BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

EEE 304

Digital Electronics Laboratory

Final Project Report

Section: B2 Group: 08

Design and Implementation of FPGA-based Vending Machine using Verilog

Course Instructors:
1. Nafis Sadik, Lecturer, Dept. of EEE, BUET
2. Rafichha Yasmin, PT, Dept. of EEE, BUET

Academic Honesty Statement:

"In signing this statement, We hereby certify that the work on this project is our own and that we have not copied the work of any other students (past or present), and cited all relevant sources while completing this project. We understand that if we fail to honor this agreement, We will each receive a score of ZERO for this project and be subject to failure of this course."

Signature: ____________ **Full Name: Nilanjona Majumder Medha** **Student ID: 2006100**	**Signature:** ____________ **Full Name: Shanta Sarker** **Student ID:2006103**
Signature: ____________ **Full Name: Sheikh Munkasir Ahmed Rafeed** **Student ID: 2006109**	**Signature:** ____________ **Full Name: Arpon Kumar Chanda** **Student ID: 2006110**

Table of Contents

1. Abstract:

The primary objective of this project is to design and implement a vending machine capable of dispensing products when the appropriate combination of cash is inserted. The vending machine will also provide change if the cash inserted exceeds the price of the selected product. A unique feature of this vending machine is its ability to allow users to specify a price range for the products they wish to purchase, enhancing its usability and flexibility.

A vending machine is an automated device that is commonly used to dispense commodities, including snacks and beverages, upon receiving payment. The efficiency, flexibility, and functionality of vending machines are improved by the incorporation of contemporary technology like Field Programmable Gate Arrays (FPGAs). Our project focuses on creating and implementing an FPGA-based vending machine with features including dynamic change computation, accurate cash management, and customizable product selection. The system will be implemented using a finite-state machine (FSM) on a field-programmable gate array (FPGA). The FSM controls the operation of the vending machine, including accepting coins, dispensing items, and providing cash returns to the user. We will implement a vending machine which is a Mealy-type FSM whose outputs depend on both the states and the primary inputs.

2. Introduction:

The finite state machine (FSM) approach is adopted for the design of vending machine. The design is achieved by formulating the Verilog code for the FSM-based machine. Finite State Machine (FSM) manages transitions between states based on user inputs and system events. The FSM ensures the orderly operation of the machine, handling cash inputs, dispensing products, and calculating change with accuracy.

<u>Specifications for our vending machine:</u>

Products

- Product 1: price 10 TK
- Product 2: price 15 TK
- Product 3: price 20 TK
- Product 4: price 25 TK

Input amounts

- 5 TK
- 10 TK
- 20 TK

One will choose one of the four products at a time. By pressing the push button, he can supply money in any combination of the three input amounts shown above. The machine will dispense the chosen product as soon as the total of the combinations equals or exceeds the price, and it will return any excess funds if the amount supplied exceeds the price. After purchasing any product, the machine will return to its initial state.

3.Design

3.1 Problem Formulation (PO(b))

3.1.1 Identification of Scope

We used the FSM technique we learned in our theory classes to accomplish our project.

3.1.2 Literature Review

Several studies have been conducted to design the vending machines. Below is a discussion of some of them. In the paper, “Design and implementation of vending machine using Verilog” [3], the FSM approach has been adopted but, in that case, the machine accepts coins of denominations five and ten TK. Also, whenever change is not available in the machine, it returns the total amount. On the other hand, we have designed a vending machine that can take more cash in, and after cash in the machine waits for another input cash in if the previously entered money is less than the product price. After a less amount of cash in if we don’t enter any cash in the next clock cycle then our machine returns the previously entered money. If the inserted cash exceeds the price of the product, the machine will dispense the product and return the excess amount.
In the paper "Implementation of FPGA-based smart vending machine."[4] the implementation of a vending machine utilizing the Finite State Machine (FSM) Model is suggested using VHDL. The state processes—such as user selection, waiting for money to be inserted, product delivery, and services are modelled in this project using the MEALY Machine Model.

3.1.3 Formulation of Problem

The target of the project is to design a vending machine that allows a user to purchase a product by following these steps:

<u>Product Selection:</u>
The user selects a product through a switch in this project. The selection determines the price of the desired product, which is processed by the system.
<u>Payment Input:</u>
The user inputs the required payment through push buttons in this project, simulating cash denominations (e.g., 5, 10, or 20 currency units). This step mimics the insertion of physical currency or digital payment in a real-world machine.
<u>LED Feedback and Cash Return:</u>
After receiving the payment, the LED corresponding to the selected product glow, indicating that the product is dispensed. If the input payment exceeds the product price, the machine calculates the excess amount and returns it to the user, which is displayed via an LED feedback mechanism for cash return.

