# ATM
# DESIGN DOCUMENT

# CONTENTS

# INTRODUCTION

This is documentation for a school project, that aims to introduce a hands-on approach to designing and verifying a digital IC for an ATM machine, in this project concepts in digital design as designing a finite state machine, register file, and the proper communication among them and all the auxiliary devices that are available.

The design has two main units “A Control Unit” that manages the auxiliary devices and outputs the proper signals to the units to achieve the wanted behavior expected from the ATM, and “RAM unit” a storage unit that holds accounts information from passwords and available balance to if this account could use the ATM service or not in the first place Figure 1

The next sections will discuss briefly the methodology we took to design it, section ‘ATM Specifications’ introduce the requirements that the design should meet, and section ‘Design Modules’ show the actual design units, their ports, and components., finally section ‘Simulation Results’ that shows the output for the design in some test cases.

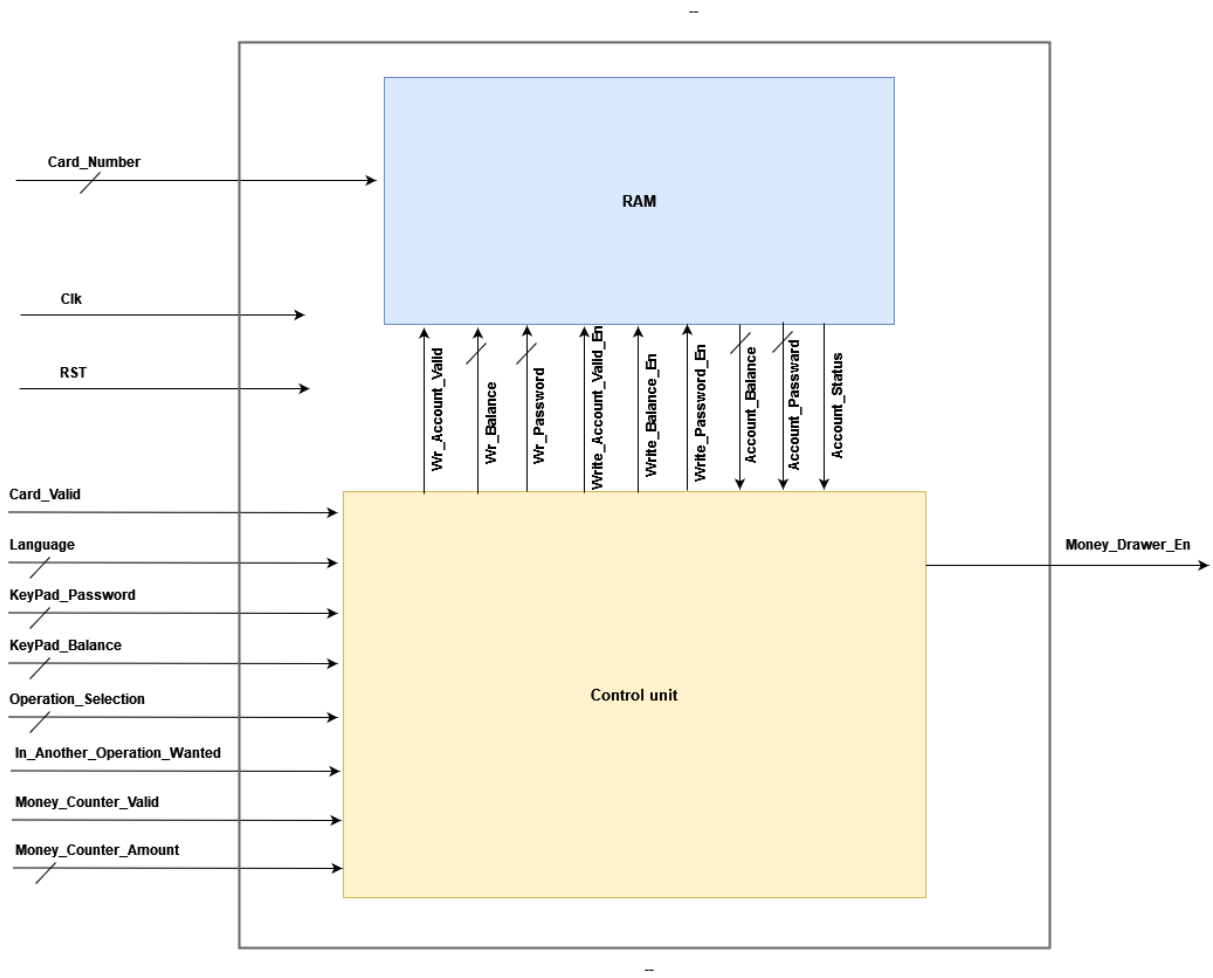


*Figure 1 ATM Control Unit*

# ATM SPECIFICATIONS

Initially, we assume that the RAM holds all the data needed at all times this data is as follows:

- 16 Bits word that represents the password for this account.
- 16 Bits word that represents the balance for this account.
- 1 Bit that represents if this account can use the ATM services or not.

