MINI PROJECT REPORT

BANKING SERVER AND ATM

Patrick George(111601015)
Prabhjot Singh(111601016)
S.M.M.Sharief(111601022)

PROBLEM DESCRIPTION:

In this project,we have implemented a Banking server and ATM using an FPGA and Verilog.Users are given an option to choose Banking server or ATM using a user input.

The context:

- There are a total of four accounts.Each account can store a maximum of Rs.10000 and each can only be accessed by entering the correct 4 pin password.

In the Banking side,we have the following functionalities:

- User can add or remove an account.
- Money can be deposited in a particular account the user chooses.

- Pin of an account can be changed to a user given value.
- An account which was blocked due to wrong pin attempts in the ATM can be unblocked in the BANK.

ATM side:

- Entered PIN will be verified before granting access to the account for the withdrawal of money.
- Two options are given in the ATM side:Check balance or withdraw money.
- The balance will be displayed with the help of 4 seven segment displays.
- Money can be withdrawn in steps-the given denominations are Rs.1000,Rs.500,Rs.100,Rs.50.
- More than 3 wrong PIN attempts after choosing an account will result in the account getting blocked,which in turn can only be unblocked in the banking side.

Contribution of team members:

Patrick George: Implemented most of the ATM module.

Prabhjot Singh: Implemented most of the Banking module.

S.M.M Sharief: Thought of the logics and helped implementing both modules and hardware implementation.

INTRODUCTION : :

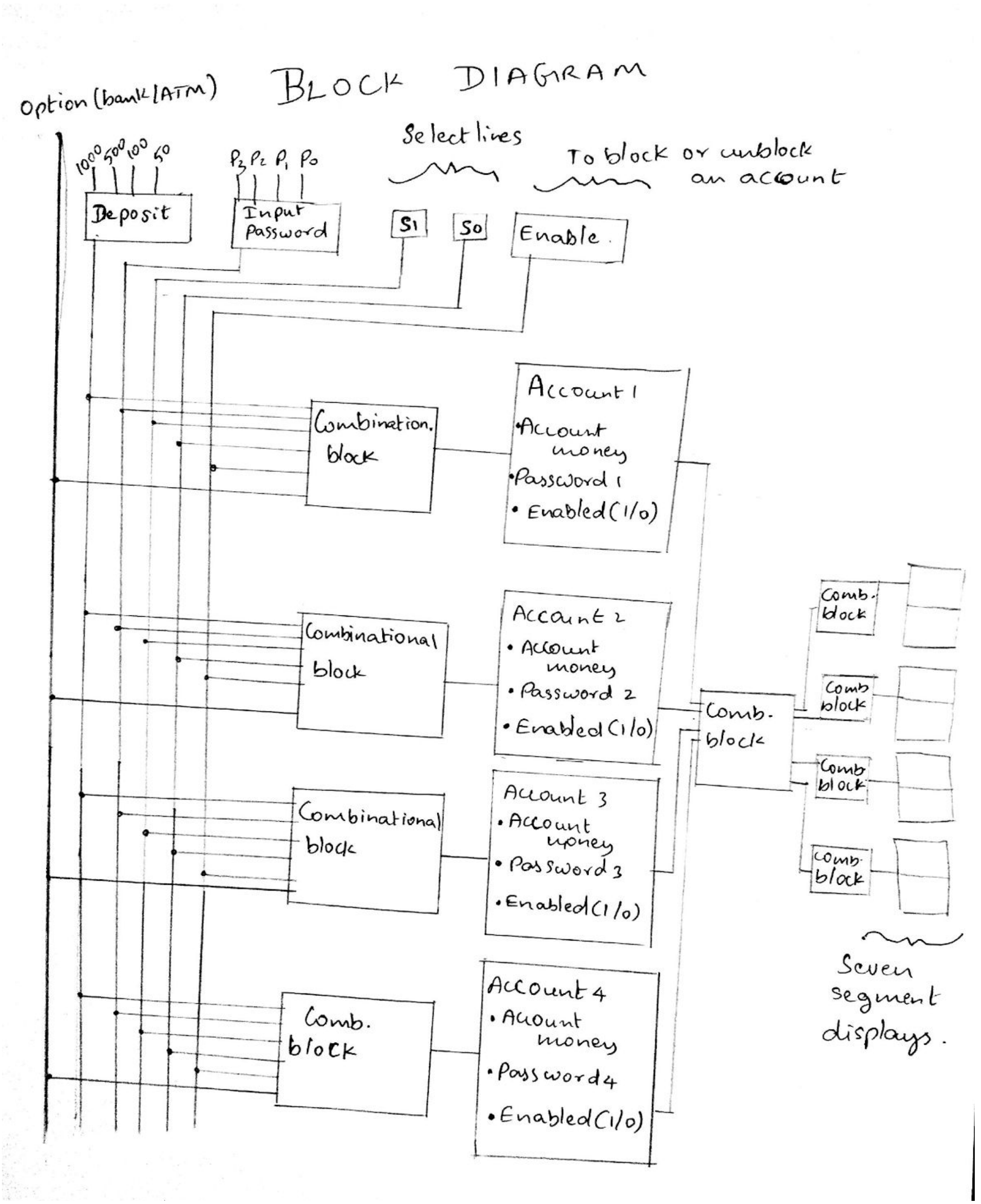
CIRCUIT BLOCK DIAGRAMS:

The circuit block diagram consists of two main parts-the ATM and the banking server.

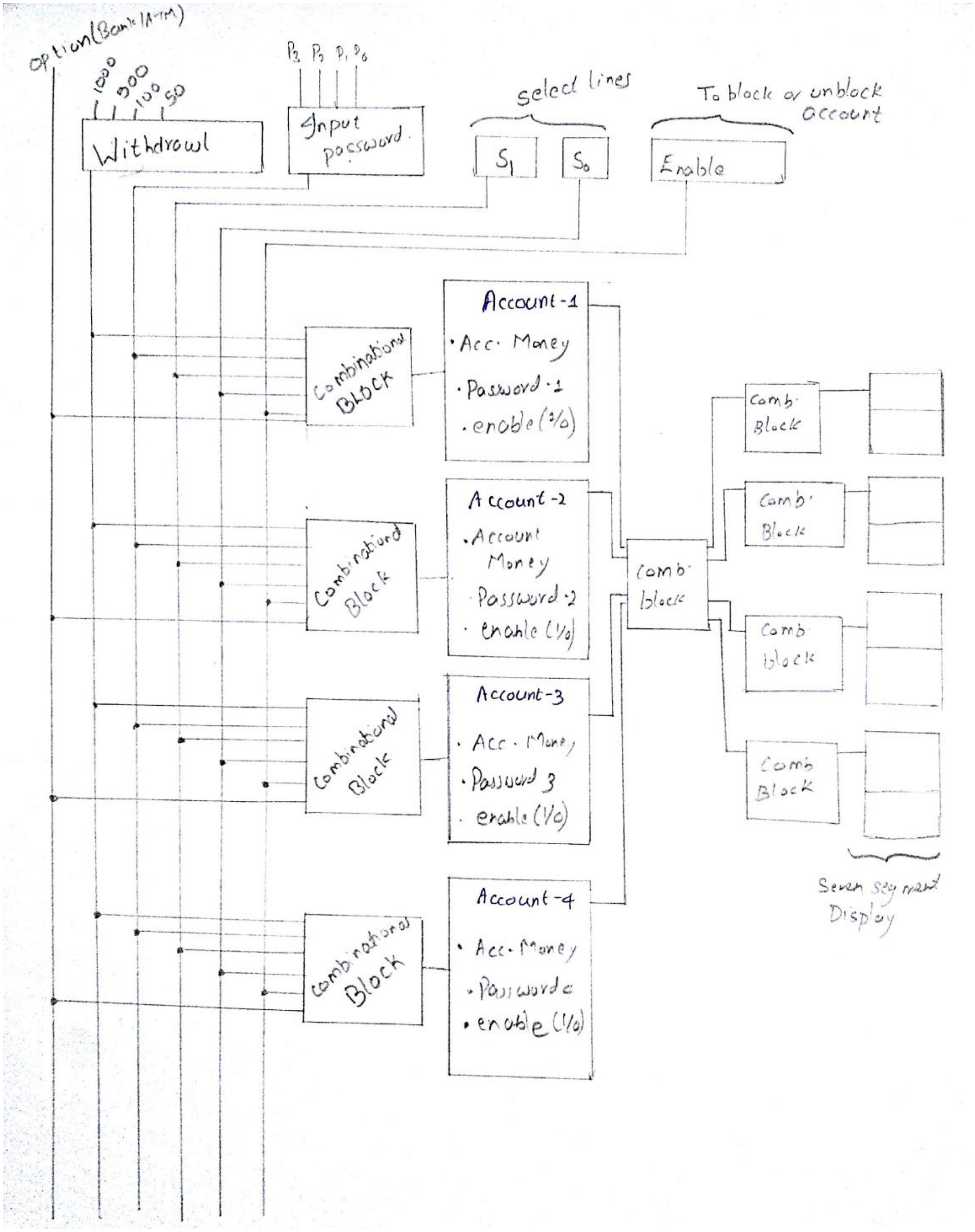
All the inputs given by the user goes into the combinational blocks and it determines how to make use of that information.

