|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A picture containing text, clock, alcohol  Description automatically generated | Academic Year: | 2022/2023 | Term: | spring 2023 | Logo  Description automatically generated |
| Course Code: | Elective-4 | Course Title: | EDA |

Cairo University

Faculty of Engineering

Electronics and Communications Engineering Department – 4th Year

**ATM - based bank system**

*Submitted to: Dr. Eman*

|  |  |  |
| --- | --- | --- |
| Name | BN | Sec |
| خالد خليفة | 10 | 2 |
| عمر ايمن عبدالمتعال | 13 | 3 |
| محمد أحمد محمد أبوسعدة | 32 | 3 |
| محمد ايمن عبدالعليم احمد | 34 | 3 |
| محمد عادل كامل | 35 | 3 |
| مينا حنا فايز وهبة | 35 | 4 |

Table of Contents

[Abstract 4](#_Toc133618107)

[Introduction 4](#_Toc133618108)

[FSM Module 5](#_Toc133618109)

[FSM Diagram 6](#_Toc133618113)

[Scan card Module (Behavioral model) 7](#_Toc133618114)

[Scan card Module 7](https://d.docs.live.net/fd69c99e1e1d1519/Desktop/4th%20year/2nd_Term/EDA/Projects/Lecture/Project_1_V4/Project_1.docx#_Toc133618115)

[States of FSM: 8](#_Toc133618116)

[S0 (idle state) 8](#_Toc133618117)

[S8 (scan card) 8](#_Toc133618118)

[S7 (lang used) 8](#_Toc133618119)

[S9 (enter pass) 8](#_Toc133618120)

[S10 (option select) 8](#_Toc133618121)

[S1 (balance check) 9](#_Toc133618122)

[S2 (Withdraw) 9](#_Toc133618123)

[S3 (deposit) 9](#_Toc133618124)

[S4 (transfer) 10](#_Toc133618125)

[S11 (anything else) 10](#_Toc133618126)

[S5 (exit) 10](#_Toc133618127)

[Verification Plan 11](#_Toc133618128)

[Reference model 12](#_Toc133618129)

[does\_card\_exist() 12](#_Toc133618130)

[Is\_pass\_correct() 12](#_Toc133618131)

[Check\_balance() 12](#_Toc133618132)

[Change\_money() 12](#_Toc133618133)

[Change\_pass() 12](#_Toc133618134)

[Verification Process: 13](#_Toc133618135)

[Assertions Section: 15](#_Toc133618136)

[Problems and Solutions: 15](#_Toc133618137)

[Problem\_1: 15](#_Toc133618138)

[Solution: 15](#_Toc133618139)

[Assertions Results: 16](#_Toc133618140)

[Coverage Section: 17](#_Toc133618141)

[1. User\_Data Covergroup 17](#_Toc133618142)

[2. FSM\_Covergroup 17](#_Toc133618143)

[Coverage Results: 18](#_Toc133618144)

[Pre\_100%\_Coverage: 18](#_Toc133618145)

# Abstract

The ATM System is the project which is used to access their bank accounts to make cash withdrawals and other transactions. Whenever user wants to make transaction, he can enter their ATM card and verified PIN, then menu of options will appear to choose the appropriate option. ATM System has capability to enter the withdrawal amounts and deposit amounts of money, display the account balance.

# Introduction

The automated teller machine (ATM ) is an automatic banking machine which allows the user to complete basic

transactions without any help of bank representatives. There are two types of automated teller machine. The basic one

allows the customer to only draw cash and receive a report of the account balance. Another one is a more complex

machine which accepts the deposit, provides credit card payment facilities and reports account information. In earlier

years all the transactions were to be done manually, still done but very rarely, as it is very difficult task. so now banks

use this to give their customers to have easy and faster transactions. This makes transactions easier faster.

The automated teller machine (ATM) is an automatic banking machine which allows the user to complete basic transactions without any help from bank representatives. There are two types of automated teller machines. The basic one allows the customer to only draw cash and receive a report of the account balance. Another one is a more complex machine which accepts deposits, provides credit card payment facilities, and reports account information.

The user is also able to perform one or more transactions. Security is the foundation of a good ATM system. This system will provide for secure connections between users and the bank servers. The whole process will be automated right from PIN validation to transaction completion. The card details and PIN database will be a secure module that will not be open to routine maintenance, the only possibility of access to this database will be through queries(questions) raised from an ATM in the presence of a valid bank ATM card. ATM Simulation System will enable two important features of an ATM, reduction of human error in the banking system and the possibility of24 hour personal banking. In earlier years all the transactions were to be done manually, still done but very rarely, as it is a very difficult task. So now banks use this to give their customers easy and faster transactions. This makes transactions easier and faster.

