SYSTEM DESIGN AND STRUCTURAL DESIGN

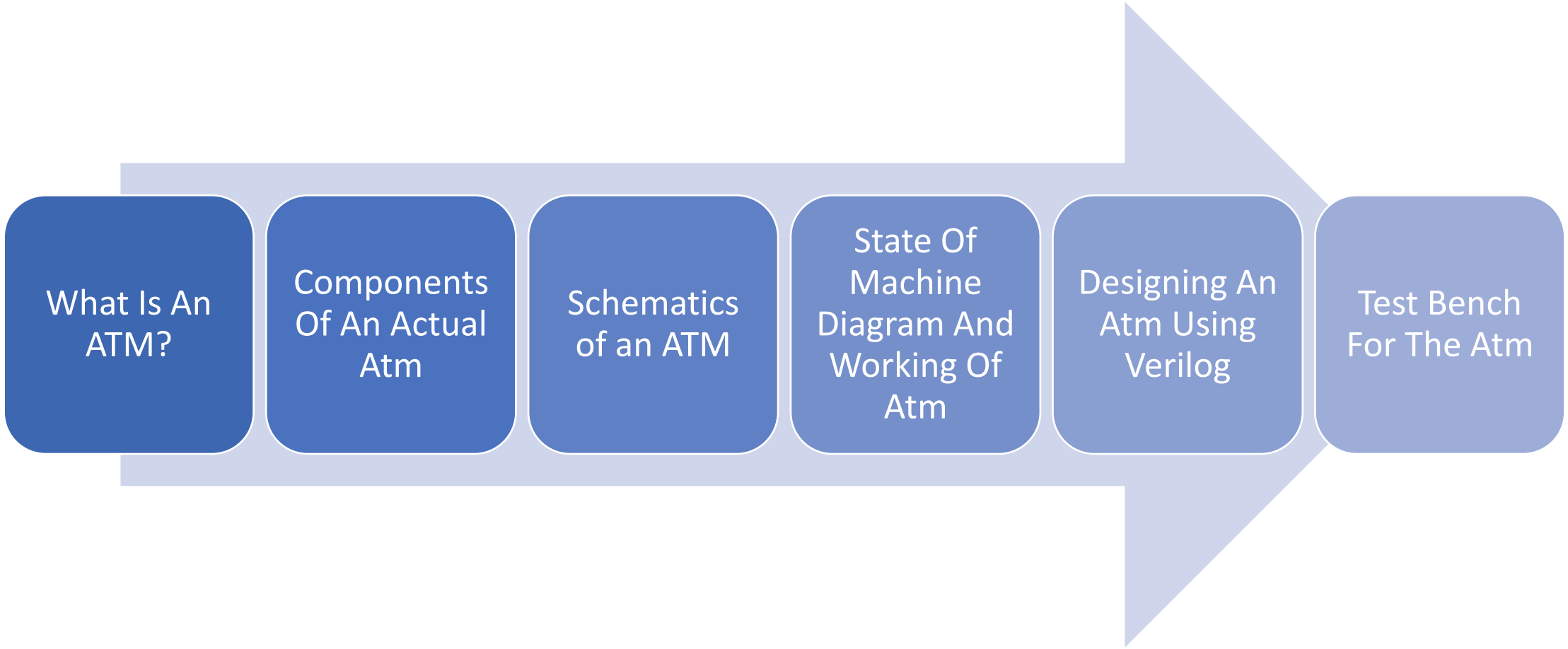
TOPIC : AUTOMATED TELLER MACHINE

PRESENTED BY :

