CHAPTER 1 – INTRODUCTION & PROJECT OVERVIEW

1.1 Problem Statement

In day-to-day life, vending machines are widely used to dispense products such as snacks, beverages, or tickets automatically after receiving payment.
In traditional vending machines, mechanical and microcontroller-based systems are used. However, these systems can be slower, power-hungry, and less adaptable for custom logic.
The goal of this project is to design a **digital vending machine** using Verilog HDL, implemented using an **FSM (Finite State Machine)**, which is simple, reliable, and easy to verify.

1.2 Objective of the Project

The primary objective is to **design, implement, and verify** a vending machine controller that:

- Accepts specific coins or tokens as input.
- Validates the inserted amount.
- Dispenses the selected product once the correct payment is received.
- Returns change (if applicable).
- Displays status (product delivered, insufficient funds, or invalid coin).

1.3 Applications of Vending Machines

- Automated food and drink dispensers.
- Public ticketing machines (bus, train, metro).
- Automated payment kiosks.
- Industrial supply dispensers (tools, safety gear).
- Educational campus vending systems for books, stationery.

1.4 Vending Machine Working Principle

1. The user selects a product.
2. The machine checks the price of the selected item.
3. The user inserts coins/tokens of allowed denominations.
4. If the amount inserted matches or exceeds the product price:
    - The machine dispenses the product.
    - If extra money is inserted, it returns the change.
5. If the inserted amount is less than the required price, the system waits for more coins.

1.5 Why Implement in VLSI / Verilog

- **Speed**: HDL-based FSM design processes signals in parallel, making the system faster.
- **Low Power**: VLSI design can be optimized for minimal power consumption.
- **Customizability**: Easily scalable for multiple products and prices.
- **Testing**: Simulation tools allow easy verification before hardware deployment.

1.6 Expected Output and Benefits

- Fully functional vending machine simulation in Verilog.
- Accurate product dispensing based on payment.
- Ability to handle different scenarios like insufficient funds, invalid coins, and change return.
- Modular design for future enhancements.

CHAPTER 2 – TECHNICAL DETAILS & IMPLEMENTATION

2.1 Functional Requirements

The vending machine design in this project supports:

- **Product Selection** – user selects one product (e.g., priced at ₹10).
- **Coin Input** – accepts coins of ₹5 and ₹10.
- **Change Return** – returns extra amount if overpaid.
- **Product Dispense** – delivers product when the required amount is inserted.
- **Invalid Coin Handling** – ignores unsupported denominations.

