```

module ATM_FSM(clk,card_insert,reset,pin_1,pin_2,pin_3,menu_sel,
               amount,acc_num,select_options,face,otp,cash_dis,
               bal_D,print_receipt,CNL,card_lock,mini_s
               );

  input wire clk, card_insert, reset;
  input wire[3:0] pin_1;
  input wire[3:0] pin_2;
  input wire[3:0] pin_3;
  input wire[1:0] menu_sel;   //menu selection
  input wire[15:0] amount;    //withdrawal or deposit amount
  input wire[3:0] face;       //face recoginition
  input wire CNL;             //cancel
  input wire select_options;
  input wire[7:0]acc_num;     //account number for pin generation
  input wire [7:0] otp;
  output reg cash_dis, bal_D, print_receipt,card_lock,mini_s;        //cash dispending
                                                                     //balance display
                                                                     //print_receipt
(ack) for deposit
                                                                     //mini-statement
                                                                     //card lock for 24
hours

  integer i,counter;
  integer t_count;

//helper signals

  reg [1:0]pin_valid[1:3];
  reg acc_valid;
  reg otp_valid,face_valid;

// States definition

parameter s0 = 4'b0000;
parameter s1 = 4'b0001;
parameter s2 = 4'b0010;
parameter s3 = 4'b0011;
parameter s4 = 4'b0100;
parameter s5 = 4'b0101;
parameter s6 = 4'b0110;
parameter s7 = 4'b0111;
parameter s8 = 4'b1000;
parameter s9 = 4'b1001;
parameter s10 = 4'b1010;
parameter s11 = 4'b1011;
parameter s12 = 4'b1100;
parameter s13 = 4'b1101;
parameter s14 = 4'b1110;
parameter s15 = 4'b1111;


reg [3:0] currState, nextState;
reg [3:0] acc_index,acc_index_Mpin;
reg EC;                             //EC:eject card
reg [31:0] balance_db [0:9];
reg [7:0] otp_db [0:9];
reg [3:0] pin_db [0:9];
reg [3:0] face_db [0:9];
reg [7:0] acc_db [0:9];
reg [15:0] transaction [0:4];
reg [15:0] balance [0:4];


initial begin
i=0;
counter=0;
t_count=0;
pin_valid[1]=2'b00;
pin_valid[2]=2'b00;
pin_valid[3]=2'b00;

// customer accounts, balance and other confidential details initialization

              acc_db[0] = 8'd1234;
              acc_db[1] = 8'd2345;
              acc_db[2] = 8'd3456;
              acc_db[3] = 8'd4567;
              acc_db[4] = 8'd5678;
              acc_db[5] = 8'd6789;
              acc_db[6] = 8'd7890;
              acc_db[7] = 8'd8901;
              acc_db[8] = 8'd9012;
              acc_db[9] = 8'd1230;

              balance_db[0] = 31'd50000;
              balance_db[1] = 31'd500;
              balance_db[2] = 31'd500;
              balance_db[3] = 31'd500;
              balance_db[4] = 31'd500;
              balance_db[5] = 31'd500;
              balance_db[6] = 31'd50000;
              balance_db[7] = 31'd500;
              balance_db[8] = 31'd500;
              balance_db[9] = 31'd500;

              pin_db[0] = 4'b1111;
              pin_db[1] = 4'b0001;
              pin_db[2] = 4'b0010;
              pin_db[3] = 4'b0011;
              pin_db[4] = 4'b0100;
              pin_db[5] = 4'b0101;
              pin_db[6] = 4'b0110;
              pin_db[7] = 4'b0111;
              pin_db[8] = 4'b1000;
              pin_db[9] = 4'b1001;

              //storing face data in digital format

              face_db[0] = 4'b1111;
              face_db[1] = 4'b0001;
              face_db[2] = 4'b0010;
              face_db[3] = 4'b0011;
              face_db[4] = 4'b0100;
              face_db[5] = 4'b0101;
              face_db[6] = 4'b0110;
              face_db[7] = 4'b0111;
              face_db[8] = 4'b1000;
