Vending Machine report

Chang Zihao
20206018
Cui Yuxuan
20206019

Contents

1 Introduction

A Finite State Machine (FSM) is a machine that can be in exactly one of a finite number of states at any given time. The FSM can transit from one state to another state in response to external inputs; the transition between states is predefined in the FSM. An FSM is defined with a list of states, conditions for each transition, and initial state. FSMs are of two types of FSM, which are the Moore machine and the Mealy machine. A Moore machine is an FSM whose outputs are determined only by its current state. This is in contrast to a Mealy machine whose outputs are determined by both its current state and inputs. We will use these two structures to design a vending machine based on these two structures.

A vending machine needs to have the functions of coin-operating, checking available items, selecting items and refunding coins.

1.1 coin-operating

When someone puts in the coin, the vending machine records the number of changed coins, so that the number of coins updated can be recorded in this clock cycle.

1.2 checking available items

Once the coins changed in one cycle, the available items will be shows up.

1.3 selecting items

When someone chooses an item, the vending machine needs to first check whether the coins are sufficient, and then record it. If there is no problem, the item is output in the next clock cycle.

1.4 refunding coins

My idea is that after each coin is thrown, after selecting the item, the remaining amount of money is calculated and the change is checked, and the currency is ready to be refunded. Every clock cycle requires the currency to be refunded. When you need to refund coins, refund one at one time.

2 Design

2.1 Module design

To design a vending machine, we should know how the vending machine works. This vending machine is a sample machine based on the FSM. The process of the machine is that input coins and then choose items like drinking, or water, then it will give changes and output items. And in the code, we divided into three modules.

2.2 input module

The input module is input coins, this part will get the coin number changes and the coin value changes, and select items changes.

2.3 output module

The output module is when a state change is detected, this part will start to operate, ready for output.

2.4 record module

The third part is at the posedge, update the data, and ready to return the coins. Every record operation only can be done at this time. Once finished, the vending machine will jump to output module if needs.

3 Implementation

Like we said in the design part, we divided the code into three-part.

3.1 input module

First set 0 at the beginning, all inputs must set to 0, because if we don't set 0, all inputs will in a high resit Then input the coin. There are three kinds of coins 100,500,1000. *i_input_coin*are three bits binary number. Then input *i_input_coin*, select the corresponding value coin, then coin number plus 1, the total value increase the corresponding coin value. Similar to the input coin select items are also used three bits binary number to control the items. And determine which item can we buy. Then the number of the item plus 1. I used the case statement and if statement here. Both of them are fine for the feature.

3.2 output module

And for the second part is to output the items. After we select items, then the number of the items is changed, after the posedge pass, it will output the items, one item at a time.

3.3 record module

The third part is the most important. At the posedge if we choose to reset, all variables should be set 0. And all inputs must set to 0, because if we don't set 0, all inputs will in a high resit And else case we first update the value, let $num\_coins = num\_coins + num\_coins\_nxt$. Then update which items can we buy with the value of the coin. At last, we must calculate the remaining value of the coin, then return the coin. The method is, first determine how much the coin remained, and compare it with the coin value. If it more than or equal to 1000 then first return a 1000 coin, then the remained value is more than 500 then return a 500 coin and so on. Then a cycle is done.

4 Evaluation

We pass all the tests except ”ItemTest, CoinTest1, and CoinTest2”.

```
# TEST                              InitialTest :
# PASSED
# TEST                          Insert100CoinTest :
# PASSED
# TEST                          Insert500CoinTest :
# PASSED
# TEST                         Insert1000CoinTest :
# PASSED
# TEST                          Select1stItemTest :
# PASSED
# TEST                          Select2ndItemTest :
# PASSED
# TEST                          Select3rdItemTest :
# PASSED
# TEST                          Select4thItemTest :
# PASSED
# TEST                             WaitReturnTest :
# PASSED
# TEST                          TriggerReturnTest :
# PASSED
# Passed = 10, Failed = 0
# 1
```

5 Discussion

We first encountered some problems in understanding the meaning of the question, but in the end by looking at the test code, we thoroughly understood the meaning of the question. And at last, we pass all the test and finished the whole project. I hope that future projects will give enough time to think and complete the code.

6 Conclusion

We implement the basic vending machine. After that, I think we totally understood the FSM, how it works, and how to make it works. To finish this project in limit time is a challenge.