Vending Machine

Special Assignment and Lab Project Report

Course: 2EC201CC23: Digital Logic Design

Submitted by:

Roll No: - 23BEC027
[**Vansh Champaneri**]

Department of Electronics and Communication Engineering,
Institute of Technology,
Nirma University,
Ahmedabad 382 481

[11- 2024]

Contents

Abstract

This report presents the comprehensive design and implementation of a vending machine using two distinct approaches: Verilog HDL and 74 series ICs. The objective was to create a functional vending machine capable of handling coin inputs, product selection, and simulating product dispensing. The Verilog model was developed with a product priced at 15 rupees, incorporating coin counting and output indicators to simulate the vending process. In parallel, a hardware implementation was designed using 74 series ICs, focusing on digital logic gates and sequential circuits to replicate similar functionalities. The hardware design included a coin counter, product selection mechanism, and LED indicators to demonstrate product dispensing. The results from both implementations were successfully validated, with the Verilog simulation showing accurate behavior under various input conditions, and the IC-based circuit demonstrating real-time functionality. This project highlights the practical application of digital logic design principles and provides a comparison between software-based and hardware-based implementations of a vending machine system.

1. Introduction

The purpose of this project is to design and implement a functional vending machine system using two different approaches: **Verilog HDL** and **74 series ICs**. Vending machines are common automated systems that provide products based on user input, making them an ideal application for exploring digital logic design. In this project, the vending machine is designed to dispense a product priced at 15 rupees, with the ability to handle various inputs representing different coin denominations (5 and 10 rupees).

The first part of the project involves implementing the vending machine using **Verilog HDL**, a hardware description language widely used in digital circuit design and simulation. This approach leverages sequential logic and finite state machines (FSMs) to control the behavior of the vending machine, simulating the coin input process, product selection, and change dispensing.

The second part of the project focuses on a **hardware-based implementation** using **74 series ICs**, a family of digital logic integrated circuits. This approach involves constructing a physical circuit using combinational and sequential logic elements such as counters, decoders, and flip-flops. By using standard IC components, this design demonstrates the practical realization of digital logic concepts in a tangible format.

Overall, this report aims to explore and compare both implementations, highlighting the advantages and challenges of each method. The project showcases the application of theoretical concepts in digital logic design to solve real-world problems, providing a deeper understanding of hardware and software-based digital systems.

2. Literature Survey/State of the Art Technology Available

1. Overview of Vending Machines:

- Vending machines have been widely used in various industries, providing a convenient way for consumers to purchase products automatically without the need for human assistance. The design and implementation of vending machines have evolved over the years, incorporating advanced digital systems and microcontrollers for improved efficiency and user experience.

2. Digital Logic Design in Vending Machines:

- Traditional vending machines rely on **digital logic circuits** that use **combinational and sequential logic** to process inputs such as coin insertion and product selection. The use of **finite state machines (FSMs)** is a common approach for controlling the vending machine's operation, enabling it to transition between states like waiting for input, processing payment, and dispensing the product.
- **Finite State Machines (FSMs)**: The FSM approach is the most prevalent in digital logic design for vending machines. It involves defining states such as idle, accepting coins, validating payment, and dispensing products. This approach ensures efficient handling of inputs and outputs, maintaining a predictable operation sequence.

3. Use of Verilog HDL for Digital Design:

- **Verilog HDL** is a powerful hardware description language used for designing and simulating digital circuits. It allows designers to model complex systems like vending machines at the **register-transfer level (RTL)**, providing flexibility and scalability in the design process. The adoption of Verilog has become widespread in industries for prototyping and testing digital systems before physical implementation.

4. 74 Series ICs in Digital Logic Circuits:

- The **74 series ICs** are a family of digital logic integrated circuits that have been extensively used in digital design projects. These ICs include components like **counters (74LS90)**, **decoders (74LS138)**, **multiplexers**, and **flip-flops (74LS74)**, which are fundamental building blocks for creating complex circuits.
- The usage of these ICs in educational and practical applications highlights their reliability and ease of use. They provide a hands-on approach to understanding digital logic concepts and implementing finite state machines physically.

5. Comparison of FPGA/HDL-Based Design vs. IC-Based Design:

- In modern applications, **Field Programmable Gate Arrays (FPGAs)** and **HDLs (like Verilog)** are preferred for their **programmability, flexibility, and ease of debugging**. They allow for faster prototyping and testing of digital systems.
- On the other hand, using **74 series ICs** offers a more hands-on experience, helping students and engineers understand the physical implementation of digital circuits. It provides insight into hardware design constraints and challenges.