Starting when the customer inserts his/her card into the card-handling device, the device decodes the account number from the card and sends it as a 16 bits word to the control unit.

After inserting the card, two scenarios could occur, first one is that the account can't use the ATM service for any reason like someone trying to log into this account by entering an incorrect password, thus the system deactivates this account to prevent any theft, the second scenario is that is account can use the AMT services and can to proceed the next

After inserting the card, a menu should appear to select the language. Normally any ATM contains 2 languages: Arabic or English. This could be also a signal called "language select" of **2-bit** [no selection-Arabic – English], the control unit should wait for a user selection or timer flag to prevent the machine from hanging

Now the ATM requests the user to insert the password for this card, the user has 5 tries to enter the password right and proceeds if not the ATM logs the account out and sets the account as inactive and the account owner should visit the nearest branch to activate the account again.

One of the **four** operation options should be selected., Options are

- Withdraw
- Deposit
- Balance services
- Change password

choosing which service the user wants leads the machine to different paths, which could be represented as **2 bits** signal, again the timer is used to prevent any hanging that could happen to the ATM

Withdraw: By selecting this option, a number should be inserted as input to determine the amount of money to be withdrawn. It will be an input to the system with a word length of 16 bits. The balance will be subtracted by the number entered and an output to show the remaining balance should also be 16 bits**,** if the amount is larger than the available balance the ATM shows a message that the balance is insufficient and proceeds to show the available balance.

Deposit: By selecting this option, the money counter device is activated to count the amount of money the user deposited and sends this count to the control unit**.** The balance will be added by the number entered and an output to show the new balance

Balance services: This option will not require an input to add or subtract to the balance, it will only output the current balance.

Change password: This option will require a 16-bit input as the new password, replacing the old password in the RAM**.**

The ATM presents the option for choosing to use another service without logging out or logging out and ending the session.

# DESIGN MODULES

As previously mentioned, there are 2 main units the control unit and a RAM, we will shine the light upon them in this section, showing the design, the ports, etc.

## 1.1 Control Unit

### 1.1.1 State Diagram

The control unit is a finite-state machine that follows the discussed sequence in the previous section, the FSM follows the state diagram stated in Figure 2

| State | Parameter | State | Parameter |
|---|---|---|---|
| Idle | 0000 | Withdraw | 0101 |
| Status_Check | 0001 | Deposit | 0111 |
| Language_Select | 0011 | Change_Password | 1111 |
| Inserting_Password | 0010 | Balance_Display | 1110 |
| DeActivate_Account | 0110 | Another_Operation | 1100 |
| Operation_Options | 0100 | _____ | _____ |

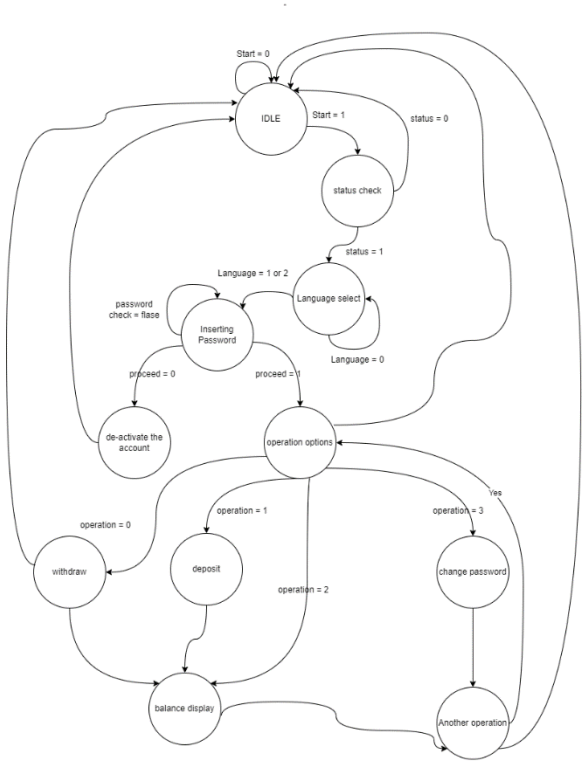


*Figure 2 State Diagram.*

### 1.1.2 Parameters

| Parameter Name | Default Value | Description |
|---|---|---|
| NUM_OF_STATES | 11 | Number of states represented in the FSM |
| NUM_OF_STATES_WIDTH | $\log_2(\text{NUM\_OF\_STATES})$ | ----- |
| PASSWORD_WIDTH | 16 Bits | The password bus width |
| BALANCE_WIDTH | 16 Bits | The balance bus width |
| NUM_OF_TRIES | 3 Tries | Number of tries for inserting the wrong password |
| NUM_OF_TRIES_WIDTH | $\log_2(\text{NUM\_OF\_TRIES})$ | -------- |
| NUM_OF_OPERATION | 4 | Number of services provided by the AMT |
| NUM_OF_OPERATION_WIDTH | $\log_2(\text{NUM\_OF\_OPERATION})$ | |

