

Design of a Vending Machine

Bang Nguyen

Contents

1	Objectives	1
2	Problem	1
3	Analysis and Design	2
3.1	Block diagram	3
3.2	Describe the problem in detail	3
3.3	State Transition Table	4
3.4	State Transition Diagram	5
3.5	Waveform	6

1 Objectives

- Review basic logic design and FSM concepts.
- Design a Vending Machine using System Verilog.

2 Problem

Vending Machine is a dispenser machine that receives coins or bills and dispenses soft drinks or snacks. A simple vending machine controller is described in Figure 1.

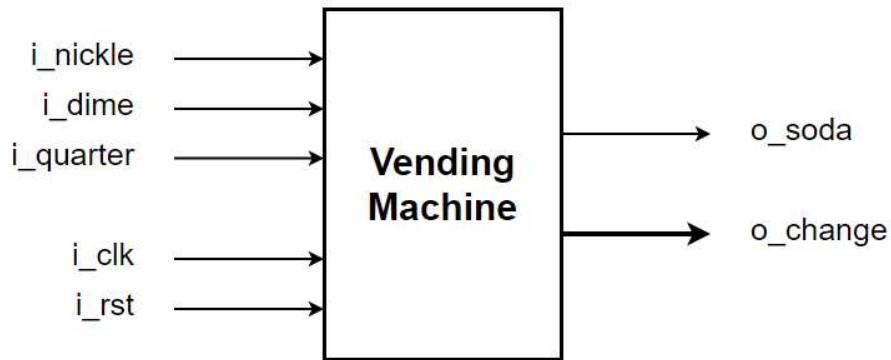


Figure 1: Vending machine's ports

Design a vending machine that meets the following specifications:

1. The vending machine is capable of accepting coins: ¢5 (Nickle), ¢10 (Dime), ¢25 (Quarter). However, it should only accept one coin at a time (per clock cycle).
2. When the deposited amount exceeds ¢20, it should dispense a soda and calculate the correct change.
3. The change is represented as a 3-bit data:

000	¢0
001	¢5
010	¢10
011	¢15
100	¢20

For 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.

3 Analysis and Design

Based on the design requirements, we proceed to build a state machine for the vending machine based on the Moore structure, and the coins accepted only include ¢5 (Nickle), ¢10 (Dime), and ¢25 (Quarter), with only one coin accepted at a time. Based on the change return mechanism, we observe that there are 9 states corresponding to 9 amounts of money

in the machine before dispensing soda and change. Each state will provide soda and change based on the current state of the machine.

3.1 Block diagram

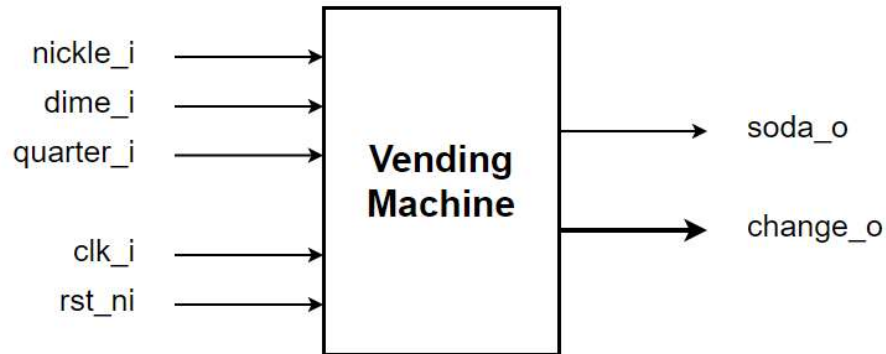


Figure 2: Block diagram of vending machine

3.2 Describe the problem in detail

State	State Description
S0	The machine received ¢0, did not dispense soda, and returned ¢0
S5	The machine received ¢5, did not dispense soda, and returned ¢0
S10	The machine received ¢10, did not dispense soda, and returned ¢0
S15	The machine received ¢15, did not dispense soda, and returned ¢0
S20	The machine received ¢20, dispense soda, and returned ¢0
S25	The machine received ¢25, dispense soda, and returned ¢5
S30	The machine received ¢30, dispense soda, and returned ¢10
S35	The machine received ¢35, dispense soda, and returned ¢15
S40	The machine received ¢40, dispense soda, and returned ¢20

Table 1: State table for possible states of a Moore state machine

Port	Port Description
clk_i	Clock signal, rising edge at the input
rst_ni	Active low reset signal at the input
nickle_i	Signal for receiving a nickle (¢5), active high

dime_i	Signal for receiving a dime (¢10), active high
quarter_i	Signal for receiving a quarter (¢25), active high
soda_o	Signal for dispensing soda, active high
change_o	Signal for returning change, active high