Finally,only seven outputs come out and a combinational block decides which SSD to send it to.

Block diagram of bank:



Block diagram of ATM:



STATE DIAGRAM:

Here,the state diagram given is related to the displaying of the account balance in the four seven segment displays.Here, for displaying in the four displays we are only using 7 outputs for the seven segments.We control the power input of each seven segment display as a function of time.

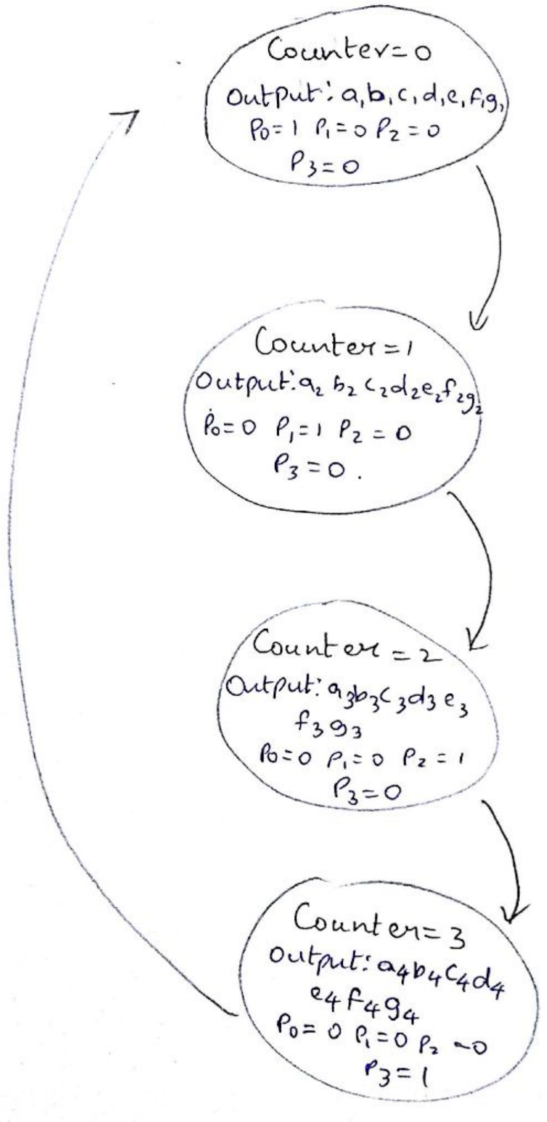
So,at a particular instant only one SSD is turned on.But,since the clock frequency is high,we can exploit the persistence of vision of humans to actually make it seem like every SSD is on at the same time.

This way,we use far less Pmods from the zybo board.

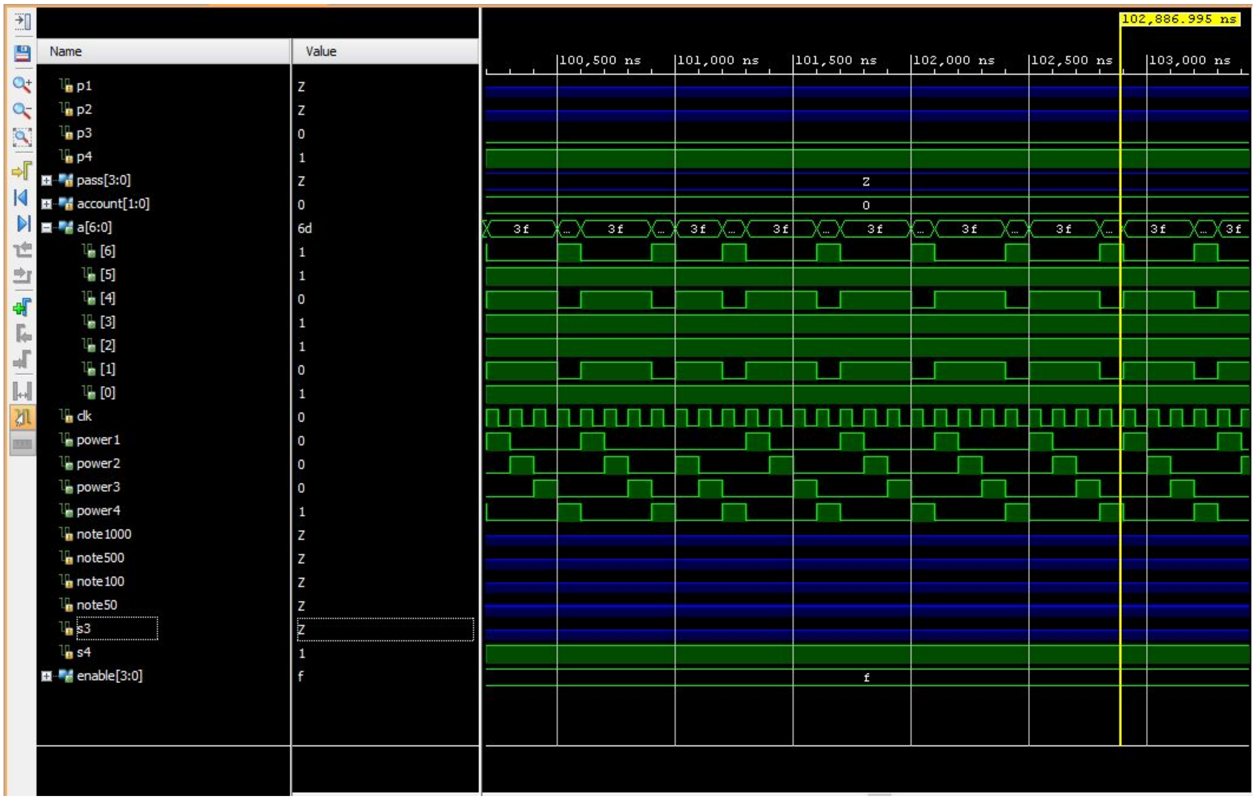
We have a counter that goes from 0 to 3 .Here a1,b1,c1,d1,e1,f1,g1 are the values to be displayed in the first SSD,while p0,p1,p2,p3 are the power input of the four SSDs respectively.

