that we gained during the development process. Even though the project was challenging at times, it was a valuable experience that helped me understand VHDL programming and digital system design in a more practical way.

Depok, December 07, 2025

Group 3

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

This project aims to utilize VHDL to implement a digital control system for an Automatic Washing Machine. Designed to simulate various functionalities of modern washing machines, such as mode selection, optional features, cost calculation, management of washing and drying cycles, as well as a door locking mechanism. This design is modular and given its synchronous design, it is suitable to be synthesized onto hardware (such as FPGA or ASIC).

Contrasting typical software-based control systems, hardware description language is capable of true parallel processing and is able to ensure precise timing. This is particularly fitting for our program since multiple operations must be coordinated simultaneously with exact timing requirements.

The design adheres to the principles of synchronous digital design, utilizing a unified clock and reset mechanism for predictable state transitions. The subsequent sections detail the system architecture, component functionality (with a focus on the Control Unit microprogram), and simulation results demonstrating the system's operational correctness and flexibility.

1.2 PROJECT DESCRIPTION

This final project is structured around eight defined VHDL components. Focusing on modularity, it is able to maximize flexibility and design organization. The washing machine system is composed of several interacting synchronous modules:

- **User Input Unit:** Captures the user's selected wash mode, upgrade option, and drying option.
- **Control Unit Microprogram:** The central intelligence, determining cycle durations, managing *start* signals for the cycles, and calculating the final price.

- **Wash Cycle Unit & Dry Cycle Unit:** Timer modules that simulate the duration of the washing and drying processes by counting clock pulses.
- **Door Lock Module:** Controls the door lock status (locked during operation, unlocked upon completion).
- **Billing Unit:** Calculates total cost based on mode and options and is able to update price dynamically as features are selected using an arbitrary currency.
- **Washing Machine (main/top):** The top-level entity that integrates and interconnects all sub-modules to form the complete system.

This microprogrammed approach models the control unit as a simpler internal computer, which fetches step-by-step micro-instructions (the microprogram) from a Control Store memory to orchestrate execution, trading speed for high flexibility and systematic debugging. Our design is able to execute a number of operational modes with the following specifications:

Mode	Code	Base Duration	Base Cost
Standard	000	30 Cycles	50 units
Quick	001	15 Cycles	30 Units
Heavy	010	45 Cycles	60 Units
Delicates	011	25 Cycles	40 Units

Table 1. Operational Modes

Users are then able to further customize their experience with an optional upgrade to the wash time that adds 10 Cycles to the wash duration at the price of 20 Units, or and additional drying feature that adds a fixed 20 Cycles to the wash phase at the price of 30 Units, these options will only activate when selected by the user. The formula for the Total Cost is as follows:

```
Total Cost = Base Cost + (Upgrade ? 20 : 0) + (Drying ? 30 : 0)

Cost Range: 30 to 110 units
- Minimum: Quick mode without options (30 units)
- Maximum: Heavy mode with upgrade and drying (110 units)
```

1.3 OBJECTIVES

The objectives of this project are as follows:

1. Designing a modular architecture, separating input, control, cycle timing, and security functions.
2. Implementing a washing machine control logic using VHDL, through the usage of implicit state management.
3. Integrating all components into a single, functional top-level entity
4. Ensuring accurate calculation of costs and cycle durations based on user selections.
5. Developing a Test Bench to thoroughly verify the system's functionality under various input conditions.