              face_db[9] = 4'b1001;

              otp_db[0] = 8'd2749;
              otp_db[1] = 8'd2175;
              otp_db[2] = 8'd2429;
              otp_db[3] = 8'd2125;
              otp_db[4] = 8'd2178;
              otp_db[5] = 8'd2647;
              otp_db[6] = 8'd2816;
              otp_db[7] = 8'd2910;
              otp_db[8] = 8'd2299;
              otp_db[9] = 8'd2689;

end


//sequential logic block for state transistion

always @ (posedge clk)
begin
      if(reset == 1'b1)
         currState <= #1 s0;
      else
         currState <= #1 nextState;
end

//combinational block for next state logic

always @ (*)
begin

   nextState = currState;

   case(currState)
```

```

        s0:if(card_insert & !card_lock & !EC)
            nextState = s8;
            else
            nextState = s0;

        s1:if(otp_valid)
           nextState = s15;
           else
           nextState = s0;

        s2:
           case(menu_sel)
           2'b00 : nextState = s6;
           2'b01 : nextState = s3;
           2'b10 : nextState = s14;
           default:nextState = s2;
           endcase

        s3:nextState = s0;

        s4:if(pin_valid[1] == 2'b11)
           nextState =  s2 ;
           else if(pin_valid[1] == 2'b01)
           nextState = s9;

        s5:if(otp_valid)
           nextState = s12;
           else
           nextState = s0;

        s6:if(!CNL)
           nextState = s7;
           else if(CNL)
           nextState = s0;
           else
           nextState = s6;

        s7:if( amount > balance_db[acc_index])
           nextState = s6;
           else if(amount > 16'd10000 && amount <= 16'd25000 && amount < balance_db[acc_index])
           nextState = s5;
           else if(amount > 16'd25000 && amount < balance_db[acc_index])
           nextState = s11;
           else if(amount <= 16'd10000 && amount < balance_db[acc_index])
           nextState = s12;

        s8:case(select_options)
           1'b0 : nextState = s4;  //banking
           1'b1 : nextState = s13; //pin generation
           endcase

        s9:if(pin_valid[2] == 2'b11)
           nextState = s2;
           else if(pin_valid[2] == 2'b01)
           nextState = s10;

       s10:if(pin_valid[3] == 2'b11)
           nextState = s2;
           else if(pin_valid[3] == 2'b01)
           nextState = s0;

        s11:if(face_valid)
            nextState = s12;
            else
            nextState = s0;

        s12:if(EC)
            nextState = s0;
            else
            nextState = s12;

        s13:if(acc_valid)
            nextState = s1;
            else
            nextState = s0;

        s14:if(EC)
            nextState = s0;
            else
            nextState = s14;

        s15:if(EC)
            nextState = s0;
            else
            nextState = s15;

        endcase
     end

     //sequential logic for outputs

     always @ (posedge clk)
     begin
        if(reset == 1'b1)
           begin
            print_receipt <= 1'b0;
            cash_dis <= 1'b0;
            bal_D <= 1'b0;
            EC <=0;
            mini_s <=1'b0;
           end
     else
        card_lock <= 1'b0;

        case(currState)

        s0:begin
            print_receipt <= 1'b0;
            cash_dis <= 1'b0;
            bal_D <= 1'b0;
             mini_s <=1'b0;
            end

        s1:if(!otp_valid)
             EC <= 1'b1;

        s2:EC <= 1'b0;

        s3:begin
             EC <= 1'b1;
             mini_s <= 1'b1;
           end

        s7:begin
            print_receipt <= 1'b0;
            cash_dis <= 1'b0;
            bal_D <= 1'b0;