Kaushal Kumawat

Under Guidance of : Dr. R.Rakesh kumar

Flow Of Presentation



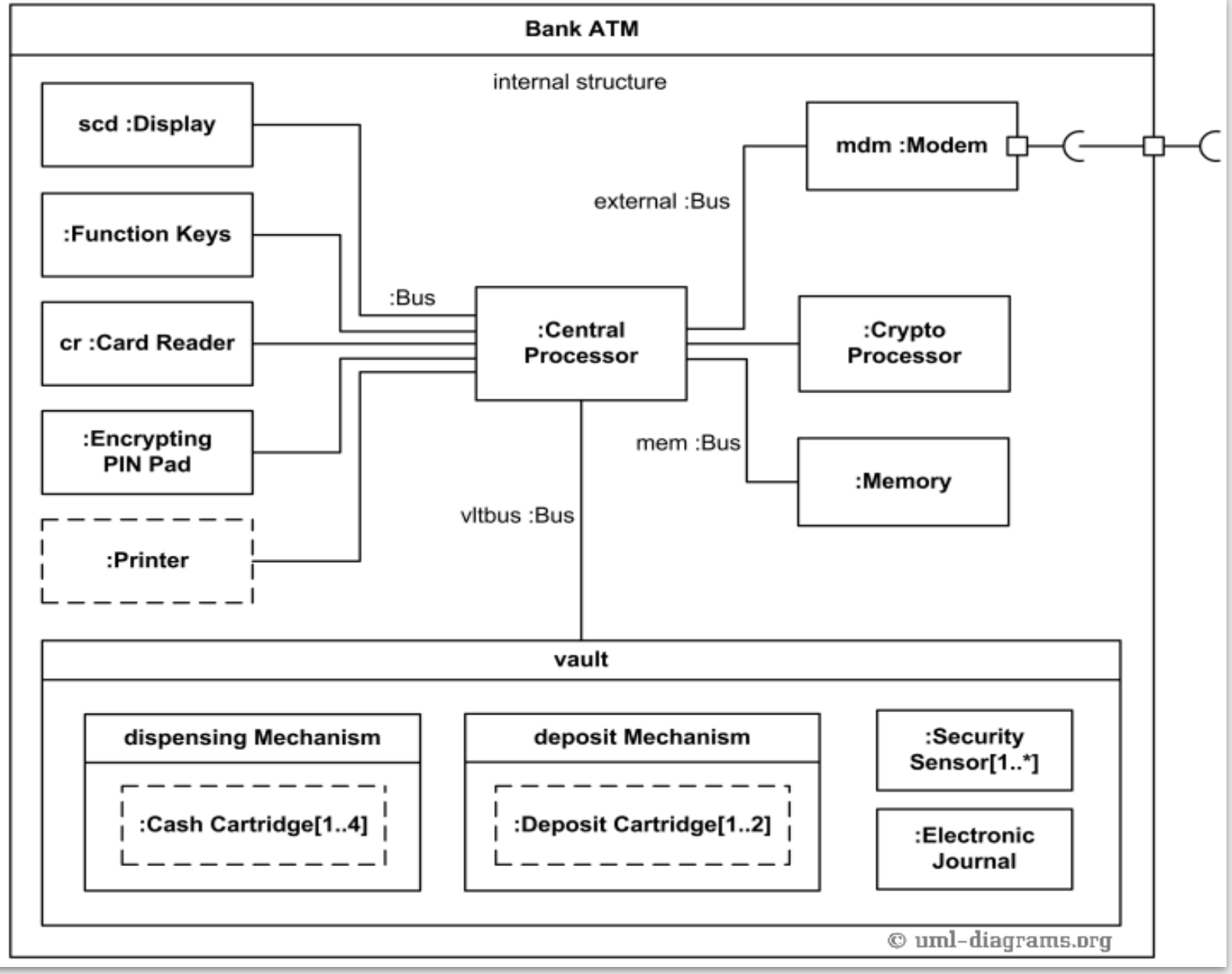
What is an ATM ?