**In this project we are implementing the core of the bank ATM design as well as verification environment.**

The ATM System is the project which is used to access their bank accounts to make cash withdrawals.

Whenever user want to make withdrawals, they can enter their ATM card and verified PIN ,then user select the

withdrawal option and enter the withdrawal amounts and it will display the amount to be withdrawn in the form of

50's,100's and 500's. The user also able to perform one or more transactions. Security is the foundation of good ATM

system. This system will provide for secure connections between users and the bank servers. The whole process will

be automated right from PIN validation to transaction completion. The card details and PIN database will be a

secure module that will not be open to routine maintenance, the only possibility of access to this database will be

through queries(questions) raised from an ATM in the presence of a valid bank ATM card. ATM Simulation System

will enable two important features of an ATM, reduction of human error in the banking system and the possibility of

24-hour personal banking.

The ATM System is the project which is used to access their bank accounts to make cash withdrawals.

Whenever user want to make withdrawals, they can enter their ATM card and verified PIN ,then user select the

withdrawal option and enter the withdrawal amounts and it will display the amount to be withdrawn in the form of

50's,100's and 500's. The user also able to perform one or more transactions. Security is the foundation of good ATM

system. This system will provide for secure connections between users and the bank servers. The whole process will

be automated right from PIN validation to transaction completion. The card details and PIN database will be a

secure module that will not be open to routine maintenance, the only possibility of access to this database will be

through queries(questions) raised from an ATM in the presence of a valid bank ATM card. ATM Simulation System

will enable two important features of an ATM, reduction of human error in the banking system and the possibility of

24-hour personal banking.

The ATM System is the project which is used to access their bank accounts to make cash withdrawals.

Whenever user want to make withdrawals, they can enter their ATM card and verified PIN ,then user select the

withdrawal option and enter the withdrawal amounts and it will display the amount to be withdrawn in the form of

50's,100's and 500's. The user also able to perform one or more transactions. Security is the foundation of good ATM

system. This system will provide for secure connections between users and the bank servers. The whole process will

be automated right from PIN validation to transaction completion. The card details and PIN database will be a

secure module that will not be open to routine maintenance, the only possibility of access to this database will be

through queries(questions) raised from an ATM in the presence of a valid bank ATM card. ATM Simulation System

will enable two important features of an ATM, reduction of human error in the banking system and the possibility of

24-hour personal banking.

The ATM System is the project which is used to access their bank accounts to make cash withdrawals.

Whenever user want to make withdrawals, they can enter their ATM card and verified PIN ,then user select the

withdrawal option and enter the withdrawal amounts and it will display the amount to be withdrawn in the form of

50's,100's and 500's. The user also able to perform one or more transactions. Security is the foundation of good ATM

system. This system will provide for secure connections between users and the bank servers. The whole process will

be automated right from PIN validation to transaction completion. The card details and PIN database will be a

secure module that will not be open to routine maintenance, the only possibility of access to this database will be

through queries(questions) raised from an ATM in the presence of a valid bank ATM card. ATM Simulation System

will enable two important features of an ATM, reduction of human error in the banking system and the possibility of

24-hour personal banking.

The ATM System is the project which is used to access their bank accounts to make cash withdrawals.

Whenever user want to make withdrawals, they can enter their ATM card and verified PIN ,then user select the

withdrawal option and enter the withdrawal amounts and it will display the amount to be withdrawn in the form of

50's,100's and 500's. The user also able to perform one or more transactions. Security is the foundation of good ATM

system. This system will provide for secure connections between users and the bank servers. The whole process will

be automated right from PIN validation to transaction completion. The card details and PIN database will be a

secure module that will not be open to routine maintenance, the only possibility of access to this database will be

through queries(questions) raised from an ATM in the presence of a valid bank ATM card. ATM Simulation System

will enable two important features of an ATM, reduction of human error in the banking system and the possibility of

24-hour personal banking.

The ATM System is the project which is used to access their bank accounts to make cash withdrawals.

Whenever user want to make withdrawals, they can enter their ATM card and verified PIN ,then user select the

withdrawal option and enter the withdrawal amounts and it will display the amount to be withdrawn in the form of

50's,100's and 500's. The user also able to perform one or more transactions. Security is the foundation of good ATM

system. This system will provide for secure connections between users and the bank servers. The whole process will

be automated right from PIN validation to transaction completion. The card details and PIN database will be a

secure module that will not be open to routine maintenance, the only possibility of access to this database will be

through queries(questions) raised from an ATM in the presence of a valid bank ATM card. ATM Simulation System

will enable two important features of an ATM, reduction of human error in the banking system and the possibility of

24 hour personal banking.

