

# **ALU project report**

**Chang Zihao**

**20206018**

**Cui Yuxuan**

**20206019**

**Contents**

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Design</b>	<b>2</b>
2.1	Module design . . . . .	2
<b>3</b>	<b>Implementation</b>	<b>3</b>
<b>4</b>	<b>Evaluation</b>	<b>4</b>
<b>5</b>	<b>Discussion</b>	<b>5</b>
<b>6</b>	<b>Conclusion</b>	<b>5</b>

# **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.

## **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 parts. The first parts are input coins, this part will get the coin number changes and the total coin value changes, then select items changes. The second parts when a state change is detected, this part will start to operate, ready for output. The third part is at the posedge, update the data, and ready to return the coins.

### 3 Implementation

Like we said in the design part, we divided the code into three-part. 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 resist. Then input the coin. There are three kinds of coins 100,500,1000.  $input_{coin}$  are three bits binary number. Then input  $input_{coin}$ , 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. Then the number of the item plus 1.

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

The third part is the most important. At the posedge if we choose to reset, all variables should be set 0. And else case we first update the value, let  $num_{coins} = num_{coins} + num_{coins_{next}}$ . Then determine 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 while project.

## **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.