- An ATM stands for Automated Teller Machine. It is an electronic banking outlet that allows customers to complete basic transactions without the need for a human bank teller. ATMs enable customers to withdraw cash, deposit checks and cash, transfer money between accounts, and check their account balance

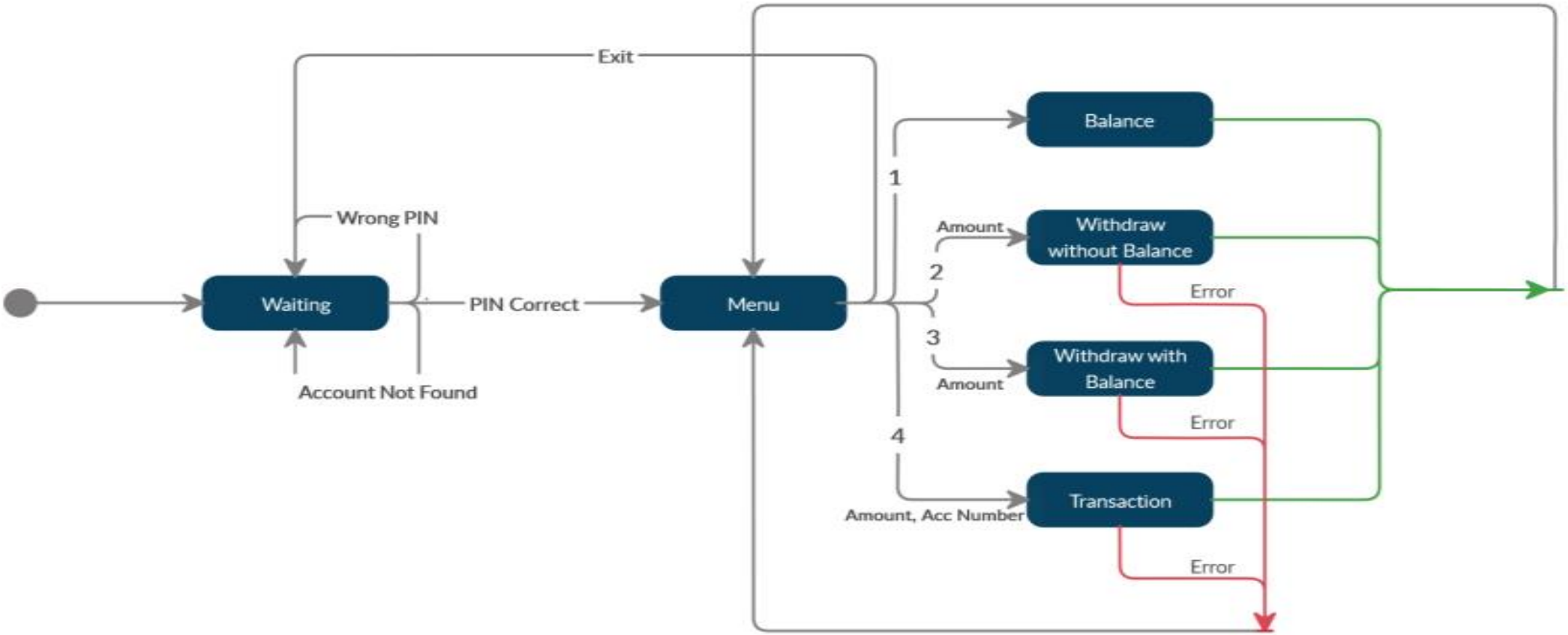
Components Of ATM

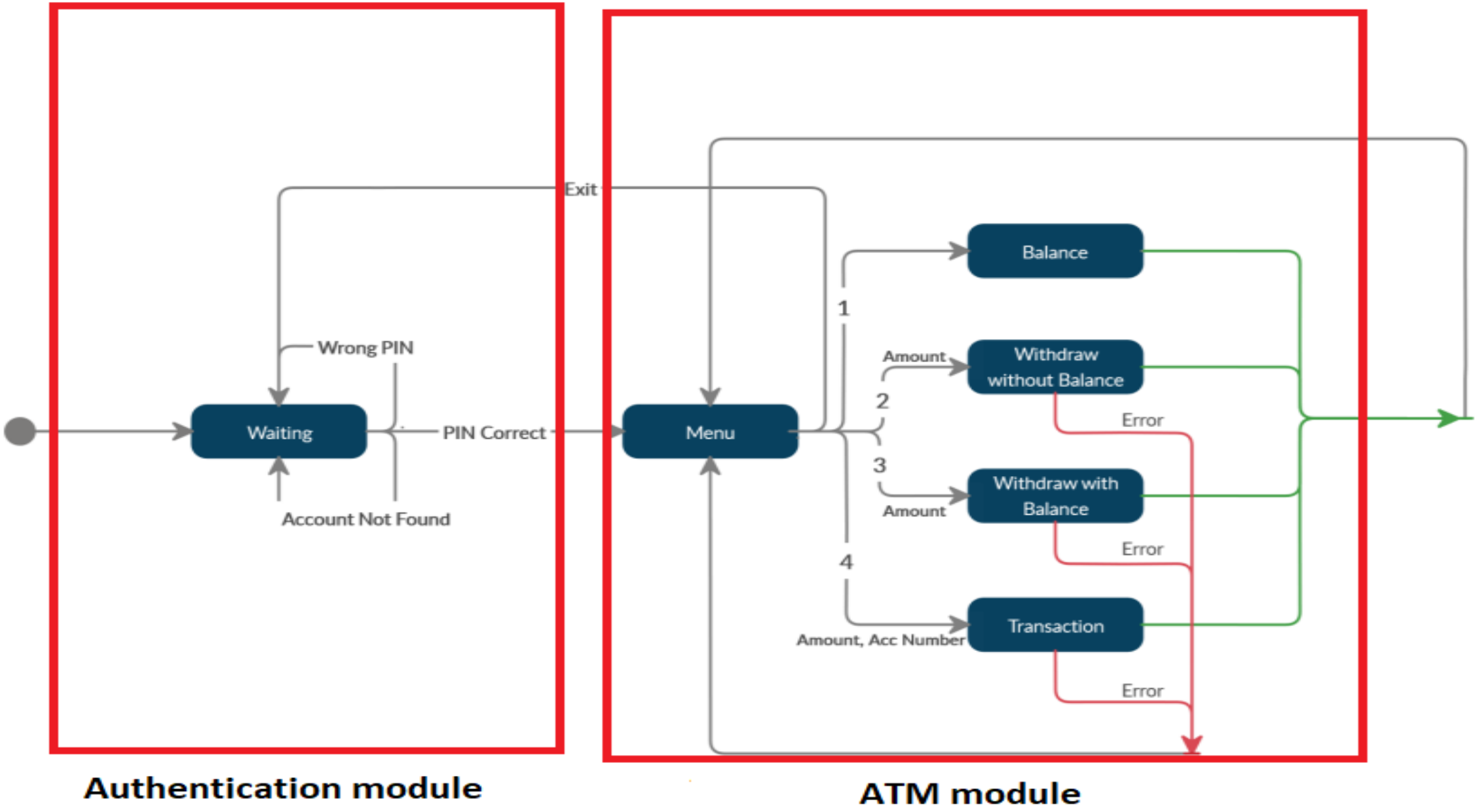
- Card Reader
- Keypad
- Display Screen
- Cash Dispenser
- Receipt Printer
- Network Connection
- Security Devices
- Power Supply

- **Schematics of an ATM**



State Of Machine Diagram




Exit
Wrong PIN
Waiting
PIN Correct
Account Not Found
Menu
1
Balance
Amount
2
Withdraw without Balance
Error
3
Withdraw with Balance
Amount
Error
4
Amount, Acc Number
Transaction
Error
Authentication module
ATM module

Some Definitions

```
`define true 1'b1
`define false 1'b0

`define FIND 1'b0
`define AUTHENTICATE 1'b1

`define WAITING                3'b000
`define MENU                   3'b010
`define BALANCE                3'b011
`define WITHDRAW               3'b100
`define WITHDRAW_SHOW_BALANCE  3'b101
`define TRANSACTION            3'b110
```

