

**DESIGN AND IMPLEMENTATION OF AUTOMATED TELLER MACHINE**

**A PROJECT REPORT**

# *Submitted by*

**PRIYANKA S (200051601025)**

**Under the guidance of**

# DR. Anusooya

# Project Mentor

# Assistant Professor */ECE*

**ELECTRONICS AND COMMUNICATION ENGINEERING**



# JULY 2023

## **ACKNOWLEDGEMENT**

I would like to thank Intel Unnati Community, for giving me the wonderful opportunity to do this Internship Program.

My sincere gratitude to Dr.Anusooya, Assistant Professor */ECE*, for offering a step-by-step guidance and clarification in all the concepts regarding the work assigned to me.

I thank all the staff members, who helped and guided me in this project.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
|  | **Topic** | **Page no** |
| **1** | **Abstract** | **4** |
| **2** | **Introduction** | **4** |
| **3** | **Software Required** | **4** |
| **4** | **Tasks peformed** | **5** |
| **5** | **Tests carried out** | **8** |
| **6** | **Final accuracy** | **14** |

**ABSTRACT:**

This project presents the design and implementation of an ATM (Automated Teller Machine) controller using Verilog hardware description language. The ATM controller is based on a Finite State Machine (FSM) architecture and provides various functionalities.ATM’S as known today,are based on a computer system with microprocessor or other programmatic components with data storage such as Hard disk or semiconductor memories,cash boxes,card readers etc which are operated by industry specific software.

**INTRODUCTION:**

The FSM ATM machine design provides a flexible and modular solution for implementing ATM functionalities in hardware. It can be synthesized and implemented on FPGA platforms to create a functional ATM prototype. The Verilog code and testbench serve as a basis for further refinement and customization to meet specific design requirements and constraints.

The Verilog implementation of the ATM controller includes modules for the main ATM controller, clock divider, and testbench for functional verification. The clock divider module divides the input clock frequency to match the desired clock frequency for the ATM controller.

The testbench validates the functionality of the ATM controller by generating test scenarios and monitoring the expected outputs.

**SOFTWARE REQUIRED:**

1. Intel Quartus Prime Lite Edition
2. Labs land

**TASKS PERFORMED:**

The designed ATM controller FSM perform the following functionalities:

1. New pin change
2. Invalid Pin entry(3 times allowed and later it gets locked for next 24 hours)
3. Withdraw
4. Deposit
5. Check balance
6. Mini statement that displays the old balance and new balance.

**1.New Pin Change:**

When the user presses the key sequence "100," the system enters the pin change state. The system checks the validity of the pin before each transaction using a 2-bit representation. The initial pin code is set as "01." If the user wants to change the pin, the system verifies the old pin. If the old pin matches, the user can enter the new pin and confirm it. If the new pin matches the confirmation, it will be updated for all future transactions.

**2.INVALID PIN ENTRY:**

The FSM system verifies the pin match before each transaction. When the card is inserted, the user is prompted to enter the pin. If the first attempt fails, the user is allowed two additional attempts. However, after three unsuccessful attempts, the ATM card is locked for the next 24 hours. After this lockout period elapses, the user can resume making transactions by entering the correct pin.

**3.WITHDRAW:**

When the user presses the key sequence "001," the system enters the withdrawal state, and further processing is carried out based on the transaction amount provided.

**a. Amount less than 6:**

For simplicity and considering the PIN constraints, a 4-bit representation is used for the transaction amount. The initial account balance is set at RS.15. If the amount requested is less than RS.5, the transaction is processed directly. The transaction amount is subtracted from the balance, and the mini statement displays both the old and new balances.

**b. Amount more than 5 and less than 11:**

When the transaction amount is greater than RS.5 and less than RS.11, the FSM system prompts for OTP verification before proceeding with the transaction. Once the OTP is successfully verified, the transaction process is initiated. However, if the OTP doesn't match, the system resends a new OTP for verification.

**c. Amount more than 10:**

When the transaction amount exceeds RS.10, the FSM system prompts for facial recognition, which is represented by a 1-bit value, before proceeding with the transaction. Once the face is successfully recognized, the transaction process is initiated.

**d.Warning:**

When the transaction amount exceeds the account balance, the FSM system enters a warning state, and no further transactions are allowed. After issuing a warning, the system returns to the transaction choice stage, where the user can select a different transaction or take appropriate action based on the available balance.

**4.DEPOSIT:**

When the user presses the key sequence "010," the system enters the deposit state, and the transaction process is initiated. The deposited amount is added to the account balance, and both the old and new balances are displayed in the mini statement.

