Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)

VENDING MACHINE CONTROLLER

The domain of the Project:

RTL Design – Digital VLSI

Team Mentors:

Mr. Satish Devarapalli (Emulation Verification Engineer , Apple)
Mr. Nikhil B. (Staff Engineer , Synopsys)

Team Members:

1. Ms. Ch. Navya - B.Tech, 4^{th} year pursuing ---- Team Leader
2. Ms. Nikhila - M.tech, 2^{nd} year pursuing --- Team member
3. Mr. K. Subramanyam - B.Tech 3^{rd} year pursuing --- Team member
4. Ms. E.Parinika - B.tech, 3^{rd} year pursuing --- Team member
5. Mr. Akash V Kashyap - B.Tech, 4^{th} year pursuing ---- Team member
6. Mr. Chetan Patil - B.Tech, 4^{th} year pursuing ---- Team member
7. Mr. Akshay Nayak - - B.Tech, 4^{th} year pursuing ---- Team member

Period of the project

March 2024 to April 2024

Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)

DECLARATION

The project titled “Vending Machine Controller” has been mentored by Nikhil sir, organised by SURE Trust, from April 2023 to August 2023, for the benefit of the educated unemployed rural youth for gaining hands-on experience in working on industry relevant projects that would take them closer to the prospective employer. I declare that to the best of my knowledge the members of the team mentioned below, have worked on it successfully and enhanced their practical knowledge in the domain.

Team Members:	Signatures
1. Ms. Ch. Navya	Ch. Navya
2. Ms. Nikhila	Nikh
3. Mr. K. Subramanyam	K. Subramanyam.
4. Ms. E.Parinika	E · Parinika
5. Mr. Akash V Kashyap	Akash
6. Mr. Chetan Patil	Chetan
7. Mr. Akshay Nayak	Akshay

Mentor’s Name : **Mr. Satish Devarapalli**
Designation— Emulation Verification Engineer , Apple

Mentor’s Name : **Mr. Nikhil B.**
Designation— Staff Engineer , Synopsys

Prof. Radhakumari
Executive Director & Founder
SURE Trust

Table of Contents

Executive Summary

This project implements a modular, configurable vending machine controller (VMC) using Verilog HDL, designed for FPGA/ASIC-based systems. The architecture supports up to 1024 items and integrates item selection, currency handling, and dispensing logic. A dual-mode operation allows dynamic switching between configuration mode via an APB interface and operational mode for real-time vending. Configuration data—such as item cost, availability, and dispense count—is stored and accessed using a dual-port memory with clock domain crossing, ensuring synchronization between the APB and system clocks. The design includes modules for item validation, currency summation with change calculation, and a state-based dispense logic. Thresholds for valid transactions are enforced through currency comparison with item cost, and internal counters ensure inventory consistency. No external memory controller or machine learning is used—functionality is fully implemented in hardware logic for efficiency and determinism. The system was verified through a dedicated testbench simulating various operational scenarios. Emphasis was placed on modularity, reusability, and robustness. This project showcases a resource-efficient, scalable hardware vending solution suitable for embedded systems and SoC platforms.

Introduction

Background and Context

A vending machine is an automated electromechanical system designed to dispense products such as snacks, beverages, or tickets to users after receiving appropriate payment. These machines operate without the need for human supervision and are commonly found in public spaces such as offices, malls, railway stations, and schools. Their ability to offer 24/7 access to a variety of items makes them a convenient solution for quick purchases. Vending machines are automated systems designed to dispense products in exchange for currency without human intervention. These machines are commonly found in public areas and provide a convenient method for purchasing items like snacks, beverages, or small goods. In a digital design or hardware description context, simulating the logic of a vending machine provides a practical example to understand the application of finite state machines (FSMs), combinational and sequential logic, and transaction handling. The vending machine is a popular choice for digital design projects because it involves key hardware design concepts such as: Coin detection and accumulation, finite state transitions, Output control (dispensing and change/refund management), handling asynchronous events (e.g., cancel button or stock status). This project involves the creation of several interconnected modules, such as item selection, inventory checking, payment and change calculation, and control logic. Implementing a vending machine in Verilog HDL (Hardware Description Language) allows students and engineers to explore real-world problems such as transaction validation, user input handling, and change return logic, all within a synthesizable digital circuit. Moreover, verifying the design using a test bench enables better understanding of simulation, waveform analysis, and design debugging. This project emphasizes how FSM-based designs are used to model real-time systems with predictable and manageable behavior, making it an ideal learning module for early-stage digital design enthusiast.

Problem Statement

To Design and verify a digital Vending Machine Controller IP that supports up to 1024 items with a configurable maximum currency input value of 100 INR. The controller should operate across three distinct modes: Reset, Configuration, and Operation. It must manage asynchronous currency inputs, perform item dispensing based on item selection and currency input, handle change return, and track item inventory and dispensed counts.

Key Functional Requirements:

1. Clock Domains: System clock: 100 MHz. , Configuration clock: 10 MHz.

2. Modes of Operation: Reset Mode: Resets all hardware elements. Configuration Mode: Load item values, available counts, and reset dispensed counts using an APB interface.Operation Mode: Accepts item selection and currency inputs, processes transactions, and handles item dispensing or change return.

3. Currency Input: Accepts multiple denominations: {5, 10, 15, 20, 50, 100}.Input is asynchronous to the system clock (10KHz to 50MHz).

4. Interfaces: APB-based configuration interface for setting up items.

Scope

The scope of this project is to design and verify a digital Vending Machine Controller IP core intended for use in programmable vending machines that can adapt to different configurations, item types, and currency inputs. The project encompasses RTL design, functional verification, and basic integration interfaces, and is suitable for ASIC/FPGA implementation.

1. Design Scope:
The project involves developing a modular and synthesizable RTL design in Verilog for a vending machine controller. The design includes separate blocks for mode control, item database management, currency input handling, transaction processing, output generation, and configuration through the APB interface.