1.4 ROLES AND RESPONSIBILITIES

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

Roles	Responsibilities	Person
Program, ReadMe, PowerPoint	Wash cycle unit, Top-level system integration	Hafizh Akbar
Program, Report, PowerPoint	Dry cycle unit, Door lock module, fix wash_cycle	Muhammad Agib
Program, ReadMe, Report	Billing unit, Testbench	Ryan Gazendra
Program, Report, PowerPoint	Control unit, User input unit, State machine design	Syifa Sarah Nuraini

Table 2. Roles and Responsibilities

CHAPTER 2

IMPLEMENTATION

2.1 EQUIPMENT

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

- **Software:** Vivado 2025.1 was utilized for VDHL entry, Register Transfer Level (RTL) analysis, synthesis, and behavioral simulation.
- **Hardware Target:** The design is synthesized targeting a standard FPGA development board (e.g., Artix-7 series), utilizing on-board clock sources and I/O switches for user interaction.
- **Schematic Viewer:** The Vivado RTL Analysis tool was used to generate and verify the hardware schematic.

2.2 IMPLEMENTATION

The system adopts a Modular Hierarchical Design, separating the control logic from the execution and safety units. The top-level entity, [washing_machine_top], integrates five main sub-modules using structural VHDL description.

Figure 2.1 (Refer to Appendix A) presents the synthesized RTL Schematic of the system. The schematic illustrates the data flow where the [control_unit] (center block) acts as the orchestrator, receiving signals from the [user_input_unit] and driving the wassh_cycle_unit], [dry_cycle_unit], and [billing_unit].

The core modules implemented are:

1. User Input Unit: Ensures signal stability and synchronization.
2. Control Unit: The central logic that calculates parameters and manages start signals.
3. Cycle Timers: Two independent counters for washing and drying.
4. Billing Unit: A dedicated combinational block for real-time cost display.
5. Door Lock Module: A safety interlock system.

2.3 Control Logic Implementation

The heart of the system is the Control Unit (control_unit_microprog.vhd). Instead of a fixed hardware sequence, this unit utilizes Behavioral VHDL to dynamically determine cycle parameters based on user selection.

- **Parameter Calculation:** Inside a synchronous process sensitive to the clock, the unit evaluates the mode_select input. A case statement assigns specific base durations and costs (e.g., Mode "010" sets 45 cycles and 60 cost units) .
- **Dynamic Feature Handling:** The logic sequentially checks for upgrade_flag and drying_flag. If selected, the logic mathematically adjusts the duration (adding 10 cycles for upgrade) and cost registers (adding 20 or 30 units) within the same clock cycle .
- **Handshaking Mechanism:** The unit generates [start_wash_out] and [start_dry_out] signals by monitoring the done status of the external timers. This ensures that the drying cycle (start_dry) only triggers after the washing cycle is complete, enforcing a strict sequential operation.

2.4 Sub-Module Description

2.4.1 User Input Unit

The [user_input_unit.vhd] serves as the interface between the physical switches and the internal logic. It implements a register that captures asynchronous inputs (mode_select, upgrade, drying) on the rising edge of the clock. This prevents metastability issues and ensures that the Control Unit receives stable, synchronized signals.

2.4.2 Wash and Dry Cycle Units

The execution of the cleaning process is handled by wash_cycle_unit and dry_cycle_unit. These are Programmable Down-Counters.

- When the Control Unit asserts the [start] signal, these units load the calculated [duration] value into an internal register.
- The counter decrements on every clock cycle until it reaches zero, at which point it asserts a [done] signal to notify the system that the phase is complete.

2.4.3 Door Lock System

Safety is implemented in [door_lock_module.vhd.] The door is locked (door_locked <= '1') by default upon system reset. The logic strictly enforces that the door acts as a "Fail-Safe"; it will only unlock (logic '0') when both the washing complete (wash_done) and drying complete (dry_done) signals are asserted high, ensuring user safety during active operation.

CHAPTER 3

TESTING AND ANALYSIS

3.1 TESTING

In order to ensure proper execution, we verified our firstly through individual components testing. Ensuring that the functionalities of our program has been properly implemented. Once the isolated modules

1. **Test Case 1**: Standard mode (000) without options - 50ns duration
2. **Test Case 2**: Quick mode (001) with upgrade - 50ns duration
3. **Test Case 3**: Heavy mode (010) with drying - 50ns duration
4. **Test Case 4**: Delicate mode (011) with both upgrade and drying - 50ns duration

The Test Bench is set to execute these four test cases, each lasting 50 ns,beginning with a reset pulse, with each clock period defined as 10 ns, providing 5 clock cycles for input stabilization between test case changes.