3.1.4 Analysis

We have 2 types of inputs:

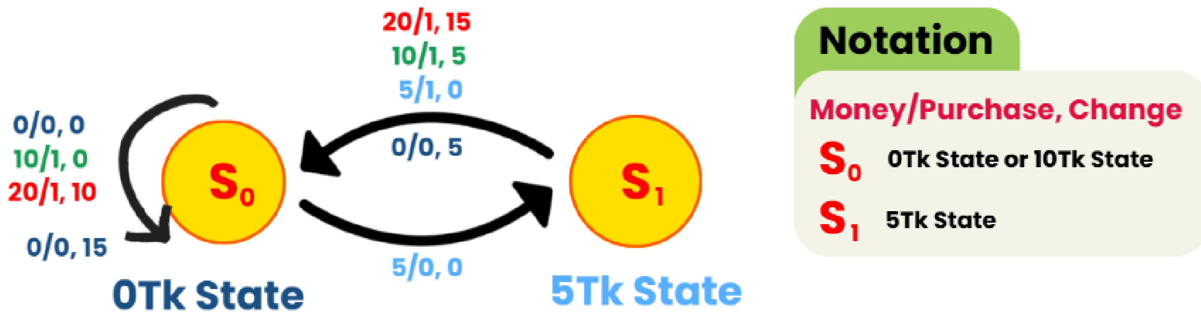
1. Product of how much money we have to select
2. How much money a consumer can pay

We have 2 types of considerable outputs:

1. Product is purchased or not
2. How much money the vending machine return to the consumer.

3.2 Design Method (PO(a))

We declared a total of 5 states for 0,5,10,15,20 taka. We designed our machine such that it can receive only 5,10 and 20 taka notes. It can return 5,10,15 and 20 according to the product's price and the amount the user gave.
Using our knowledge from EEE303, we designed the state diagram for each product and made their state table and state assigned table so implement them in Verilog code. As for each input, our output is found different, it has become a mealy type FSM module.
State diagram for 10 taka product:



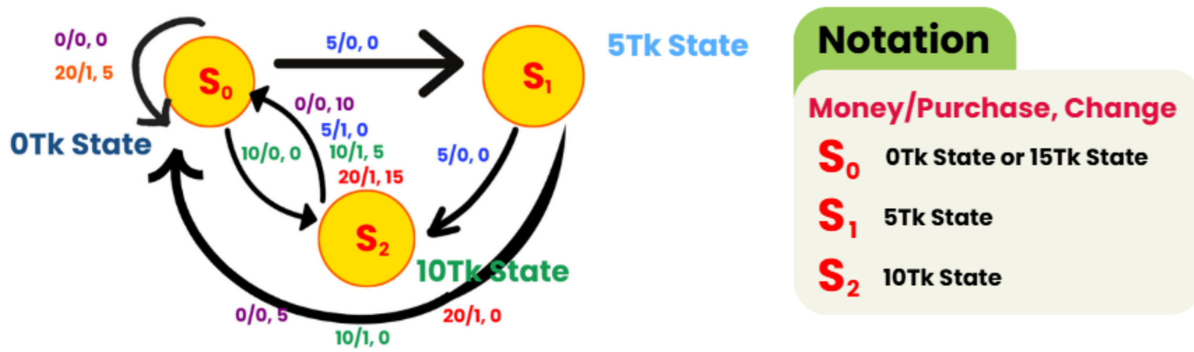
State table for 10 taka product:

Present state	Next state				Purchase				Cash return			
	0 tk	5 tk	10 tk	20 tk	0 tk	5 tk	10 tk	20 tk	0 tk	5 tk	10 tk	20 tk
State0 //0taka	S0	S1	S0	S0	N	N	Y	Y	0	0	0	10
State1 //5taka	S0	S0	S0	S0	N	Y	Y	Y	5	0	5	15

State assigned table for 10 taka product:

Present state	Next state				Purchase				Cash return			
	00	01	10	11	00	01	10	11	00	01	10	11
000 //0tk	000	001	000	000	0	0	1	1	000	000	000	010
001 //5tk	000	000	000	000	0	1	1	1	001	000	001	011

State diagram for 15 taka product:



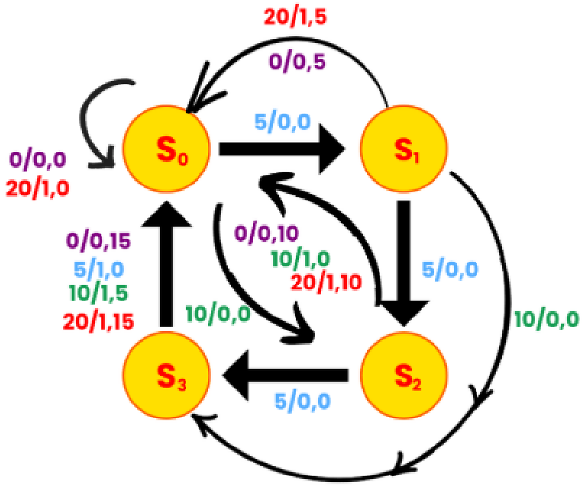
State table for 15 taka product:

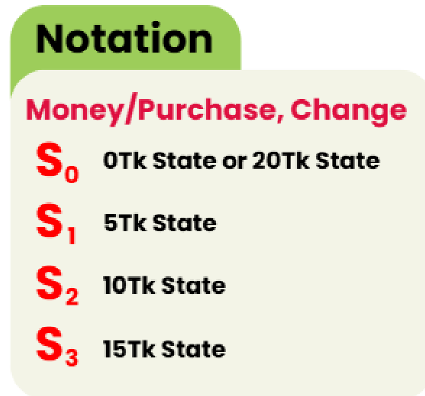
Present state	Next state				Purchase				Cash return			
	0 tk	5 tk	10 tk	20 tk	0 tk	5 tk	10 tk	20 tk	0 tk	5 tk	10 tk	20 tk
State0 //0taka	S0	S1	S2	S0	N	N	N	Y	0	0	0	5
State1 //5taka	S0	S2	S0	S0	N	N	Y	Y	5	0	0	10
State2 //10taka	S0	S0	S0	S0	N	Y	Y	Y	10	0	5	15