            card_lock <= 1'b0;
           end

        s14:begin
            print_receipt <= !amount ? 1'b0 : 1'b1;
            EC <= 1'b1;
            end

        s12:begin
             cash_dis <= 1'b1;
             #20 bal_D <= 1'b1;
             EC <= 1'b1;
            end

        s13:if(!acc_valid)
             EC <= 1'b1;


        s10:begin
              if(pin_valid[3] == 2'd01)
               EC <= 1'b1;
               for(i=0;i<24;i=i+1)
               #10 card_lock <= 1'b1;
            end

        s15: EC <= 1'b1;
```

```

default:begin
         print_receipt <= 1'b0;
         cash_dis <= 1'b0;
         bal_D <= 1'b0;
         card_lock <= 1'b0;
         EC <=0;
         mini_s <=1'b0;
        end

endcase
end

//combinational block for performing operations

always @(currState)
begin

   case(currState)

   s0:$display("insert card");

   s1:begin
      $display("enter the otp sent to registerd mobile number");
      otp_valid = ( otp == otp_db[acc_index_Mpin]) ? 1'b1 : 1'b0;
      if(!otp_valid)
         $display("Invalid otp");
      end

   s2:begin
      $display("enter 00 for withdrawal");
      $display("enter 01 for  mini_statement");
      $display("enter 10 to deposit money");
   end

   s3:begin
        for ( i = 0; i<5 ; i=i+1)
        $display(" %d . amount_transacted = %d      main_balance = %d",i,transaction[i], balance[i]);
        $display("balance is %d",balance_db[acc_index]);
      end

   s4:begin
        $display("enter the 4 digit pin number");
         for (i=0;i<9 && pin_1!=pin_db[i];i=i+1)
          if(i==9) pin_valid[1] = 2'b01;

         if(i<9)
            begin
            acc_index = i;
            pin_valid[1] = 2'b11;
            end
         else
            pin_valid[1] = 2'b01;
      end

   s5:begin
      $display("enter the otp sent to mobile number");
      otp_valid = (otp == otp_db[acc_index]) ? 1'b1 : 1'b0;
      end

   s6:$display("if u wish to cancel press CNl or else continue");

   s8:begin
       $display("press '0' for banking");
       $display("press '1' for pin generation");
       end

   s9:begin
     $display("incorrect pin_1");
     $display("enter pin again");
     for (i=0;i<9 && pin_2!=pin_db[i];i=i+1)
       if(i==9)  pin_valid[2] = 2'b01;

     if(i<9) begin
        acc_index = i;
        pin_valid[2] = 2'b11;
```

```
         end
         else
            pin_valid[2] = 2'b01;
         end

    s10:begin
        $display("incorrect pin_2");
        $display("enter pin again");
        for (i=0;i<9 && pin_3!=pin_db[i];i=i+1)
          if(i==9) pin_valid[3] = 2'b01;

        if(i<9) begin
           acc_index = i;
           pin_valid[3] = 2'b11;
           end
        else
           begin
           $display("card has been locked for 24 hours");
           pin_valid[3] = 2'b01;
           end
        end

    s11:begin
        $display("recognizing face");
        face_valid = (face == face_db[acc_index]) ? 1'b1 : 1'b0;
        if(!face_valid)
           $display("face does not match");
        end

    s12:begin
        $display("old_balance is %d",balance_db[acc_index]);
        balance_db[acc_index] = balance_db[acc_index] - amount;
        $display("new_balance is %d",balance_db[acc_index]);

        //storing transaction details
        transaction[t_count] = amount;
        balance[t_count] = balance_db[acc_index];
        t_count = (t_count+1) % 5;
        end

    s13:begin
        $display("enter the Account number");
        for (i=0;i<9 && acc_num!=acc_db[i];i=i+1)
          if(i==9) acc_valid <= 1'b0;

        if(i<9) begin
          acc_index_Mpin = i;