Authentication module

```
module authentication(
  input [11:0] accNumber,
  input [3:0] pin,
  input action,
  input deAuth,
  output reg  wasSuccessful,
  output reg [3:0] accIndex
);


  reg [11:0] acc_database [0:9];
  reg [3:0] pin_database [0:9];

  //initializing the database with arbitrary accounts
  initial begin
    acc_database[0] = 12'd2749; pin_database[0] = 4'b0000;
    acc_database[1] = 12'd2175; pin_database[1] = 4'b0001;
    acc_database[2] = 12'd2429; pin_database[2] = 4'b0010;
    acc_database[3] = 12'd2125; pin_database[3] = 4'b0011;
    acc_database[4] = 12'd2178; pin_database[4] = 4'b0100;
    acc_database[5] = 12'd2647; pin_database[5] = 4'b0101;
    acc_database[6] = 12'd2816; pin_database[6] = 4'b0110;
    acc_database[7] = 12'd2910; pin_database[7] = 4'b0111;
    acc_database[8] = 12'd2299; pin_database[8] = 4'b1000;
    acc_database[9] = 12'd2689; pin_database[9] = 4'b1001;
    end
```

```
always @ (deAuth) begin
  if(deAuth == `true)
    wasSuccessful = 1'bx;
end
//looping through the database, trying to find a match for the given accNumber and pin
// if action is set to find then it'll simply ry to find a match for the given accNumber and returns its index
integer i;
always @(accNumber or pin) begin
    wasSuccessful = `false;
    accIndex = 0;

    //loop through the data base
    for(i = 0; i < 10; i = i+1) begin

        //find a match for accNumber
        if(accNumber == acc_database[i]) begin
            if(action == `FIND) begin
              wasSuccessful = `true;
              accIndex = i;
            end
```

```
                if(action == `AUTHENTICATE) begin
                    if(pin == pin_database[i]) begin
                        wasSuccessful = `true;
                        accIndex = i;

                    end
                end
            end
        end
    end

endmodule

//
```

ATM module

```
module ATM(
  input clk,
  input exit,
  input [11:0] accNumber,
  input [3:0] pin,
  input [11:0] destinationAcc,
  input [2:0]menuOption,
  input [10:0] amount,
  output reg error,
  output reg [10:0] balance
  );

  //initializing the balance database with an arbitrary amount of money
  reg [15:0] balance_database [0:9];
  initial begin
    $display("Welcome to the ATM");
     balance_database[0] = 16'd500;
     balance_database[1] = 16'd500;
     balance_database[2] = 16'd500;
     balance_database[3] = 16'd500;
     balance_database[4] = 16'd500;
     balance_database[5] = 16'd500;
     balance_database[6] = 16'd500;
     balance_database[7] = 16'd500;
     balance_database[8] = 16'd500;
     balance_database[9] = 16'd500;

  end
```

```
reg [2:0] currState = `WAITING;

wire [3:0] accIndex;
wire [3:0] destinationAccIndex;
wire isAuthenticated;
wire wasFound;

reg deAuth = `false;

authentication authAccNumberModule(accNumber, pin, `AUTHENTICATE, deAuth, isAuthenticated, accIndex);
authentication findAccNumberModule(destinationAcc, 0, `FIND, deAuth, wasFound, destinationAccIndex);

//main block of module with asynchronous exit
always @(posedge clk or isAuthenticated or menuOption or exit) begin

  //restart At error
error = `false;
  if(exit == `true) begin
    //transition to the waiting state
    currState = `WAITING;
    //deathenticate the current user
    deAuth = `true;
    #20;
  end
```

```
if(currState == `MENU) begin
  //set the selected option as the current state
  if((menuOption >= 0) & (menuOption <= 7))begin
    currState = menuOption;
  end else
  currState = menuOption;
end


//switch case for the menu options
//the rest is pretty straight forward
  case (currState)


  `WAITING: begin
    if (isAuthenticated == `true) begin
      currState = `MENU;
      $display("Logged In.");
    end
    else if(isAuthenticated == `false) begin
      $display("Account number or password was incorrect");
      currState = `WAITING;
    end
  end
```

```
`BALANCE: begin
  balance = balance_database[accIndex];
  $display("Account %d has balance %d", accNumber, balance_database[accIndex]);
  currState = `MENU;
end

`WITHDRAW: begin
    if (amount <= balance_database[accIndex]) begin
      balance_database[accIndex] = balance_database[accIndex] - amount;
      balance = balance_database[accIndex];
      currState = `MENU;
      error = `false;
    end
    else begin
      currState = `MENU;
      error = `true;
    end
end

`WITHDRAW_SHOW_BALANCE: begin
    if (amount <= balance_database[accIndex]) begin
      balance_database[accIndex] = balance_database[accIndex] - amount;
      balance = balance_database[accIndex];
      currState = `MENU;
      error = `false;
      $display("Account %d has balance %d after withdrawing %d", accNumber, balance_database[accIndex], amount);
    end
    else begin
      currState = `MENU;
      error = `true;
    end
end
```

```
    `TRANSACTION: begin
      if ((amount <= balance_database[accIndex]) & (wasFound == `true) & (balance_database[accIndex] + amount < 2048)) begin
          currState = `MENU;
          error = `false;
          balance_database[destinationAccIndex] = balance_database[destinationAccIndex] + amount;
          balance_database[accIndex] = balance_database[accIndex] - amount;
          $display("Destination account %d after transaction has a total balance of %d", destinationAcc,
          balance_database[destinationAccIndex]);
      end
      else begin
          currState = `MENU;
          error = `true;
      end
    end

  endcase


end

endmodule
```