2. Verification Scope:
Verification includes writing test benches in Verilog to validate functionality in all operating modes. Test scenarios include edge cases like exact payment, overpayment with change return, underpayment, and out-of-stock conditions. Timing constraints such as dispensing within 10 system clocks are checked using assertions and functional coverage.

3. Interface Scope:
The design implements and simulates key interfaces including the system clock and reset, APB configuration interface, currency input and item selection interfaces, and the item dispense and change output interface.

4. Parameterization and Scalability:
The controller is parameterized to support up to 1024 items and currency values up to ₹100. Data widths and memory sizes are configurable, allowing reuse across different vending machine configurations.

Limitations

While the Vending Machine Controller IP is designed to be flexible and parameterizable, the following limitations are present in the current specification

1. **Fixed Currency Denominations:** Only specific denominations are supported: 5, 10, 15, 20, 50, and 100 INR.The design cannot handle coins or custom currency values without modifying the RTL.

2. Single Item Dispense per Transaction: The controller supports dispensing only one item per transaction.Multiple item purchases in a single session are not supported, requiring repeated item selections.

3. Limited User Interaction Handling: No support for:Transaction cancellation after currency insertion.Timeouts if a user stops inserting currency midway.Authentication or payment via digital methods (e.g., cards, QR codes).

4. No Security or Fault Detection Logic: The design does not include:Tamper detection, Fake note detection Error handling for failed dispense mechanisms

5. Asynchronous Input Handling Complexity: Currency and item selection inputs are asynchronous to the system clock, which may require synchronizer logic that can be sensitive to metastability if not carefully designed.

6. No Real-Time Monitoring or Logging: No provision for real-time transaction logging or external monitoring/debug interfaces. Tracking is limited to internal counters for dispensed and available items.

7. APB Configuration Interface Only: Configuration is limited to an APB interface at 10 MHz, which may not be ideal for high-speed or real-time dynamic configuration use cases.

Innovation

This project showcases innovation by applying digital design principles to replicate real-world transaction behavior in a fully automated vending system. Unlike basic FSM implementations, this design integrates several advanced features that enhance functionality and user experience, such as:

Dynamic Coin Recognition: The machine intelligently handles multiple denominations, allowing users to insert coins in any combination without requiring exact payment.

Smart Change Logic: The design ensures that excess payment is correctly calculated and returned as change, emulating real-world vending behavior more accurately.

Transaction Flexibility: Introducing a cancel button adds a layer of user control, enabling refunds mid-transaction—something not typically seen in minimal FSM designs.

By combining FSM design with real-world usability features, this project bridges the gap between theoretical digital logic and practical embedded system applications—offering a hands-on experience in building user-centric digital systems.

Project Objectives

Project Objectives and Expected Outcomes

1. Develop a Configurable Vending Machine Controller IP

Design and implement a flexible, reusable, and parameterized digital IP that can control vending machine operations, including item selection, currency input, and item dispensing.

Expected Outcome: A synthesizable Verilog IP core that supports up to 1024 configurable items, multiple currency denominations, and seamless integration into various vending machine systems via a standard APB configuration interface.

2. Implement Dual Operating Modes with APB-Based Configuration

Design two operational modes — configuration and operation — where the configuration mode allows programming of items and their values via APB, and the operation mode handles item dispensing based on user input.

Expected Outcome: A robust dual-mode controller where item data can be preloaded using APB in configuration mode and vending logic operates in real time during operation mode, ensuring clean mode switching and accurate state retention.

3. Handle Real-Time Item Dispense and Change Return

Develop logic to manage user interaction: selecting items, inserting currency, verifying price, dispensing items, and returning change accurately and efficiently.

Expected Outcome: A low-latency (under 10 system clock cycles) item dispense logic that processes asynchronous currency inputs, calculates total inserted value, matches item cost, and returns the correct change if needed.

4. Enable Smart Empty Item Detection and User Feedback

Detect when a selected item is unavailable and provide immediate feedback along with returning any inserted currency.

Expected Outcome: A controller that detects unavailable items, returns inserted currency on the first input, and signals the user through an empty item code, ensuring graceful error handling and user

experience.

5. Maintain Per-Item Inventory and Dispense Tracking

Track available and dispensed item counts per item to maintain real-time inventory and support analytics.

Expected Outcome: Real-time counters (avail_items, disp_items) for each item stored in internal memory, with support for resetting and updating via configuration mode.

6. Support Asynchronous Input Handling for Currency and Item Selection

Develop mechanisms to handle human-paced asynchronous inputs (10KHz–50MHz) for currency and item selection without metastability or data corruption.

Expected Outcome: A robust synchronization system that reliably captures and processes asynchronous user input signals in the 100MHz system clock domain, ensuring glitch-free operation.

7. Optimize for Hardware Efficiency and Integration

Design a resource-efficient controller that meets performance and area goals suitable for embedding in low-to-mid complexity SoCs.

Expected Outcome: An RTL implementation that uses efficient FSMs, memory structures, and minimal combinational logic, making the IP lightweight, portable, and integration-ready.

8. Ensure Real-Time Transaction Processing with Deterministic Latency

Design the vending controller to process currency input, item validation, dispense decision, and change return within a predictable, deterministic number of clock cycles suitable for real-time applications.

Expected Outcome: A real-time vending controller capable of making complete dispense decisions within less than 10 system clock cycles after receiving valid currency input, ensuring a responsive user experience under strict timing requirements.

9. Demonstrate Real-Time System Behavior in Noisy and Bursty Input Scenarios

Simulate and verify the system's response to bursty or noisy inputs, such as rapid currency insertion, repeated item selections, or user errors, to ensure it maintains correct state and consistent output.

