



PROJECT REPORT ON “ATM MACHINE USING VERILOG CODE”



UNDER THE GUIDANCE OF:

Mr. ADITYA KUMAR HOTA

Dr. BANDAN KUMAR BHOI

PREPARED BY:

OMM JYOTI BAI (Regd. NO. 2002070072)

BISWADARSHEE BEHERA (Regd. NO. 2002070070)

HIMANSHU MAROTHIA (Regd. NO. 2002070045)

SANGRAM JENA (Regd. NO. 2103070006)

VEER SURENDRA SAI UNIVERSITY OF TECHNOLOGY

BURLA, SAMBALPUR

ODISHA-768018

ABSTRACT

This report attempts to understand the design of an Automated Teller Machine (ATM) system, a device used by bank customers to process account transactions. Typically, a user inserts into the ATM a special plastic card that is encoded with information on a magnetic strip. The strip contains an identification code that is transmitted to the bank's central computer by modem. To prevent unauthorized transactions, a personal identification number (PIN) must also be entered by the user using a keypad. The computer then permits the ATM to complete the transaction; most machines can dispense cash, accept deposits, transfer funds, and provide information on account balances. Banks have formed cooperative, nationwide networks so that a customer of one bank can use an ATM of another for cash access. Some ATMs will also accept credit cards for cash advances. The first ATM was installed in 1969 by Chemical Bank at its branch in Rockville Centre, New York. A customer using a coded card was dispensed a package containing a set sum of money.

VEER SURENDRA SAI UNIVERSITY OF TECHNOLOGY



CERTIFICATE

This is to certify that project entitled "**ATM MACHINE USING VERILOG**" presented by **OMM JYOTI BAI(2002070072)**, **BISWADARSHEE BEHERA(2002070070)**, **HIMANSHU MAROTHIA (2002070045)** and **SANGRAM JENA(2103070006)** of 5th Semester in Department of Electronics and Telecommunication, VSSUT, Burla has been successfully delivered during the academic year 2022-23 towards the fulfillment of the requirements of bachelor's degree.

Dr. Bandan Kumar Bhoi

Assistant Professor

Department of E&TC

VSSUT, Burla

Mr. Aditya Kumar Hota

Assistant Professor

Department of E&TC

VSSUT, Burla

INTRODUCTION

An automated teller machine (ATM) is a specialized computer that allows you to complete bank transactions without the need to see a bank representative. Many ATMs are conveniently accessible any time of day or night and can be used for everything from withdrawing or depositing money to checking your account balance to transferring money between accounts.

Here we'll delve deeper into what an ATM is, common transactions performed at ATMs and important things to know before using them.

What is an ATM?

ATMs are machines that dispense cash and allow you to make other banking transactions. An ATM typically consists of a screen, a card reader, a keypad, a cash dispenser and a printer.

ATMs can be found in many locations throughout the U.S. and the world. On-premise ATMs are located at financial institutions such as banks and credit unions, while off-premise ones are commonly offered at places like airports, grocery stores and gas stations.

Using an ATM simply involves inserting your bank-issued ATM card, entering your PIN and following the prompts on the screen to complete your desired transaction.

Examples of ATM transactions

Various common banking transactions that are often carried out at an ATM include:

Withdrawing cash

The most common ATM transaction is the withdrawal of funds from one's account. Banks typically impose limits on the maximum amount that can be taken out each day. This

amount can vary from bank to bank, as well as among different accounts offered by a single bank.

Transferring funds

You may be able to use an ATM to transfer money between accounts you hold with your bank. For instance, if you wish to transfer \$200 from your savings account to your checking account, this can often be done by selecting the “transfer” option at the ATM. Like balance inquiries, transferring funds between accounts is also something you can accomplish using your bank’s mobile app or website.