The ATM System is the project which is used to access their bank accounts in order to make cash withdrawals.

Whenever user want to make withdrawals, they can enter their ATM card and verified PIN ,then user select the

withdrawal option and enter the withdrawal amounts and it will display the amount to be withdrawn in the form of

50's,100's and 500's. The user also able to perform one or more transactions. Security is the foundation of good ATM

system. This system will provide for secure connections between users and the bank servers. The whole process will

be automated right from PIN validation to transaction completion. The card details and PIN database will be a

secure module that will not be open to routine maintenance, the only possibility of access to this database will be

through queries(questions) raised from an ATM in the presence of a valid bank ATM card. ATM Simulation System

will enable two important features of an ATM, reduction of human error in the banking system and the possibility of

24 hour personal banking.

The ATM System is the project which is used to access their bank accounts in order to make cash withdrawals.

Whenever user want to make withdrawals, they can enter their ATM card and verified PIN ,then user select the

withdrawal option and enter the withdrawal amounts and it will display the amount to be withdrawn in the form of

50's,100's and 500's. The user also able to perform one or more transactions. Security is the foundation of good ATM

system. This system will provide for secure connections between users and the bank servers. The whole process will

be automated right from PIN validation to transaction completion. The card details and PIN database will be a

secure module that will not be open to routine maintenance, the only possibility of access to this database will be

through queries(questions) raised from an ATM in the presence of a valid bank ATM card. ATM Simulation System

will enable two important features of an ATM, reduction of human error in the banking system and the possibility of

24 hour personal bank

# FSM Module

card\_accepted

clk

**FSM Module**

Insuffucient\_flag

account\_balance

Ic

Rst

destination\_number

credit\_number

mt clk

new\_password

en\_password

y0, y1,y2,y3,y4,y5,y6,y7,y11,y10,y8,y12

Keyboard

Option

en\_ammount\_money

Enter

|  |  |  |  |
| --- | --- | --- | --- |
| **Signal** | **Type** | **Width** | **Description** |
| Clk | **Input** | 1 | Clock of the system |
| Rst | **Input** | 1 | Reset |
| Ic | **Input** | 1 | Insert card |
| mt | **Input** | 1 | More transaction |
| credit\_number | **Input** | 64 | Number of credit |
| destination\_number | **Input** | 64 | Destination number |
| en\_password | **Input** | 16 | Enter password |
| new\_password | **Input** | 16 | New password |
| en\_ammount\_money | **Input** | 16 | Enter amount of money |
| Option | **Input** | 3 | Screen options |
| Keyboard | **Input** | 4 | Keyboard for enter the different options |
| Enter | **Input** | 1 | Enter key |
| card\_accepted | **Output** | 1 | Accepted the card |
| account\_balance | **Output** | 16 | Account balance |
| Insuffucient\_flag | **Output** | 1 | Withdraw value more than account balance |
| y0,y1,y2,y3,y4,y5,y6,y7,y11,y10,y8,y12 | **Output** | 1 | Flags to show the current state |

# FSM Diagram

Figure 1 FSM diagram

# Scan card Module (Behavioral model)

card\_index

credit\_number

# Scan card Module

card\_pass

credit\_pass\_update

card\_balance

credit\_balance\_update

card\_found\_flag

|  |  |  |  |
| --- | --- | --- | --- |
| **Signal** | **Type** | **Width** | **Description** |
| credit\_number | **Input** | 64 | Number of credit |
| credit\_pass\_update | **Input** | 16 | Update of old password |
| credit\_balance\_update | **Input** | 14 | Update of amount of balance |
| card\_index | **Output** | 4 | Index of card |
| card\_pass | **Output** | 16 | Password of the card |
| card\_balance | **Output** | 14 | Amount of balance |
| card\_found\_flag | **Output** | 1 | Flag refer to that the card that enter is founded |

**Note:**

Behavioral model to act as ATM database.

# States of FSM:

## S0 (idle state)

The idle state nothing happened on it, and it is the state that I return to it at the end if:

1. The card does not exist.
2. Enter wrong password.
3. Time out and the user does not use any option
4. If enter exit option

## S8 (scan card)

The scan card state responsible for handling card and check its existence and get its information from server data base (account balance and pass)

## S7 (lang used)

The lang used state decides the language to be used through the transaction. The user in

This state selects the required language by pressing any number in the keyboard input.

## S9 (enter pass)

The enter pass state responsible for comparing entered pass with pass from database.

## S10 (option select)

The option select state responsible for choosing the desired options:

(Withdraw – deposit- balance check-new pass- transfer -exit – anything else).