3.2 RESULT

The simulation results, visualized in the waveform (see Figure 2.1 below), confirm the system's behavior across the defined test cases.

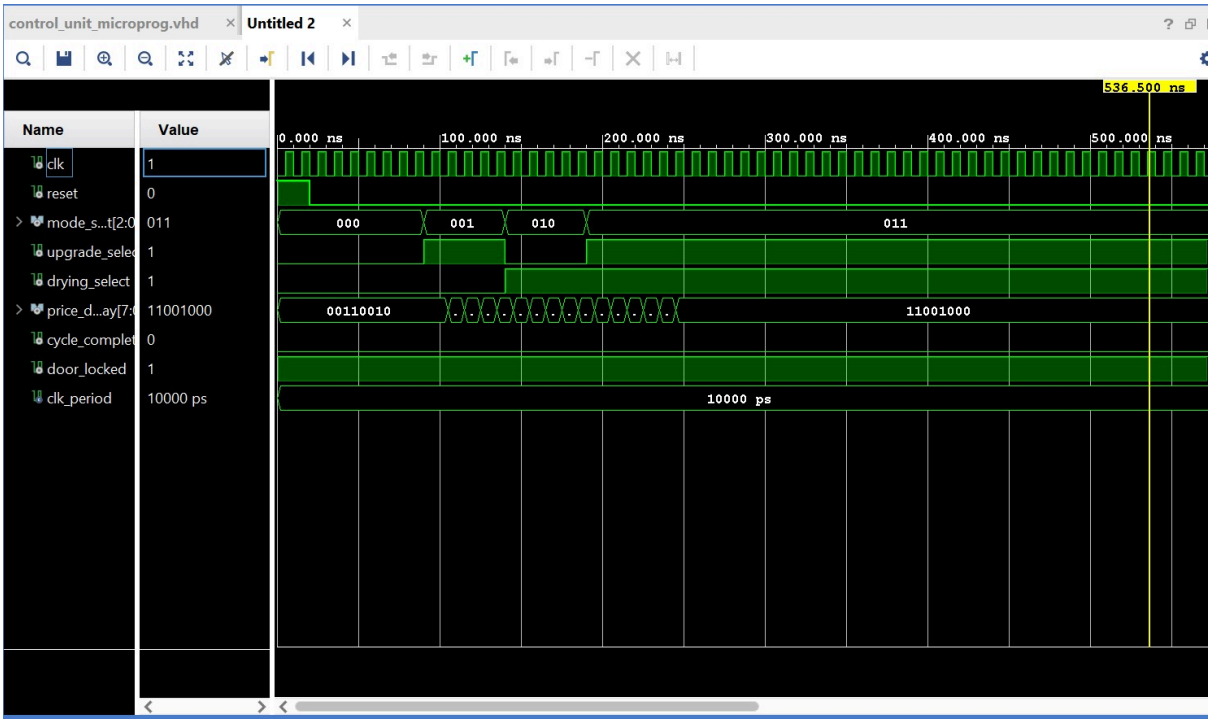

Fig 2. Testing Result

Observation of Signal Behaviour:

- 0 ns - 20 ns : High Reset Signal, forcing internal state variables to default values. For instance, the price defaults to 50.
- 20 ns - 70 ns : The mode select is “000”. The price output is correctly calculated as “00110010” (usigned 50), matching the base cost for standard mode.
- 70 ns - 120 ns : The mode select is “001”, and upgrade select is set to “1”. The price output is correctly calculated as “00110010” (usigned 50).
- 120 ns - 170 ns : The mode select is “010”, and dry select is set to “1”. The price output is correctly calculated as “11001000” (usigned 90).
- 170 ns - 220 ns : The mode select is “011”, and upgrade as well as dry select is set to “1”. The price output is correctly calculated as “11001000” (usigned 90).
- The door_locked signal remains high ('1') throughout the simulation, as the cycle completion signals (wash_done_in and dry_done_in) are not asserted by the test bench.

