**VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY  
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY**

**DEPARTMENT OF ELECTRONICS**



**EE3043: COMPUTER ARCHITECTURE**

**Milestone 1: Vending Machine**

|  |  |  |  |
| --- | --- | --- | --- |
| Instructor | : | Dr. Tran Hoang Linh |  |
| TA | : | M.S. Cao Xuan Hai |  |
| Group | : | 23 |  |
| Members | : | Doan Dinh Nam  Giang Anh Duc  Hoang Minh Chien | * 2110368 * 2011103 * 2112932 |

*Ho Chi Minh City, Octorber 07, 2024*

**PREFACE**

This report is submitted as part of the coursework for the *Computer Architecture* class, focusing on the design and implementation of a vending machine using Verilog/SystemVerilog. Throughout this milestone, we had the opportunity to apply theoretical concepts related to finite state machines (FSMs) and digital logic design in a practical, hands-on environment.

We would like to express our gratitude to our instructor and teaching assistants for their guidance.

*Ho Chi Minh City, October 07, 2024*

**Team Leader**

*Nam*

Nam Doan Dinh

nam.doanlqd@hcmut.edu.vn

**CONTENTS**

[1. PROBLEM 2](#_Toc179234868)

[2. DESIGN STRATEGY 3](#_Toc179234869)

[Description 3](#_Toc179234870)

[2.1. 3](#_Toc179234876)

[2.2. I/O Description 4](#_Toc179234921)

[3. VERIFICATION STRATEGY 5](#_Toc179234934)

**LIST OF FIGURES**

[Figure 1: Block diagram 2](#_Toc179234743)

[Figure 2: Expected Waveform 2](#_Toc179234744)

[Figure 3: Moore machine with 9 states of Vending machine 3](#_Toc179234745)

[Figure 4: Vending Machine’s block diagram 4](#_Toc179234746)

[Figure 5: Wave form of Vending Machine with random input coins 6](#_Toc179234747)

[Figure 6: Tcl console window show the change of values ($monitor) 6](#_Toc179234748)

1. PROBLEM

A vending machine is a machine capable of accepting coins or paper money and dispensing soft drinks or snacks. In this exercise, you need to design a vending machine that meets the following requirements:

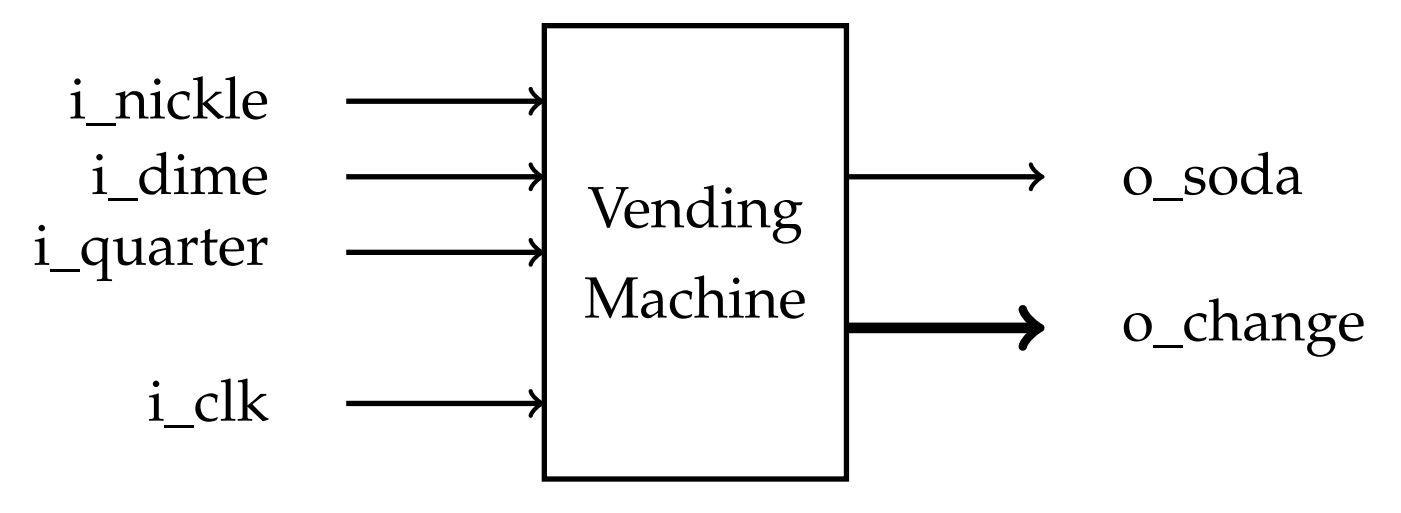


Figure : Block diagram

1. It can accept coins: ¢5 (Nickel), ¢10 (Dime), ¢25 (Quarter), but only one coin at a time (or per clock).
2. When the deposit exceeds ¢20, it dispenses a soda and a change.
3. Change is a 3-bit binary data.

000 ¢0

001 ¢5

010 ¢10

011 ¢15

100 ¢20

In this example, the system accepts a dime and then a quarter as input from the customer. In the subsequent cycle, the system dispenses a soda and provides a change of ¢15.

*That is, a can of soda costs*  ¢*20.*

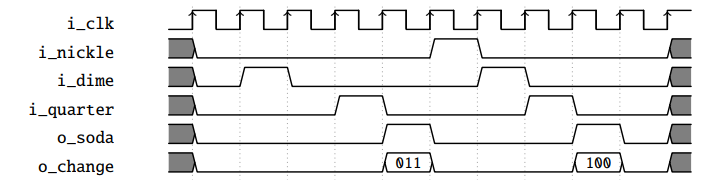


Figure 2: Expected Waveform

1. DESIGN STRATEGY
   1. Description

**State Diagram**

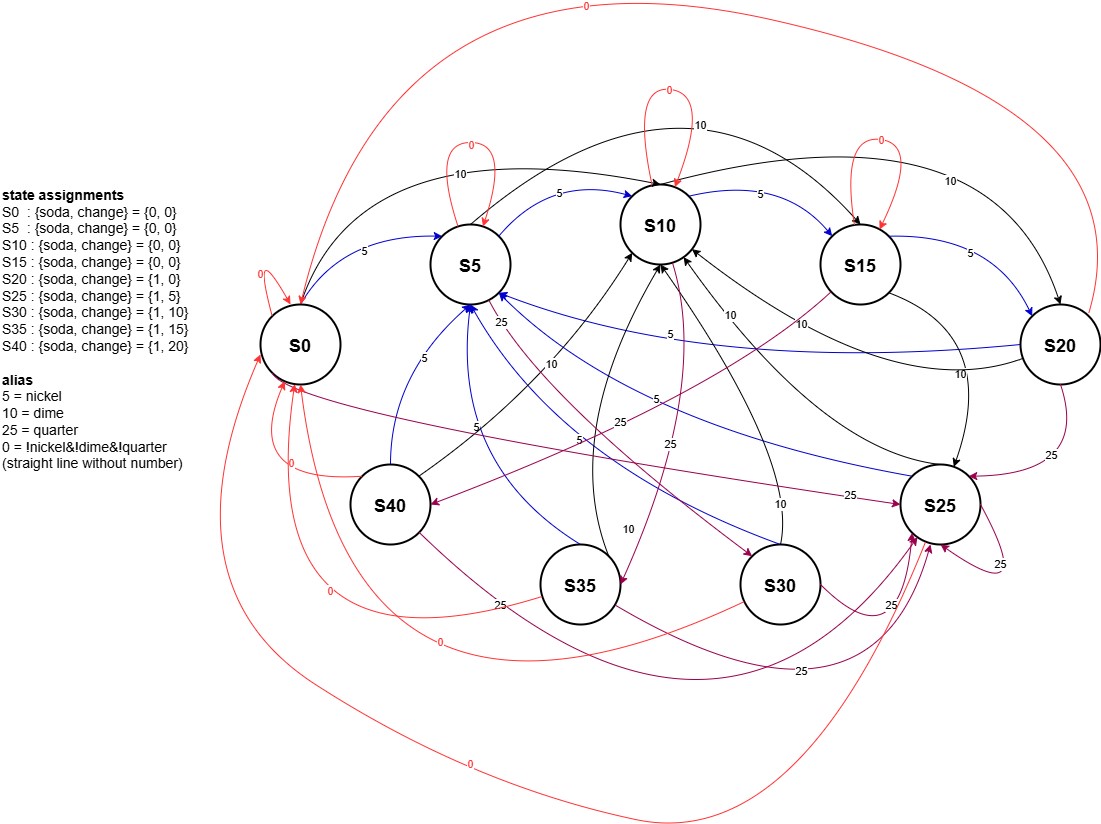


Figure 3: Moore machine with 9 states of Vending machine

We first consider there is 9 possible state that the vending machine have:

* S … represent for the total moneys inside the machine.
* {soda, change} represent for the outputs attached with that current state.

**State assignments**

|  |  |
| --- | --- |
| State | State Description |
| S0 | The machine receives a total of 0 cents, does not release soda, returns 0 cents |
| S5 | The machine receives a total of 5 cents, does not release soda, returns 0 cents |
| S10 | The machine receives a total of 10 cents, does not release soda, returns 0 cents |
| S15 | The machine receives a total of 15 cents, does not release soda, 0 cents is lost |
| S20 | The machine receives a total of 20 cents, releases soda, and returns 0 cents |
| S25 | The machine receives a total of 25 cents, releases soda, and returns 5 cents |
| S30 | The machine receives a total of 30 cents, releases soda, and returns 10 cents |
| S35 | The machine receives a total of 35 cents, releases soda, and returns 15 cents |
| S40 | The machine receives a total of 40 cents, releases soda, and returns 20 cents |

**Block Diagram**

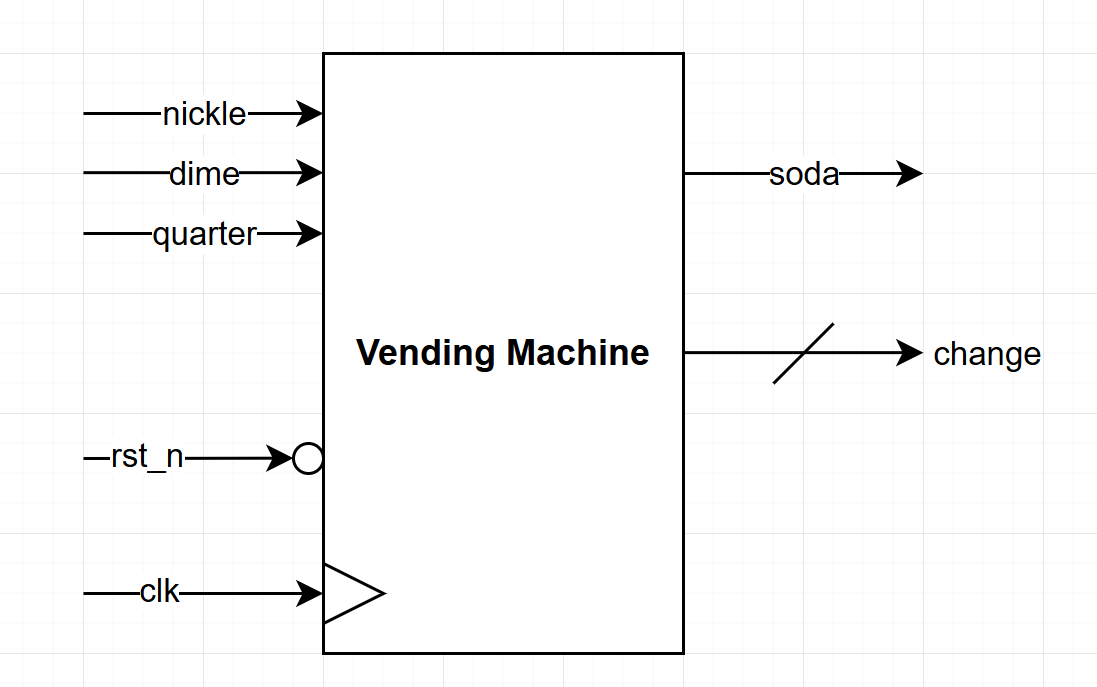
****

Figure : Vending Machine’s block diagram

* 1. I/O Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **I/O** | **Width** | **Description** |
| clk | Input | 1 | Operation Clock |
| rst\_n | Input | 1 | Asynchronous reset. Active Low |
| nickle | Input | 1 | nickle = 1: machine received one nickle  nickle = 0: machine not received nickle |
| dime | Input | 1 | dime = 1: machine received one dime  dime = 0: machine not received dime |
| quarter | Input | 1 | quarter = 1: machine received one quarter  quarter = 0: machine not received quarter |
| soda | Output | 1 | output a soda |
| change | Output | 3 | output a change |