5**.SHOW BALANCE:**

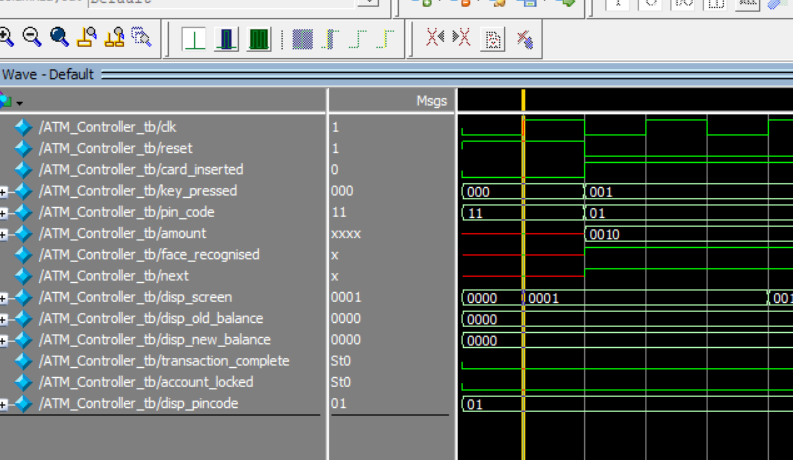
When the user presses the key sequence "011," the system displays the account balance. The balance is shown on the LEDs, providing the user with the current available balance in the account. This feature allows the user to conveniently check their account balance before proceeding with any further transactions.

**TESTS CARRIED OUT:**

**1.RESET:**

If the reset signal is pressed, the FSM transitions to the "IDLE" state, and all the output signals are set to zero. This ensures a clean and reset state for the ATM controller. The system is ready to accept new card insertions and user interactions, starting the ATM transaction flow from the initial state. It provides a reliable and consistent starting point for users, allowing them to initiate a new transaction without any residual effects from previous operations.