          acc_valid = 1'b1;
        end
        else
          acc_valid = 1'b0;
        if(!acc_valid)
          $display("the entered account number does not exist");
        end

    s14:begin
            $display("place the amount");
            $display("transaction completed");
            balance_db[acc_index] = balance_db[acc_index] + amount;

            //soring transaction details
            transaction[t_count] = amount;
            balance[t_count] = balance_db[acc_index];
            t_count = (t_count+1) % 5;
        end

    s15:begin
           $display("enter new Mpin");
           pin_db[acc_index_Mpin] = pin_1;
           $display("pin updated successfully");
         end

    endcase
    end

    endmodule
```

```
//Test bench
module ATM_SM_tb;

  reg clk;
  reg card_insert, reset;
  reg [3:0] pin_1;
  reg [3:0] pin_2;
  reg [3:0] pin_3;
  reg [2:0] menu_sel;
  reg [15:0] amount;
  reg [7:0] acc_num;
  reg select_options;
  reg [3:0] face;
  reg [7:0] otp;
  reg CNL;
  wire cash_dis,bal_D,print_receipt;
  wire card_lock;
  wire mini_s;

  // Instantiate the ATM_SM module
  ATM_FSM atm_sm (
    .clk(clk),
    .card_insert(card_insert),
    .reset(reset),
    .pin_1(pin_1),
    .pin_2(pin_2),
    .pin_3(pin_3),
    .menu_sel(menu_sel),
    .amount(amount),
    .acc_num(acc_num),
    .select_options(select_options),
    .face(face),
    .otp(otp),
    .cash_dis(cash_dis),
    .bal_D(bal_D),
    .print_receipt(print_receipt),
    .CNL(CNL),
    .card_lock(card_lock),
    .mini_s(mini_s)
  );

initial begin
clk = 0;
reset = 1;

//test case 1                       card_lock after 3 pin attempts failed
//========================================================================================
==========


#10   CNL=0;reset = 0 ;card_insert = 1;
#10 pin_1 = 4'b1110;
#10 pin_2 = 4'b1110;
#10 pin_3 = 4'b1110;
 select_options = 0;
 menu_sel = 2'b00;
 amount = 16'd33000;
 face = 4'b1111;



//test case 2                         deposit
//========================================================================================
==========
#400 reset=1;
#10 CNL=0;reset = 0 ;
 card_insert = 1;
 select_options = 0;
 pin_1 = 4'b1111;
 menu_sel = 2'b10;
 amount = 16'd2000;

//test case 3                      pin generation / pin change
//========================================================================================
==========

#80 reset=1;
```

```
#10   CNL=0;reset = 0 ;
 card_insert = 1;
 select_options = 1;
 acc_num = 8'd1234;
 otp = 8'd2749;
 pin_1 = 4'b1010;

// verifying whether the pin is updated

#200 reset=1;
#10   CNL=0;reset = 0 ;
 card_insert = 1;
 select_options = 0;
 pin_1 = 4'b1010;
 menu_sel = 2'b01;

//test case 4                      withdrawal
//=======================================================================================
==========
#80   reset =1 ;
#10 CNL=0;reset = 0;
 card_insert = 1;
 pin_1 = 4'b1010;
 select_options = 0;
 menu_sel = 2'b00;
 amount = 16'd33000;
 face = 4'b1111;

#150 reset =1;
#10 CNL=0;reset = 0;
 card_insert = 1;
 pin_1 = 4'b1010;
 select_options = 0;
 menu_sel = 2'b00;
 amount = 16'd12000;
 otp = 8'd2749;

#150 reset =1;
#10 CNL=0;reset = 0;
 card_insert = 1;
 pin_1 = 4'b1010;
 select_options = 0;
 menu_sel = 2'b00;
 amount = 16'd5000;


//test case 5                      printing recent 5 transactions
//=======================================================================================
==========

#150 reset=1;
#10   CNL=0;reset = 0 ;
 card_insert = 1;
 select_options = 0;
 pin_1 = 4'b1010;
 menu_sel = 2'b01;


#500 $finish;
  end

  always #5 clk = ~clk;

endmodule
```