Expected Outcome: A validated system that performs reliably under burst traffic and erratic user behavior without misfires, false triggers, or state corruption — proving its real-time robustness in practical use cases.

Methodology and Results

Methods/Technology Used

1. Verilog HDL (Hardware Description Language)

Used to design the core logic of the vending machine controller, including finite state machines, dual-port memory access, and synchronous/asynchronous interfaces.

2. Parameterized Dual-Port RAM Architecture

Implemented in Verilog to allow concurrent access from the APB configuration interface (Port A) and real-time vending logic (Port B) without contention.

3. AMBA APB Protocol Integration

Used to interface the controller with a host processor for configuration. This enabled the loading of item values, counts, and reading system status during configuration mode.

4. Simulation and Functional Verification in Vivado Simulator

Comprehensive testbenches were written to validate FSM transitions, APB read/write operations, currency handling, item selection, and corner cases such as empty item dispensing.

Tools/Software Used

Xilinx Vivado Design Suite-Used for RTL design, synthesis, implementation, simulation, and verification of the vending machine controller. Vivado provided an integrated environment for constraint management, timing analysis, and bitstream generation.

Project Architecture

1. System Clock & Reset Initialization

The system is initialized with a 100 MHz system clock for real-time vending operations and a 10 MHz APB clock for configuration. The design uses an active-low reset (prstn) to initialize all internal registers, memories, and FSMs.

2. Configuration Mode via APB Interface

When the system enters **configuration mode** (cfg_mode = 1), a host processor or external controller writes to the internal item memory and configuration registers via the **AMBA APB**

protocol. This includes:

- Loading available item counts
- Setting item values (in Rs)
- Resetting dispensed item counters

All APB transactions are handled synchronously with respect to the pclk.

3. Dual-Port RAM Management

A **parameterized dual-port memory** (item_memory[]) stores item information such as:

- Item price
- Available item count
- Dispensed item count
- **Port A (APB Clock)** is used for configuration
- **Port B (System Clock)** is used for runtime vending operations

This allows simultaneous config and operation access without data corruption.

4. Item Selection Input Handling (Asynchronous Interface)

Users select an item using an item_select signal and validate it with item_select_valid. This signal may come from a slow interface (10kHz–50MHz) and is **synchronized** to the 100 MHz system clock for safe processing.

5. Currency Input Handling (Asynchronous Interface)

Currency values (₹5, ₹10, ₹15, ₹20, ₹50, ₹100) are input using the currency_value and currency_valid signals. Like item selection, these inputs are also asynchronous and synchronized before being accumulated for processing.

6. Vending Decision Logic (FSM in Operation Mode)

Once enough currency is inserted for a selected item:

- The system checks item availability

- If the inserted value exceeds the item cost, it **returns the change** (via currency_change)
- If the item is not available, it **returns the full currency immediately** and outputs an **empty item indicator**

7. Real-Time Output Signal Generation

When an item is dispensed:

- The item_dispense_valid signal is pulsed for one system clock
- item_dispense holds the item number
- If applicable, currency_change outputs the change amount

These outputs are **single-cycle pulses** for synchronous detection by external systems.

8. Inventory Tracking and Status Calculation

After each dispense or config update:

- Internal counters update disp_items and avail_items
- A running sum of available items is used to generate a compact status register (vending_machine_cfg)
- This register helps host systems monitor machine stock levels in real-time

Mode	Condition	Description
Reset Mode	rstn = 0	Resets all internal states and registers
Config Mode	rstn = 1, cfg_mode = 1	Host (vendor) loads item values and stock through the APB interface
Operation Mode	rstn = 1, cfg_mode = 0	Handles currency input, item selection, and vending logic

Configuration Flow (Vendor/APB-controlled)

1. Enter Config Mode (cfg_mode = 1)
2. Through APB interface:
   - Set no_of_items in Vending_machine_cfg register.

- For each item in Item_cfg[]:
  - Set item_val (cost)
  - Set avail_items (stock)
  - Clear disp_items count

Currency Flow (User interaction)

1. User selects item using item_select + item_select_valid
2. Inserts currency (currency_value + currency_valid)
3. Currency is accumulated until >= item value
4. If item is available:
   - Item is dispensed
   - item_dispense and item_dispense_valid are asserted
   - Change is returned if applicable (currency_change)
   - disp_items count incremented; avail_items decremented
5. If item is unavailable:
   - An empty item indication is flagged
   - Full inserted amount is returned as change

Module	Responsibility
Mode Controller	Transitions between reset, config, and op modes
APB Config Interface	Handles vendor configuration via APB
Item Configuration Memory	Holds: cost, available stock, and dispensed count
Currency Accumulator	Adds inserted money asynchronously until sufficient
Item Selection FSM	Manages user choice, ensures it is valid
Dispense Logic	Determines if item can be dispensed and handles change
Change Logic	Computes how much change to return
Output Generator	Provides outputs (item_dispense, currency_change) in sync with system clock

Registers and Memory

A. Vending_machine_cfg (Addr: 0x4000_0000)

- no_of_items: RW – total number of configured items

B. Item_cfg[]

Each entry:

- item_val [15:0] – item cost
- avail_items [23:16] – remaining stock
- disp_items [31:24] – count of dispensed items

Interface Summary

General

- clk: 100MHz system clock
- rstn: active-low reset
- cfg_mode: high during config mode

Currency Interface

- currency_value, currency_valid: input from user (10kHz–50MHz)