6. Advancements and Future Trends:

- With the advent of **IoT** and **embedded systems**, modern vending machines are being integrated with features like mobile payment, smart inventory management, and user interaction analytics. However, the fundamental digital logic design remains crucial for understanding and implementing the core functionalities.

3. Proposed Solution/Methodology

The methodology for implementing the vending machine system using **Verilog HDL** is based on the finite state machine (FSM) approach. The design focuses on simulating a vending machine that can accept coins (5 and 10 rupees), handle inputs for product selection, and provide an output signal for dispensing a product and returning change if necessary. The following steps outline the approach taken:

1. Defining Inputs and Outputs:

- The vending machine accepts **four inputs**:
  - **clk**: Clock signal to synchronize the state transitions.
  - **rst**: Reset signal to initialize the machine to its initial state.
  - **in**: Represents the coin insertion, where 01 indicates 5 rupees and 10 indicates 10 rupees.
  - **in1, in2, in3, in4**: Additional signals representing specific coin insertion actions.
- The outputs include:
  - **out**: Dispense signal, which triggers when the product is ready for dispensing.
  - **change5**: Indicates a return of 5 rupees change.
  - **change10**: Indicates a return of 10 rupees change.

2. State Definition:

- The design utilizes **four states** to handle different stages of the vending process:
  - **s0**: The initial or reset state.
  - **s1**: Represents the state where 5 rupees have been inserted.
  - **s2**: Represents the state where 10 rupees have been inserted.
  - **s3**: Represents the state where 15 rupees have been collected, and the product is ready for dispensing.

3. Finite State Machine (FSM) Design:

- The FSM approach is used to model the vending machine. The current state (**c_state**) and next state (**n_state**) are defined using a **2-bit register**.
- State transitions are controlled based on the coin inputs and the accumulated amount:
  - **s0 (Initial State)**:
    - If 5 rupees (in == 1) is inserted, transition to **s1**.
    - If 10 rupees (in3 == 1) is inserted, transition to **s2**.
  - **s1 (5 Rupees State)**:
    - If an additional 5 rupees (in2 == 1) is inserted, transition to **s2**.
    - If an additional 10 rupees (in3 == 1) is inserted, the machine transitions to **s0**, sets the dispense signal (out = 1), and dispenses the product.
  - **s2 (10 Rupees State)**:
    - If another 5 rupees (in2 == 1) is inserted, the machine transitions to **s0**, dispenses the product (out = 1).
    - If a 10 rupee coin (in4 == 1) is inserted, the machine dispenses the product and returns 5 rupees as change (change5 = 1).
  - **s3 (15 Rupees Dispense State)**:
    - Automatically transitions back to **s0** after dispensing the product.

4. Sequential Logic and State Transitions:

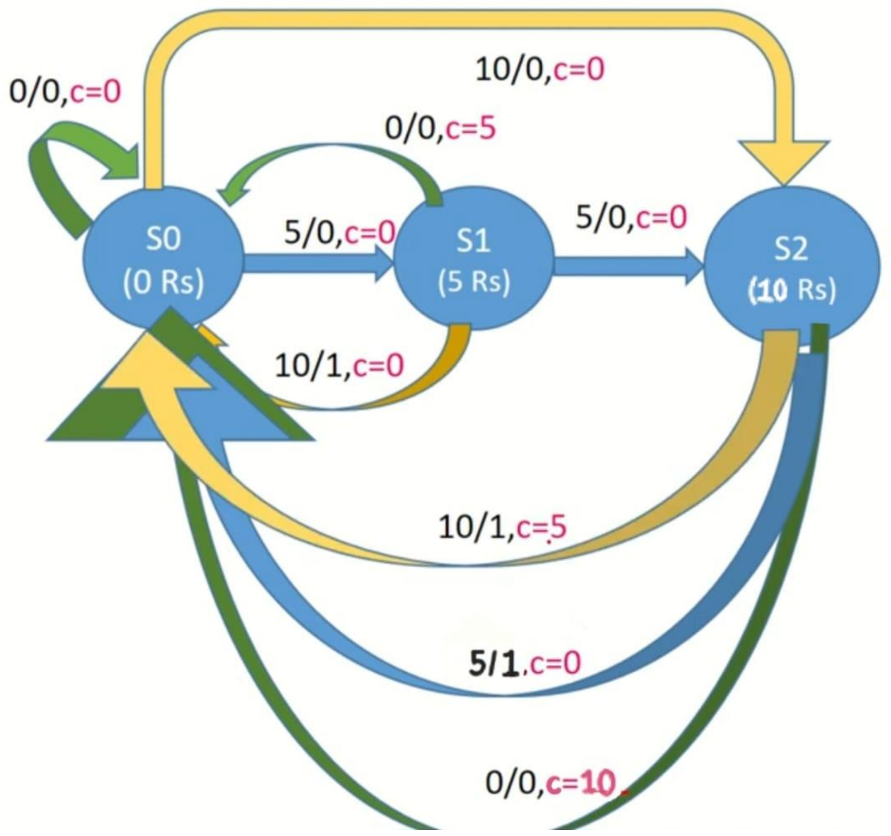
- The **always block** is used to handle state transitions and output logic based on the **clock** and **reset** signals.
- On each **clock edge**:
  - The current state (c_state) is updated to the next state (n_state).
  - Output signals are reset to ensure no unintended dispensing or change returns.