State assigned table for 15 taka product:

Present state	Next state				Purchase				Cash return			
	00	01	10	11	00	01	10	11	00	01	10	11
000 //0tk	000	001	010	000	0	0	0	1	000	000	000	001
001 //5tk	000	010	000	000	0	0	1	1	001	000	000	010
010 //10tk	000	000	000	000	0	1	1	1	010	000	001	011

State diagram for 20 taka product:





State table for 20 taka product:

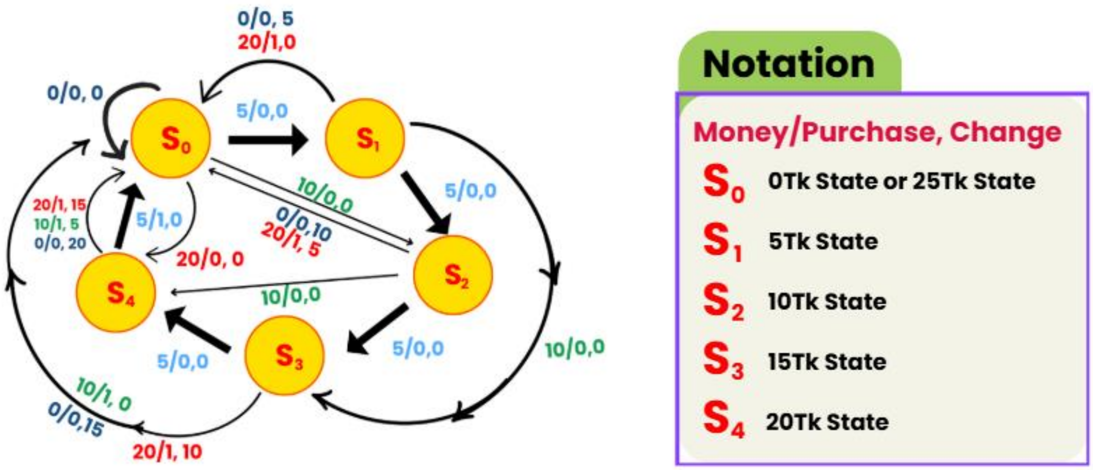
Present state	Next state				Purchase				Cash return			
	0 tk	5 tk	10 tk	20 tk	0 tk	5 tk	10 tk	20 tk	0 tk	5 tk	10 tk	20 tk
State0 //0taka	S0	S1	S2	S0	N	N	N	Y	0	0	0	0
State1 //5taka	S0	S2	S3	S0	N	N	N	Y	5	0	0	5
State2 //10taka	S0	S3	S0	S0	N	N	Y	Y	10	0	0	10
State3 //15taka	S0	S0	S0	S0	N	Y	Y	Y	15	0	5	15

State assigned table for 20 taka product:

Present state	Next state				Purchase				Cash return			
	00	01	10	11	00	01	10	11	00	01	10	11
000 //0tk	00 0	001	010	000	0	0	0	1	000	000	000	000
001/5 //5tk	00 0	010	011	000	0	0	0	1	001	000	000	001
010 //10tk	00 0	011	000	000	0	0	1	1	010	000	000	010

011 //15tk	000	000	000	000	0	1	1	1	011	000	001	011

State diagram for 25 taka product:



State table for 25 taka product:

Present state	Next state				Purchase				Cash return			
	0 tk	5 tk	10 tk	20 tk	0 tk	5 tk	10 tk	20 tk	0 tk	5 tk	10 tk	20 tk
State0 //0taka	S0	S1	S2	S4	N	N	N	N	0	0	0	0
State1 //5taka	S0	S2	S3	S0	N	N	N	Y	0	0	0	0
State2 //10taka	S0	S3	S4	S0	N	N	N	Y	10	0	0	5
State3 //15taka	S0	S4	S0	S0	N	N	Y	Y	15	0	0	10
State4 //20taka	S0	S0	S0	S0	N	Y	Y	Y	20	0	5	15

State assigned table for 25 taka product:

Present state	Next state				Purchase				Cash return			
	00	01	10	11	00	01	10	11	00	01	10	11
000 //0tk	000	001	010	S4	0	0	0	0	000	000	000	000
001 //5tk	000	010	011	S0	0	0	0	1	000	000	000	000
010 //10tk	000	011	100	S0	0	0	0	1	010	000	000	001
011 //15tk	000	100	000	S0	0	0	1	1	011	000	000	010
100 //20tk	000	000	000	S0	0	1	1	1	100	000	001	011