Item Selection

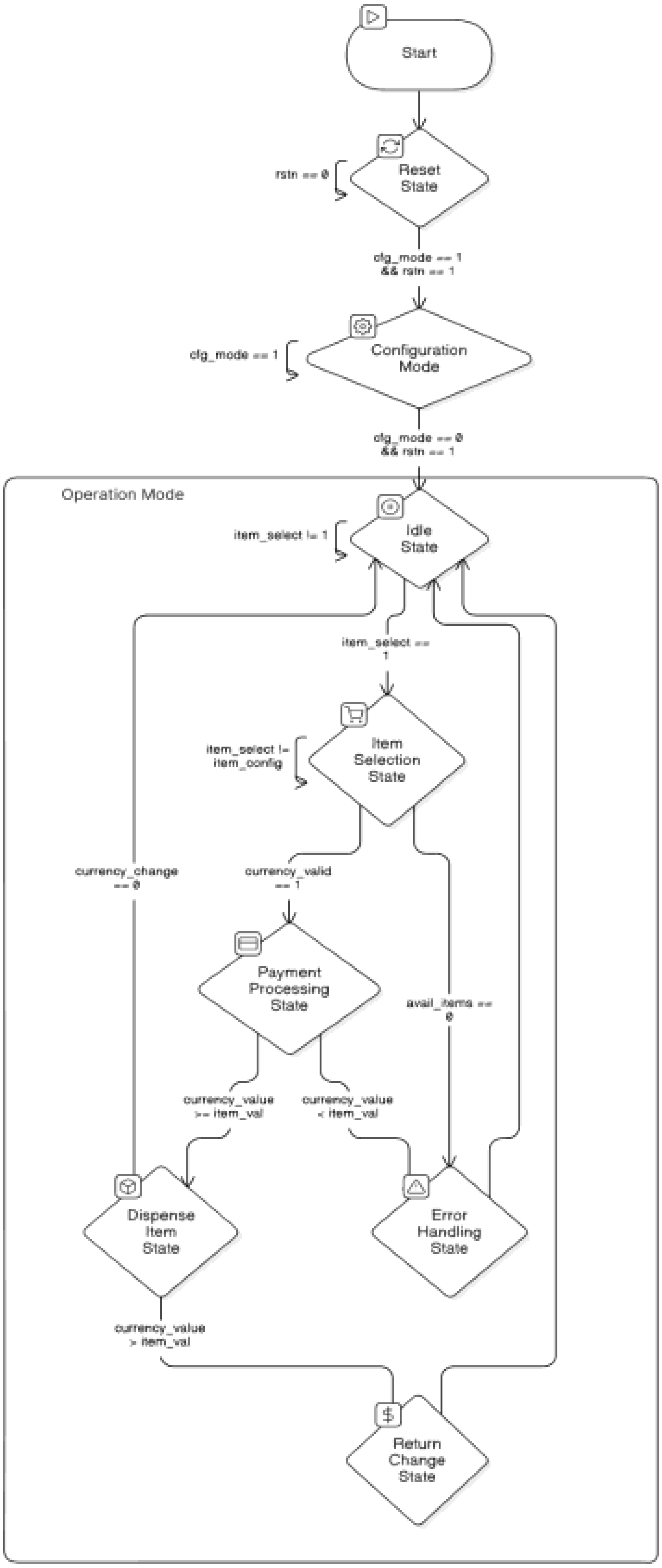
- item_select, item_select_valid: user's choice of item

Output

- item_dispense, item_dispense_valid: single-cycle pulse when item is dispensed
- currency_change: value of change returned

APB Interface (Configuration)

- pclk: config clock (10MHz)
- prstn: synchronous reset
- paddr, psel, pwrite, pwdata, prdata, pready: standard APB signals