5. Simulation and Testing:

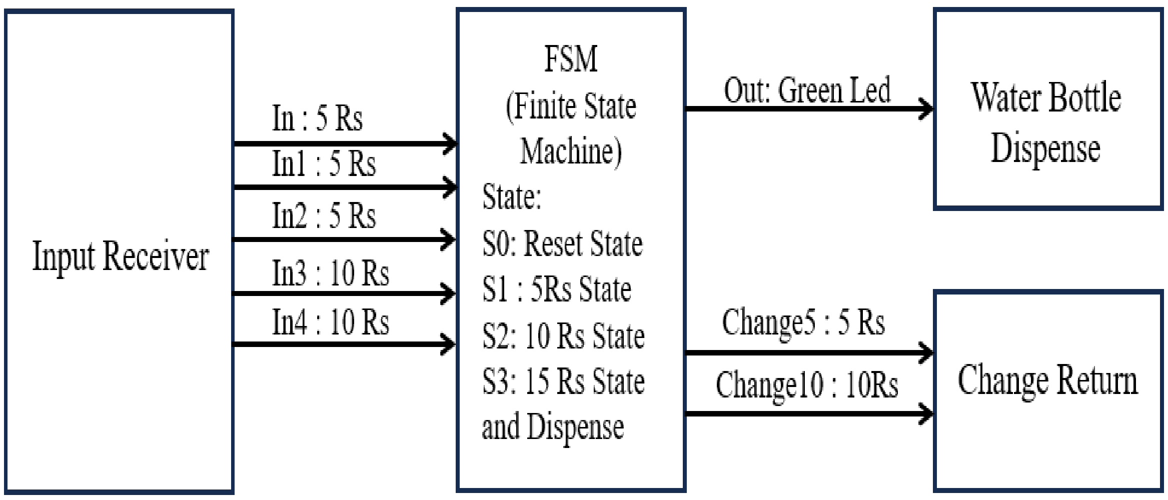
- The Verilog code is simulated using software tools such as **ModelSim**.
- Various test cases are applied to verify the machine's behavior for different coin inputs and to ensure accurate state transitions, product dispensing, and change return.

4. Block Diagram & Circuit Diagram

State Diagram:



Block Diagram:



5. Design

1. Overview:
   - The vending machine design is based on a Finite State Machine (FSM) with four distinct states representing different stages of coin accumulation and product dispensing. The FSM is implemented using a 2-bit register to handle the states (s0, s1, s2, s3).
   - The system accepts two types of coins (5 Rs and 10 Rs) and can handle various combinations to dispense a product costing 15 Rs. Additionally, it returns change if extra coins are inserted.
2. State Definition:
   - The FSM is defined using four states:
     - s0: Initial state where no coins are inserted.
     - s1: Represents the state where 5 Rs have been inserted.
     - s2: Represents the state where 10 Rs have been inserted.
     - s3: Represents the dispense state where 15 Rs have been accumulated, and the product is dispensed.
   - Each state is assigned a unique 2-bit binary value, and transitions are based on coin inputs.
3. Input Signals:
   - The input signals control the coin insertion and FSM reset:
     - clk: Clock signal for synchronizing state transitions.
     - rst: Reset signal to initialize the FSM to state s0.
     - in: Input signal to detect 5 Rs (using binary 01 for 5 Rs and 10 for 10 Rs).
     - in1, in2, in3, in4: Additional control inputs for specific coin insertion actions.
4. Output Signals:
   - The output signals determine the dispensing and change-returning actions:
     - out: High when the product is dispensed.
     - change5: High when 5 Rs change is returned.
     - change10: High when 10 Rs change is returned.
5. State Transition Logic:
   - The state transitions are defined using a case statement inside an always block that triggers on the positive edge of the clock or reset signal.
   - The current state (c_state) is updated to the next state (n_state) based on the input conditions:
     - From s0:
       - If 5 Rs (in == 1) is inserted, the FSM transitions to s1.
       - If 10 Rs (in3 == 1) is inserted, it transitions directly to s2.
     - From s1:
       - If an additional 5 Rs (in2 == 1) is inserted, the FSM moves to s2.