Operation of previous states:

1-First, the operation started from idle state (S0)

2-The user pressed insert card (IC) input and entered card value (card\_number)

3-The operation transferred to (scan\_card) state to check the existence of the card. (S8)

4-The card number was correct so now we are in (lang\_used) state we pressed 1 in the keyboard to select Arabic. (S7)

5-we then move to (enter\_pass) state we write the password and press enter if it is true, we move to option select if wrong we have number of trials that we should not exceed. (S9 to S10)

A picture containing graphical user interface

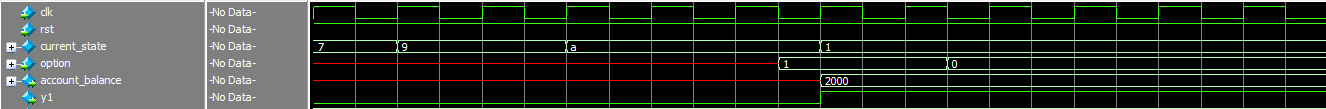
Description automatically generated-y0, y8,y7,y6,y10 are output flags indicating which state we are at now.

## S1 (balance check)

The balance check state gets all information about updated account balance.

1-During (select\_option) state we enter option 1 to select check balance.

2-The output account balance will take the value of the entered card number.

 3-y1 flag indicates that we are in this state.

## S2 (Withdraw)

The withdraw state gets account balance deducted by the desired entered amount of money.

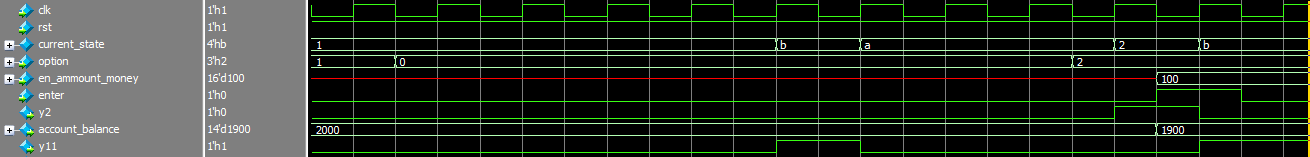
If balance account has the capacity to make this withdraw operation if not go to option state to check account balance.

1-From (select\_option) state we enter option 2 to select (withdraw) S2.

2-Enter the amount of money(en\_amount\_money) and press (enter) to save it.

3-output (account\_balance) will change in the waveform= 2000-100=1900.but the

Withdrawal value must be less than the account balance.

 4-y2 flag indicates that we are in this state.

## S3 (deposit)

The deposit state gets account balanced increased by the desired entered amount of money.

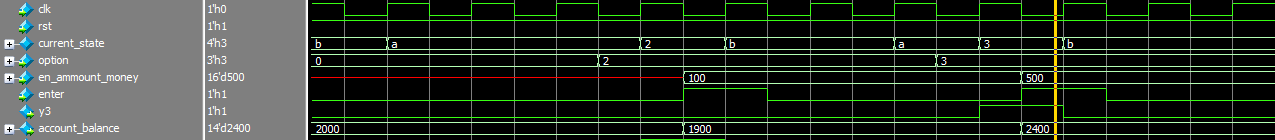
If entered amount has exceed max value of deposit operation go to option state

1-From (select\_option) state we enter option 3 to select (deposit) S3.

2-Enter the amount of money(en\_amount\_money) and press (enter) to save it.

3-output (account\_balance) will change in the waveform= 2000+500=2400.but the

deposit value must be less than the deposit max value.

 4-y3 flag indicates that we are in this state.

## S4 (transfer)

The transfer state checks the existence of destination account and perform withdraw operation on source account and deposit operation to destination account.

1-From (select\_option) state we enter option 4 to select (transfer) S4.

2-Enter the amount of money(en\_amount\_money) and destination card number and press (enter) to save it.

3-output (account\_balance) will change in the waveform= 2400-500=1900.but the

deposit value must be less than the deposit max value.

4-y4 flag indicates that we are in this state.A screenshot of a computer

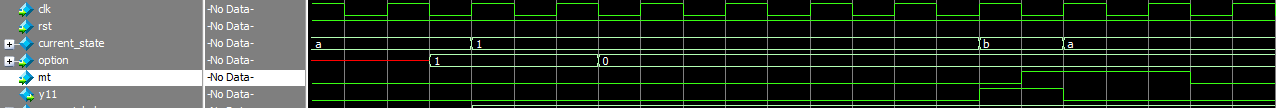
Description automatically generated with medium confidence

## S11 (anything else)

The anything else state to let opportunities for user to do more transaction or not.

1-After (check\_balance) state we go to (anything\_else) state.

2-we press mt (more transaction) if we want to continue and choose another option.

 3-y11 flag indicates that we are in this state.

## S5 (exit)

The exit state option to finish transaction and go to idle state.