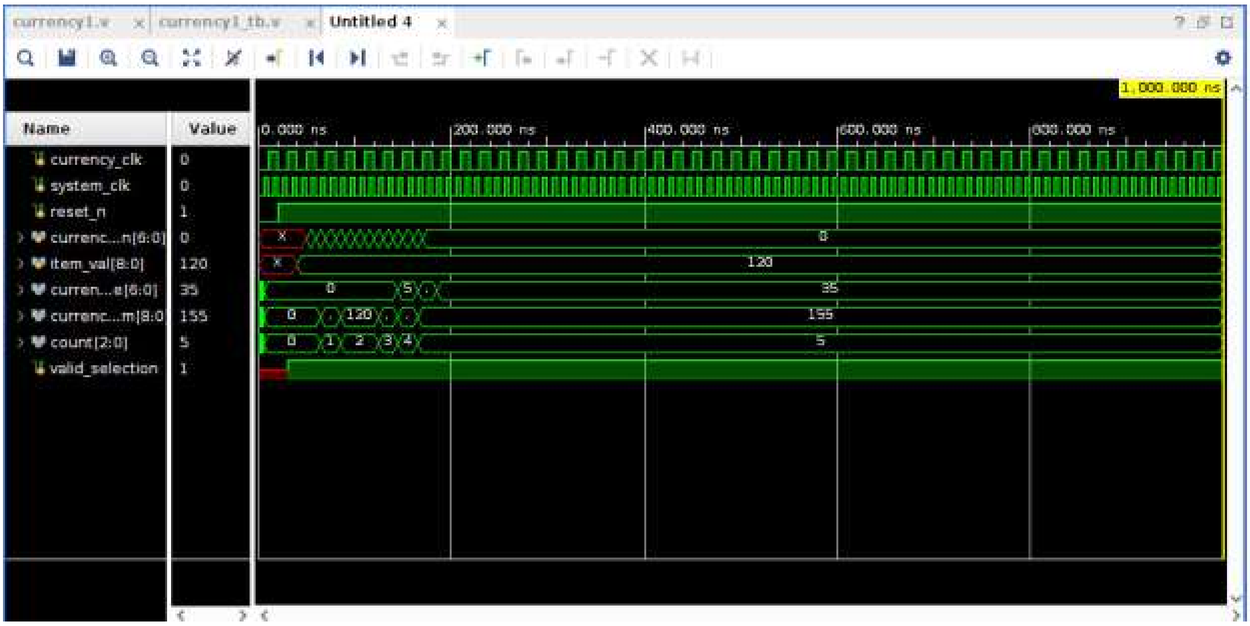


Flow Chart of Vending Machine Controller

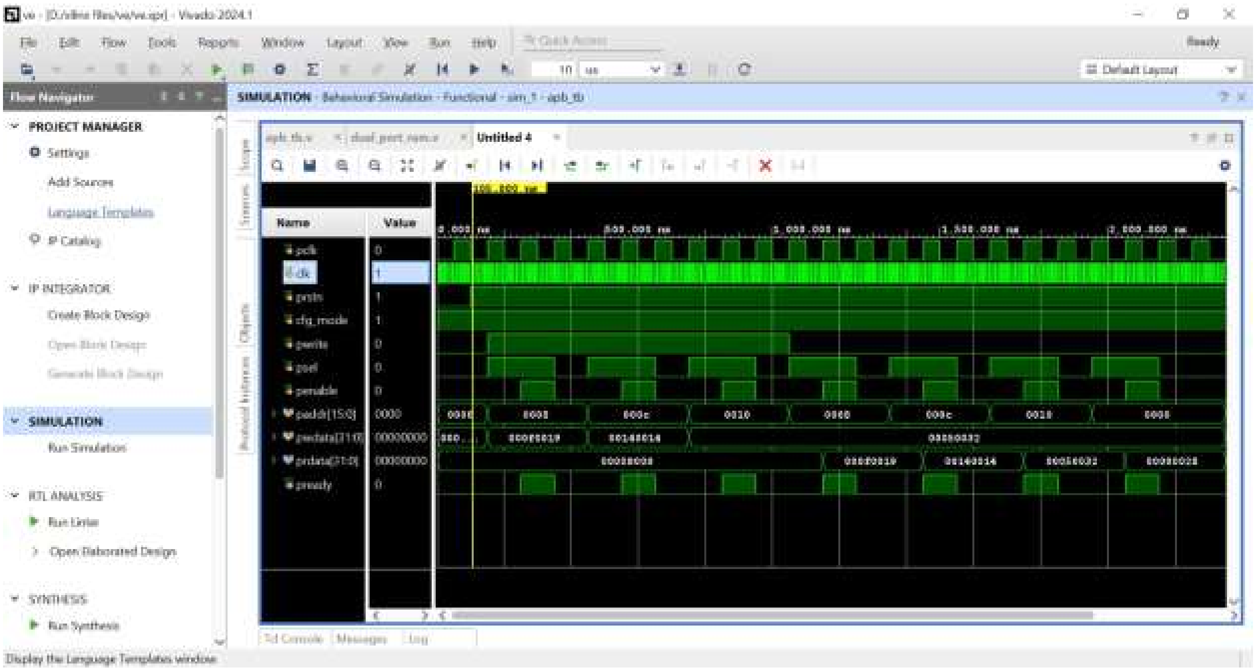
Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)