STATE DIAGRAM:

State diagram for the display in the four seven segment displays using multiplexing



Timing Diagram :



Name	Value
p1	Z
p2	Z
p3	0
p4	1
pass[3:0]	Z
account[1:0]	0
a[6:0]	6d
[6]	1
[5]	1
[4]	0
[3]	1
[2]	1
[1]	0
[0]	1
clk	0
power1	0
power2	0
power3	0
power4	1
note1000	Z
note500	Z
note100	Z
note50	Z
s3	Z
s4	1
enable[3:0]	f

Logic explanation:

Since we had to implement two modules ,i.e. Bank and ATM , we dedicated a zybo switch to choose between these two.

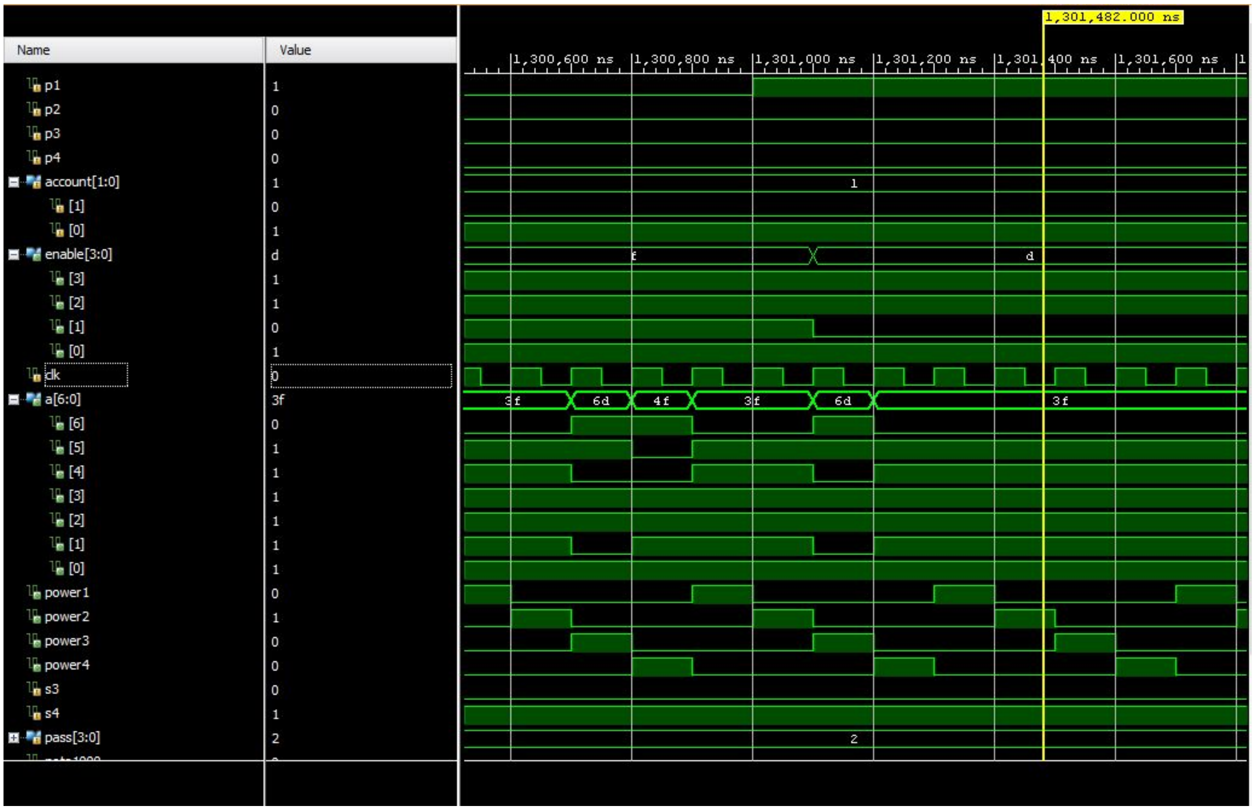
This also helped in making use of the same inputs from the main module for different options in Bank and ATM.

The logics for implementing various options have been explained in the relevant sections.

Simulations for some of the options :

For disabling an account in Bank:

Account 2 disabled ,i.e. enable[1] goes to 0.



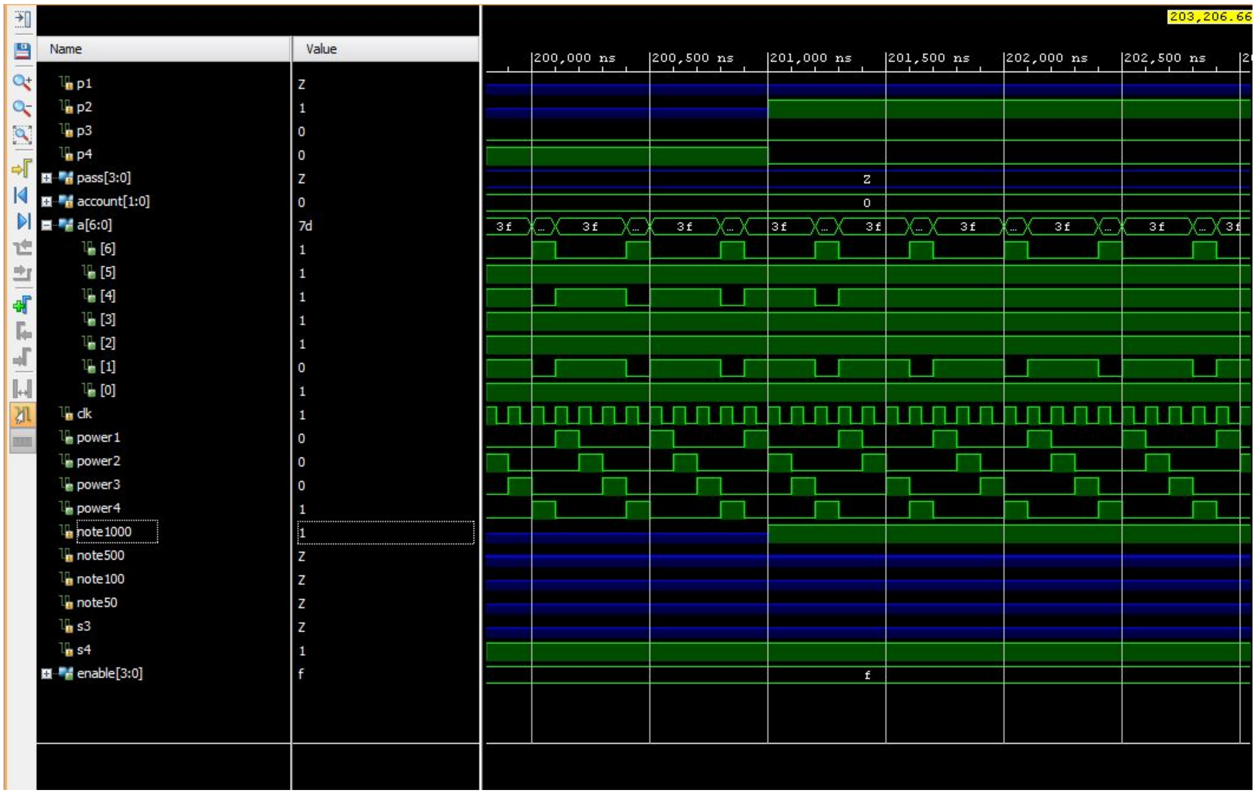
Name	Value
p1	1
p2	0
p3	0
p4	0
account[1:0]	1
[1]	0
[0]	1
enable[3:0]	d
[3]	1
[2]	1
[1]	0
[0]	1
clk	0
a[6:0]	3f
[6]	0
[5]	1
[4]	1
[3]	1
[2]	1
[1]	1
[0]	1
power1	0
power2	1
power3	0
power4	0
s3	0
s4	1
pass[3:0]	2

For depositing money to an account in bank:

When button p2 is pressed

Rs. 1000 are deposited to first account (can be seen in seven segment output)