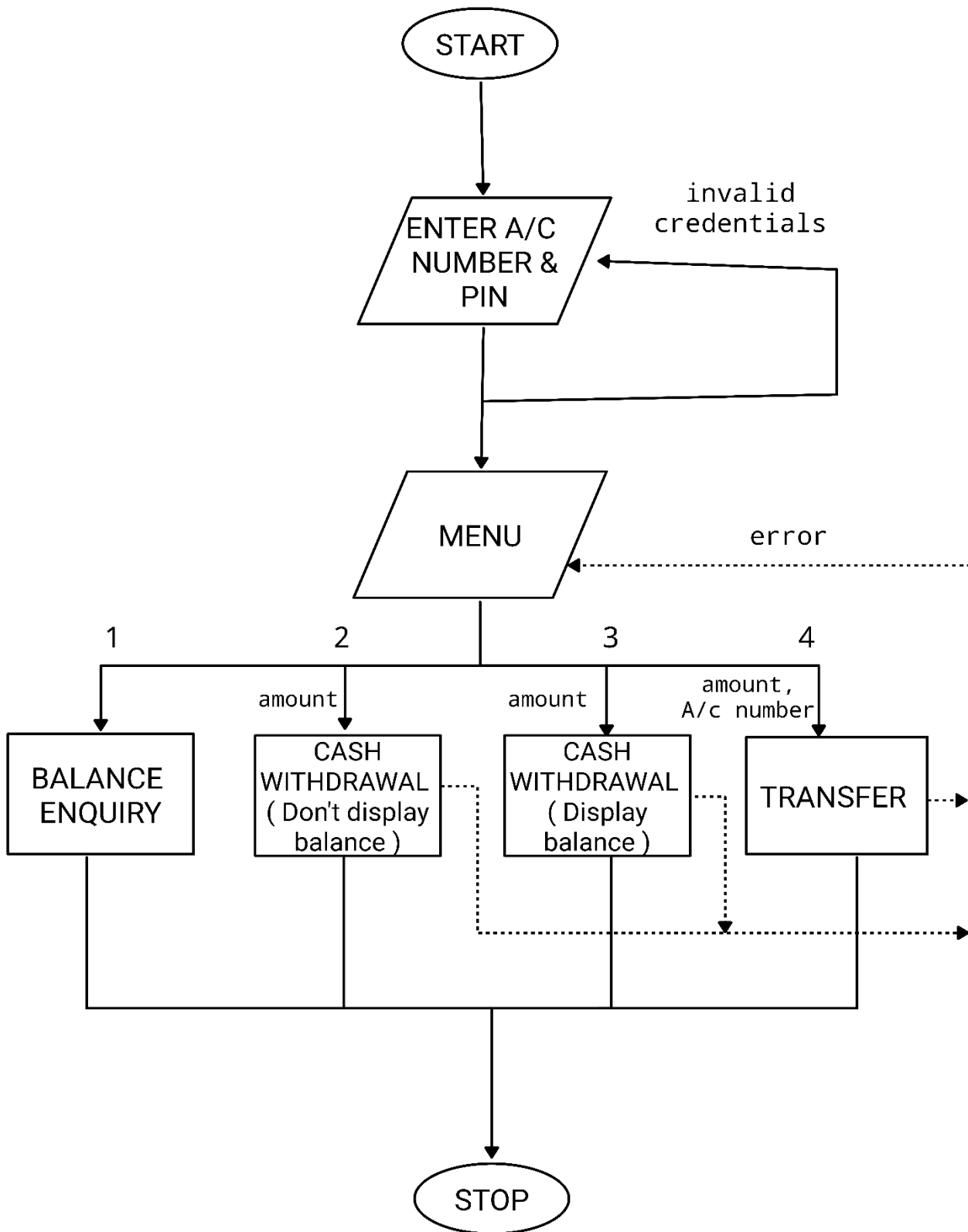
Balance inquiries

You can also visit an ATM to view your current account balance. This feature may come in handy if you wish to know how much you’re able to spend when using your debit card or writing a check. Alternatively, your account balance is something you can view by logging onto your bank’s mobile app or website. Knowing your balance can help you keep from overspending or overdrawing your account.