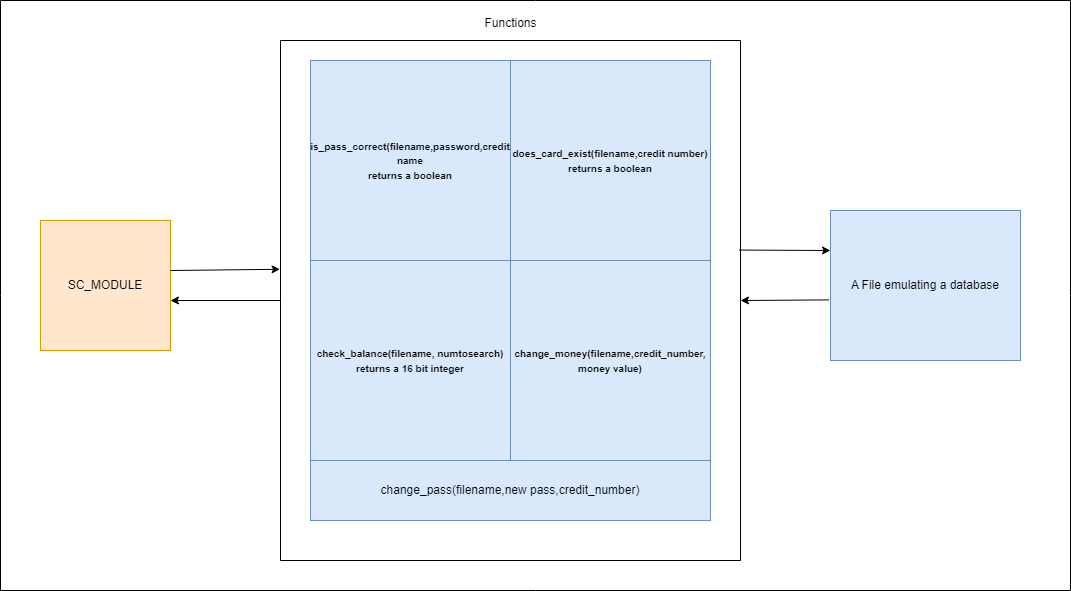
# Verification Plan

After reading the design specs we wrote a plan for the verification process that would guarantee that the design performs all required operations correctly by using the assertion propertied mentioned below as well as the cover states.



# Reference model

The Reference model was designed based on the previous implantation , it was designed using c++ and systemc library, it have the components as following.



The main module of ATM is defined using systemc macro and have the same states as mentioned before, there is an external file which contains all the credit details and used to emulate the database and the ATM module is interfacing with the file using those functions.

## does\_card\_exist()

checks whether the card exists or not

## Is\_pass\_correct()

It checks if the password is correct, it takes filename, password and credit numberand returns a bool which tells if the password is correct or not.

## Check\_balance()

This returns the current balance for a credit number.

## Change\_money()

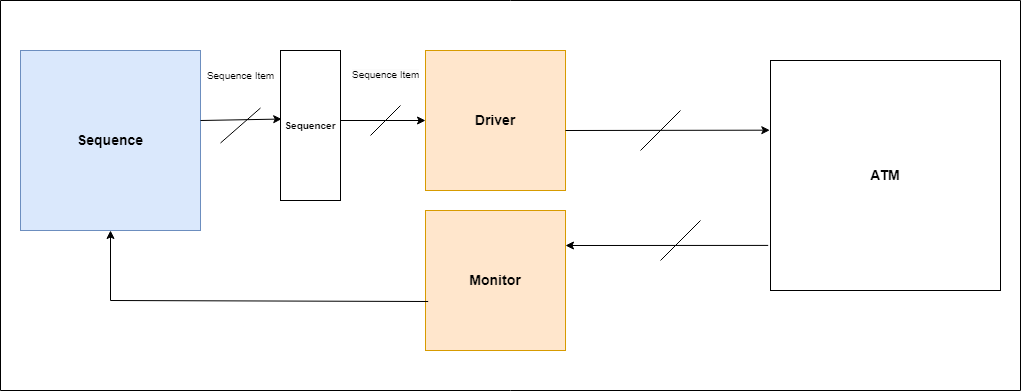
This can change the money balance In a user account whether in a deposit , transfer or withdraw operation.

## Change\_pass()

Used to change password for a given new password

# Verification Process:

The verification Process starts with creating an input Stimulus, that could cover all the possible states and this was done using the UVM methodology and system Verilog classes but making a complete randomization would take a lot of simulation time and is mostly wouldn’t get us to the wanted coverage not even close to it so a constraint randomization was made so that we should be able to attend this coverage but how the hierarchy of injecting the stimulus was made.