Balance increases to Rs. 6000 from Rs. 5000



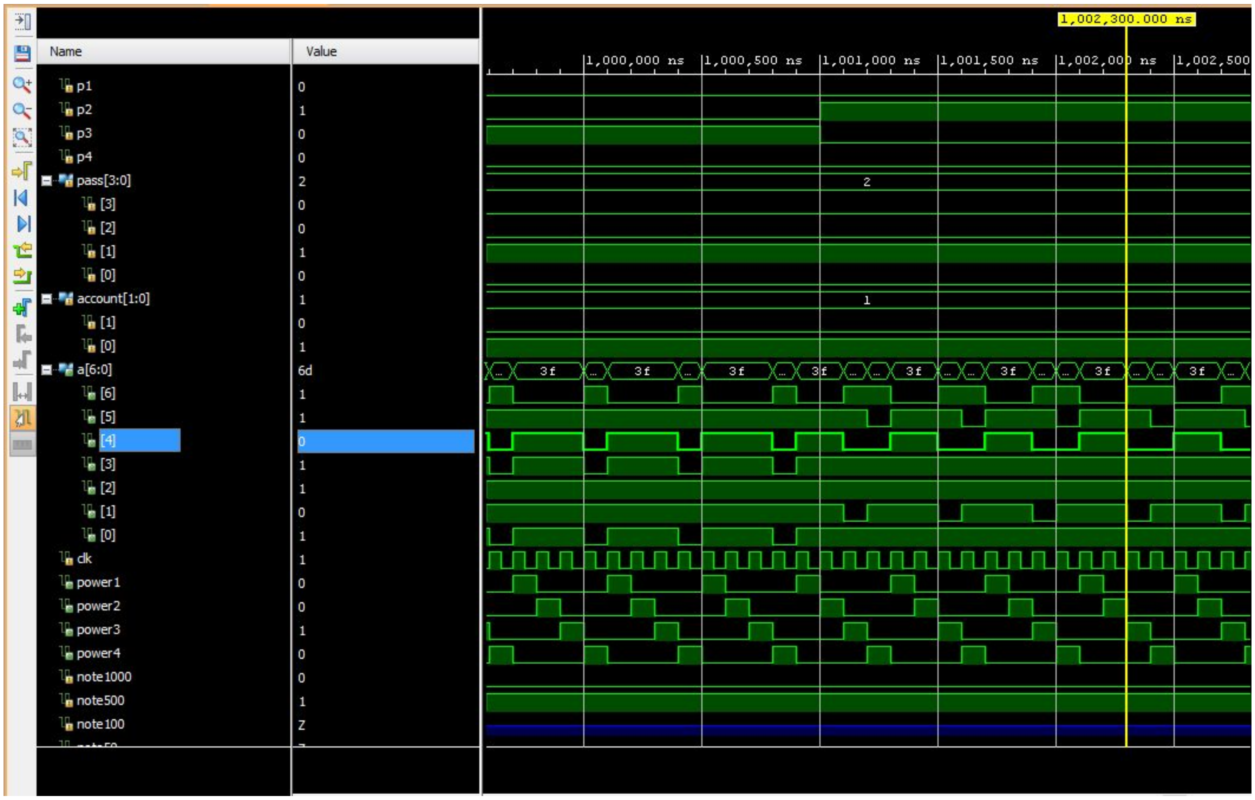
Name	Value
p1	Z
p2	1
p3	0
p4	0
pass[3:0]	Z
account[1:0]	0
a[6:0]	7d
[6]	1
[5]	1
[4]	1
[3]	1
[2]	1
[1]	0
[0]	1
clk	1
power1	0
power2	0
power3	0
power4	1
note1000	1
note500	Z
note100	Z
note50	Z
s3	Z
s4	1
enable[3:0]	f

Withdrawing money from ATM:

When button p2 is pressed:

Rs. 500 is withdrawn from account 3

Balance changes to Rs. 3500 from Rs. 4500 and is displayed on the seven segment display.

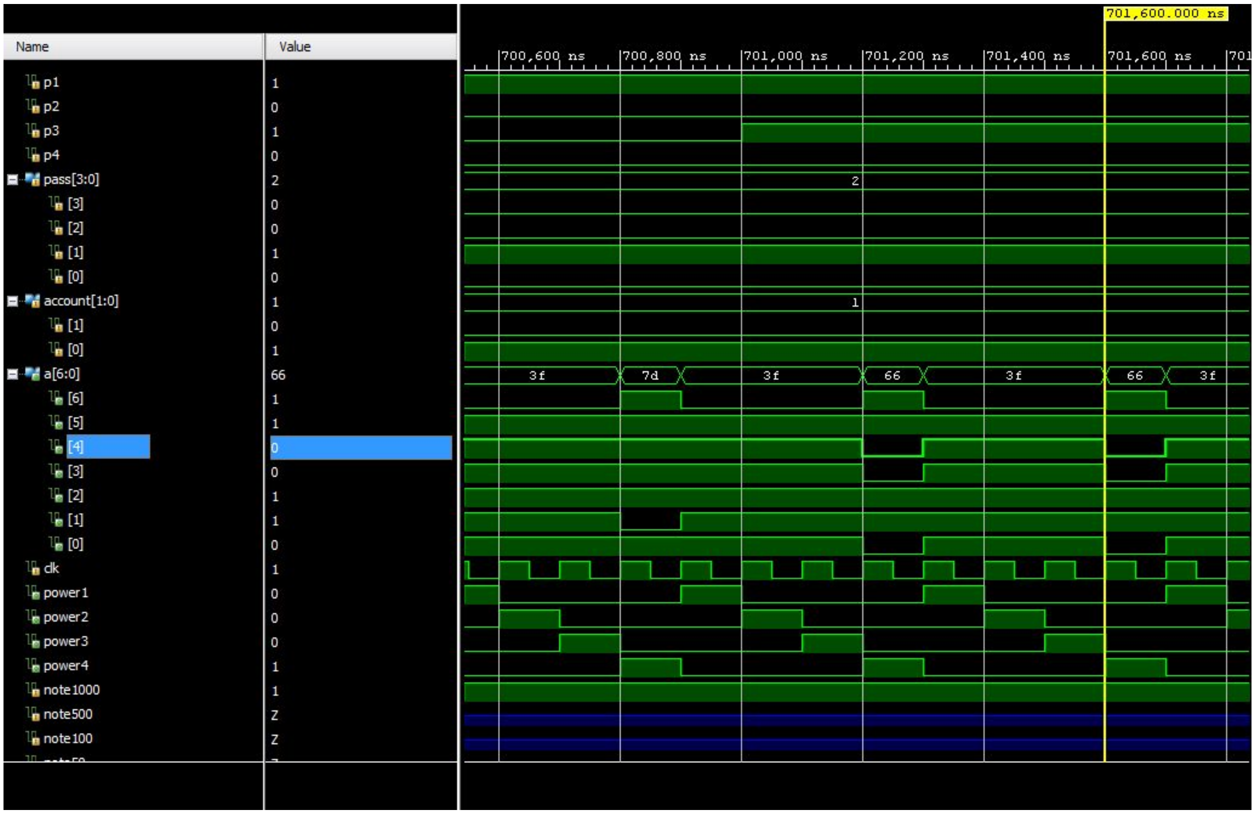


Name	Value
p1	0
p2	1
p3	0
p4	0
pass[3:0]	2
[3]	0
[2]	0
[1]	1
[0]	0
account[1:0]	1
[1]	0
[0]	1
a[6:0]	6d
[6]	1
[5]	1
[4]	0
[3]	1
[2]	1
[1]	0
[0]	1
clk	1
power1	0
power2	0
power3	1
power4	0
note1000	0
note500	1
note100	Z

Displaying balance in ATM :