ALGORITHM

- Start the program.
- Take account number and pin as input.
- If account number is valid & pin is matched then take input value for menu. Else go to login page again.
- For menu value as 1, display the balance.
- For menu value as 2, take withdraw amount value as input & give cash output without displaying the remaining balance.
- For menu value as 3, take withdraw amount value as input & give cash output & display the remaining balance.
- For menu value as 4, take account number & cash amount to be transferred as input & transfer the amount specified from the user account to the account number entered.
- For invalid credentials or invalid cash amount value give an error & go to the menu page.
- End the program.

FLOWCHART



PROGRAM CODE

```
//ATM MACHINE CODE FOR ICS PROJECT (2022-23)
/* BY:-HIMANSHU MAROTHIA-2002070045 (himanshumarothia04@gmail.com)
   BISWADARSHEE BEHERA-2002070070 (biswadarsheebehera@gmail.com)
   OMM JYOTI BAI-2002070072 (ommjyotib@gmail.com)
   SANGRAM JENA-2103070006 (jsangram062@gmail.com)
*/
module VERIFY(
    input [31:0] AcNumber,
    input [11:0] Pin,
    input Action,
    input LogOut,
    output reg isVerified,
    output reg [3:0] AcIndex);
    reg [31:0] DATABASE_AcNumber [0:11];
    reg [11:0] DATABASE_PinNumber [0:11];

    //Creating the database for account number & pin.
    initial begin
        DATABASE_AcNumber[0] = 32'd2002070045; DATABASE_PinNumber[0] =
12'd1000;
        DATABASE_AcNumber[1] = 32'd2002070070; DATABASE_PinNumber[1] =
12'd1001;
        DATABASE_AcNumber[2] = 32'd2002070072; DATABASE_PinNumber[2] =
12'd1010;
        DATABASE_AcNumber[3] = 32'd2103070006; DATABASE_PinNumber[3] =
12'd1011;

        DATABASE_AcNumber[4] = 32'd2002070073; DATABASE_PinNumber[4] =
12'd1100;

        DATABASE_AcNumber[5] = 32'd2002070074; DATABASE_PinNumber[5] =
12'd1101;

        DATABASE_AcNumber[6] = 32'd2002070077; DATABASE_PinNumber[6] =
12'd1110;
```



```
    DATABASE_AcNumber[7] = 32'd2103070004; DATABASE_PinNumber[7] =  
12'd1111;  
    DATABASE_AcNumber[8] = 32'd2103070005; DATABASE_PinNumber[8] =  
12'd1234;  
    DATABASE_AcNumber[9] = 32'd2002020029; DATABASE_PinNumber[9] =  
12'd1990;  
    DATABASE_AcNumber[10] = 32'd2002030155; DATABASE_PinNumber[10] =  
12'd1357;  
    DATABASE_AcNumber[11] = 32'd2002020078; DATABASE_PinNumber[11] =  
12'd1590; end
```

```
always @ (Logout) begin  
    if(Logout == 1)  
        isVerified = 1'bx; end
```

```
//Using linear search algorithm to find a match for the given account number and pin.  
integer i;  
always @(AcNumber or Pin) begin  
    isVerified = 0;  
    AcIndex = 0;  
    for(i = 0; i < 12; i = i+1)  
begin  
    //Action 0 : Check validity of account number.  
    //Action 1: Authenticate the account number with its pin.  
    if(AcNumber == DATABASE_AcNumber[i])  
begin  
    if(Action == 0)  
begin  
        isVerified = 1;  
  
        AcIndex = i;  
  
end  
    if(Action == 1)  
begin  
        if(Pin == DATABASE_PinNumber[i])  
begin
```

```

        isVerified = 1;
        AcIndex = i;
    end
end
end
end
end
endmodule

```

```

module ATM_MACHINE(input clk, input Cancel, input [31:0] AcNumber,
input [11:0] Pin, input [31:0] ReceiverAcNumber, input [2:0] Menu,
input [15:0] Amount, output reg error, output reg [15:0] Balance);

```

```

//Creating the database for account balance.