The sequence first generates a sequence item to send it to the DUT this sequence item is a class which contains all the signals in the designed and it is randomized or not, the following signals that are in an instance of sequence item of a class is as following

|  |  |  |
| --- | --- | --- |
| Signal | Type | Constraints |
| rst | **Input - direct only** | **No constraints** |
| ic | **Input - direct only** | **No constraints** |
| enter | **Input - direct only** | **No constraints** |
| mt | **Input- Randomized** | **No constraints** |
| credit\_number,  destination\_number | **Input - Randomized** | **Constraints inside a given array of Credit Numbers** |
| en\_password | **Input - Randomized** | **Constraints inside a given array of passwords with probability of success higher** |
| new\_password | **Input - Randomized** | **No constraints** |
| en\_ammount\_money | **Input – Randomized** | **Constraints to be less than 10,000 EGP** |
| option | **Input - Randomized** | **No constraints** |
| keyboard | **Input – Randomized** | **No constraints** |

And the rest of the signals are defined as static output but why its static and why there is a connection between the monitor and the sequence class, that connection is actually a flag based that the monitor rises every time it reads values from the DUT and put in the output of the sequence item class and based on the state of the machine the sequence or the generator decides which constraint randomization values should be put.

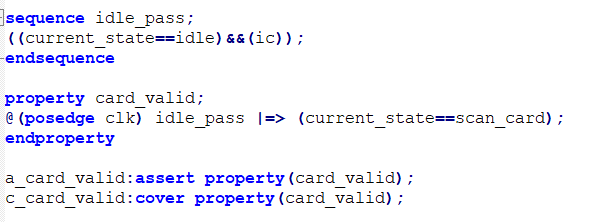
But what are the sequence that the generator could possibly generate, the following table represent the sequences which can the sequence generate.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| First state | 2nd state | 3rd state | 4th state | 5th state | 6th state | 7th state | 8th state | Probability |
| Idle | **Scan card** | **Enter lang** | **Enter pass** | **Enter option** | **Balance check** | **Anything else** | **Enter option** | **Normal** |
| Idle | **Scan card** | **Enter lang** | **Enter pass** | **Enter option** | **Balance check** | **Anything else** | **Idle** | **Normal** |
| Idle | **Scan card** | **Enter lang** | **Enter pass** | **Enter option** | **withdraw** | **Anything else** | **Enter option** | **Normal** |
| Idle | **Scan card** | **Enter lang** | **Enter Pass** | **Enter option** | **withdraw** | **Idle** |  | **Normal** |
| Idle | **Scan card** | **Enter lang** | **Enter pass** | **Enter option** | **withdraw** | **Anything else** |  | **Normal** |
| Idle | **Scan card** | **Enter lang** | **Enter pass** | **Enter option** | **New pass** | **Anything else** | **Enter option** | **Normal** |
| Idle | **Scan card** | **Enter lang** | **Enter pass** | **Enter option** | **New pass** | **Anything else** | **Idle** | **Normal** |
| Idle | **Scan card** | **Enter lang** | **Enter pass** | **Enter option** | **New pass** | **Idle** |  | **Low -may not occure** |
| Idle | **Scan card** | **Enter lang** | **Enter pass** | **Enter option** | **deposit** | **Anything else** | **Enter option** | **Normal** |
| Idle | **Scan card** | **Enter lang** | **Enter pass** | **Enter option** | **deposit** | **Idle** |  | **Normal** |
| Idle | **Scan card** | **Enter lang** | **Enter pass** | **Enter option** | **deposit** | **Anything else** | **Idle** | **Normal** |
| Idle | **Scan card** | **Idle** |  |  |  |  |  | **Not applied** |
| Idle | **Scan card** | **Enter lang** | **Enter pass** | **Enter option** | **Idle** |  |  | **Low – may not occure** |
| Idle | **Scan card** | **Enter lang** | **Enter pass** | **Enter option** | **Transfer op** | **Anything else** | **Idle** | **Normal** |
| Idle | **Scan card** | **Enter lang** | **Enter pass** | **Enter option** | **Transfer op** | **Idle** |  | **Normal** |
| Idle | **Scan card** | **Enter lang** | **Enter pass** | **Enter option** | **Transfer op** | **Anything else** | **Enter option** | **Normal** |
| Idle | **Scan card** | **Enter lang** | **Enter pass** | **Enter option** | **Exit** | **Idle** |  | **Normal** |

# Assertions Section:

We have written 12 assertions properties that verify the control unit which is a Finite State Machine. We wrote properties that check for each and every state that whenever the right conditions occur this state does it’s operation correctly and goes to the expected next state.

**Example:**

****

This is the most simple property , it checks that whenever the current state is idle and a user inserts his card, we move to the scan card state.