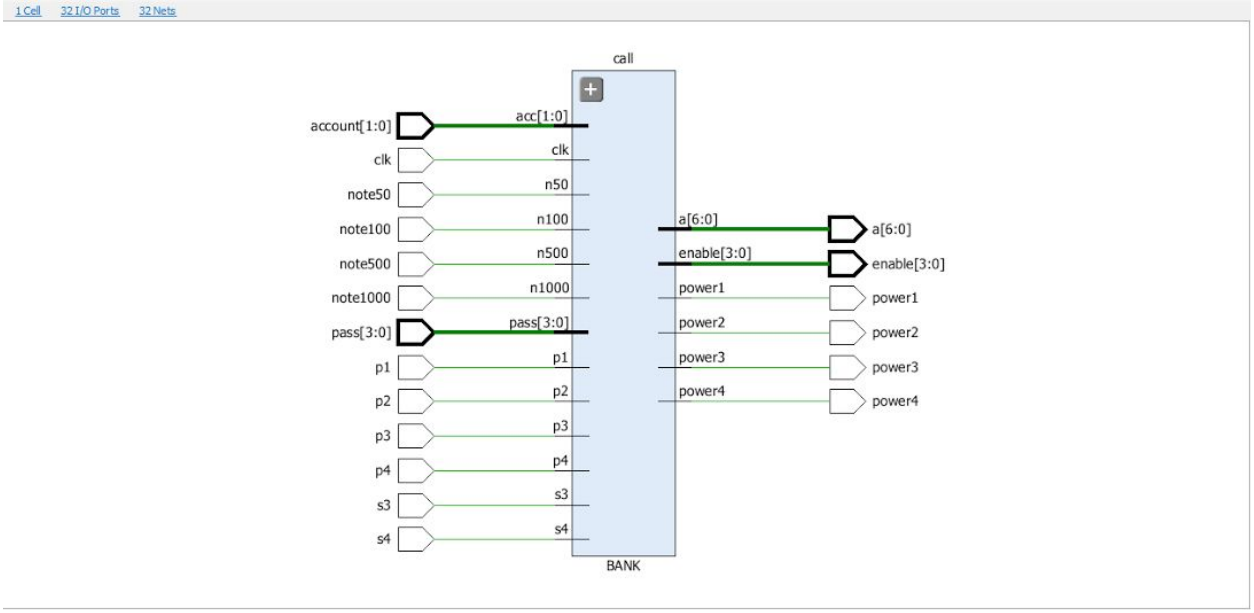
On pressing p1 after reset(p3)

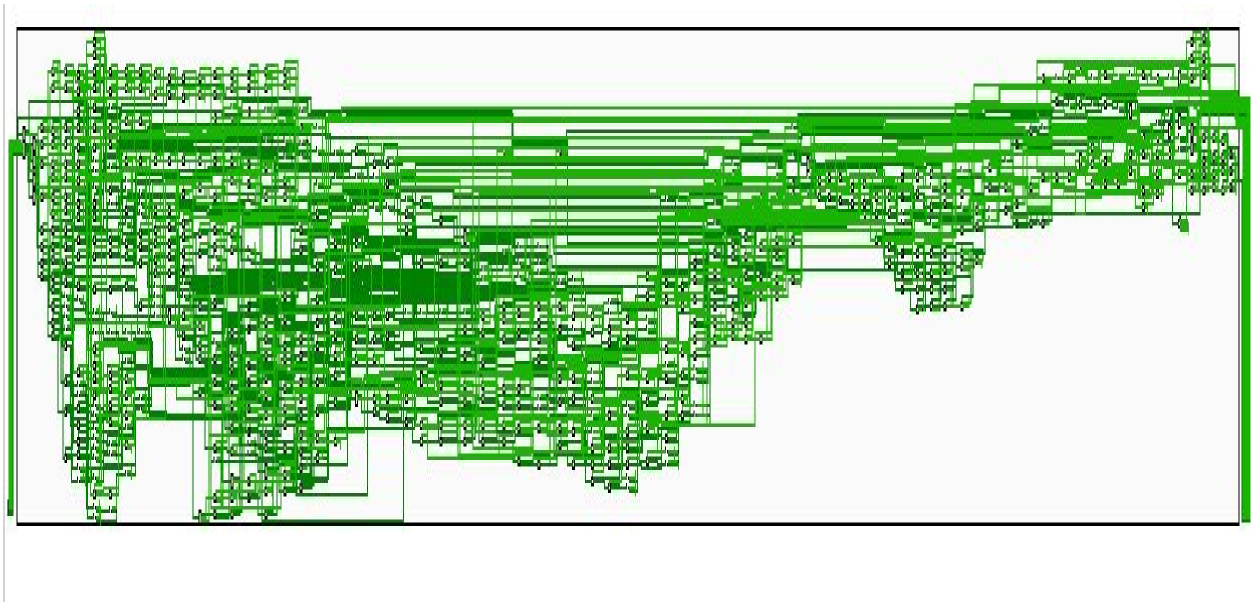
Account 2 balance is displayed on the seven segment display,i.e. Rs. 4000



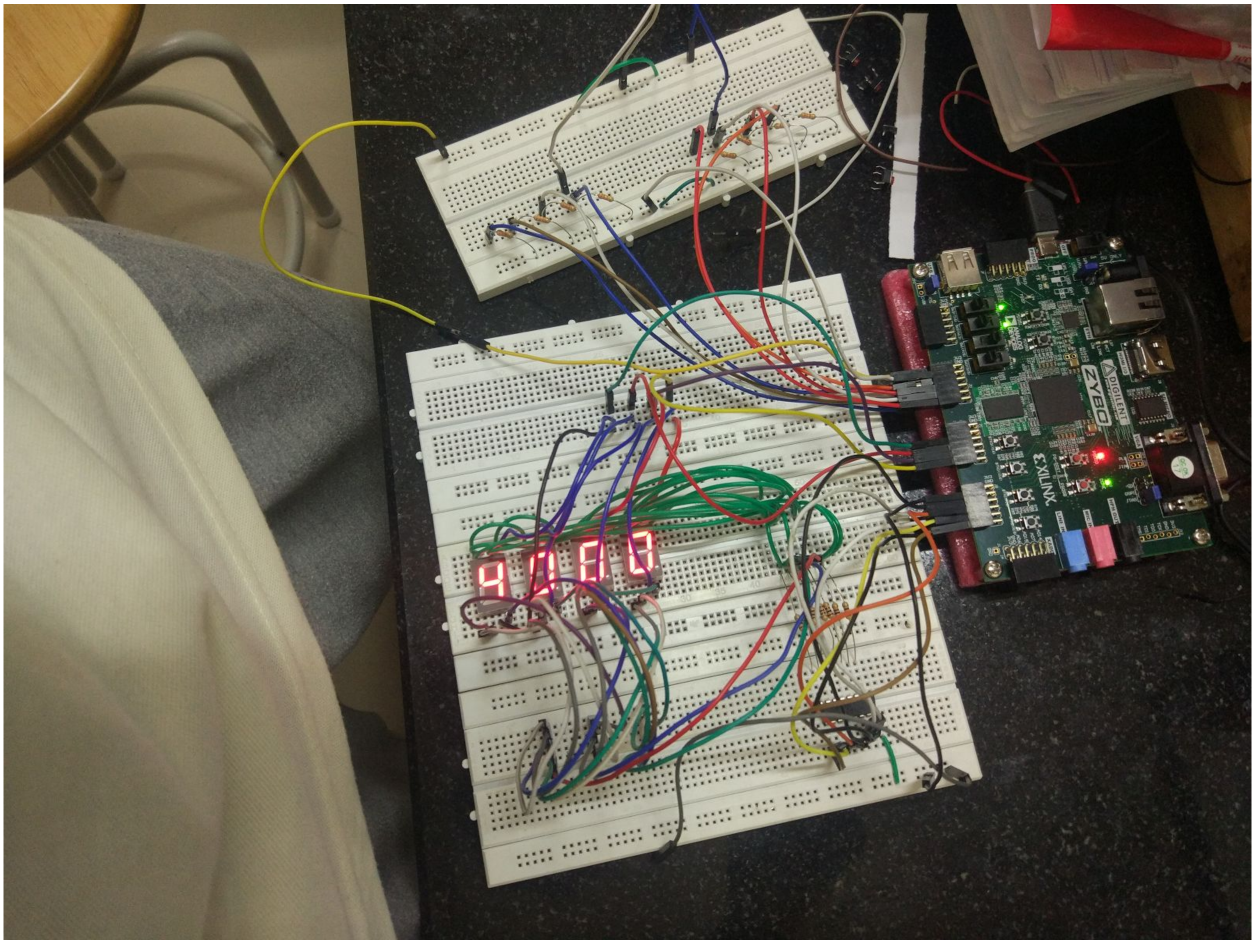
Name	Value
p1	1
p2	0
p3	1
p4	0
pass[3:0]	2
[3]	0
[2]	0
[1]	1
[0]	0
account[1:0]	1
[1]	0
[0]	1
a[6:0]	66
[6]	1
[5]	1
[4]	0
[3]	0
[2]	1
[1]	1
[0]	0
clk	1
power1	0
power2	0
power3	0
power4	1
note1000	1
note500	Z
note100	Z