RESULTS

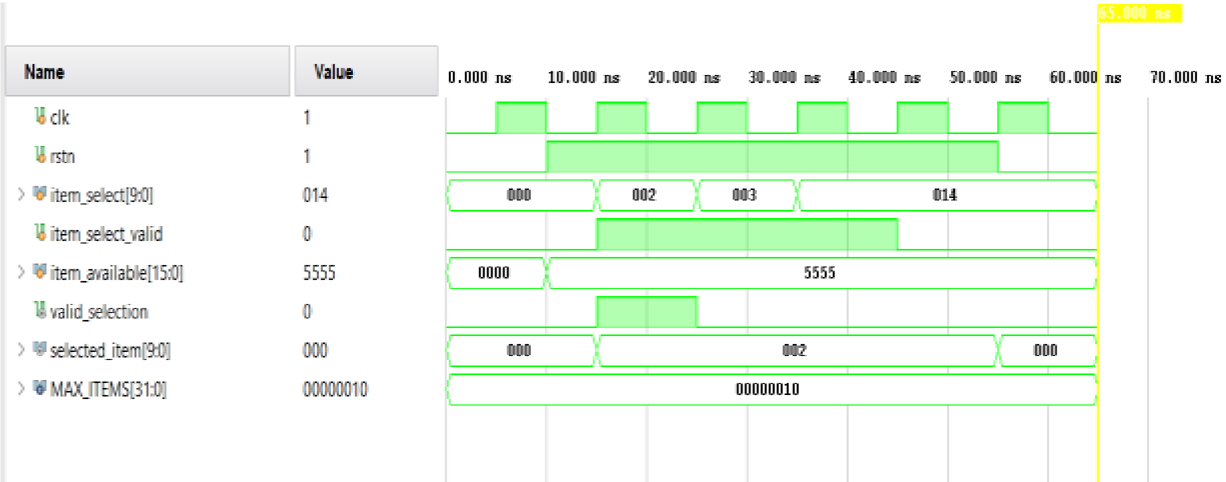
1. Currency Module



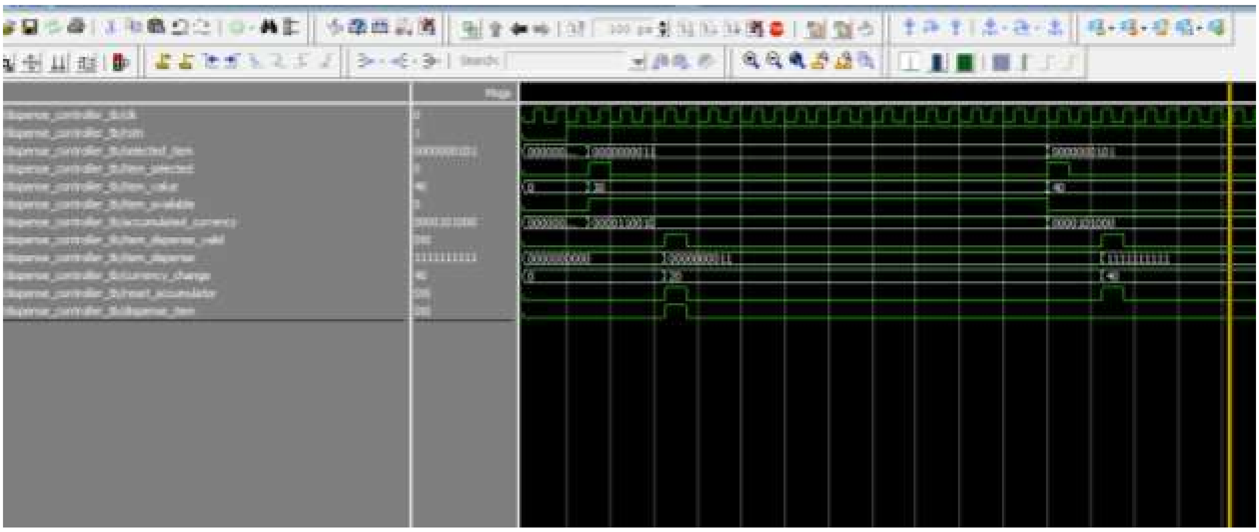
2. Dual Port RAM Module



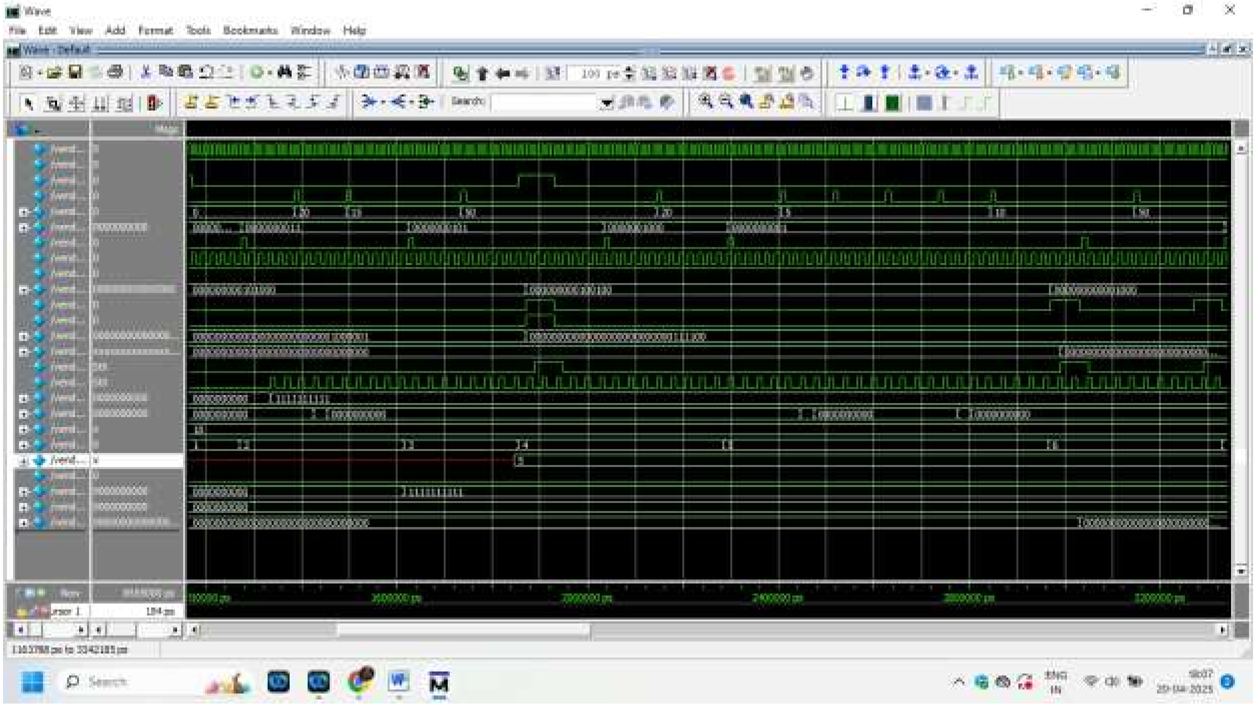
3. Item Selection Module



4. Item Dispense Module



5. Top Module



GITHUB LINK: [*https://github.com/sure-trust/CHILUKA-NAVYA-g3-integrated-vlsi/tree/main/FinalProject/VendingMachine*](https://github.com/sure-trust/CHILUKA-NAVYA-g3-integrated-vlsi/tree/main/FinalProject/VendingMachine)

Learning and Reflection

The successful execution of our Vending Machine Controller project using Verilog was not only a technical milestone but also a valuable learning experience for every member of our team. As we progressed through the stages of design, coding, simulation, debugging, and final integration, we encountered a wide range of challenges that contributed to our individual and collective growth. These experiences helped us gain a deeper understanding of digital design concepts, hardware description languages, and the importance of synchronization across multiple clock domains. In addition to the technical learning, we also developed essential soft skills such as task coordination, teamwork, time management, and documentation. This section outlines the specific learnings and reflections of each team member, showcasing both the technical skills acquired and the overall experience of working together on a real-world digital system design.

Learning :

1. NAVYA (TEAM LEADER):

a. Led the development of the **APB protocol**, including writing individual testbenches for thorough verification and validation.

b. Played a crucial role in connecting the team members, ensuring smooth communication and collaboration throughout the project.

c. Worked relentlessly to resolve issues and guide the team, contributing significantly to the successful completion of the project.

2.K.SUBRAMANYAM :

a. Gained in-depth understanding of the **APB (Advanced Peripheral Bus) protocol**, especially in the context of configuration interfaces.

b. Discovered and implemented the use of a **dual-port RAM** to allow direct access to configuration data during operation mode while maintaining APB-based configuration.

c. Improved problem-solving skills by exploring architectural alternatives and optimizing data access paths.