**a.WAVEFORM OUTPUT:**



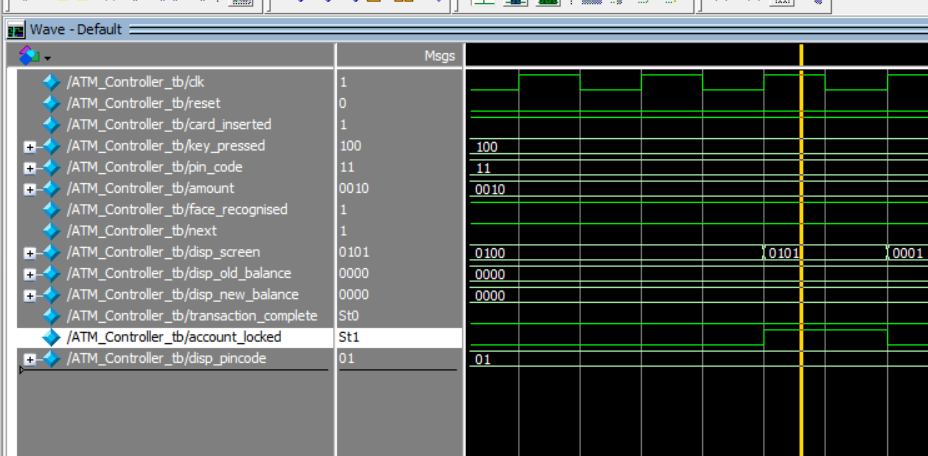
* **b.TRANSCRIPT RESULTS:**



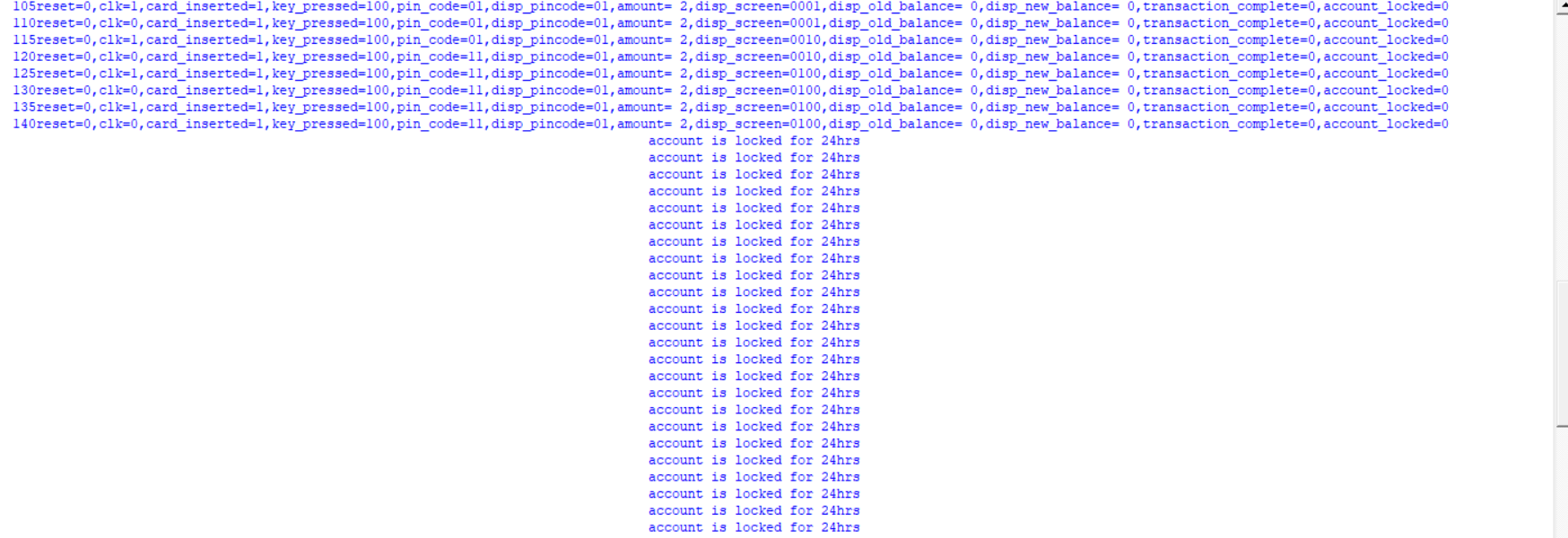
**2.INVALID PIN:**

After three unsuccessful PIN attempts, the account is locked for the next 24 hours. During this lockout period, the user will not be able to perform any transactions or access the account**.**

**a.WAVEFORM OUTPUT:**



**b.TRANSCRIPT:**

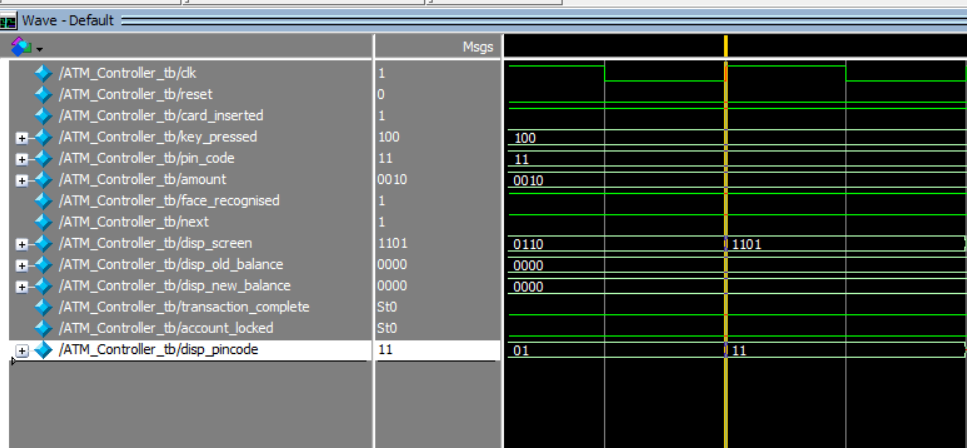


The ‘**pincode’** variable represents the PIN entered by the user, while the ‘**disp\_pincode’** variable holds the actual password. In this case, the password is initially set as "01," but if the user enters "11" as the pincode and exceeds three attempts, the account will be locked. The system will display a message indicating that the account is locked for 24 hours that is displayed in transcript. After the 24-hour lockout period elapses, the user can attempt to access the account again.