RTL Schematic:





Hardware picture:



PROJECT CODE : :

Module for Bank and ATM : :

```
module BANK(

    input p1,
    input p2,
    input p3,
    input p4,

    input [3:0]pass,

    input [1:0]acc,

    input s3,
    input s4,
    output [6:0]a,   //for seven segment display

    input clk,

    output power1,
    output power2,
    output power3,
    output power4,

    input n1000,n500,n100,n50,
    output reg [3:0]enable
    );

    wire clk1;
            clk_c popo(clk,clk1);

    reg [13:0]acc1_money;
    reg [13:0]acc2_money;
    reg [13:0]acc3_money;
    reg [13:0]acc4_money;
```

```
 reg [3:0]dig1;   //stores the decimal digits of the account money
 reg [3:0]dig2;
 reg [3:0]dig3;
 reg [3:0]dig4;

 reg [13:0]acc1;
 reg [13:0]acc2;
 reg [13:0]acc3;
 reg [13:0]acc4;

 reg [3:0]pass1;
 reg [3:0]pass2;
 reg [3:0]pass3;
 reg [3:0]pass4;

 reg [13:0]withdraw;
 reg [3:0]dig;
 reg key;



 reg count;
// reg [3:0]enable;

 initial
  begin

      acc1 = acc1_money;
      acc2 = acc2_money;
      acc3 = acc3_money;
      acc4 = acc4_money;

      pass1= 'b0001;
      pass2= 'b0010;
      pass3= 'b0100;
      pass4= 'b1000;

      acc1_money= 'd5000;
      acc2_money= 'd4000;
```

```
         acc3_money= 'd3000;
         acc4_money= 'd1000;

         enable[0]=1;
         enable[1]=1;
         enable[2]=1;
         enable[3]=1;

         count=0;
         withdraw = 'b00000000000000;



         dig = 'd0;
         key = 'b0;

     end
     always @(posedge clk)
     begin
////////////////////
//bank module
//////////////////

if(s4==1)
begin

     //options available: (can choose option using
   //  buttons ( [3:0]push) )
             //1: add or remove an account
                 //initial
                 //begin

             //choose to remove/add using one switch (s3)
          if(p1==1)
          begin
             if(s3==0)   //remove account
             begin
                 //choose account no. using 2 switches(s0,s1);
                 if(acc=='b00)
```

```
    begin
        enable[0] = 0;
        acc1_money= 'd0;
    end
    else if(acc=='b01)
    begin
        enable[1] = 0;
        acc2_money= 'd0;
    end
    else if(acc=='b10)
    begin
        enable[2] = 0;
        acc3_money= 'd0;
    end
    else if(acc=='b11)
    begin
        enable[3] = 0;
        acc4_money= 'd0;
    end
end
else if(s3==1)      //add account
begin
    if(acc=='b00)
    begin
        enable[0] = 1;
    end
    else if(acc=='b01)
    begin
        enable[1] = 1;
    end
    else if(acc=='b10)
    begin
        enable[2] = 1;
    end
    else if(acc=='b11)
    begin
        enable[3] = 1;
    end
end
```

```
         end

   // 2: deposit money
    if(p2==1)
    begin
        //choose account no. using 2 switches(s0,s1);

        //take input for money using exeternal switches dedicated to
different denominations
        // external switches : n1000,n500,n100,n50
      if(count ==0)
      begin
        if(acc=='b00 && enable[0]==1)
        begin
            if( n1000 == 1)
                if(acc1_money<8999)
                    acc1_money <= acc1_money + 'd1000;
            if( n500 == 1  )
                if(acc1_money<9499)
                    acc1_money <= acc1_money + 'd500;
            if( n100==1 )
                if(acc1_money<9899)
                    acc1_money <= acc1_money + 'd100;
            if(n50==1 )
                if(acc1_money<9949)
                    acc1_money <= acc1_money + 'd50;
        end
        else if(acc=='b01&& enable[1]==1)
        begin
            if( n1000 == 1 )
                if(acc2_money<8999)
                    acc2_money <= acc2_money + 'd1000;
            if( n500 == 1   )
                if(acc2_money<9499)
                    acc2_money <= acc2_money + 'd500;
             if( n100==1 )
                if(acc2_money<9899)
                    acc2_money <= acc2_money + 'd100;
            if(n50==1 )
```

```
                if(acc2_money<9949)
                    acc2_money <= acc2_money + 'd50;
        end
        else if(acc=='b10&& enable[2]==1)
        begin
            if( n1000 == 1 )
                if(acc3_money<8999)
                    acc3_money <= acc3_money + 'd1000;
            if( n500 == 1 )
                if(acc3_money<9499)
                    acc3_money <= acc3_money + 'd500;
             if( n100==1)
                if(acc3_money<9899)
                    acc3_money <= acc3_money + 'd100;
            if(n50==1 )
                if(acc3_money<9949)
                    acc3_money <= acc3_money + 'd50;
        end
        else if(acc=='b11&& enable[3]==1)
        begin
            if( n1000 == 1 )
                if(acc4_money<8999)
                    acc4_money <= acc4_money + 'd1000;
           if( n500 == 1  )
                if(acc4_money<9499)
                    acc4_money <= acc4_money + 'd500;
             if( n100==1)
                if(acc4_money<9899)
                    acc4_money <= acc4_money + 'd100;
            if(n50==1)
                if(acc4_money<9949)
                    acc4_money <= acc4_money + 'd50;
        end
        count=1;
        end
    end

/////////
//reset
```

```
/////////
    if(p4 == 1)
    begin
        count=0;
        acc1 = acc1_money;
        acc2 = acc2_money;
        acc3 = acc3_money;
        acc4 = acc4_money;
    end


  //  3: changing the pin of an account
       if(p3==1)
            //choose account no. using 2 switches(s0,s1);
           begin
            if(acc=='b00)
                 pass1<=pass;
            else if(acc=='b01)
                 pass2<=pass;
            else if(acc=='b10)
                 pass3<=pass;
            else if(acc=='b11)
                 pass4<=pass;
        end




        if(acc == 'b00)
                    begin
                       // dig = dig1;
                       dig1 = acc1%'d10;
                       acc1 = acc1/10;
                       dig2 = acc1%'d10;
                       acc1 = acc1/10;
                       dig3 = acc1%'d10;
                       acc1 = acc1/10;
                       dig4 = acc1%'d10;
                       acc1 = acc1/10;
                       acc1 = acc1_money;

                    end
                if(acc == 'b01)
                    begin
                        //dig = dig2;
                        dig1 = acc2%'d10;
                        acc2 = acc2/10;
                        dig2 = acc2%'d10;
                        acc2 = acc2/10;
                        dig3 = acc2%'d10;
                        acc2 = acc2/10;
                        dig4 = acc2%'d10;
                        acc2 = acc2/10;

                        acc2 = acc2_money;

                    end
                if(acc == 'b10)
                    begin
                        //dig = dig3;
                        dig1 = acc3%'d10;
                        acc3 = acc3/10;
                        dig2 = acc3%'d10;
                        acc3 = acc3/10;
                        dig3 = acc3%'d10;
                        acc3 = acc3/10;
                        dig4 = acc3%'d10;
                        acc3 = acc3/10;