## Problems and Solutions:

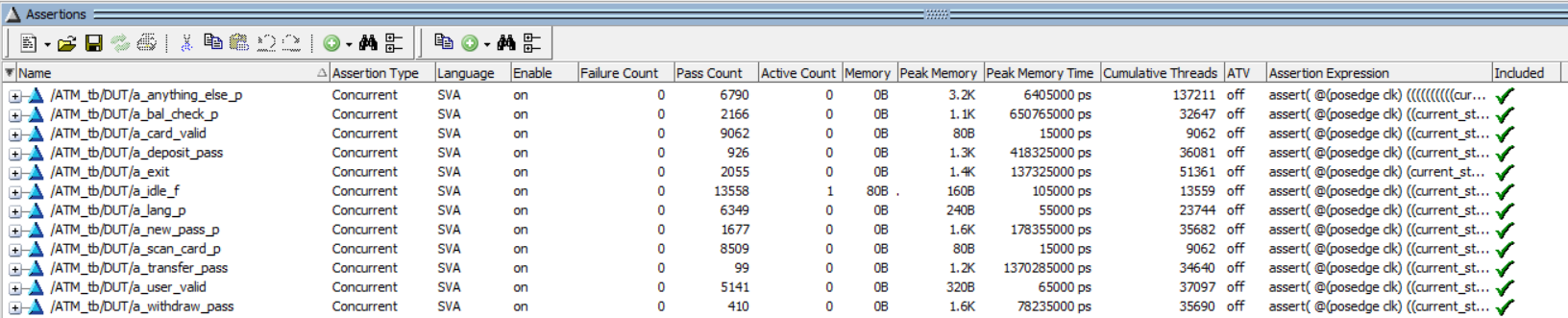
Problem\_1: At the beginning we wrote large and overlapping properties, which were difficult to read and debug.

Solution:We followed some guidelines written in the lecture slides which solved the situation such as:

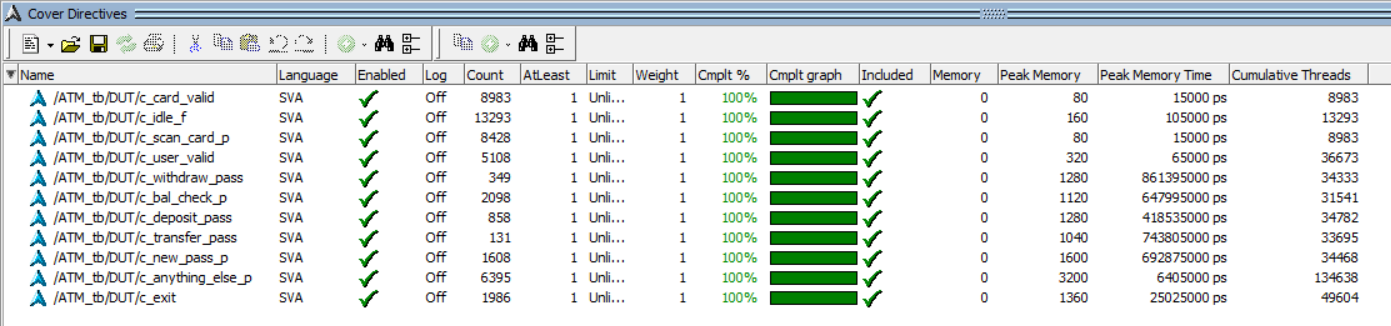
* Keep sequences and properties as simple as possible.
* Use sequences to simplify properties.
* Build complex properties out of simple, and short sequences.

# Assertions Results:

We have 12 assertion properties .. All of them passed ..