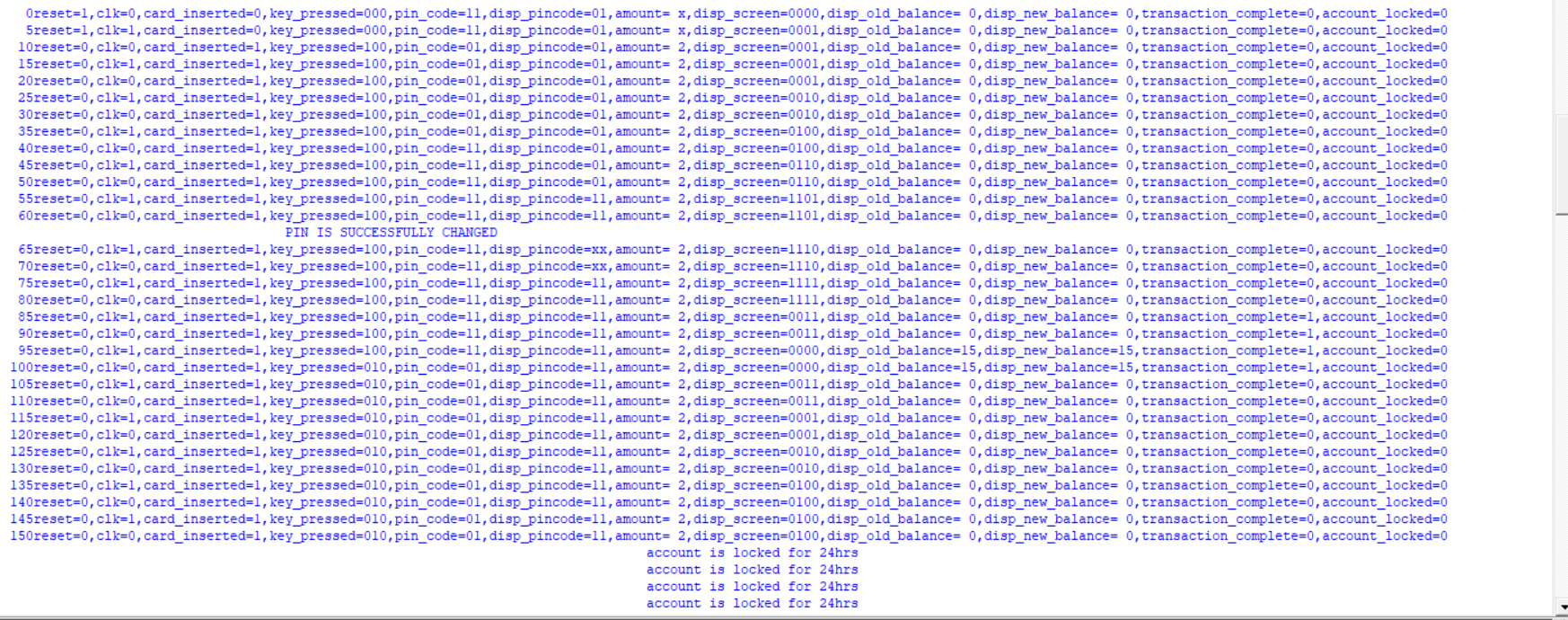
**3.PIN CHANGE:**

When the user wishes to change the PIN, they are allowed to update it.

**a.WAVEFORM OUTPUT:**



**b.TRANSCRIPT:**



**PINCODE** refers to the PIN entered by the user, while **DISP\_PINCODE** represents the actual password for that user. Initially, the password was set as "01". After changing the PIN, the new password "11"is updated in **DISP\_PINCODE**, and it is also displayed for verification.

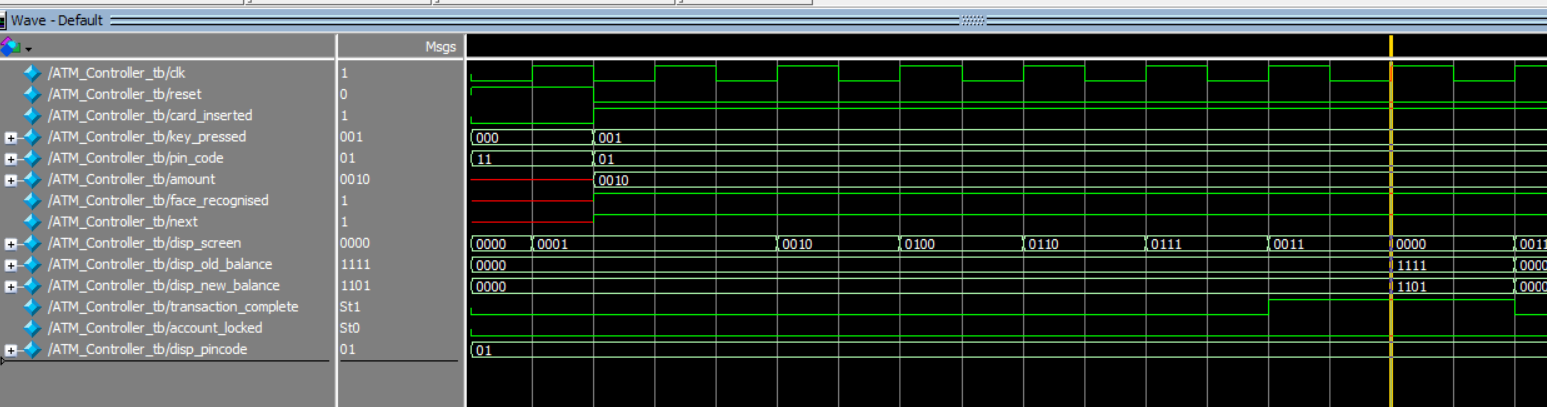
When attempting the next transaction, if the user enters the old PIN code ("01"), but fails to provide the correct PIN after three attempts, the account is locked. This indicates that the new PIN has been successfully updated. The account will remain locked for a 24 hrs, as per the system's lockout policy, to ensure account security.