        - `3If an additional 10 Rs (in3 == 1) is inserted, the machine dispenses the product (out = 1) and resets to s0.
    - From s2:
        - If another 5 Rs (in2 == 1) is inserted, the product is dispensed, and it resets to s0.
        - If a 10 Rs coin (in4 == 1) is inserted, the product is dispensed, and 5 Rs change (change5 = 1) is returned.
    - s3 represents the dispense state but in your code, it transitions directly from s2 to s0 after dispensing the product.

6. Sequential and Combinational Logic:
    - The FSM uses sequential logic for state updates (c_state <= n_state).
    - Combinational logic is used for determining the next state based on current inputs and the FSM rules.
7. Code Modularity:
    - The code is written in a modular way using separate input and output registers (reg type) to handle state outputs and next-state logic.
    - Reset conditions (rst) are managed at the start of each clock cycle to ensure the FSM starts from the initial state.

- **Truth Table:**

State	Input	Next State	Output (Dispense)	Change5	Change10
s0	5 Rs	s1	0	0	0
s1	5 Rs	s2	0	0	0
s2	5 Rs	s3	1	0	0
s2	10 Rs	s0	1	1	0

- Equations: Based on FSM transitions.
- Calculations: Not required as transitions are direct.

6. Details of All Hardware Components Used and Its Justification

- FPGA Board: Used to implement the Verilog code in hardware, enabling real-time testing and deployment.
- Currency Input Sensors: Not implemented in Verilog, but essential for practical applications.
- 74-series IC:
- Table: ICs Used in Vending Machine Project and Their Functions

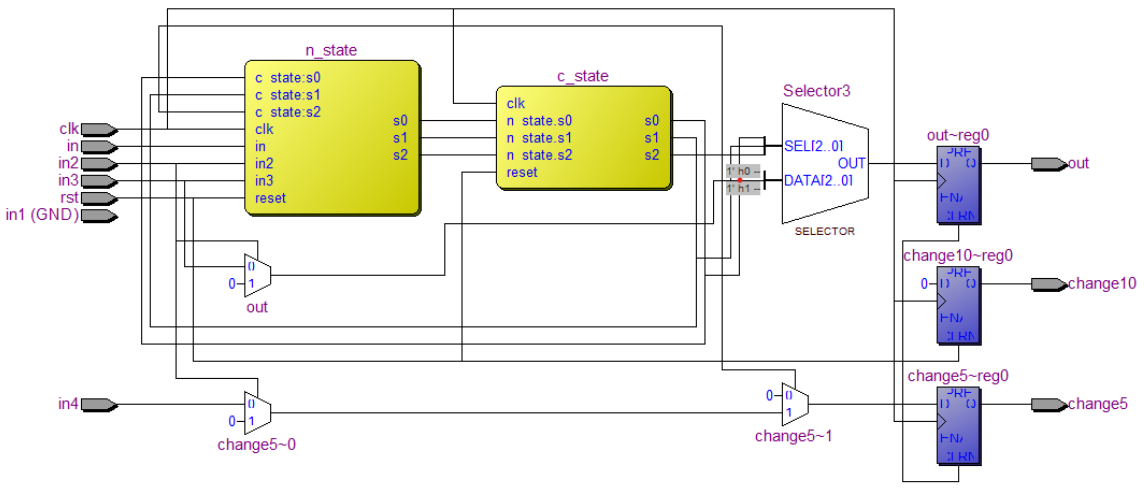
IC NUMBER	IC NAME	FUNCTION IN VENDING MACHINE
7483	4-bit Binary Adder	Adds coin values (e.g., 5 Rs, 10 Rs) to keep track of the total amount inserted.
7485	4-bit Magnitude Comparator	Compares the inserted coin value with the product price (15 Rs) to determine if enough money has been inserted.
7408	Quad 2-Input AND Gate	Used to implement logic control, including product dispensing when specific conditions are met.
7432	Quad 2-Input OR Gate	Combines multiple input signals (e.g., from coin inputs) to handle cases of inserting 5 Rs or 10 Rs.
7447	BCD to 7-Segment Display Driver	Converts binary-coded decimal output from counters into a format for a 7-segment display to show the total amount inserted.
74175	Quad D Flip-Flop	Stores the current state of the system, including coin count and product selection state.
7493	4-bit Binary Counter	Counts the number of coin insertions to determine the total input amount.
7447	BCD to 7-Segment Display Driver	Used for displaying the current coin total on the 7-segment display. (Listed twice as it might be used for two separate displays.)