```

```

reg [15:0] DATABASE_Balance [0:11];

```

```

initial begin

```

```

    $display("Welcome to the ATM");

```

```

    DATABASE_Balance[0] = 16'd1000;

```

```

    DATABASE_Balance[1] = 16'd340;

```

```

    DATABASE_Balance[2] = 16'd339;

```

```

    DATABASE_Balance[3] = 16'd60000;

```

```

    DATABASE_Balance[4] = 16'd1000;

```

```

    DATABASE_Balance[5] = 16'd1000;

```

```

    DATABASE_Balance[6] = 16'd1000;

```

```

    DATABASE_Balance[7] = 16'd1000;

```

```

    DATABASE_Balance[8] = 16'd1000;

```

```

    DATABASE_Balance[9] = 16'd1000;

```

```

    DATABASE_Balance[10] = 16'd1000;

```

```

    DATABASE_Balance[11] = 16'd1000; end

```

```

parameter S0=3'b000,S1=3'b001,S2=3'b010,S3=3'b011,S4=3'b100,S5=3'b101;

```

```

reg [2:0] PS = 3'b000;

```

```

wire [3:0] AcIndex;

```

```

wire [3:0] ReceiverAcIndex;

```

```

wire isVerified;

```

```

wire isFound;

```

```

reg LogOut = 0;

```

```
VERIFY ValidateCredentials(AcNumber, Pin,1, LogOut, isVerified, AcIndex);  
VERIFY FindAccountNumber(ReceiverAcNumber, 0,0, LogOut, isFound,  
ReceiverAcIndex);
```

```
always @(*) begin
```

```
    error = 0;
```

```
    if(Cancel == 1) begin
```

```
        PS = S0;
```

```
        LogOut = 1; #50; end
```

```
    if(PS == S1) begin
```

```
        if((Menu >= 0) & (Menu <= 7))begin
```

```
            PS = Menu; end
```

```
        else PS = Menu; end
```

```
    case (PS)
```

```
    S0: begin
```

```
        if (isVerified == 1) begin
```

```
            PS = S1;
```

```
            $display("Account verified successfully"); end
```

```
        else if(isVerified == 0) begin
```

```
            $display("Invalid credentials");
```

```
            PS = S0; end end
```

```
    S2: begin
```

```
        Balance = DATABASE_Balance[AcIndex];
```

```
        $display("Avialable Balance : %d", DATABASE_Balance[AcIndex]);
```

```
        PS = S1; end
```

```
    S3: begin
```

```
        if (Amount <= DATABASE_Balance[AcIndex]) begin
```

```
            DATABASE_Balance[AcIndex] = DATABASE_Balance[AcIndex] - Amount;
```

```
            Balance = DATABASE_Balance[AcIndex];
```

```
            PS = S1; error = 0; end
```

```
        else begin
```

```
            PS = S1; error = 1; end end
```

```
    S4: begin
```

```
        if (Amount <= DATABASE_Balance[AcIndex]) begin
```

```

        DATABASE_Balance[AcIndex] = DATABASE_Balance[AcIndex] - Amount;
        Balance = DATABASE_Balance[AcIndex];
        PS = S1; error = 0;
        $display("Transaction Successful \n Available Balance :
%d",DATABASE_Balance[AcIndex]); end
    else begin
        PS = S1; error = 1; end end

S5: begin
    if ((Amount <= DATABASE_Balance[AcIndex]) & (isFound == 1) &
(DATABASE_Balance[ReceiverAcIndex] + Amount < 65536)) begin
        PS = S1; error = 0;
        DATABASE_Balance[ReceiverAcIndex] =
DATABASE_Balance[ReceiverAcIndex] + Amount;
        DATABASE_Balance[AcIndex] = DATABASE_Balance[AcIndex] - Amount;
        $display("Transaction Successful \n Available Balance : %d",
DATABASE_Balance[ReceiverAcIndex]); end
    else begin
        PS = S1; error = 1;
    end
end
end
endcase

