**The Simulation Project**

**Digital Logic (Fall 2019)**

**Deadline: before 9th of Bahman 1398**

Design and implement a digital circuit to control an ATM machine. This machine must provide the following functionalities:

1. Charge bank account
2. Withdraw money from account.
3. Return the current balance.
4. Transfer money to another account using the card number
5. Verify the password.

Notes:

* For simplicity reasons, the ATM machine only accepts the operations of 1 user. (So we only need to save 1 password and 1 account password). The design of memory mechanism is up to you (You can implement any kind of register/flip flop combinations).
* As you might know, an ATM machine has 24 buttons: 8 control buttons (4 at each side of the screen) and 16 buttons to enter numeric values and submit/clear etc. entered values. The ATM machine that you must design has 4 control buttons. The entered number is a binary number ranging from 0 to 1024. So you have 14 input bits (4 control and 10 number bits). If you need more control bits you can use more signals but you should completely specify why you used those bits. Also the device has some output bits: We need one output bit to prompt the user of incorrect password (It is asserted if the user enters an incorrect password at the first step in flow and it is “don’t care” otherwise). 1 bit must be used to indicate that a user is requesting more than the account’s current balance (asserted only if user is withdrawing money and balance is not enough, “don’t care” otherwise). Other input bits are introduced in the following notes when needed.
* The ATM’s flow of function is as following: The user inputs a number using input bits and hits a control button (which sets first control bit to 1). The ATM machine must check the password and in case of invalidity, sets WRONG\_PASSWORD bit to 1. Otherwise, user can choose from functionalities, we mentioned above using the control bits.
* Charging flow: user enters a number as money value using number bits. The ATM machine must add the entered value to the previous account balance and save it. Then, returns to the functions menu.
* Withdraw flow: User can select a fixed amount from given values; 3 control bits correspond to 3 different values to be used. If users enters a number and uses other control bit, the entered value is withdrawn from account. You must be careful that the accounts balance shall not be zero; so you must assert INSUFFICENT\_BALANCE output bit.
* Return balance flow: User can select either to view the balance or get a printed receipt. In case of choosing a printed solution, the user is charged a fixed amount. Also the GET\_BALANCE\_MODE output bits are set according to the following table:

|  |  |
| --- | --- |
| mode | GET\_BALANCE\_MODE |
| view | 10 |
| printed | 01 |
| Any other case | “don’t care” |

* Transfer Money: The user enters the account number of the second party and then enter the money value for transferring. The ATM machine validates the account number with the name of second party. If yes, transfer the money. That money should be reduced from the account balance.
* The system may need other input/output bits. It is up to you to add new input/output bits if you need so feel free to do so if you need to.
* The project must be documented and explained clearly; so report all the steps of implementation, flow of the system functionality and details of any functionality you provide. Further implementation of new and rich features is much approved and applaud.