- The cycle_complete signal remains low ('0'), indicating the washing process is active or pending, which is expected since the timers have not completed their count.

3.3 ANALYSIS

The results of these simulations strongly indicate that the core control and billing logic of the Washing Machine controller functions correctly. The validation covers both parameter calculation and the structural integrity of the VHDL design.

The control unit module is able to successfully map different user modes and feature flags like the drying and upgrade flags to the correct total price output. This validates the cost calculation logic, where fixed costs (20 units for upgrade, 30 units for drying) are dynamically added to the variable base costs of the four operational modes.

The implementation of a modular structure facilitates easy maintenance and debugging, demonstrating sound hardware description language engineering practice. The simulation successfully tests the safety policy enforced by the door_lock_module. The door_locked signal remains high ('1') throughout the simulated period, which is the required fail-safe default state during operation. This validates the logic that the door will not unlock until an explicit completion signal is received. This confirms that the system is ready to enforce the safety constraint, only allowing the door to unlock ('0') when both the washing and drying phases are completely done (wash_done_s AND dry_done_s = '1').

CHAPTER 4

CONCLUSION

This project effectively used VHDL to design, implement, and verify a full washing machine control system. Key concepts of digital design, such as modularity, synchronous operation, state management, and hardware description language programming, are demonstrated by the system.

With simplified functionality for educational purposes, the architecture is modeled after real washing machine controllers found in commercial products. Professional embedded systems development directly benefits from the concepts of modular design, synchronous operation, and safety mechanisms.

The system's modular architecture, comprised of dedicated units for User Input, Control, Washing Cycle, Drying Cycle, Billing, and Door Lock, proved highly effective. This separation of concerns simplifies complex logic, as evidenced by the successful verification of the dynamic cost calculation and cycle duration setting within the control_unit_microprog. By using a microprogrammed-like approach, the Control Unit can flexibly orchestrate the system, allowing for systematic adjustments to operational modes and pricing without necessitating a complete redesign of the core timing mechanisms.

REFERENCES

[1] IEEE Standard VHDL Language Reference Manual, IEEE Std 1076-2019, IEEE, 2019.

[2] Chu, P. P. (2011). *FPGA Prototyping by VHDL Examples: Xilinx MicroBlaze MCS SoC Edition*. John Wiley & Sons.

[3] ModelSim User's Manual, Mentor Graphics Corporation, 2023.

[4] "Digital Sistem Design (PSDD-SG)." Digilab DTE. [Online]. Available: https://learn.digilabdte.com/books/digital-sistem-design-psddsg. [Accessed: Dec. 12, 2025].

APPENDICES

Appendix A: Project Schematic

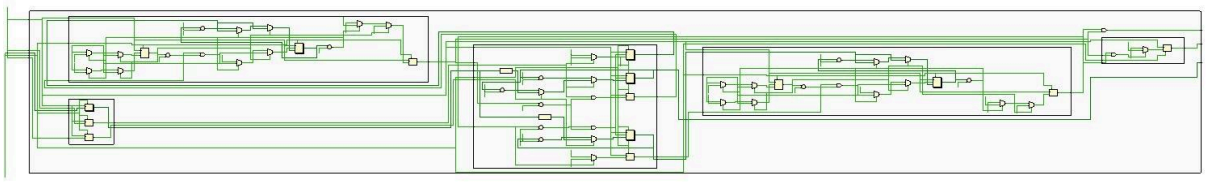

Figure 2.1

Appendix B: Documentation

Put the documentation (photos) during the making of the project