```

TEST BENCH CODE

```
module ATM_TESTBENCH();
  reg [31:0] AcNumber;
  reg [11:0] Pin;
  reg [31:0] ReceiverAcNumber;
  reg [2:0] Menu;
  reg [15:0] Amount;
  reg clk, Cancel;
  wire error;
  wire [15:0] Balance;
  parameter S0=3'b000,S1=3'b001,S2=3'b010,S3=3'b011,S4=3'b100,S5=3'b101;

  ATM_MACHINE ATM1(clk, Cancel, AcNumber, Pin, ReceiverAcNumber, Menu,
Amount, error, Balance);
  initial begin clk = 1'b0; end
  always @(error) begin
    if(error == 1)
      $display("An Error Occurred Due To Invalid Operation"); end

  initial begin
    AcNumber = 32'd2103070006; // Invalid pin.
    Pin = 12'd1110; #50

    AcNumber = 32'd2103070006; // Valid pin.
    Pin = 12'd1011; #50

    Menu = S2; // Balance enquiry.
    clk=~clk; #5 clk=~clk; #50

    Menu = S4; Amount = 15000; // Cash withdraw & balance display.
    clk=~clk; #5 clk =~clk; #50

    Menu = S2; // Balance enquiry
    clk=~clk; #5 clk=~clk; #50

    Menu = S3; Amount = 50000; // Balance underflow.
    clk=~clk; #5 clk =~clk; #50
```

```
Menu = S2; // Balance enquiry.  
clk=~clk; #5 clk=~clk; #50
```

```
Menu = S5; Amount = 10000; // Money transfer.  
ReceiverAcNumber = 2002070072;  
clk=~clk; #5 clk=~clk; #50
```

```
//Exit the system  
Cancel = 1; #50 Cancel = 0;
```

```
AcNumber = 32'd2002070072; // New user login.  
Pin = 12'd1010; #50
```

```
Menu = S2;  
clk=~clk; #5 clk=~clk; #50;
```