### 1.1.3 Port Map

| Port Name | Direction | Width | Description |
|---|---|---|---|
| clk | Input | 1 Bit | System clock |
| RST | Input | 1 Bit | System asynchronous reset |
| In_Start | Input | 1 Bit | The start signal comes from the card-handling device indicates a valid card has been inserted |
| In_Language | Input | NUM_SUPPORTED_LANGUAGES_WIDTH | The language selected by the user<br>• No selection=0<br>• Arabic = 1<br>• English = 2 |
| In_Keypad_Password | Input | PASSWORD_WIDTH | The password entered by the user |
| In_Keypad_Balance | Input | BALANCE_WIDTH | The amount of money that the user wants to withdraw |
| Operation_Selection | Input | NUM_OF_OPERATION_WIDTH | The service which the user wants:<br>• Withdraw = 0<br>• Deposit =1<br>• Show Balance=2<br>• Change Password=3 |
| In_Another_Operation_Wanted | Input | 1 Bit | If the user wants to conduct another operation |

| Port Name | Direction | Width | Description |
|---|---|---|---|
| In_Account_Passward | Input | PASSWORD_WIDTH | Account's password from the RAM |
| In_Account_Passward | Input | BALANCE_WIDTH | Account's balance from the RAM |
| Account_Status | Input | 1 Bit | Account's status from the RAM |
| Out_Write_Password_En | Output | 1 Bit | Write new password enable to the RAM |
| Out_Write_Balance_En | Output | 1 Bit | Write new balance enable to the RAM |
| Out_Write_Account_Valid_En | Output | 1 Bit | Write new status enable to the RAM |
| Out_Wr_Password | Output | PASSWORD_WIDTH | Write new password to the RAM |
| Out_Wr_Balance | Output | BALANCE_WIDTH | Write new balance to the RAM |
| Out_Wr_Account_Valid | Output | 1 Bit | Write new status to the RAM |
| In_Money_Counter_Valid | Input | 1 Bit | Money counter device valid signal |
| In_Money_Counter_Amount | Input | BALANCE_WIDTH | Money counter device counted amount |
| Out_Money_Drawer_En | Output | 1 Bit | Money drawer enable signal |

## 1.2 RAM

The RAM unit is represented by a register file that holds the account data and interacts with the control unit.

### 1.2.1 Parameters

| Parameter Name | Default Value | Description |
|---|---|---|
| SAVED_ACCOUNTS | 10 | Number of Accounts that the RAM can store |
| ACCOUNT_NUMBER_WIDTH | $\log_2(\text{SAVED\_ACCOUNTS})$ | ----- |
| PASSWORD_WIDTH | 16 Bits | The password bus width |
| BALANCE_WIDTH | 16 Bits | The balance bus width |

### 1.2.2 Port Map

| Port Name | Direction | Width | Description |
|---|---|---|---|
| clk | Input | 1 Bit | System clock |
| RST | Input | 1 Bit | System asynchronous reset |
| In_Address | Input | ACCOUNT_NUMBER_WIDTH | Account number found on the card coming from the card handling device |
| Out_Re_Password | Output | PASSWORD_WIDTH | Account's password from the RAM |
| Out_Re_Balance | Output | BALANCE_WIDTH | Account's balance from the RAM |
| Out_Re_Account_Valid | Output | 1 Bit | Account's status from the RAM |
| In_Write_Password_En | Input | PASSWORD_WIDTH | Write a new password enable |
| In_Write_Balance_En | Input | BALANCE_WIDTH | Write a new balance enable |
| In_Write_Account_Valid_En | Input | 1 Bit | Write a new status enable |
| In_Wr_Password | Input | PASSWORD_WIDTH | Write a new password value |
| In_Wr_Balance | Input | BALANCE_WIDTH | Write a new balance value |
| In_Wr_Account_Valid | Input | 1 Bit | Write a new status value |

# SIMULATION

In this section a few simulation results are shown, not for full functional verification purpose

## 1.3 Test Case #1

In this test case, a withdrawal scenario will be tested, by trying to withdraw 90 EGP.

In Figure 3, we can notice the state transition from Idle to status check to password, when the user enters the password that is right and matches the password stored in the RAM, then the user request withdraw service and withdraws 90 EGP from an available balance of 1841 EGP, thus the remaining balance is 1751 EGP

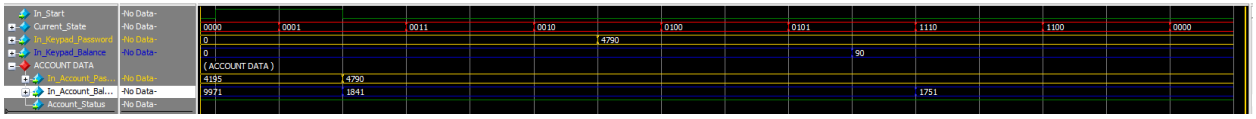


*Figure 3 Test Case #1 Waveform*

## 1.4 Test Case #2

In this test case, a user changes the account password to 8998 instead of 4790, and we can see that the new password is stored in the RAM replacing the old password


*Figure 4 Test Case #2 Waveform*