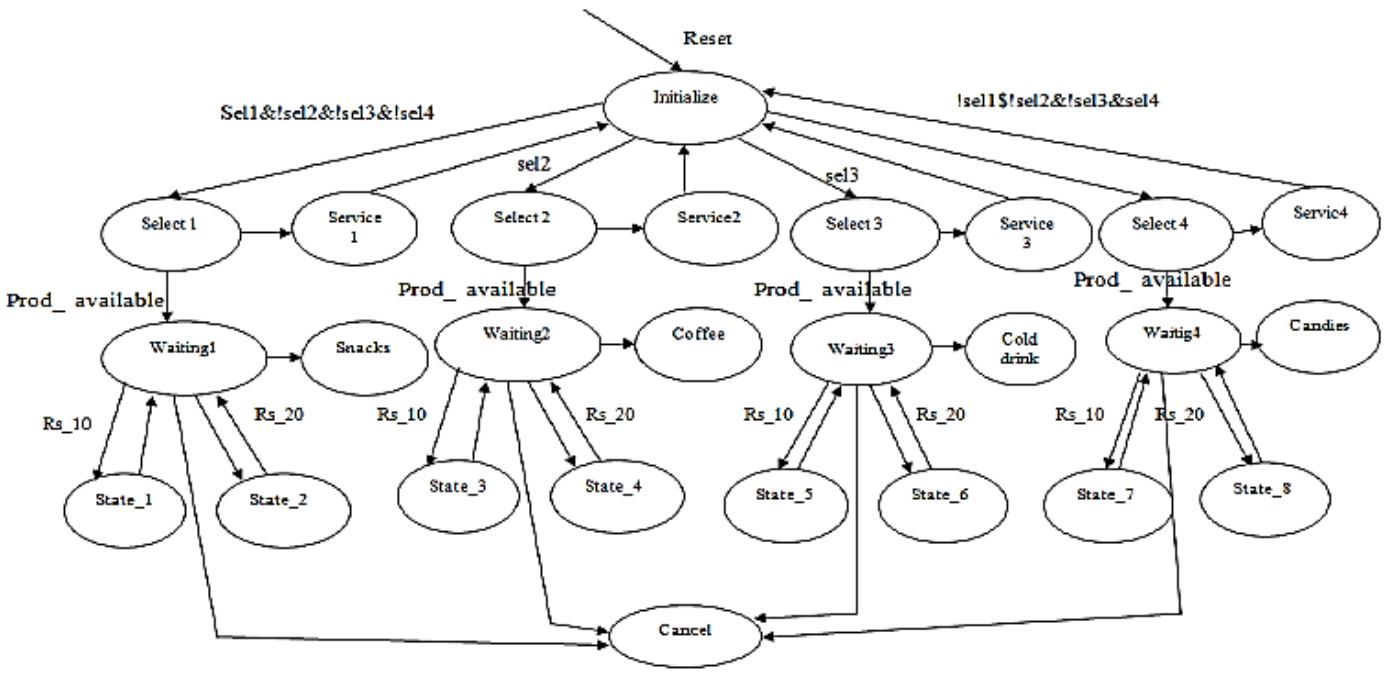
2.2 FSM (Finite State Machine) Design

The machine is designed using a **Moore FSM** where outputs depend only on the current state.

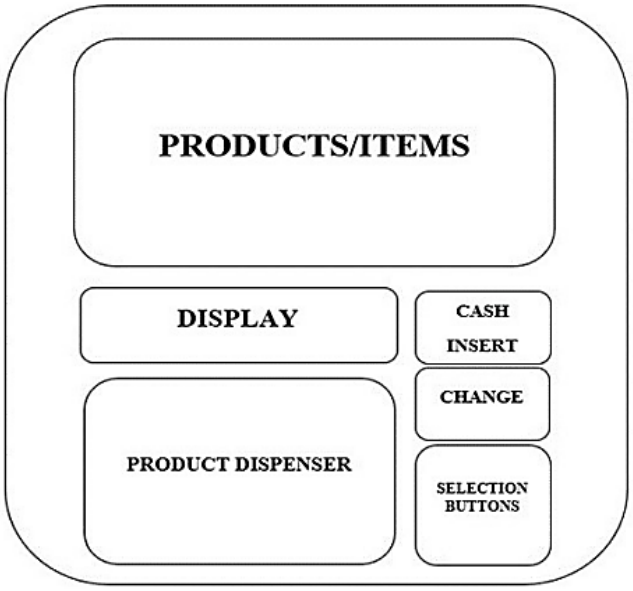
FSM States:

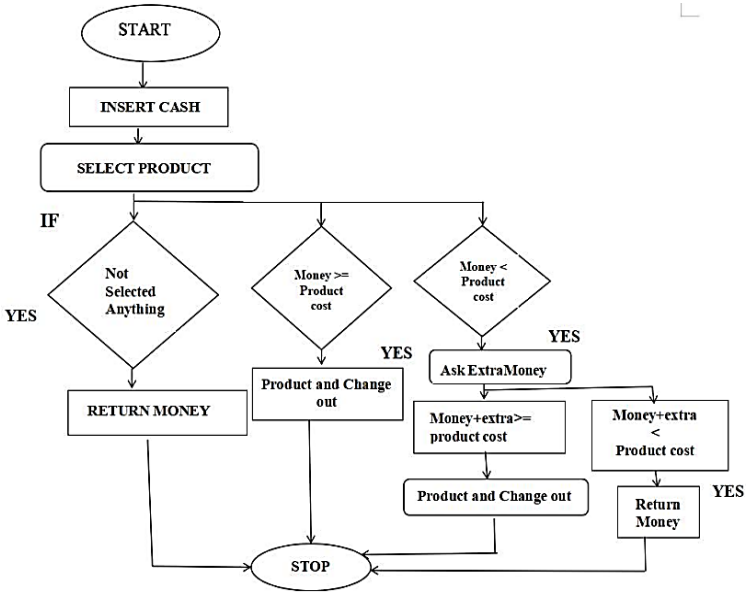
1. **Idle** – waiting for product selection.
2. **Wait_Coin** – waiting for user to insert coins.
3. **Coin_Check** – verifies if the coin is valid and adds it to the balance.
4. **Dispense_Product** – activates the product dispensing mechanism.
5. **Return_Change** – returns excess balance to user.
6. **Reset** – clears all values and returns to Idle.