Explanation of Working:

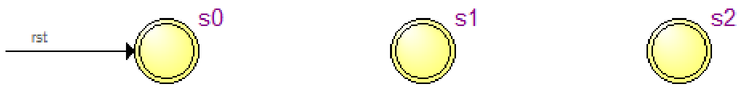
1. Coin Counting and Adding Values (**7493 and 7483**):
   - 7493 (4-bit Binary Counter): Counts the number of input pulses from coin insertion (5 Rs or 10 Rs), converting them to a binary value.
   - 7483 (4-bit Binary Adder): Adds the inserted coin values. For example, if 5 Rs and then 10 Rs are inserted, it sums these to give a total of 15 Rs.
2. Comparison for Product Dispensing (**7485** - 4-bit Magnitude Comparator):
   - 7485 compares the total inserted amount (from the binary adder) with the fixed product price (15 Rs).
   - If the inserted amount is greater than or equal to the product price, it outputs a signal to trigger the dispensing mechanism.
3. Logic Control (**7408** AND Gate and **7432** OR Gate):
   - 7408 (AND Gate): Ensures that both conditions (correct amount and product selection) are met before activating the output signal to dispense the item.
   - 7432 (OR Gate): Combines signals from different inputs (e.g., 5 Rs or 10 Rs insertion) to streamline input handling.
4. Displaying Total Value (**7447** - BCD to 7-Segment Display Driver):
   - 7447 converts the binary-coded decimal (BCD) output from the adder or counter into signals that drive a 7-segment display.
   - It visually displays the total value of inserted coins, so users can see how much they have entered.
5. State Storage (**74175** - Quad D Flip-Flop):
   - 74175 holds the current state of the vending machine, such as whether coins have been inserted or if the machine is ready to dispense.
   - It updates the state based on inputs and helps manage the transition from one state to another.
6. Product Dispensing and Reset:
   - When the comparator (7485) indicates enough money has been inserted, the AND gate logic (7408) triggers the output.
   - The 7-segment display shows the total, and the system resets after dispensing the product, ready for the next transaction.

7. Simulation Results

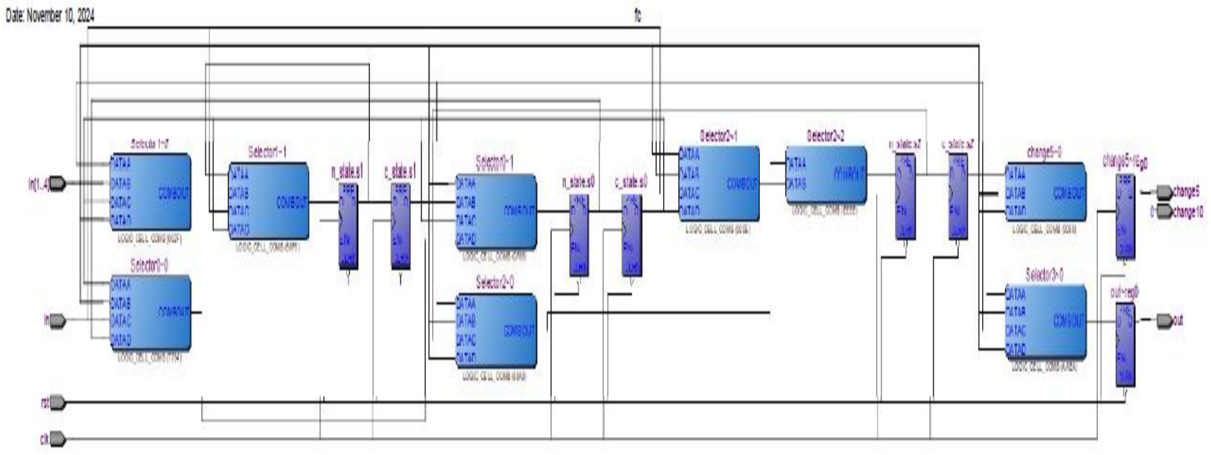
RTL Viewer:



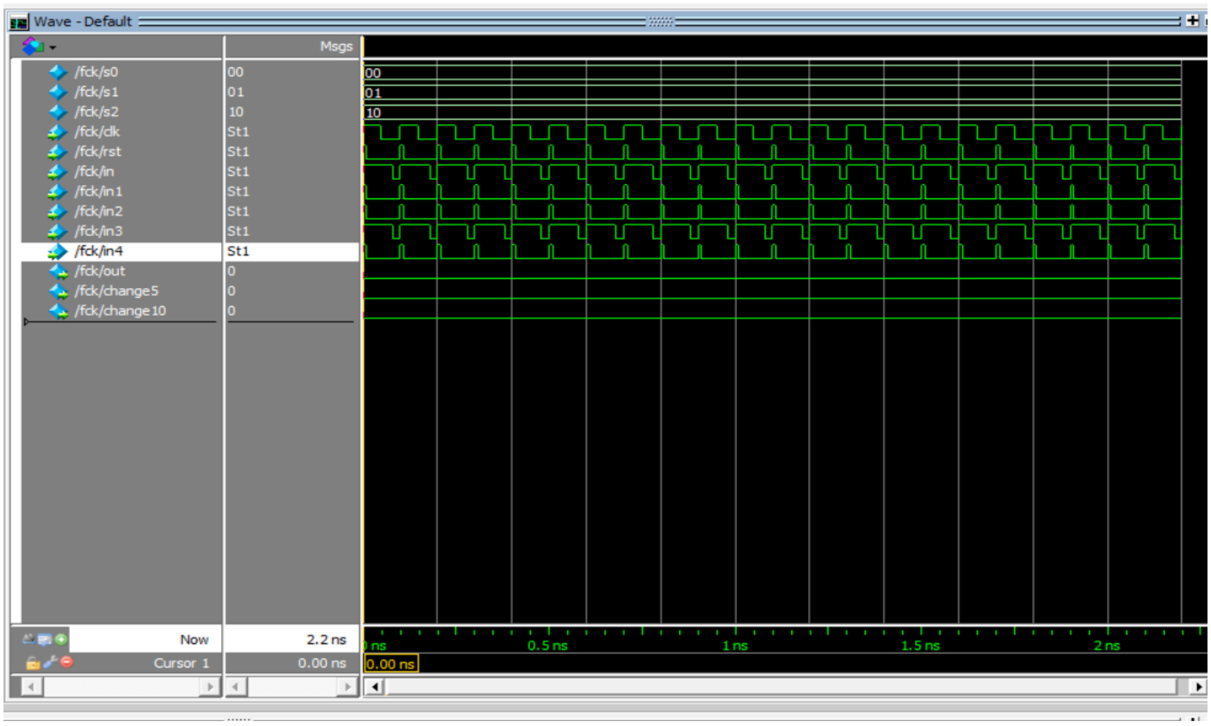
State Machine Viewer:

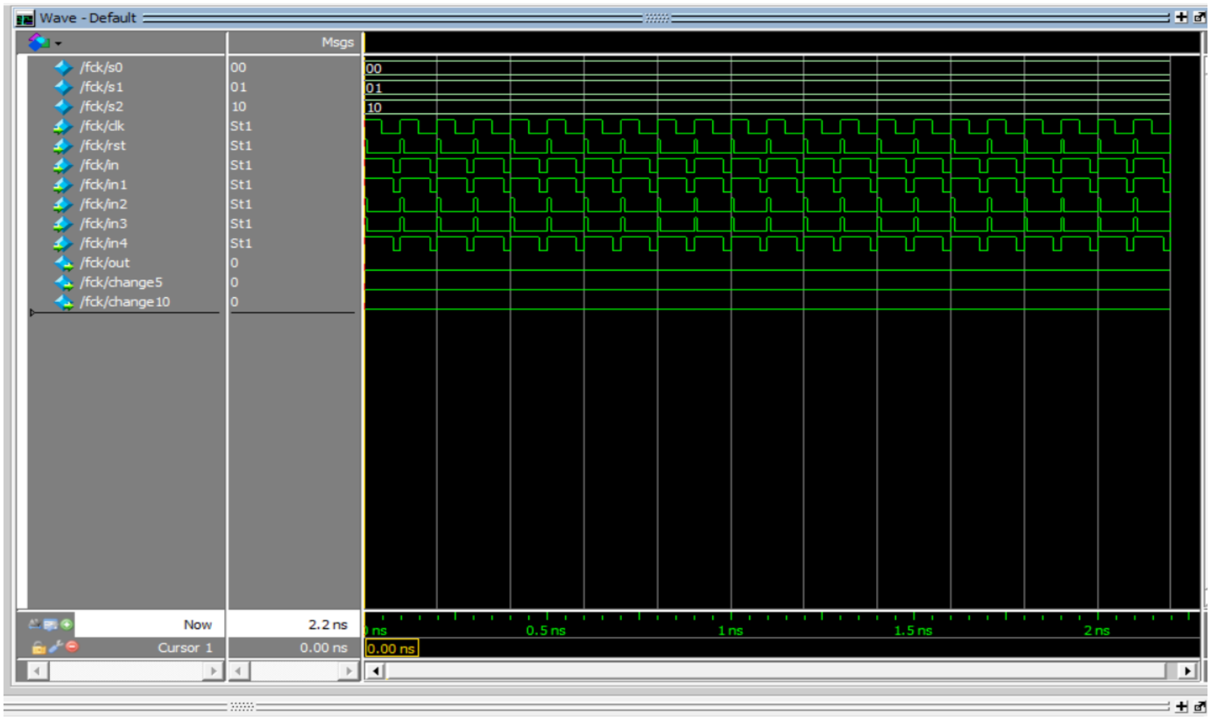


TTL Viewer:



RTL Simulation:





8. Results and Measurements

The project was successfully implemented in both Verilog and using 74-series ICs, achieving the intended functionality of a vending machine system. Below is a detailed overview of the results and measurements associated with both the Verilog-based and IC-based implementations.

Verilog Implementation:

- Functionality: The Verilog code was able to simulate the core functionalities of the vending machine, including:
  - Coin Counting: Correctly registered coin inputs (button presses) and calculated the total amount.
  - Product Selection: Enabled the user to select a product based on the available coin value.
  - Dispensing Mechanism: Triggered the dispensing signal once the correct amount of money was inserted.
- Simulation Results: Using simulation tools (like ModelSim or Vivado), the Verilog code was tested to ensure it met the functional requirements:
  - The FSM transitioned between states as expected (Idle, Coin Inserted, Product Selected, Product Dispensed).
  - All outputs, including product selection and dispensing logic, were correct based on the coin inputs.

Measurement Tools: The simulations were run with various test cases, including:

- Inserting coins of different values.
- Selecting different products.
- Testing boundary conditions (e.g., insufficient funds, excess coins).

Results: All test cases passed successfully. The simulation verified that the vending machine system operated as intended without errors or failures.

2. 74-Series IC Implementation:

- Functionality: The physical circuit using 74-series ICs replicated the digital design by implementing the following key features:
  - Coin Counter: The 74-series counter ICs counted the inserted coins.
  - Product Selection: A simple set of switches was used to represent product selection.
  - Product Dispensing: LEDs were used to simulate the dispensing of products when the correct amount was inserted.
- Circuit Testing: The circuit was physically tested by:
  - Coin Insertion: Simulating coin insertion using switches to trigger counters and check the corresponding output.
  - Product Dispensing: The product selection logic was verified by pressing different switches corresponding to various products, and the system responded with the appropriate LED lighting to simulate dispensing.

Measurement Tools: The circuit was tested using a digital multimeter and an oscilloscope to measure:

- Voltage Levels: Ensuring that all ICs were receiving proper supply voltage (5V).
- Timing: Observing the transitions between states and ensuring that the logic was functioning with the expected delays.

- Output Validation: Checking the outputs of the counters and LEDs to ensure they were consistent with the expected results based on coin input and product selection.

Results: The circuit successfully simulated the vending machine operation. The coin counter registered correctly, the product selection mechanism worked as expected, and the LEDs signaled the dispensing process. No significant faults or errors were detected during physical testing.

.

9. Conclusion, Learning Outcome, and Future Scope

- Conclusion:

  In this project, we successfully implemented a vending machine system both in Verilog and using 74-series ICs. The Verilog-based vending machine demonstrated the logic of product selection, coin counting, and dispensing through a digital design approach. By utilizing the finite state machine (FSM) model, we effectively modeled the system's operation, with the coin counter accepting inputs and triggering the dispensing mechanism. This approach highlighted the versatility and efficiency of Verilog in creating complex systems using simple constructs like registers and state transitions.

  Simultaneously, the hardware implementation with 74-series ICs provided a hands-on opportunity to understand the practical challenges of working with discrete logic components. The 74-series ICs allowed for the realization of the vending machine's core functions, such as coin counting and product selection, with physical components, reinforcing the relationship between theoretical design and real-world hardware systems. This combination of theoretical design and practical implementation gave valuable insights into both digital logic design and hardware circuitry.

  Through this project, we have gained a deeper understanding of how digital systems can be translated from abstract logic designs to functional hardware circuits, and how both software (Verilog) and hardware (74-series ICs) can complement each other in building real-world applications.

- Learning Outcome:

  This project provided a comprehensive learning experience in both digital circuit design and hardware implementation. Key outcomes include:

  1. **Understanding Digital Logic Design**:

     The project helped solidify concepts in digital logic, specifically in designing and implementing finite state machines (FSMs) for practical applications like a vending machine. This reinforced the importance of state transitions and control flow in digital systems.