3. NIKHILA :

a. Initially attempted to use **file handling** for simulation purposes but faced issues accessing memory registers, so dropped the approach

b. Later contributed to the design by working on **APB access through Port B** of the dual-port RAM to improve operation mode data handling .

c. Faced significant **debugging challenges** during simulation, learning how to identify and fix issues like incorrect signal assignments and unexpected output behavior.

4. PARINIKA:

a. Took responsibility for implementing the **currency handling module** in Verilog, which involved validating inputs, tracking balances, and coordinating with the item dispensing logic.

b. Learned how to systematically **debug Verilog code**, especially in situations where simulation didn't show direct errors but behaved incorrectly.

c. While working on the micro-architecture, explored and learned how to select appropriate combinational blocks like **multiplexers and decoders** to simplify control logic.

d. Developed skills in designing **state machines** and managing control flow in currency validation and refund logic.

5. AKSHAY :

a. Developed the **Item Dispense Module**, which involved implementing the logic to dispense items based on valid user input and available stock.

b. Gained experience in integrating **item dispense logic with other system components**, such as currency validation and item selection, to ensure smooth operation.

c. Contributed to **debugging and testing** the module to ensure that the correct item was dispensed and that stock was updated accurately.Also helped in top level integration and testing.

6. AKASH:

a. Worked on the development of the **Item Selection Module**, which involved implementing the logic for selecting items based on user input and available stock

b. Faced challenges with **register access and synchronization** between different clock domains, which required careful attention to timing and data integrity.

c. Learned valuable lessons in **system-level design**, particularly in ensuring that individual modules work together seamlessly within the overall system architecture.Also helped in top level integration and testing.

7. CHETAN :

a. Led the **top-level integration** of all modules (item selection, item dispense, currency handling, APB interface, and dual-port RAM) into a cohesive system.

b. Faced challenges in ensuring **seamless communication** between modules and resolving issues related to signal integrity, timing, and clock domain synchronization.

c. Worked extensively on **debugging and testing** the integrated system to ensure that all components worked together as expected and the system operated smoothly.

d. Learned the importance of **team collaboration**, as integration required close coordination with all team members to ensure the individual modules were aligned with the overall project objectives.

OVERALL LEARNING EXPERIENCE :

1. NAVYA (TEAM LEADER):

*"Throughout this project, I took responsibility for developing and testing the **APB protocol**. This involved writing individual testbenches to ensure thorough verification and validation of the protocol's functionality. It was a challenging task, but it gave me deeper insights into how to approach testing and debugging complex systems. In addition to technical tasks, I also focused on **team coordination**, ensuring that communication between team members remained smooth and that everyone was aligned with the project goals. I found it crucial to create an environment where everyone could collaborate effectively and share ideas.Despite facing challenges along the way, I worked relentlessly to guide the team and help resolve issues. Seeing the project succeed after all the hard work made the experience extremely rewarding, and it taught me a lot about both technical development and team leadership."*

2.K.SUBRAMANYAM :

"During the course of this project, I was primarily responsible for implementing the APB protocol and handling the configuration interface of the vending machine system. One of the key challenges I encountered was ensuring that the configuration mode, which must be handled strictly through the APB interface, did not introduce unnecessary delays during operation mode. Initially, I struggled with accessing the configuration data efficiently during operation without involving the APB interface, as this would have impacted performance.After analyzing the problem, I realized that using a ***dual-port RAM*** *would be the ideal solution. This allowed the system to write configuration data through the APB interface and later access the same data directly in operation mode without delay. I worked on implementing this memory design and carefully addressed the register-level addressing and synchronization required across different clock domains.This part of the project taught me not only the technical depth of memory interfacing and protocol handling in Verilog, but also how to solve real-time challenges by thinking critically and exploring architectural alternatives. I also tried to working on top level integration of all design modules .It was a significant learning experience that strengthened my understanding of efficient hardware design and system-level integration"*

3. NIKHILA :

"At the start of the project, I initially thought of using ***file handling*** *for simulation to manage and access data, but quickly ran into issues. The simulation couldn't properly access the memory registers defined in the design, which led me to discard that approach. As I continued working on the project, I faced various debugging challenges. A lot of time was spent identifying and resolving issues with the simulation, such as incorrect signal assignments and unexpected behavior in the outputs. I improved my skills in* ***debugging Verilog code****, learning to identify subtle issues that could cause significant problems in the simulation results.*

Later, I shifted focus to ***APB access through Port B*** *of the dual-port RAM, which allowed more efficient data handling between configuration and operational modes. This experience taught me valuable lessons about troubleshooting, testing, and adapting my approach to achieve a functional design."*

4. PARINIKA:

"*I worked on the* ***currency handling module*** *of the vending machine project. It was a challenging yet rewarding experience. Initially, I ran into a lot of issues—sometimes the simulation would run without any errors, but the values wouldn't store properly, or the outputs wouldn't come out as expected. In a few cases, I even encountered* ***infinite loops*** *during simulation, which took a lot of time to debug and fix. These challenges taught me how to be patient and methodical while debugging* ***Verilog code****. I learned how even a small mistake in signal assignment or timing could affect the entire design's behavior. While* *designing the control logic, I realized the importance of selecting the right combinational circuits, like* ***multiplexers and decoders****, to make the architecture more efficient and easier to control Working on the* ***micro-architecture and state machines*** *for currency validation, balance tracking, and refund logic helped me strengthen my understanding of both digital design principles and real-world hardware implementation. Overall, it was a great learning experience that pushed me to think critically and improve my design and debugging skills."*