Table 2: Describing the ports on the state machine

Binary	Money
000	¢0
001	¢5
010	¢10
011	¢15
100	¢20

Table 3: Description of the ports on the state machine

3.3 State Transition Table

Current State	Input			Next State	Output	
	nickle_i (¢5)	dime_i (¢10)	quarter_i (¢25)		soda_o	change_o
S0	0	0	0	S0	0	000
	1	0	0	S5	0	000
	0	1	0	S10	0	000
	0	0	1	S25	1	001
S5	0	0	0	S5	0	000
	1	0	0	S10	0	000
	0	1	0	S15	0	000
	0	0	1	S30	1	010
S10	0	0	0	S10	0	000
	1	0	0	S15	0	000
	0	1	0	S20	1	000

	0	0	1	S35	1	011
S15	0	0	0	S15	0	000
	1	0	0	S20	1	000
	0	1	0	S25	1	001
	0	0	1	S40	1	100
S20	0	0	0	S0	0	000
S25	0	0	0	S0	0	000
S30	0	0	0	S0	0	000
S35	0	0	0	S0	0	000
S40	0	0	0	S0	0	000

Table 4: The state transition table of the state machine

3.4 State Transition Diagram

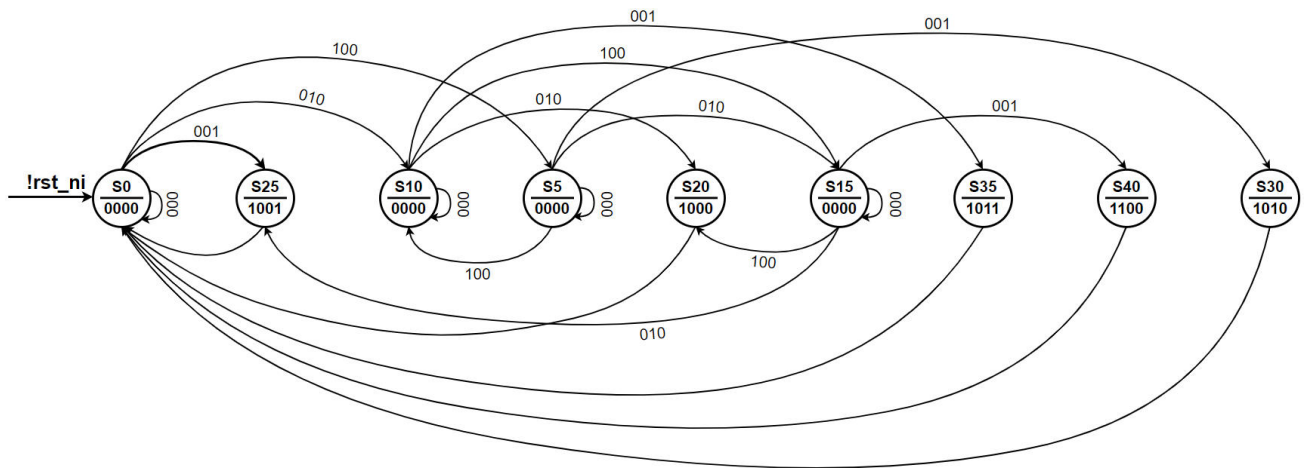


Figure 3: Finite State Machine

This Finite State Machine (FSM) represents a vending machine that accepts nickels (¢5), dimes (¢10), and quarters (¢25) to dispense soda when the total inserted amount reaches or exceeds ¢20. The machine has several states labeled S0 to S40, where the number indicates the cumulative amount inserted in cents. At each state, the user can insert one of the three coins, transitioning the machine to a new state based on the amount. When the state reaches S20 or beyond, the machine dispenses a soda (soda = 1) and returns the correct change if the inserted amount exceeds ¢20 (change = {¢0, ¢5, ¢10, ¢15, ¢20}).

For example, if the state is S25, the machine dispenses a soda and returns €5 in change. If the state is S40, it dispenses a soda and returns €20. This FSM ensures that only one soda is dispensed per transaction and handles change appropriately.

3.5 Waveform

Observing the simulation waveform, we can see that when the signal `rst_ni = 0`, the input signals `nickle_i`, `dime_i`, and `quarter_i` all have a value of 0. When the signal `rst_ni = 1`, these input signals are randomly assigned values (0 or 1) on each rising edge of the clock. Only one of the three signals has a value of 1, while the other two signals have a value of 0. At the same time, the next state (`next_state`) is updated from the current state (`current_state`) according to the designed FSM.

In conclusion, based on the simulation waveform, it can be confirmed that the vending machine design meets all the specified design requirements.



Figure 4: Waveform on ModelSim