                        acc3 = acc3_money;

                    end
                if(acc == 'b11)
                    begin
                        //dig = dig4;
                        dig1 = acc4%'d10;
                        acc4 = acc4/10;
```

```
                        dig2 = acc4%'d10;
                        acc4 = acc4/10;
                        dig3 = acc4%'d10;
                        acc4 = acc4/10;
                        dig4 = acc4%'d10;
                        acc4 = acc4/10;

                        acc4 = acc4_money;

                    end
    end



else if(s4 == 0 )
begin
////////////////
//atm module
/////////////////

if( n1000 == 1 && key == 0)
begin
     withdraw = withdraw + 'd1000;
     key = 'b1;
end

if(n500 == 1 && key == 0)
 begin
        withdraw = withdraw + 'd500;
        key = 'b1;
   end
if( n100 == 1 && key == 0)
    begin
           withdraw = withdraw + 'd100;
           key = 'b1;
      end
if(n50 == 1 && key == 0)
    begin
          withdraw = withdraw + 'd50;
          key = 'b1;
     end

if( p1 ==1 )    //For just displaying the account balance
begin
    if(count == 0)
    begin


            count = 1;
            if(pass == pass1 && acc == 'b00 && enable[0]==1)
                begin

                    dig1 = acc1%'d10;
                    acc1 = acc1/10;
                    dig2 = acc1%'d10;
                    acc1 = acc1/10;
                    dig3 = acc1%'d10;
                    acc1 = acc1/10;
                    dig4 = acc1%'d10;
                    acc1 = acc1/10;

                   // dig = dig1;
                end
              else if(pass == pass1 && acc == 'b00 && enable[0]==0)
              begin
                    dig1='d0;
                    dig2='d0;
                    dig3='d0;
                    dig4='d0;
              end

            if(pass == pass2 && acc == 'b01 && enable[1]==1)
                begin
                dig1 = acc2%'d10;
                acc2 = acc2/10;
                dig2 = acc2%'d10;
                acc2 = acc2/10;
                dig3 = acc2%'d10;
                acc2 = acc2/10;
                dig4 = acc2%'d10;
                acc2 = acc2/10;

                    //dig = dig2;
                end
                else if(pass == pass2 && acc == 'b01 && enable[1]==0)
                              begin
                                    dig1='d0;
                                    dig2='d0;
                                    dig3='d0;
                                    dig4='d0;
                              end


            if(pass == pass3 && acc == 'b10 && enable[2]==1)
                begin
                dig1 = acc3%'d10;
                acc3 = acc3/10;
                dig2 = acc3%'d10;
                acc3 = acc3/10;
                dig3 = acc3%'d10;
                acc3 = acc3/10;
                dig4 = acc3%'d10;
                acc3 = acc3/10;
                    //dig = dig3;
                end
                else if(pass == pass3 && acc == 'b10 && enable[2]==0)
                              begin
                                    dig1='d0;
                                    dig2='d0;
                                    dig3='d0;
                                    dig4='d0;
                              end

            if(pass == pass4 && acc == 'b11 && enable[3]==1)

                begin
                dig1 = acc4%'d10;
```

```
                acc4 = acc4/10;
                dig2 = acc4%'d10;
                acc4 = acc4/10;
                dig3 = acc4%'d10;
                acc4 = acc4/10;
                dig4 = acc4%'d10;
                acc4 = acc4/10;
                    //dig = dig4;
                end
                else if(pass == pass4 && acc == 'b1 && enable[3]==0)
                              begin
                                    dig1='d0;
                                    dig2='d0;
                                    dig3='d0;
                                    dig4='d0;
                              end

    end

end
if( p3 == 1)     //This is like a reset
begin
    count = 0;
     acc1 = acc1_money;
      acc2 = acc2_money;
      acc3 = acc3_money;
      acc4 = acc4_money;
    withdraw = 'b00000000000000;
    key = 'b0;

end

if( p2 ==1 )     //For withdrawing the money
    begin
        if(count == 0)
        begin

            if(pass == pass1 && acc == 'b00 && enable[0]==1 &&
acc1_money>=withdraw)
```

```
                begin
                    acc1_money=acc1_money - withdraw;
                    acc1 = acc1_money;
                end

            if(pass == pass2 && acc == 'b01 && enable[1]==1 &&
acc2_money>=withdraw)
                begin
                    acc2_money=acc2_money - withdraw;
                    acc2 = acc2_money;
                end
                else if(pass!=pass2)

            if(pass == pass3 && acc == 'b10 && enable[2]==1 &&
acc3_money>=withdraw)
            begin
                acc3_money=acc3_money - withdraw;

                acc3 = acc3_money;
            end

            if(pass == pass4 && acc == 'b11 && enable[3]== 1&&
acc4_money>=withdraw)
            begin
                acc4_money=acc4_money - withdraw;

                acc4 = acc4_money;
            end


            if(pass == pass1 && acc == 'b00)
                begin
                   // dig = dig1;
                   dig1 = acc1%'d10;
                   acc1 = acc1/10;
                   dig2 = acc1%'d10;
                   acc1 = acc1/10;
                   dig3 = acc1%'d10;
                   acc1 = acc1/10;
```

```
       dig4 = acc1%'d10;
       acc1 = acc1/10;
        count = 1;
    end
if(pass == pass2 && acc == 'b01)
    begin
        //dig = dig2;
        dig1 = acc2%'d10;
        acc2 = acc2/10;
        dig2 = acc2%'d10;
        acc2 = acc2/10;
        dig3 = acc2%'d10;
        acc2 = acc2/10;
        dig4 = acc2%'d10;
        acc2 = acc2/10;
         count = 1;

    end
if(pass == pass3 && acc == 'b10)
    begin
        //dig = dig3;
        dig1 = acc3%'d10;
        acc3 = acc3/10;
        dig2 = acc3%'d10;
        acc3 = acc3/10;
        dig3 = acc3%'d10;
        acc3 = acc3/10;
        dig4 = acc3%'d10;
        acc3 = acc3/10;
        count = 1;

    end
if(pass == pass4 && acc == 'b11)
    begin
        //dig = dig4;
        dig1 = acc4%'d10;
        acc4 = acc4/10;
        dig2 = acc4%'d10;
        acc4 = acc4/10;
```

```
                    dig3 = acc4%'d10;
                    acc4 = acc4/10;
                    dig4 = acc4%'d10;
                    acc4 = acc4/10;
                    count = 1;
                end
        end
end


end
end

display inst_1(a,dig1,dig2,dig3,dig4,clk1,power1,power2,power3,power4);
//For the seven segment displays


endmodule
```

Module clock : :

```
module clk_c(clk,clk1); //We have to get the desired frequency
//to clk1 from given 125 Mhz
// inbuilt clock frequency
input clk;
output clk1;
reg clk1;
reg [26:0]q;
 initial
 begin
 q<=27'd000000000;
 clk1<=0;
 end
 always@(posedge clk)
 begin
 if(q==27'd12500)
 begin
 clk1<=~clk1;
 q<=27'd000000000;
 end
 else
 begin
 q<=q+1;
 end
 end
endmodule
```