5. AKSHAY :

" In this project, I was responsible for designing the ***Item Dispense Module****, ensuring that the correct item was dispensed only after validating the user's input and currency. A major challenge was coordinating with the* ***currency handler*** *to update item stock accurately and handle invalid transactions without disrupting the system. This experience improved my understanding of module interaction and real-time control logic in Verilog. I also contributed to the top-level integration of all modules, where I focused on ensuring smooth communication and debugging system-level issues. Working on both module design and integration helped me strengthen my skills in timing analysis, signal coordination, and collaborative development. Overall, it was a valuable learning experience that enhanced my technical and problem-solving abilities."*

6. AKASH:

"In this project, I worked on the ***Item Selection Module****, focusing on selecting items based on user input and available stock. One of the key challenges I faced was maintaining data consistency and ensuring smooth synchronization across different clock domains. To address this, I carefully managed signal integrity and timing. I gained valuable experience in designing Verilog-based state machines to control the selection logic efficiently. During testing, I encountered and resolved issues related to incorrect item dispensation and handled edge cases like item unavailability. These challenges helped me sharpen my debugging and problem-solving skills. I also learned the importance of seamless interaction between modules in a system-level design. Overall, this project was a great opportunity to strengthen my technical abilities and improve collaboration within a team environment."*

7. CHETAN :

"My main responsibility in this project was handling the ***top-level integration*** *of all the modules—item selection, item dispensing, currency handler, APB interface, and the dual-port RAM. This role gave me a comprehensive view of the entire system, and it came with the challenge of ensuring that all modules communicated correctly and the data flow was consistent.One of the most critical tasks was resolving issues related to* ***signal mismatches and timing coordination*** *between modules, especially in different clock domains. I worked closely with other teammates to debug integration issues and verify that each module was functioning properly within the larger design.*

Conclusion and Future Scope

Objectives :

The objectives of the Vending Machine Controller are to define the architecture, interfaces, and operational behavior of a digital IP controller used in vending machines. Here are the key objectives:

1. **Design a reusable IP** for a vending machine controller that can be integrated into various vending systems with flexible item configurations.
2. **Support for multiple items** (up to 1024) and variable currency denominations (5, 10, 15, 20, 50, 100 INR).
3. **Efficient operation** with a latency of less than 10 system clock cycles between receiving valid currency and dispensing an item.
4. **Three operation modes**:
   - **Reset Mode**: Hardware reset state.
   - **Configuration Mode**: Loading item data and resetting counters via an APB interface.
   - **Operation Mode**: Handling item selection, currency insertion, and dispensing.
5. **Interface standardization**, including:
   - System clock and reset.
   - Currency input.
   - Item selection input.
   - Item dispense and change output.
   - APB-based configuration interface.
6. **Inventory tracking**, including:
   - Available item count.
   - Dispensed item count per item type.
7. **Robust user experience**, ensuring:
   - Immediate change return if a selected item is out of stock.
   - Clear signaling for item dispensing and empty item conditions.
8. **Parameterization**, allowing customization of:
   - Number of items.

    - Maximum currency value.

Achievements:

1. **Reusable & Scalable IP Design**
   - Developed a parameterized digital controller IP that can support up to 1024 unique items and currency denominations up to 100 INR.

2. **Multi-Mode Operation**
   - Implemented a clear mode-based architecture:
     - **Reset Mode**: For initialization.
     - **Configuration Mode**: For vendor-based item loading via APB.
     - **Operation Mode**: For runtime transactions.

3. **Asynchronous Currency Interface**
   - Designed to handle currency inputs at lower human-scale frequencies (10KHz – 50MHz), independent of the 100MHz system clock.

4. **Smart Inventory & Transaction Tracking**
   - Keeps track of:
     - Total dispensed items
     - Remaining items
     - Item-specific values

5. **Real-Time Error Handling**
   - Built-in mechanism to:
     - Dispense change immediately if the selected item is out of stock.
     - Signal "empty item" scenarios properly.

6. **Configurable via Standard APB Bus**
   - Allows integration with other system components and configuration via a widely-used protocol (APB @ 10MHz).

Conclusion :

The Vending Machine Controller is a comprehensive and scalable digital IP core designed to meet the needs of modern vending applications. With its support for asynchronous currency input, configurable item and value parameters, and efficient state-based operation, the controller ensures accurate, low-latency responses for real-time transactions. Its structured design, featuring clear modes for reset, configuration, and operation, enhances maintainability and ease of integration. The APB-based configuration interface further enables flexible customization, making the IP suitable for a wide range of product vending systems. This design not only improves operational efficiency but also provides a solid foundation for future enhancements in vending automation.

Future Scope :

Vending machines have evolved beyond simple snack dispensers into sophisticated,smart systems with a variety of innovative features. The future of vending technology is centeredaround enhancing user convenience, promoting sustainability, and incorporating cutting-edge technologies.

1. Blockchain-based Transaction Security

Use of blockchain: To ensure transparency and security in transactions. Every purchase made through the vending machine could be recorded in a decentralized ledger, preventing fraud and ensuring customers can verify their purchase history.

2. Solar-powered Vending Machines

Sustainability-driven: Vending machines powered by solar energy, ideal for outdoor environments or locations without easy access to electricity. This could be a unique feature for eco-conscious businesses or remote areas.

3. AI-driven Personalized Recommendations

Smart product suggestions: Using AI and machine learning, vending machines could analyze customer preferences based on past purchases or even on real-time data (like the weather or time of day) to recommend personalized products, improving the user experience.

4. Voice and Gesture-controlled Interfaces

Hands-free operation: For improved accessibility and hygiene, vending machines can incorporate voice commands or gesture recognition technology. This allows users to interact with the machine without physically touching it, which could be particularly appealing in healthcare, public places, or during post-pandemic times.

5. Recycling and Incentive Systems

Sustainable vending: Machines that offer rewards or discounts for users who deposit usedpackaging (like bottles or cans) in exchange for products or credits. This could be an effective way to encourage recycling while providing customers with additional value