3.3 Circuit Diagram

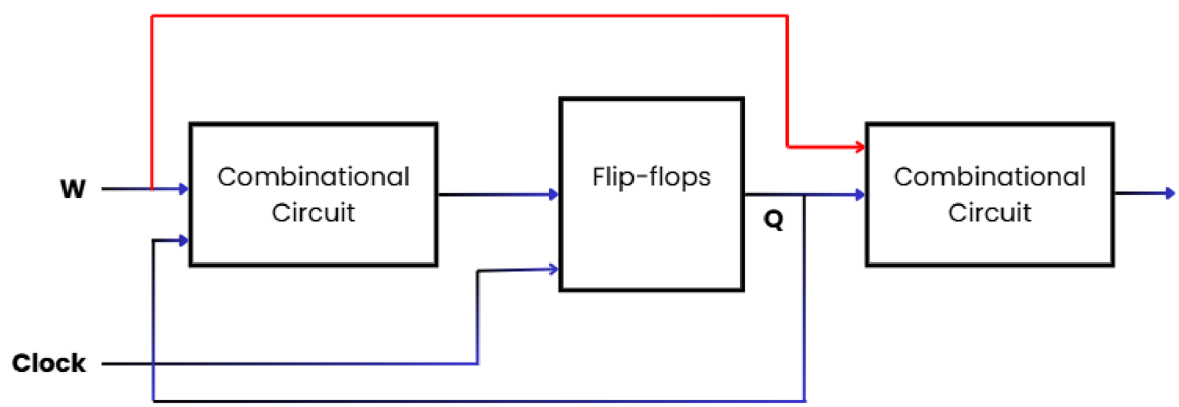
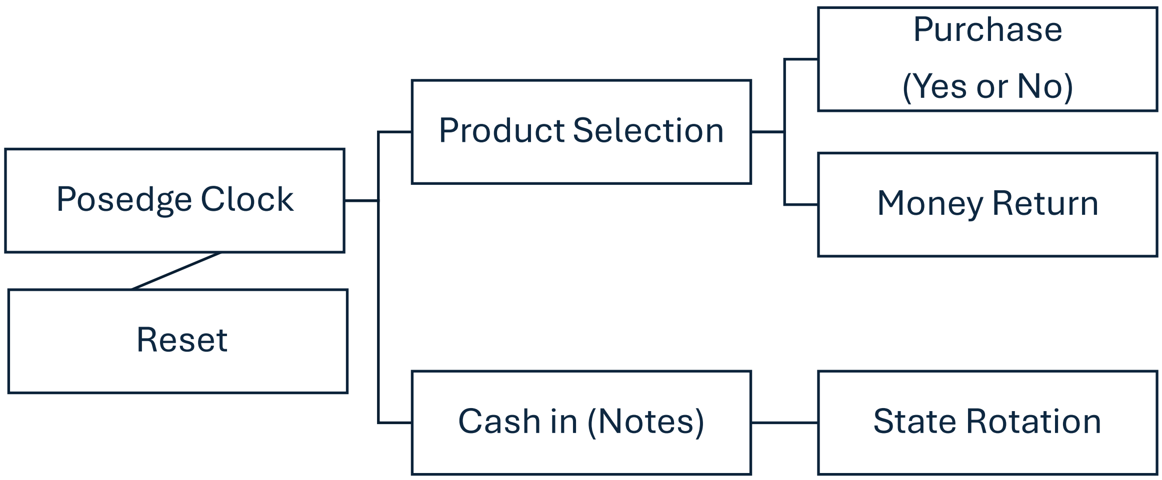


Fig: Mealy type FSM sequential circuit schematic

Using the encoded state table we could construct the combinational logic relation between all of the inputs and outputs by converting them into K map. In our project, we didn't need any hardware implementation to avoid complications and that's why we skipped this part for further scope.

3.4 Simulation Model



3.5 Hardware Design

We didn’t use any extra hardware setup to accomplish this project. After writing appropriate code in Quatus the Lab’s PC, we assigned push button and LEDs of the CPLD board from the computer and displayed expected outputs as shown above.

3.6 Full Source Code of Firmware

vending_machine.v*

```
module vending(clock, reset, cash_in, purchase, present_state, next_state, cash_return,LED_COM,N);
    input clock, reset;
    input [1:0] cash_in;
    input [2:0]N;
    inout LED_COM;
    assign LED_COM=1;
    output reg purchase;
    reg n;
    output reg [2:0] present_state, next_state;
    output reg [2:0]  cash_return;
    parameter state0 = 3'b000,   //0tkfinal state
              state1 = 3'b001,   // 5tk state
              state2 = 3'b010,   //10tk state
              state3 = 3'b011,   //15tk state
              state4 = 3'b100;   //20tk state
            //  state4=3'b101;    //25 taka state

              // cash in : 2'b00=0 taka,2'b01=5taka,2'b10=10 taka ,2'b11=20taka
            //cash return :2'b000=0 taka,2'b001=5 taka,2'b010=10 taka,2'b011=15 taka,2'b100=20 taka


    always @(posedge clock)
    begin
    if(reset==1)
    begin
                present_state = 0;
                next_state = 0;
                purchase = 0;
                cash_return = 0;
            end
            else
            begin
    /*case (N)
    tenn==10;
   fifteenn==15;
    twentyn==20;
   endcase*/
        if(N==3'b010)  // 10taka prouduct
        begin
            if(reset == 1)
            begin
                present_state = 0;
                next_state = 0;
            end
            else
            begin
                    present_state = next_state;
                    case(present_state)
                    state0: if(cash_in == 2'b00)
                                begin
                                    next_state = state0;
                                    purchase = 0;
                                    cash_return = 0;
                                end
                            else if(cash_in == 2'b01)
                                begin
                                    next_state = state1;
                                    purchase = 0;
                                    cash_return = 0;
                                end
                            else if(cash_in == 2'b10)
                                begin
                                    next_state = state0;
                                    purchase = 1;
                                    cash_return = 0;
                                end
                            else if(cash_in == 2'b11) //cash in 20 taka
                                begin
                                    next_state = state0;
                                    purchase = 1;
                                    cash_return = 3'b010; //return 10 taka
                                end

                    state1:if(cash_in == 2'b00) // 5 taka state
                                begin
                                    next_state = state0;
                                    purchase = 0;
                                    cash_return = 3'b001;
                                end
```

(Remaining added in the GITHUB link)

4. Implementation

4.1 Description

In this project, we designed and implemented a vending machine system using Verilog in Quartus, with the functionality demonstrated on an FPGA board. The vending machine operates based on a finite-state machine (FSM), which efficiently manages key processes, including coin acceptance, product dispensing, and user feedback.