****

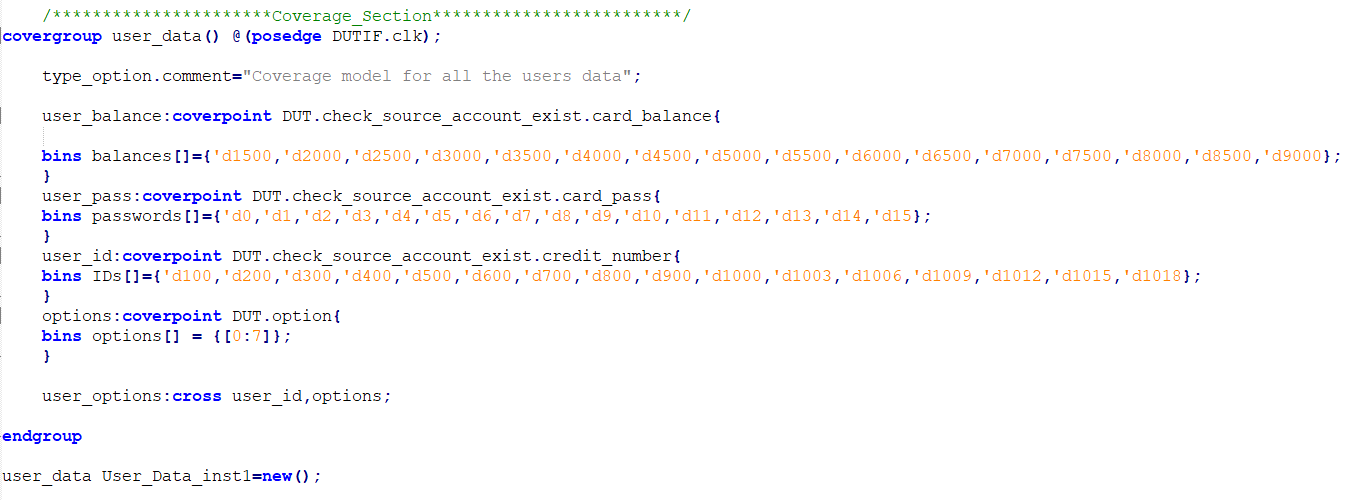
We ensured that all of them have been covered as well to make sure there are vacuous states ..

****

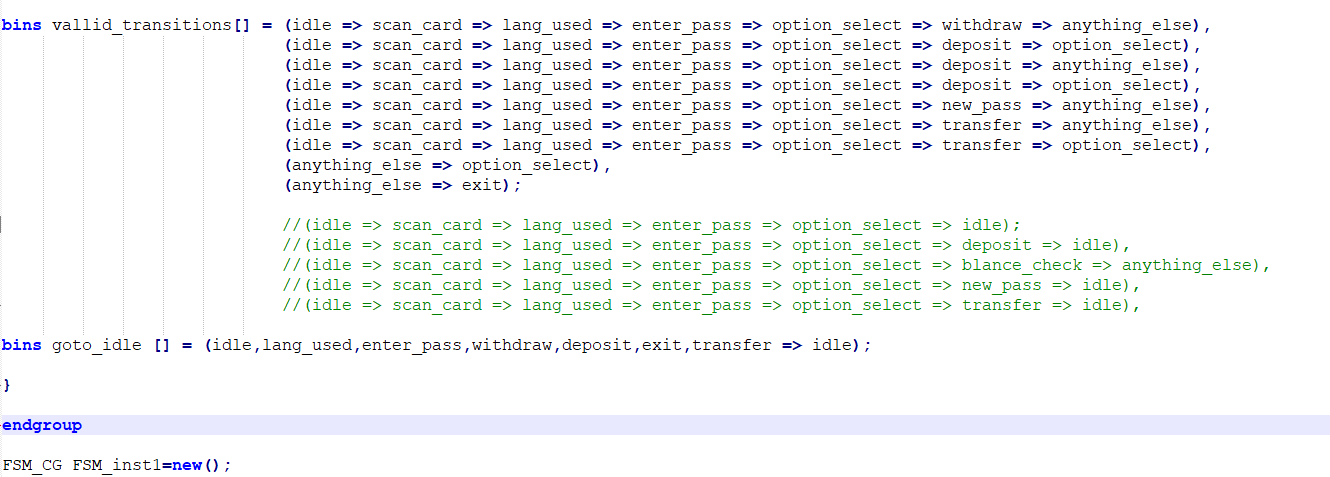
# Coverage Section:

We have 2 covergroups , below is a brief discussion about each one:

1. User\_Data Covergroup: In this covergroup we check that all the data in the users database have been accessed at least once by the testbench stimulus .. This makes us sure that all users have entered their credit numbers correctly and entered a correct password for their accounts and lastly checked their balances correctly ..

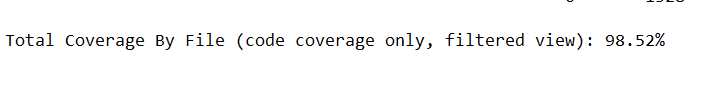


1. FSM\_Covergroup: In this covergroup we check that all states have been covered by the testbench stimulus , as well as some transitions that we care about.

****

# Coverage Results:

With respect to the code coverage , the percentage got from the coverage report is as follows:



With respect to the total Functional Coverage the results got from the coverage report is as follows:



With respect to the assertion properties coverage, the results got from the coverage report are as follows:



## Pre\_100%\_Coverage:

At the beginning the coverage percentage was not 100 % , we found that some transitions in the FSM have not been covered by the testbench stimulus, so we changed the input constraints in the testbench to solve this problem, and at some other cases we changed the priority of some inputs that would cover the uncovered states.