Module Display : :

```
module display(
    output [6:0]a,
    input [3:0]digit1,
    input [3:0]digit2,
    input [3:0]digit3,
    input [3:0]digit4,
    input clk,
    output reg power1,
    output reg power2,
    output reg power3,
    output reg power4
    );  //Here a[0] = a,a[1] = b etc

inst_1(a,dig1,dig2,dig3,dig4,clk1,power1,power2,power3,power4);

    reg [1:0]counter;
    initial
    begin
        counter = 'b00;
    end
    reg p,b,c,d;
```

```
assign a[0]= p | c | b&d | ~b&~d;
assign a[1]= ~b | ~c&~d | c&d ;
assign a[2]= b | ~c | d;
assign a[3]= p | ~b&c | ~c&b&d | c&~d | ~b&~d;
assign a[4]= c&~d | ~b&~d;
assign a[5]= p | ~d&~c | b&~c | b&~d;
assign a[6]= p | ~b&c | c&~d | b&~c;


always@(posedge clk)
begin

        if(counter==0)
        begin
               power1=1;
               power2=0;
               power3=0;
               power4=0;

               p=digit1[3];
               b=digit1[2];
               c=digit1[1];
               d=digit1[0];

               counter = 'b01;
        end
       else if(counter==1)
                    begin
                           power1=0;
                           power2=1;
                           power3=0;
                           power4=0;

                           p=digit2[3];
                           b=digit2[2];
                           c=digit2[1];
                           d=digit2[0];
```

```
                          counter = 'b10;
                        end
                       else if(counter==2)
                                    begin
                                           power1=0;
                                           power2=0;
                                           power3=1;
                                           power4=0;

                                           p=digit3[3];
                                           b=digit3[2];
                                           c=digit3[1];
                                           d=digit3[0];

                                              counter = 'b11;

                                    end
                                    else if(counter==3)
                                                begin
                                                       power1=0;
                                                       power2=0;
                                                       power3=0;
                                                       power4=1;

                                                       p=digit4[3];
                                                       b=digit4[2];
                                                       c=digit4[1];
                                                       d=digit4[0];

                                                      counter = 'b00;

                                                end
    end

endmodule
```

Main Module : :

```
module kakashi(
    input p1,p2,p3,p4,             //zybo push buttons

    input [3:0]pass,              //external switches to enter password
    input [1:0]account,           //Selecting the account (first two zybo
switches

    output [6:0]a,   //for seven segment display

    input clk,         //zybo clock

    output power1,power2,power3,power4        //power for the seven
segment displays

    input note1000,note500,note100,note50,   //input for denominations

    input s3,s4,                  // zybo switches for separate functions
in Bank and ATM

    output [3:0]enable            //output to LEDs for the enabled
accounts
    );


BANK
call(p1,p2,p3,p4,pass,account,s3,s4,a,clk,power1,power2,power3,power4,note
1000,note500,note100,note50,enable);

endmodule
```

Constraint file:

```
#zybo push buttons
set_property -dict {PACKAGE_PIN Y16 IOSTANDARD LVCMOS33 } [get_ports
{p1}];
set_property -dict {PACKAGE_PIN V16 IOSTANDARD LVCMOS33 } [get_ports
{p2}];
set_property -dict {PACKAGE_PIN P16 IOSTANDARD LVCMOS33 } [get_ports
{p3}];
set_property -dict {PACKAGE_PIN R18 IOSTANDARD LVCMOS33 } [get_ports
{p4}];
```

```
#set_property -dict {PACKAGE_PIN R18 IOSTANDARD LVCMOS33 } [get_ports
{p3}];

#zybo first two switches
set_property -dict {PACKAGE_PIN W13 IOSTANDARD LVCMOS33 } [get_ports
{account[0]}];
set_property -dict {PACKAGE_PIN T16 IOSTANDARD LVCMOS33 } [get_ports
{account[1]}];

set_property -dict {PACKAGE_PIN P15 IOSTANDARD LVCMOS33 } [get_ports
{s3}];
set_property -dict {PACKAGE_PIN G15 IOSTANDARD LVCMOS33 } [get_ports
{s4}];

#enalbe leds (on zybo)

set_property -dict {PACKAGE_PIN D18 IOSTANDARD LVCMOS33 } [get_ports
{enable[0]}];
set_property -dict {PACKAGE_PIN G14 IOSTANDARD LVCMOS33 } [get_ports
{enable[1]}];
set_property -dict {PACKAGE_PIN M15 IOSTANDARD LVCMOS33 } [get_ports
{enable[2]}];
set_property -dict {PACKAGE_PIN M14 IOSTANDARD LVCMOS33 } [get_ports
{enable[3]}];

#zybo clock
set_property -dict {PACKAGE_PIN L16 IOSTANDARD LVCMOS33 } [get_ports
{clk}];

#JB ( seven segment )
set_property -dict {PACKAGE_PIN T20 IOSTANDARD LVCMOS33 } [get_ports
{a[0]}];
set_property -dict {PACKAGE_PIN U20 IOSTANDARD LVCMOS33 } [get_ports
{a[1]}];
set_property -dict {PACKAGE_PIN V20 IOSTANDARD LVCMOS33 } [get_ports
{a[2]}];
set_property -dict {PACKAGE_PIN W20 IOSTANDARD LVCMOS33 } [get_ports
{a[3]}];
set_property -dict {PACKAGE_PIN Y18 IOSTANDARD LVCMOS33 } [get_ports
{a[4]}];
set_property -dict {PACKAGE_PIN Y19 IOSTANDARD LVCMOS33 } [get_ports
{a[5]}];
set_property -dict {PACKAGE_PIN W18 IOSTANDARD LVCMOS33 } [get_ports
{a[6]}];

#JC  ( multiplex power to seven segment)
set_property -dict {PACKAGE_PIN V15 IOSTANDARD LVCMOS33 } [get_ports
{power1}];
set_property -dict {PACKAGE_PIN W15 IOSTANDARD LVCMOS33 } [get_ports
{power2}];
set_property -dict {PACKAGE_PIN T11 IOSTANDARD LVCMOS33 } [get_ports
{power3}];
set_property -dict {PACKAGE_PIN T10 IOSTANDARD LVCMOS33 } [get_ports
{power4}];

#JD  1st row     ( external switches )
set_property -dict {PACKAGE_PIN T14 IOSTANDARD LVCMOS33 } [get_ports
{pass[0]}];
set_property -dict {PACKAGE_PIN T15 IOSTANDARD LVCMOS33 } [get_ports
{pass[1]}];
set_property -dict {PACKAGE_PIN P14 IOSTANDARD LVCMOS33 } [get_ports
{pass[2]}];
set_property -dict {PACKAGE_PIN R14 IOSTANDARD LVCMOS33 } [get_ports
{pass[3]}];

#JD 2nd row ( external push buttons )
set_property -dict {PACKAGE_PIN U14 IOSTANDARD LVCMOS33 } [get_ports
{note1000}];
set_property -dict {PACKAGE_PIN U15 IOSTANDARD LVCMOS33 } [get_ports
{note500}];
set_property -dict {PACKAGE_PIN V17 IOSTANDARD LVCMOS33 } [get_ports
{note100}];
set_property -dict {PACKAGE_PIN V18 IOSTANDARD LVCMOS33 } [get_ports
{note50}];
```

CONCLUSION : :

In this project,we learnt how to make use of many modules and integrate it to make it work in a synchronous way.We also found a way to reduce the number of Pmods by multiplexing the outputs for the four seven segment displays.We made use of the same switches for inputs for different modules according to our requirements.Overall,it was quite a job to write the code but the hardware implementation was moderately easy.