Test Bench

```
`define true 1'b1
`define false 1'b0

`define FIND 1'b0
`define AUTHENTICATE 1'b1

`define WAITING                3'b000
`define GET_PIN                3'b001
`define MENU                   3'b010
`define BALANCE                3'b011
`define WITHDRAW               3'b100
`define WITHDRAW_SHOW_BALANCE  3'b101
`define TRANSACTION            3'b110
`define DONE                   3'b111

module atm_tb();

  reg clk, exit;
  reg [11:0] accNumber;
  reg [3:0] pin;
  reg [11:0] destinationAccNumber;
  reg [2:0] menuOption;
  reg [10:0] amount;
  wire error;
  wire [10:0] balance;
```

```
ATM atmModule(clk, exit, accNumber, pin, destinationAccNumber, menuOption, amount, error, balance);

initial begin
  clk = 1'b0;
end

 always @(error) begin
    if(error == `true)
      $display("Error!, action causes an invalid operation.");
 end

initial begin

  //incorrect PIN
  accNumber = 12'd2278;
  pin = 4'b0100;
      #30
  //valid credentials
  accNumber = 12'd2178;
  pin = 4'b0100;

  #30
  //withdraw some money and then show the balance
  amount = 100;
menuOption = `WITHDRAW_SHOW_BALANCE;
  clk = ~clk;#5clk = ~clk;
```

```
  #30
  //show the balance
menuOption = `BALANCE;
  clk = ~clk;#5clk = ~clk;

  #30
  //withdraw too much money, resulting in an error
  amount = 2500;
menuOption = `WITHDRAW;
  clk = ~clk;#5clk = ~clk;

  #30
  //the balance wont change because an error happened during withdrawal
menuOption = `BALANCE;
  clk = ~clk;#5clk = ~clk;

  #30
  //transfer some money to the destination account with number 2816
  amount = 50;
  destinationAccNumber = 2816;
menuOption = `TRANSACTION;
  clk = ~clk;#5clk = ~clk;
```

```
    #30
    //transfer too much money to the destination account with number 2816 which exceeds 2047 and cuases an error
    amount = 2550;
    destinationAccNumber = 2816;
  menuOption = `TRANSACTION;
    clk = ~clk;#5clk = ~clk;
    #30

    //exit the system
    exit = 1;
    #30
    exit = 0;
    #30

    //log in using the account with number 2816
    accNumber = 12'd2816;
    pin = 4'b0110;
    #30

    //you'll see that the balance is more than the default value because we had trasnsferred some money to this account a while ago
    menuOption = `BALANCE;
    clk = ~clk;#5clk = ~clk;
    #30;

  end

endmodule
```

Test Bench Output

```
Welcome to the ATM
Account number or password was incorrect
Logged In.
Account 2178 has balance   400 after withdrawing  100
Account 2178 has balance   400
Error!, action causes an invalid operation.
Account 2178 has balance   400
Error!, action causes an invalid operation.
Logged In.
Account 2816 has balance   500
```

Output Waveform