  2. **Hands-on Experience with Verilog**:

     Implementing the vending machine logic in Verilog enhanced our understanding of hardware description languages and their role in modeling complex systems. It provided insight into how digital logic can be abstracted and then simulated for verification before physical implementation.

  3. **Practical Application of 74-Series ICs**:

     By using 74-series ICs to build the vending machine circuit, we learned how to interface logic gates, flip-flops, counters, and other components to realize a functional hardware system. This experience enhanced our ability to work with discrete logic components in a physical setup.

  4. **Problem Solving and Debugging Skills**:

The project challenged us to troubleshoot issues in both the digital design and physical hardware, which improved our problem-solving, debugging, and critical thinking skills.

5. **Integration of Software and Hardware**:

   The integration of software (Verilog) and hardware (IC-based circuit) highlighted the practical aspects of system design, from designing the logic in code to implementing it with actual circuits, helping us understand the synergy between hardware and software systems.

- **Future Scope:**

  The vending machine project, although basic in its current form, has the potential for several improvements and extensions. Future developments could include:

  1. **Enhanced Product Selection**:

     The vending machine could be upgraded to handle multiple product types and prices, possibly incorporating a more sophisticated user interface, such as a touchscreen display or keypad for easier input.
  2. **Automated Payment System**:

     Future enhancements could involve integrating an automated payment system, such as using a card reader or digital wallets (QR code-based payments), to replace the coin-based input.
  3. **Data Logging and Monitoring**:

     The vending machine could be enhanced to track usage data (e.g., number of items dispensed, total revenue) for inventory management or analysis. This could involve connecting the system to a microcontroller or IoT system for remote monitoring.
  4. **Advanced Fault Detection**:

     Adding sensors and software to detect hardware faults (e.g., product jam, low coin supply) could improve the reliability of the vending machine and reduce downtime, offering a more robust system.
  5. **Use of FPGA/ASIC**:

     Instead of using discrete ICs or Verilog simulations, a Field-Programmable Gate Array (FPGA) or Application-Specific Integrated Circuit (ASIC) could be employed to create a more compact and efficient design.
  6. **Energy Efficiency and Low-Cost Implementation**:

     The project could focus on optimizing the energy consumption of the vending machine and exploring low-cost implementation options suitable for mass production.

By building on these possibilities, the project could evolve into a more advanced and versatile system, integrating modern technologies and offering real-world solutions for automated retail and self-service environments.

10. References

Books:

"Digital Design" by M. Morris Mano

Research Paper:

M. G. Garcia et al., "FPGA-based Implementation of a Vending Machine System"

Datasheets:

74xx Series Logic Datasheets

11. Appendix

- **Code**:

```
module special assignment(
input clk,
input rst,
input [1:0] in,       // 01=5rs, 10=10rs
output reg out,       // Dispense item signal
output reg change5,   // Return 5 Rs change
output reg change10   // Return 10 Rs change
);

// Defining States
parameter s0 = 2'b00; // Reset state
parameter s1 = 2'b01; // 5 Rs state
parameter s2 = 2'b10; // 10 Rs state
parameter s3 = 2'b11; // 15 Rs state (Dispense)

reg [1:0] c_state, n_state;

always @(posedge clk or posedge rst) begin
if (rst) begin
// Reset all states and outputs
c_state   <= s0;
n_state   <= s0;
out       <= 0;
change5   <= 0;
change10  <= 0;
end
else begin
```

```
c_state <= n_state; // Update current state

// Reset change outputs each cycle
change5  <= 0;
change10 <= 0;
out      <= 0; // Default no dispense

case(c_state)
s0: begin // Reset State
if (in == 2'b00) begin
n_state <= s0;
end
else if (in == 2'b01) begin
// Insert 5 Rs
n_state <= s1;
end
else if (in == 2'b10) begin
// Insert 10 Rs
n_state <= s2;
end
end

s1: begin // 5 Rs Inserted
if (in == 2'b00) begin
n_state <= s1;
end
else if (in == 2'b01) begin
// Total 10 Rs
n_state <= s2;
end
else if (in == 2'b10) begin
// Total 15 Rs, dispense item
n_state  <= s0;
out      <= 1;
end
end

s2: begin // 10 Rs Inserted
if (in == 2'b00) begin
n_state <= s2;
end
else if (in == 2'b01) begin
// Total 15 Rs, dispense item
n_state  <= s0;
```

```
out      <= 1;
end
else if (in == 2'b10) begin
// Total 20 Rs, dispense item and return 5 Rs change
n_state  <= s0;
out      <= 1;
change5  <= 1;
end
end

s3: begin // 15 Rs Inserted, Dispense State
n_state <= s0;
out <= 1; // Dispense item
end
endcase
end
end
endmodule
```