**4.WITHDRAW:**

* **Amount less than Rs.6**

For amount less than RS.6 no aunthetication is required.When the key pressed is 001 then it goes for withdraw state.

**a.WAVEFORM OUTPUT:**



**b.TRANSCRIPT:**

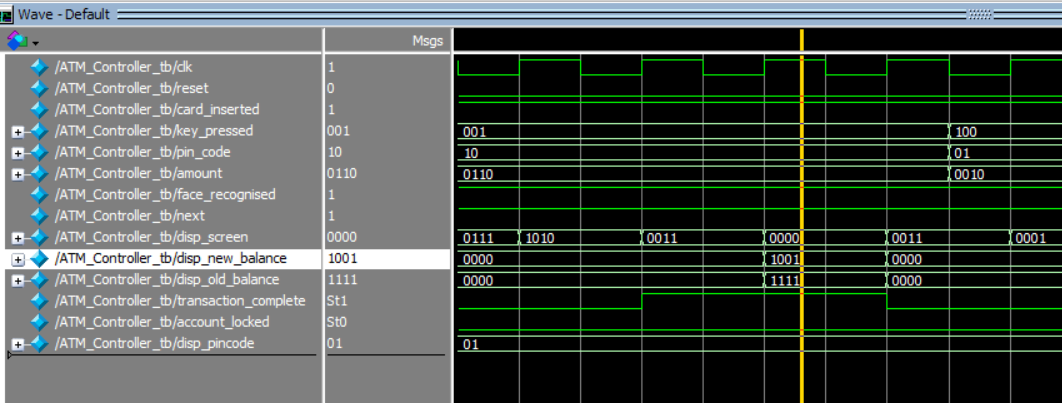


In this scenario, the amount given is RS.2, which is less than RS.6, so no authentication is required. After pressing the key sequence "001," the FSM enters the withdrawal state. Based on the transaction amount, the system proceeds with the transaction process, and cash is dispensed from the cash dispenser. Following the successful transaction, a mini statement displaying the new and old balances is displayed. The user then removes the card, and finally, the system transitions to the idle state.

* **Amount more than 5 and less than 11**

For amounts greater than RS.5 and less than RS.11, OTP verification is required.

**a.WAVEFORM OUTPUT:**



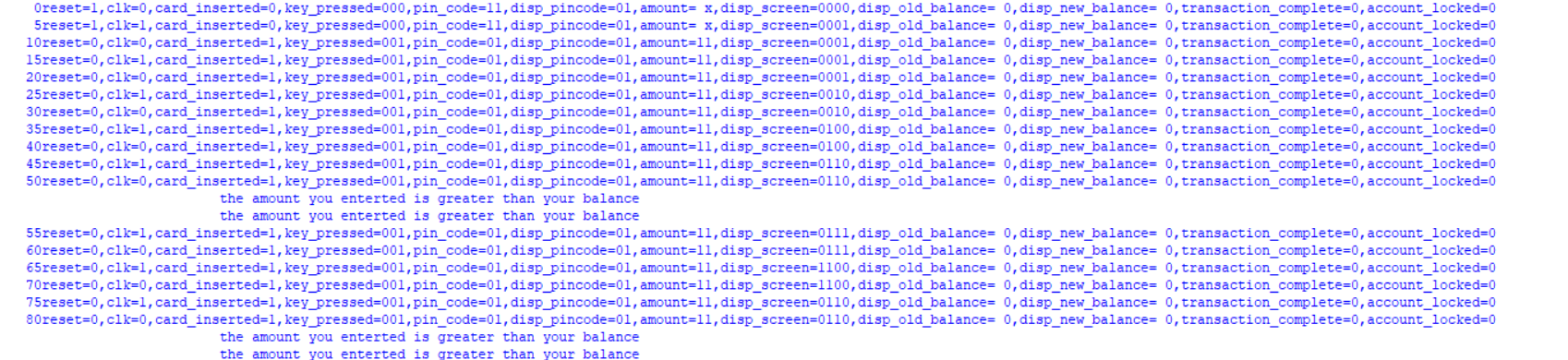
**b.TRANSCRIPT:**



Since the amount entered is RS.6, which falls between RS.5 and RS.11, OTP verification is required. After entering the amount, the system will prompt for OTP verification. If the entered OTP matches the generated OTP, the system will proceed with the transaction process. In this case, as the OTP is verified, the system will display a message indicating that the OTP is verified, and the transaction can continue. If the entered OTP does not match the generated OTP, the system will resend a new OTP for verification.

