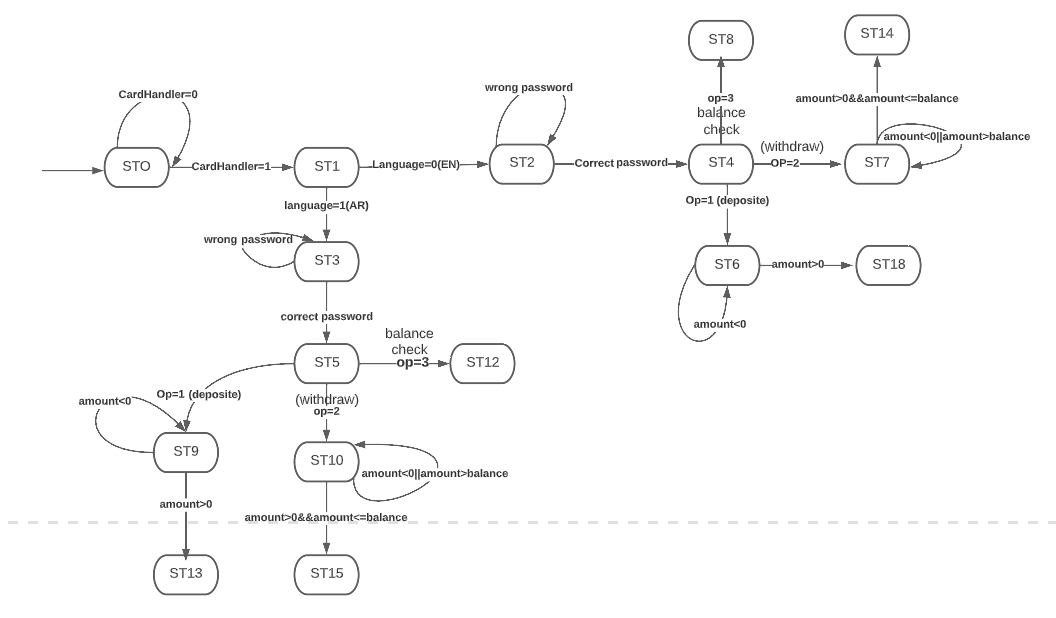
EDA Project (Phase 1)

# Design

## State Machine



## Assumptions

* Card Reader is already implemented and turns flag card handler=1 when detected
* We have only one user with initial Known balance (1000000) and certain password defined (4’b1111) in real world this info is gathered form database
* Money counting always returns true when deposited or withdraw
* Our output is the string to be displayed on the screen
* ST0: str = ”Insert your card”
* ST1: str= “choose language”
* ST2(EN), ST3(AR): str= ”enter password”
* ST4(EN), ST5(AR): “str=Choose service
* Deposited
* Withdraw
* Balance “
* ST8, ST12: str= (“your balance is %d”, balance)
* ST13(EN), ST18(AR): str = (”Deposit succeeded, Your balance now is %d”, balance +amount)
* ST14, ST15: str= (“Withdraw succeeded, your balance now is %d”, balance-amount)
* Exit key (in = 4’b0000) from st6,st7,st8,st12,st14 sends you to ST4
* Exit key (in = 4’b0000) from st9,st10,st11,st13,st15 sends you to ST5

# Test

## Test plan:

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Current state | Action | Next state |
| 1. | ST0 | (Cardhandler = 1)  Enter the card | ST1 |
| 2. | ST1 | (Language = 2 (EN))  Choose English | ST2 |
| 3. | ST2 | Enter wrong password | ST2 |
| 4. | ST2 | Enter correct password (4’b1111) | ST4 |
| 5. | ST4 | Choose Deposit operation (op=1) | ST6 |
| 6. | ST6 | Choose amount <= 0 (invalid amount) | ST6 |
| 7. | ST6 | Choose amount > 0 (valid amount) | ST12 |
| 8. | ST12 | Press Exit key (in = 4’b0000) | ST4 |
| 9. | ST4 | Choose Deposit operation (op=2) | ST7 |
| 10. | ST7 | Choose amount <= 0 or > balance (invalid amount) | ST7 |
| 11. | ST7 | Choose amount > 0 and <= balance (invalid amount) | ST14 |
| 12. | ST14 | Press Exit key (in = 4’b0000) | ST4 |
| 13. | ST4 | Choose Deposit operation (op=3) | ST8 |
| 14. | ST8 | Reset = 1 then Reset =0 (Async reset) | ST0 |
| 15. | ST0 | (Cardhandler = 1)  Enter the card | ST1 |
| 16. | ST1 | (Language = 1 (AR))  Choose Arabic | ST3 |
| 17. | ST3 | Enter wrong password | ST3 |
| 18. | ST3 | Enter correct password (4’b1111) | ST5 |
| 19. | ST5 | Choose Deposit operation (op=1) | ST9 |
| 20. | ST9 | Choose amount <= 0 (invalid amount) | ST9 |
| 21. | ST9 | Choose amount > 0 (valid amount) | ST13 |
| 22. | ST13 | Press Exit key (in = 4’b0000) | ST5 |
| 23. | ST5 | Choose Deposit operation (op=2) | ST10 |
| 24. | ST10 | Choose amount <= 0 or > balance (invalid amount) | ST10 |
| 25. | ST10 | Choose amount > 0 and <= balance (invalid amount) | ST15 |
| 26. | ST15 | Press Exit key (in = 4’b0000) | ST5 |
| 27. | ST5 | Choose Deposit operation (op=3) | ST11 |
| 28. | ST11 | Reset = 1 then Reset =0 (Async reset) | ST0 |

## Report

**Statement coverage:** 100% all statements are covered in the testbench

**Branch coverage:** some branches are not covered like exit from ST9 , ST10 , ST6 , ST7 and reset from all states except ST5 , ST4 ( 63.82% (30/47) if we consider async reset on each state is branch , if we neglect it 87.5% (28/32) )

## Wave form