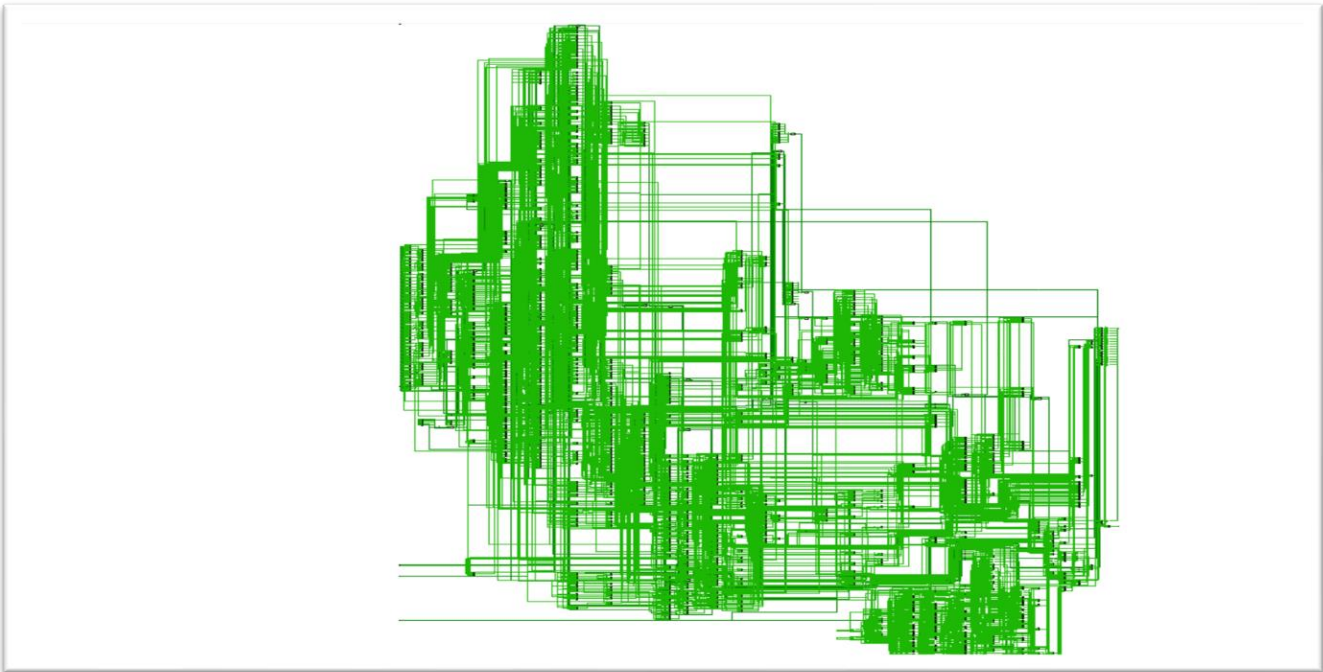


FIG. 1 SCHEMATIC DIAGRAM

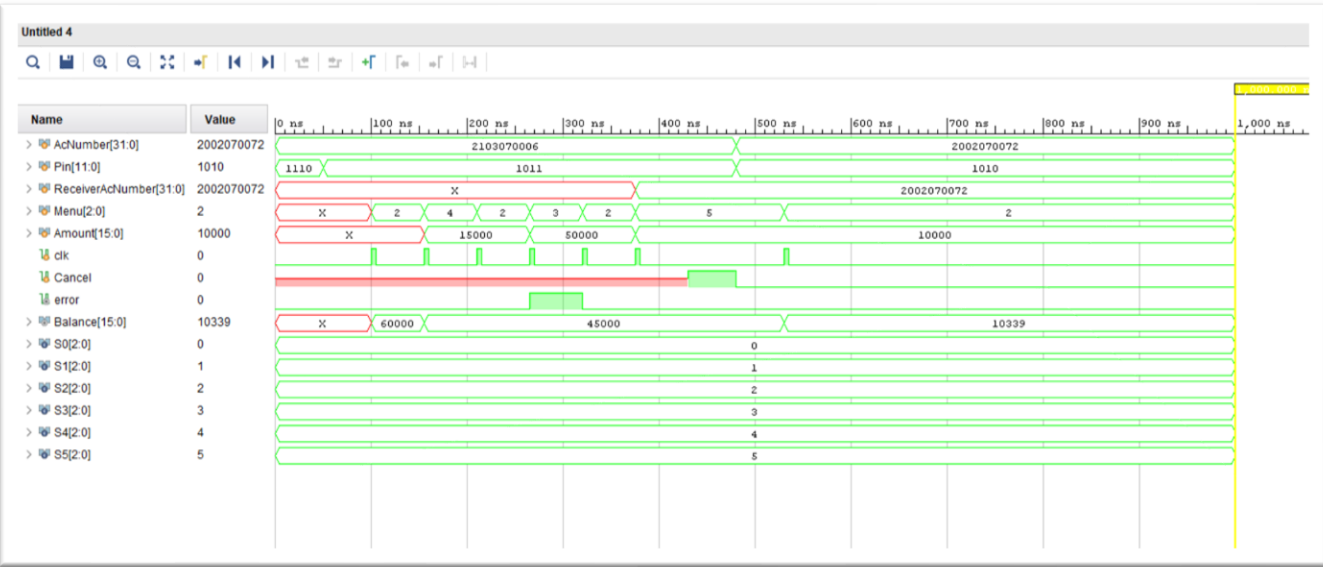


FIG. 2 OUTPUT WAVEFORM

Hierarchy			
Name	Slice LUTs (63400)	Slice Registers (126800)	Bonded IOB (210)
ATM_MACHINE	750	211	113