The machine's states are visually represented using LEDs, and switches are utilized for input, reset, and clock control. A successful purchase is indicated by the illumination of a green LED, while the return of change for excess cash is displayed using a sequence of LEDs. These LED patterns represent binary sequences corresponding to the machine's states and the amount of change returned, as detailed further in the user manual.

The implementation demonstrates the potential of FSMs in creating efficient and user-friendly systems on FPGA platforms.

5. Design Analysis and Evaluation

5.1 Novelty

The vending machine designs from FPGA-based are flexible and quicker than the machine designed from CMOS-based. The vending machines designed FPGA-based are easier to program and can be reconfigured anytime without changing the whole machine design architecture if the designers want to enhance the design of the machine. This flexibility is not possible in the case of an Embedded-based machine. In this vending machine designed using FPGA, the user can add a greater number of items with other facilities.
Our vending machine has multi-price product handling capability. The system dynamically handles different product prices (10, 15, 20, and 25 units of currency) using the N input parameter. Our vending machine has optimized cash handling capability also. The vending machine supports multiple denominations for cash input (2'b00 for 0, 2'b01 for 5, 2'b10 for 10, and 2'b11 for 20). Automatic calculation of cash returns ensures that excess payment is efficiently refunded,

enhancing user experience. A reset feature ensures that the vending machine can be returned to its initial state instantly, providing reliability during operation or maintenance.

5.2 Design Considerations (PO(c))

5.2.1 Considerations to public health and safety

The vending machine designed in this project can be utilized to dispense essential items such as first aid kits, personal protective equipment (PPE), or emergency supplies. This functionality becomes especially valuable during times when general shops are closed, ensuring that people can still access lifesaving and critical items. By allowing users to select a product and make payments seamlessly, the system provides an efficient and reliable solution for dispensing essential goods, catering to both normal and emergency situations.

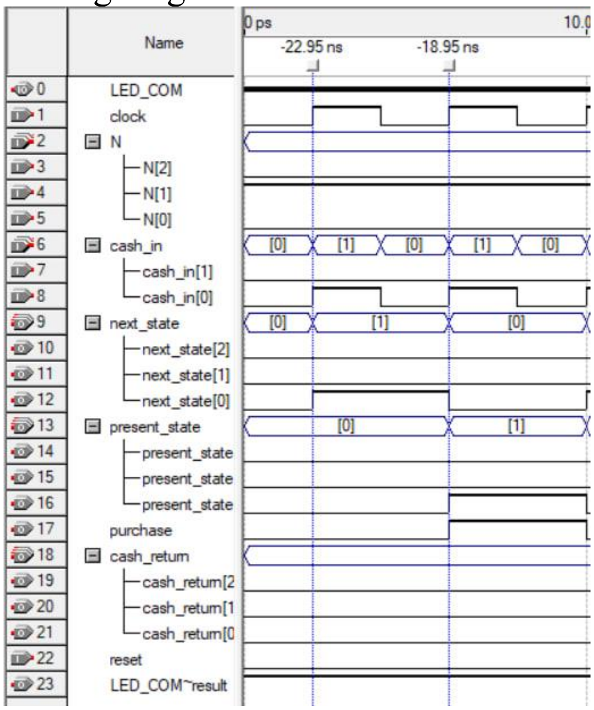
5.2.2 Considerations to cultural and societal needs

We avoided products, imagery, or advertising that may be culturally insensitive or offensive, ensuring that our project respects and appreciates local cultural norms and values.

5.3 Investigations (PO(d))

Test case:
Product: 10 taka
Cash in: 5 taka and 5 taka
FPGA display:

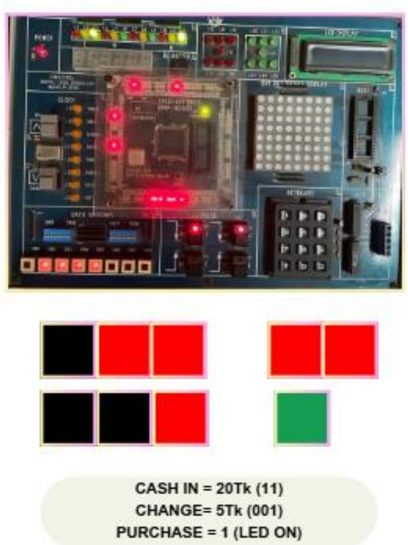
Timing diagram:



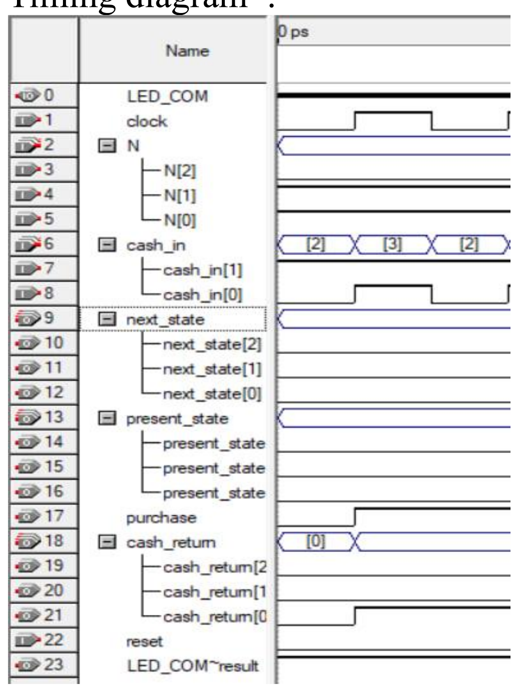
Product: 15 taka

Cash in: 20 taka

FPGA display:



Timing diagram :

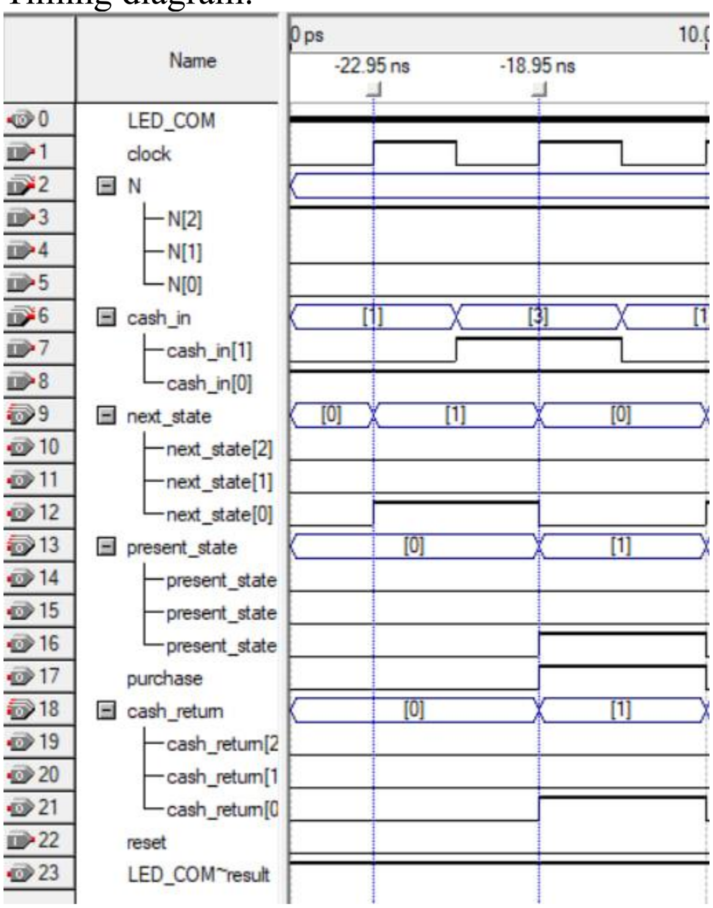


Product: 20 taka

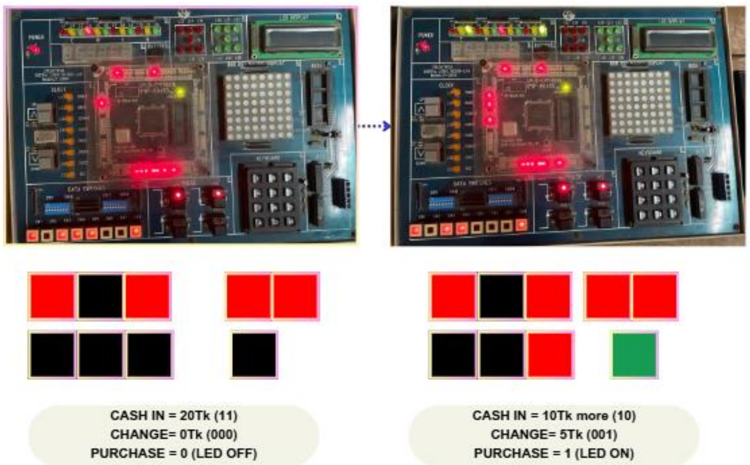
Cash in: 5 taka and 20 taka

FPGA display:

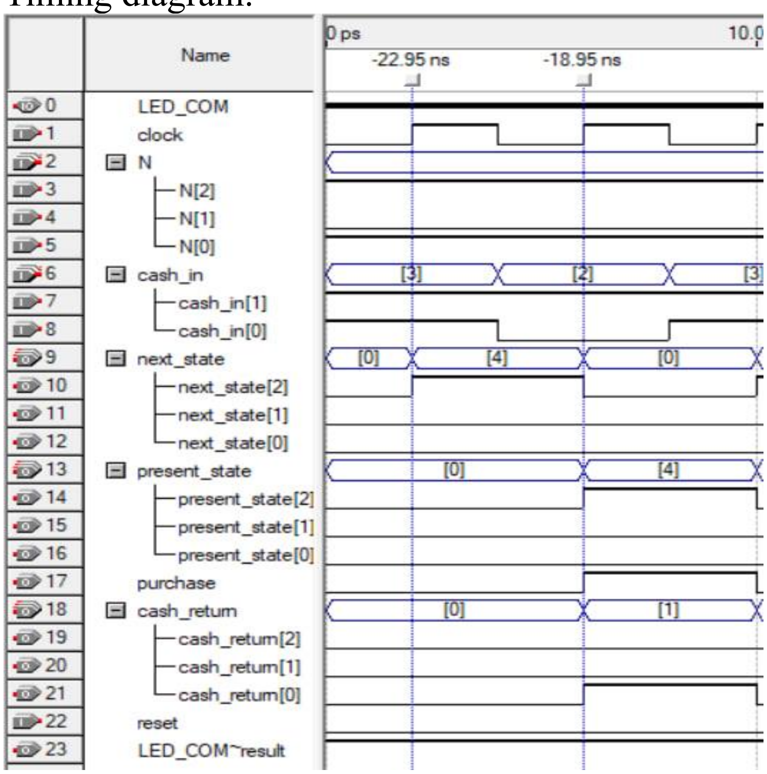
Timing diagram:



Product: 25 taka
Cash in: 20 taka and 10 taka
FPGA display:



Timing diagram:



5.3.1 Results and Analysis

The implemented vending machine successfully demonstrated its core functionalities on the FPGA board, aligning with the expected outcomes derived from the FSM-based design. Key results and observations are summarized as follows:

State Transition Accuracy:

The finite-state machine accurately transitioned between states based on the input signals. The states, corresponding to cash denominations were clearly represented by the LED lighting sequences

Input and Output Functionality:

The switches provided inputs for coin insertion, reset, and clock signals, enabling correct state transitions. The green LED indicated successful purchases, while LED sequences accurately displayed returned change, aligning with the design's binary specifications.

Return of Change:

The system correctly calculated and displayed the change for excess cash inputs. For example, when a 20 TK coin was inserted for a product costing 15 TK, the remaining 5 TK was indicated through the LED sequence (001).

The project demonstrated the effectiveness of FSMs in implementing a real-world digital system through the use of Verilog and the FPGA board. The use of LEDs and switches provided a simple yet effective mechanism for state representation and user interaction.

5.4 Limitations of Tools (PO(e))

In our vending machine, if the reset button is pressed at any point during operation, the machine will reset and display the cash-in value as 0, regardless of any cash that has already been inserted. Additionally, the machine does not return the previously inserted cash if reset button is pressed, which is a significant limitation of its design.

Moreover, the vending machine only accepts cash denominations of 5 TK, 10 TK, and 20 TK, limiting its flexibility to handle other amounts. Another drawback is the lack of a display to indicate the cash inserted and the cash returned (in TK), which reduces the user-friendliness of the vending machine.

5.5 Impact Assessment (PO(f))

5.5.1 Assessment of Societal and Cultural Issues

A variety of products have been have been offered , keeping in mind the diverse cultural preferences and dietary habits of the consumers.

5.5.2 Assessment of Health and Safety Issues

Products will have clear labeling including manufacturing date, expiring date , ingredients list and nutritional information, which will help the customers to decide according to their physical need and prohibit them from consuming any product particularly harmful for them.

5.5.3 Assessment of Legal Issues

We ensured that items dispensed by the vending machine are safe for consumption, meet all relevant food safety standards in manufacturing and went through due process of authorized food safety protocol .

5.6 Sustainability Evaluation (PO(g))

We considered factors such as energy use, waste generation, and resource consumption.

5.7 Ethical Issues (PO(h))

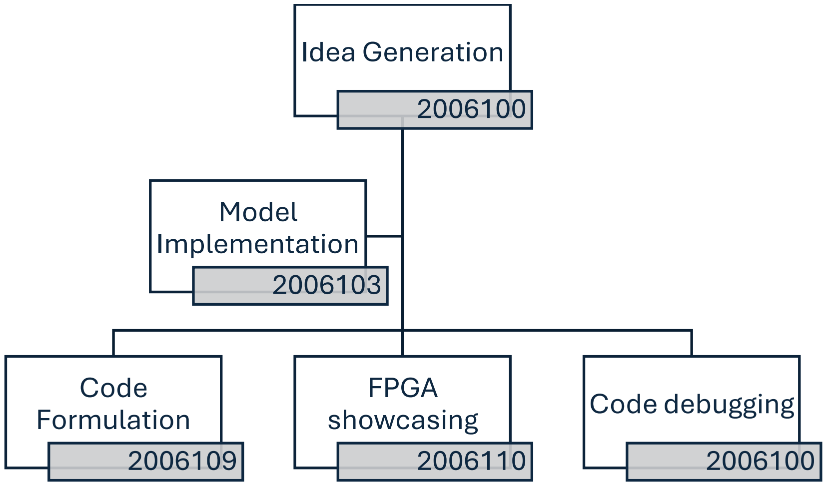
No product has been stored that can be used to bring chaos in the society

6. Reflection on Individual and Team Work (PO(i))

6.1 Individual Contribution of Each Member

ID	Statement
2006100	Idea generation and Code implementation
2006103	Code implementation and debugging
2006109	State diagram and state table design
2006110	State diagram, design and debugging

6.2 Model of Team Work



6.3 Diversity Statement of Team

Collaborative Approach

The key to our team's success is cooperation. Every member was invited to share their thoughts and offer helpful criticism since we placed a high value on open communication. Team meetings were held regularly to review progress, resolve issues, and coordinate our objectives.

Division of Labor

To maximize efficiency and capitalize on individual strengths, our team has adopted a well-structured division of labor. Each member is assigned specific roles and responsibilities.

7. Communication to External Stakeholders (PO(j))

7.1 Executive Summary:

A vending machine algorithm has been designed an implemented on FPGA by a group of 4 members.