* **Amount greater than the balance:**

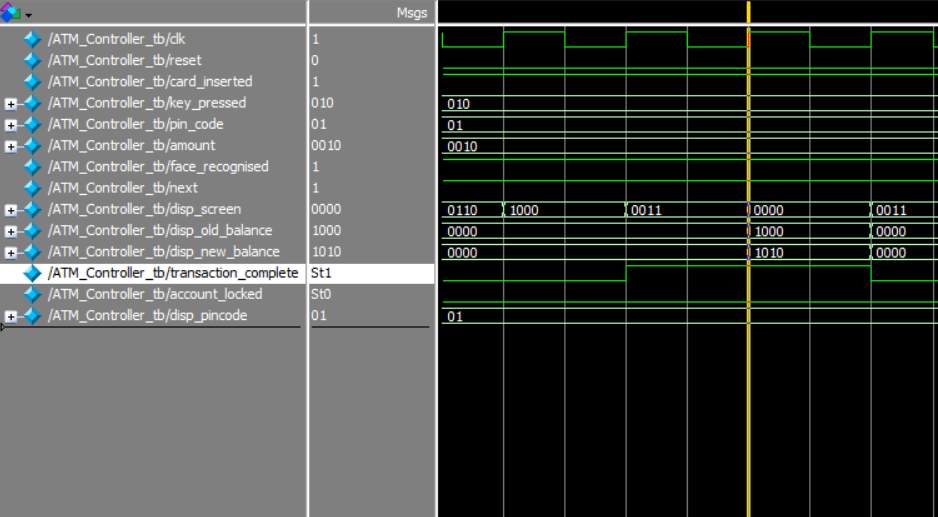
**a.transcript:**



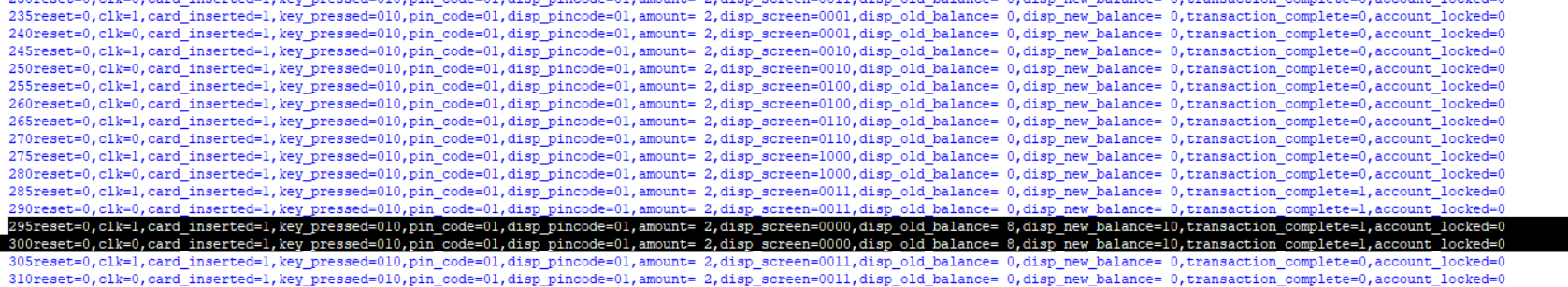
Here the entered amount is greater than the account balance,so the system enters a warning state and displays a warning indicating that the amount exceeds the available balance. After issuing the warning, the system returns to the transaction choice state, allowing the user to select a different transaction or take appropriate action based on the available balance.