1. VERIFICATION STRATEGY

To create an item list, we use the random function to generate a random value for the triplet {nickle, dime, quarter} so that at a time only 1 coin is inserted, meaning the value of the 3-bit binary code { nickle, dime, quarter} can only be one-hot codes.

We choose the $urandom\_range function to randomize the value for the random variable, then use the case statement to create 4 case items corresponding to the 4 random variables to have the code for the coins: 3'b000, 3'b001, 3'b010, 3' b100.

**Waveform**

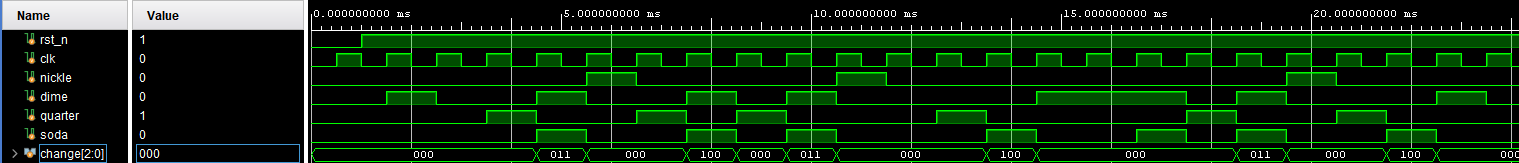


Figure : Wave form of Vending Machine with random input coins

**Tcl console**

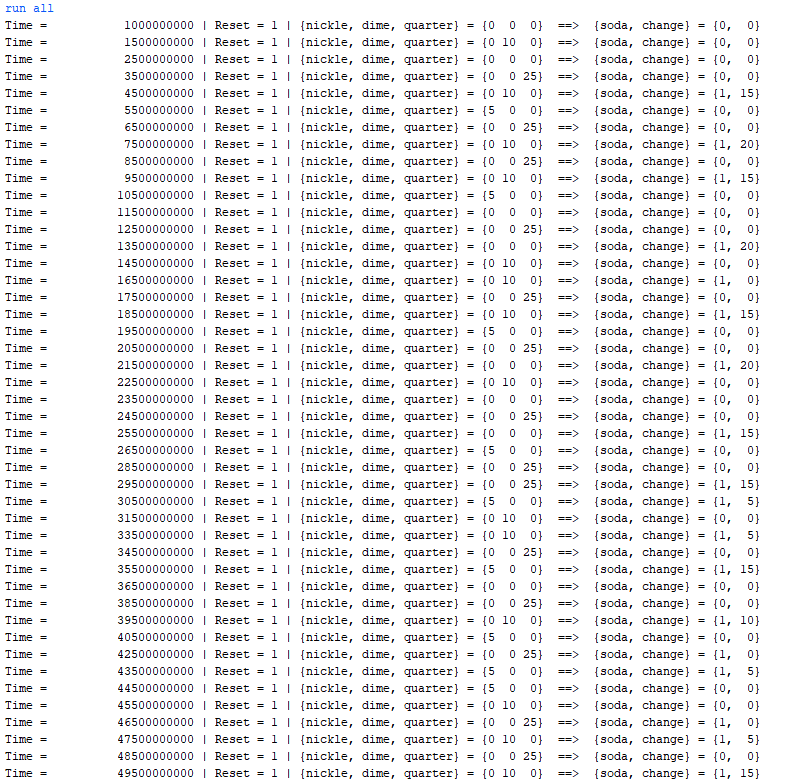


Figure : Tcl console window show the change of values ($monitor)

* Through careful comparing with expected results, we found that the design met the requirements.