FIG 3. UTILISATION REPORT

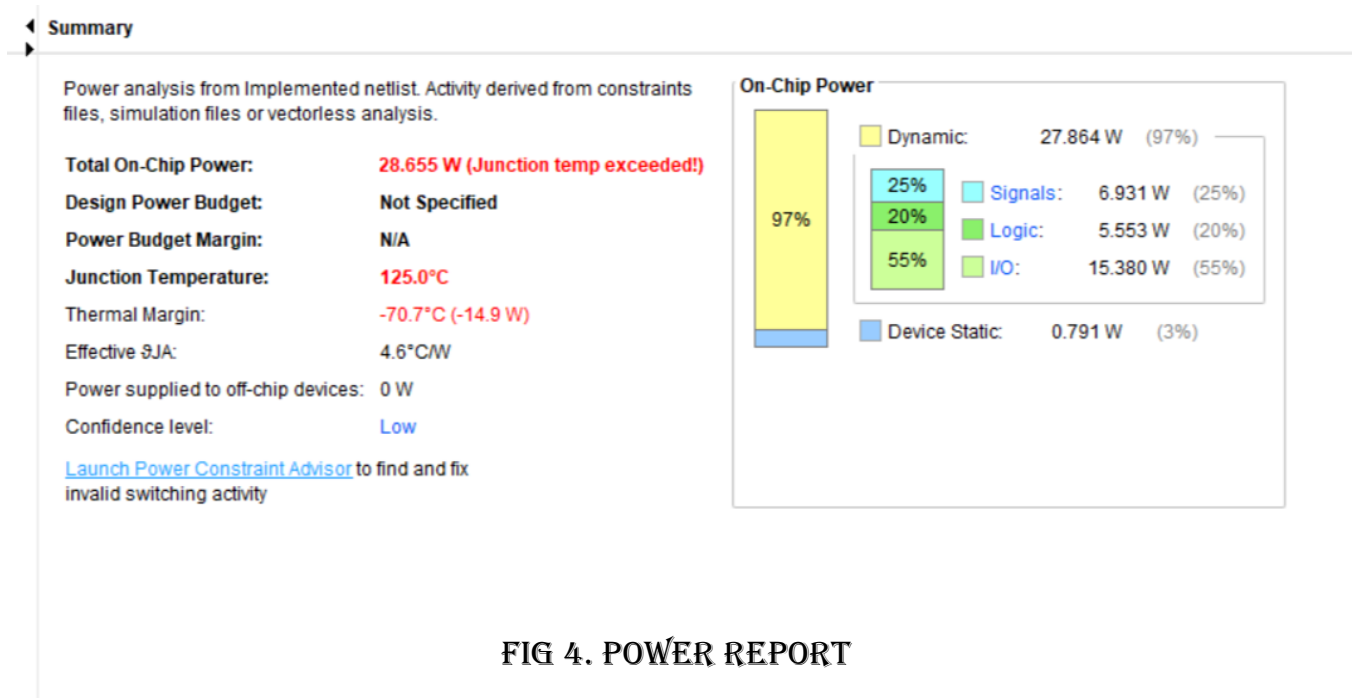


FIG 4. POWER REPORT

CONCLUSION

This project represents a thorough analysis of creating the functionality of ATM machine using Verilog HDL. This whole project was designed, tested and implemented virtually through Xilinx (VIVADO Version 17.2) software. Simulating the test bench code, we finally obtained the desired output and checked the features of the ATM machine we implemented. Consequently, we found that the power consumption for the whole function turns out to be 27.864 W.

UTILISATION	Slice LUTs	Slice Registers
	750	211
ON CHIP POWER	27.864 W	



THANK YOU!