**5.DEPOSIT:**

**a.WAVEFORM OUTPUT:**



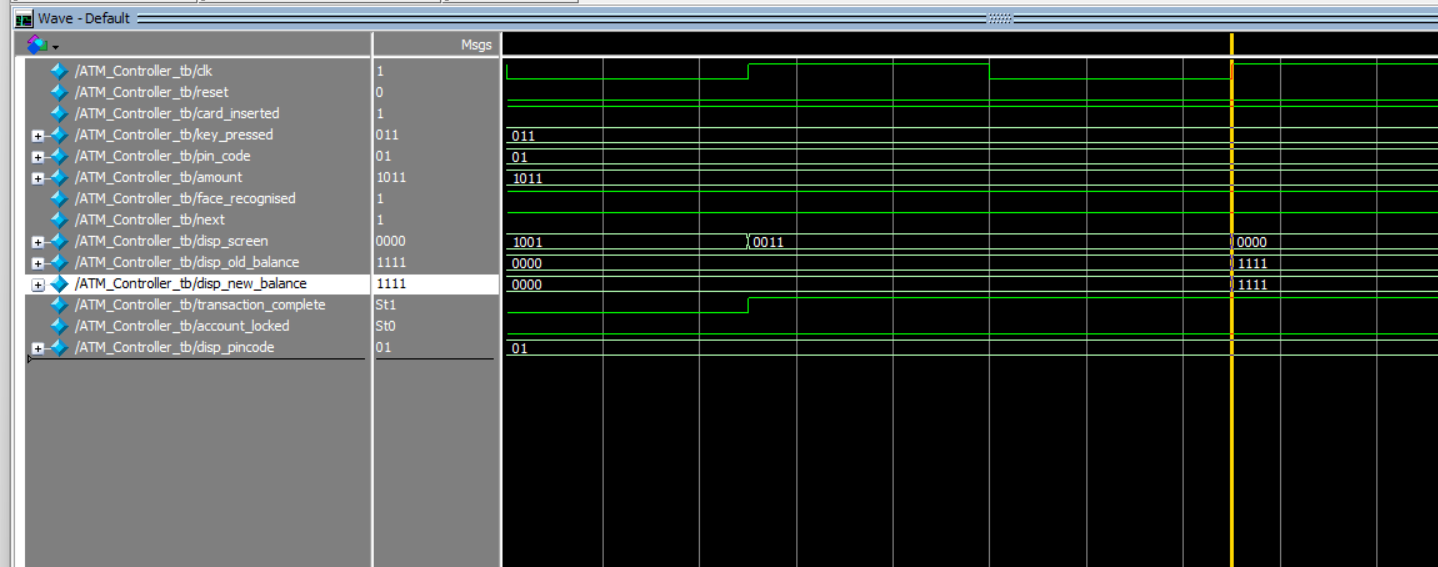
**b.TRANSCRIPT:**



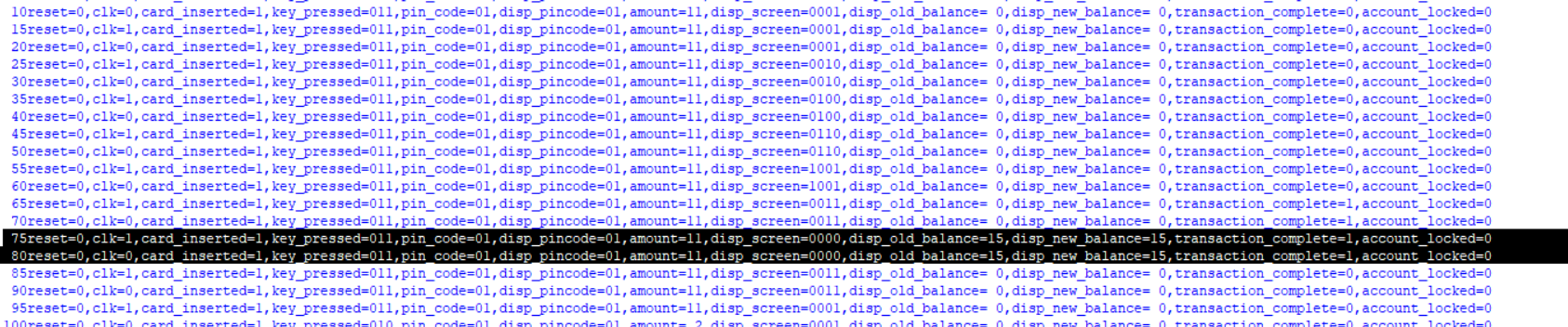
When the key sequence "010" is pressed, the system enters the deposit process, and further processing is performed. The user inserts the money, which is then added to the account balance. In this case, the user enters an amount of RS.2, which is added to the balance. Finally, both the new and old balances are displayed.