2.3 FSM Diagram



2.4 Block Diagram





2.5 Input/Output Signal Description Table

Signal Name	Direction	Description
clk	Input	System clock
reset	Input	Active high reset
coin[1:0]	Input	Coin value: 00=No coin, 01=₹5, 10=₹10
select	Input	Product selection signal
dispense	Output	High when product is dispensed
change	Output	High when change is returned
invalid_coin	Output	High if coin is not supported

2.6 RTL Design Flow (Step-by-Step)

1. Define the states and state transitions for FSM.
2. Write Verilog code for FSM with non-blocking assignments for sequential logic.
3. Map coin and selection inputs to FSM transitions.
4. Assign outputs based on states (Moore design).
5. Write a testbench to verify all cases (exact payment, overpayment, underpayment, invalid coin).
6. Run simulation in tools like ModelSim / Vivado / Icarus Verilog.

2.7 Testbench Development & Simulation Setup

- **Testbench Goals**:
  - Test exact price scenario (₹10 coin for ₹10 product).
  - Test overpayment (₹5 + ₹10, return ₹5 change).
  - Test underpayment (₹5 only → wait).
  - Test invalid coin (ignored).

2.8 Handling Edge Cases

- **Invalid Coin** – ignored and machine stays in Wait_Coin state.
- **No Product Selected** – machine stays in Idle.
- **Multiple Coins in One Cycle** – only first coin is processed.
- **Change Return Logic** – returns change if overpayment is made.

CHAPTER 3 – RESULTS & CONCLUSION

3.1 Simulation Waveforms

The vending machine design was simulated using a Verilog testbench.
The waveform shows:

- **clk**: Continuous system clock signal.
- **reset**: Resets the FSM to the Idle state at simulation start.
- **select**: Goes high when the user selects a product.
- **coin**: Represents inserted coins (₹5 or ₹10).
- **dispense**: Goes high when the required amount is received, triggering product delivery.
- **change**: Goes high when excess payment is detected, triggering change return.
- **invalid_coin**: Goes high if an unsupported coin is inserted.

3.2 Output Analysis Table

Test Case	Product Selected	Coins Inserted	Total Amount	Expected Output	Simulation Result
1	Yes	₹10	₹10	Dispense product	Pass
2	Yes	₹5 + ₹5	₹10	Dispense product	Pass
3	Yes	₹5 + ₹10	₹15	Dispense product + ₹5 change	Pass
4	Yes	₹5 only	₹5	Wait for more coins	Pass
5	Yes	₹2 (Invalid)	₹0	Invalid coin message	Pass
6	No	₹10	₹10	No action (Idle)	Pass

3.3 Limitations of Current Design

- Supports only one product with a fixed price.
- Only two-coin denominations accepted (₹5 and ₹10).
- No stock-checking mechanism implemented.

3.4 Possible Improvements

- Multi-product selection with individual prices.
- Digital display for price and balance.

- Low-power design for battery-operated systems.
- Integration with digital payment methods (UPI, card).
- Stock availability check.

3.5 Conclusion

The vending machine was successfully designed, implemented, and simulated using Verilog HDL. The FSM-based approach made the control logic modular, simple, and easily scalable. Simulation results matched the expected behaviour for all tested scenarios, confirming correct functionality. This project demonstrates the effectiveness of using HDL-based FSM design in developing real-time, automated systems for commercial and industrial applications.

REFERENCES

[1] **Design of a Vending Machine Using Verilog HDL** – FSM-based design, simulation, and FPGA implementation with Genus & Encounter. *(2023)*

🔗 Link

[2] **Efficient Change System for Vending Machines** – Verilog FSM with accurate change handling and modular design. *(2025)*

🔗 Link

[3] **Multi-Level Vending Machine FSM Using Kintex-7 FPGA** – Low-power multi-product design in Verilog. *(2024)*

🔗 Link

[4] **FPGA-Based Vending Machine Optimization** – RTL power reduction and coding style comparison in Verilog. *(2025)*

🔗 Link