A customer can select 1 of 4 products at a time , inserting money in the vending machine in any combination of the 3 input system mentioned above . As soon as the sum of the combination becomes equal or greater than the price of the selected product, the product will be delivered and return money will be automatically calculated and returned , which will be showed by combination of LED lights on the FPGA board .

The machine will go back to reset state at any point of the process after pushing the "Reset" button.

7.2 User Manual

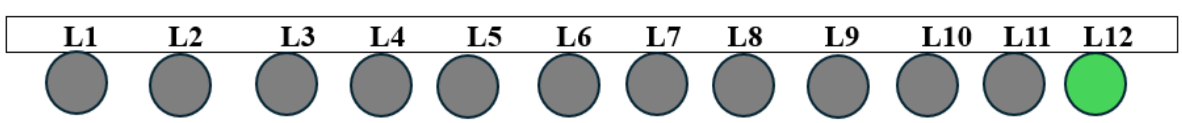
To buy any product
SW7 (RESET) must be 0!!
Have to press SW8 (Positive edge) after every input note.
Firstly, User will select the price of the product using SW1, SW2, SW3:

Input options			Selected product
SW1	SW2	SW3	
			10 TK product
			15 TK product
			20 TK product
			25 TK product

Then the user will provide cash in using SW4, SW5:

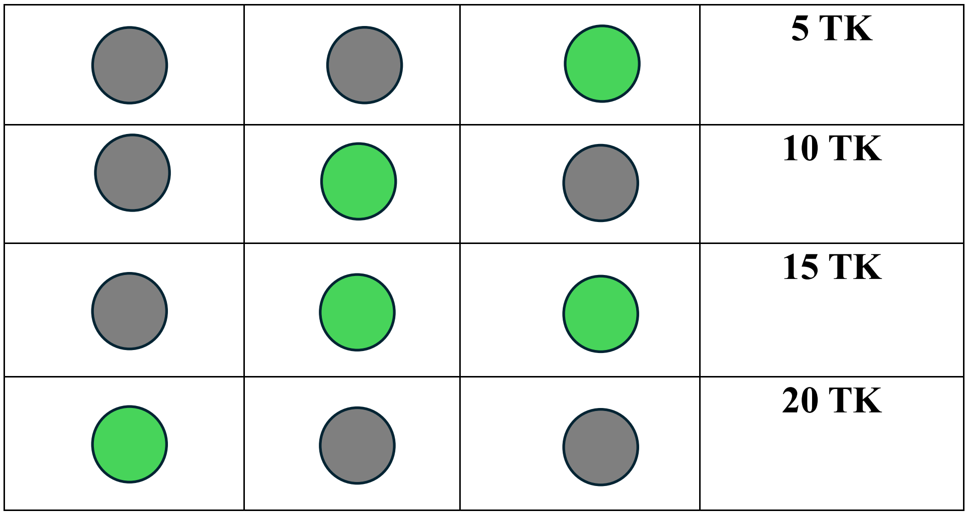
Input options		Cash in
SW4	SW5	
		0 TK
		5 TK
		10 TK
		20 TK

Ouput LEDs:



LED 12 (L12) will be on if purchase is successful!
L1, L2, L3 indicates the amount of cash return.

L1	L2	L3	Cash Return
			0 TK



7.3 GitHub Link

https://github.com/NilanjonaMedha/Design-of-FPGA-based-Vending-Machine.git

7.4 YouTube Link

https://youtu.be/uL-mNh7mbbI?si=D5ILRwp714Aemg3v

8. Project Management and Cost Analysis (PO(k))

8.1 Bill of Materials

We were able to complete a cost-free project due to the use of FPGA and Verilog . The CPLD/ FPGA board setup was already in the lab so we didn't need to buy a single wire or IC to accomplish the project.

8.2 Timeline of Project Implementation

Date	Performance
03/10/2024	Proposal submission
10/10/2024	Project assigned by instructor
11/10/2024	Development of design ideas
01/11/2024	Code implemented for particular product
04/11/2024	**First Update**
01/12/2024	Display setup in CPLD board
04/12/2024	Updated code for better performance
09/12/2024	**Final Presentation**

9. Future Work (PO(l))

Several improvements and extensions can be explored in future iterations of the project:

The vending machine can be designed with better cash management and a better return process. A function can be provided that, upon pressing the reset button, returns the money that was previously inserted, guaranteeing that user funds are not lost while the system is in use. For more flexibility and user convenience, improve the cash-handling mechanism to accept a larger variety of currencies, such as coins and smaller currency notes.

An easy-to-use display can be included in the vending machine to provide real-time information about the amount of money inserted, returned, and the progress of the current transaction, add an LCD display. The system will become more engaging, and the user experience will be much enhanced.

A user interface improvement can be included, such as incorporating touch buttons or a keypad to allow users to input custom cash amounts or select from predefined options for products and price ranges.

10. References

1.M. Morris Mano "Digital logic and computer design" 3rd edition pearson.

2.Stephen Brown and Zvonko Vranesic "Fundamentals of Digital Logic with Verilog Design" 3rd edition.

3. Muhammad Ali Qureshi "Design and implementation of vending machine using Verilog" 2nd international conference on networking and information technology. IPCSIT vol.17, pp-6 (2011).

4.B. Jyothi, Isarah, A.srinivas "Implementation of FPGA based smart vending machine " International Journal of Engineering Research and application (IJERA) 2248-9622. National Conference on Development Advances and trend in Engineering Science (NCDATES 09th and 10th January 2015