**6.SHOW BALANCE:**

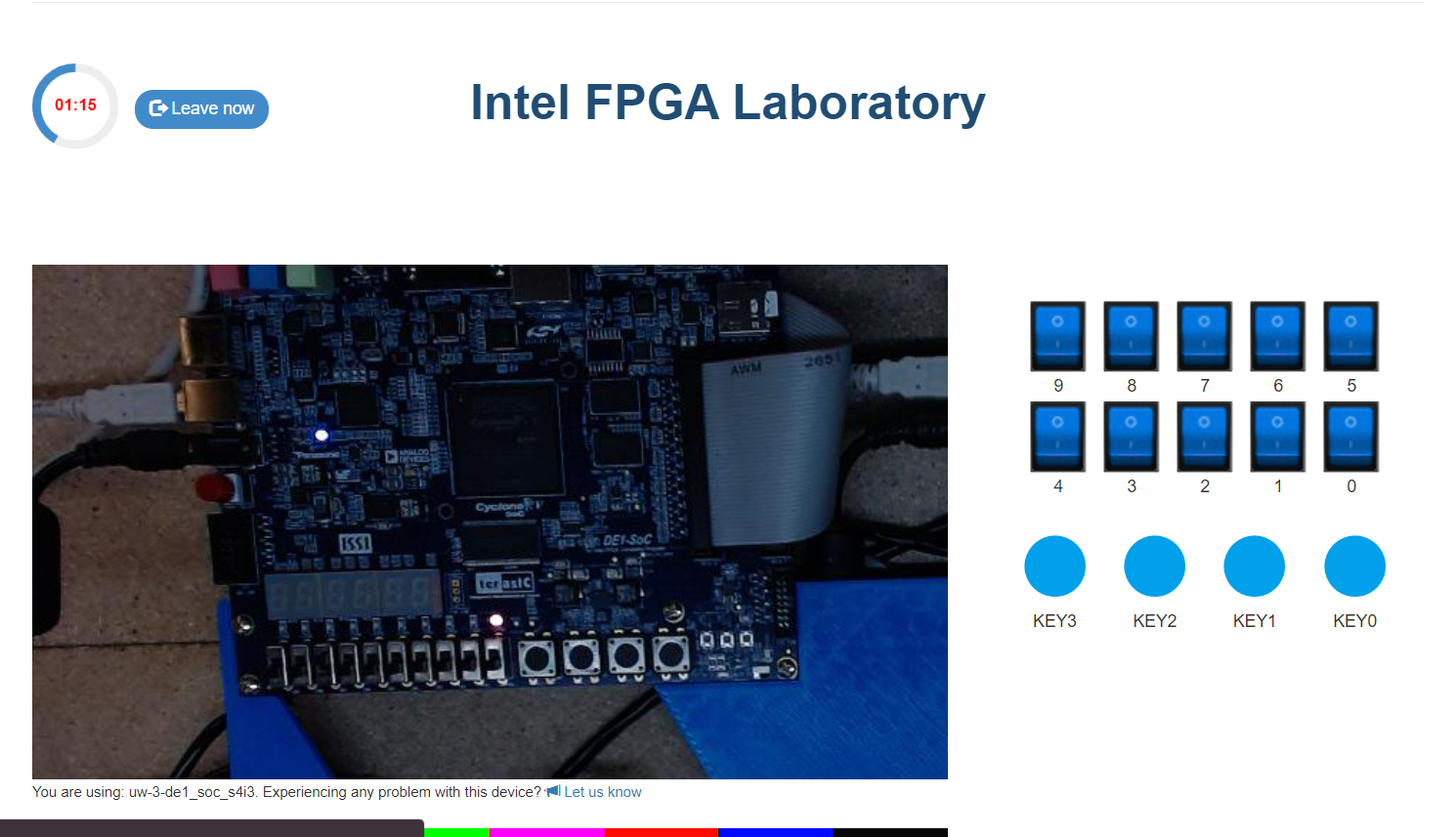
**a.WAVEFORM OUTPUT:**



**b.TRANSCRIPT:**

 When the key sequence "011" is pressed, the system enters the show balance state. In this state, the system displays both the old balance and the new balance to provide the user with a comprehensive view of the account's financial status. This information helps the user keep track of their transactions and monitor the changes in their account balance over time. It enhances transparency and enables users to make informed decisions regarding their financial activities.

**7.IMPLEMENTATION RESULTS:**

****

**NOTE:**After uploading the code in the board,I m getting the first ouput and when I change the input I couldn’t see any changes.I also tried the code which intel group have sent(mealy 100 pattern detector) and I ended up facing the same issue.

So I tried to add clock divider and integrate that with the main code.But from Thursday night I m facing server issues.Once the server is ready I will try to implement the design on the board and attach the outputs as soon as possible.

**FINAL ACCURACY**:

Several tests were conducted to verify the functionality and accuracy of the ATM controller. These tests included simulating different scenarios, such as valid and invalid PIN entry, withdrawal and deposit transactions with varying amounts, balance display, PIN change process, and handling of error conditions like account lockout. The final accuracy of the ATM controller was evaluated based on the successful completion of these tests, ensuring that the desired functionality was achieved.The implementation part I will try to upload once the server is ready.