# Vending Machine

EE 271, Electrical Engineering, College of Engineering, San Jose State University
Professor: Dr. Binh Le
Johan Saltin
ID: 010933728
Talal Khalil
ID: 008796281
Narues Wagokh
ID:013768144

## Abstract

The following document includes the project of designing a simple vending machine in Verilog and implementing the design on the DE10-Lite FPGA using Intel's Quartus-lite software. The vending machine design comprises of seven states with five available products and three available amount selections. In addition, the testbench, simulations, and waveforms for different test cases are provided.

## Introduction

Customers are in need of consumables at any time. Therefore, vending machines are in desperate need of all-nighter students, night shift workers, and many more. In the following project, a 7 state machine is therefore created to satisfy five different products and three different input amounts, with a change system implemented as well.

The requirements are the following: the vending machine needs to be designed with 7 different states; it has to be a finite state machine; an FPGA DE10-Lite needs to act as a vending machine; it has to be fully written in Verilog code; there have to be five different products; there have to be three input values, depending on cost of product and preference; the LEDs have to be used at preference; the seven segment display or an LCD has to display at preference; in conclusion, a report, video, and Quartus project has to be submitted.

For the following project, the provided switches, LEDs, buttons, and the seven-segment displays were used to create the "ultimate" vending machine experience.

The state machine design and schematic are seen in Figures 1 and 2.

## Working Specifics

The designed vending machine offers five products (A, B, C, D, E) and contains three possible amount selections (quarter, fifty, dollar).

Operation flow of the vending machine:

A. State 0: Waiting for Product Selection

The vending machine is initially be in a "waiting" state for the user to select a product where a "HELLO" message will be displayed on the 7-Segment display. The selection of a product results in a transition to State 1.

B. State 1: Price Assignment

After a user selects the desired product, the price of the product is assigned accordingly and will be displayed on the 7-Segment display. The confirmation of the selected product results in a transition to State 2.

C. State 2: Check Product Availability

The availability of the selected product is checked. If the selected product is available, a "yes" message will be displayed on the 7-Segment display and the confirmation from the user results in a transition to State 3. Otherwise, an "Error" message is displayed and a transition to State 0 occurs.

D. State 3: Money Insertion

The total inserted amount of money is continuously calculated until it either exceeds the maximum amount of money allowed (200 cents) or until the user indicates being done with inserting money. In either case, a transition to State 4 occurs.

E. State 4: Check if Inserted Money is Sufficient

The total inserted amount of money is checked against the price of the selected product. If the total is less than the price of the selected product price, the money inserted will be returned as change. If the total exceeds the price of the selected product, the insufficient flag will be set to zero and a transition to State 5 occurs.

F. State 5: Dispense Product

The selected product will be dispensed and confirming receival of the product by the user results in a transition to State 6.

G. State 6: Return Change

The change is calculated, and the value will be displayed on the 7-Segment display,

## State Diagram

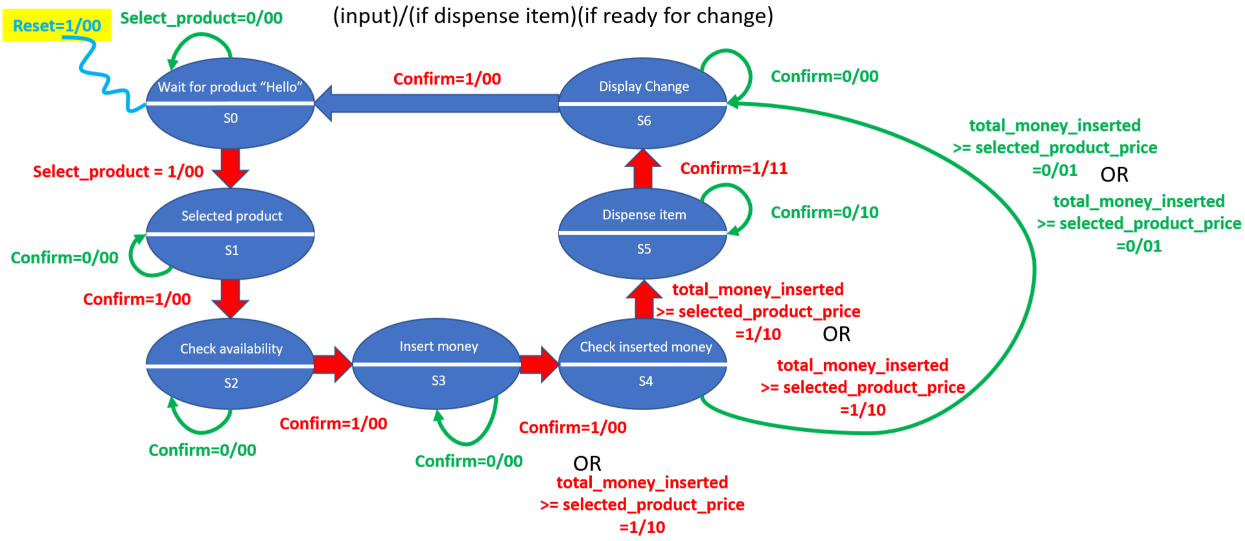

Reset=1/00
Select_product=0/00
(input)/(if dispense item)(if ready for change)
Wait for product "Hello"
S0
Confirm=1/00
Display Change
S6
Confirm=0/00
total_money_inserted >= selected_product_price =0/01
OR
total_money_inserted >= selected_product_price =0/01
Select_product = 1/00
Confirm=1/11
Selected product
S1
Dispense item
S5
Confirm=0/10
Confirm=0/00
Confirm=1/00
total_money_inserted >= selected_product_price =1/10
OR
total_money_inserted >= selected_product_price =1/10
Check availability
S2
Insert money
S3
Check inserted money
S4
Confirm=1/00
Confirm=1/00
Confirm=0/00
Confirm=0/00
OR
total_money_inserted >= selected_product_price =1/10

Figure 1. State Diagram

## Schematic

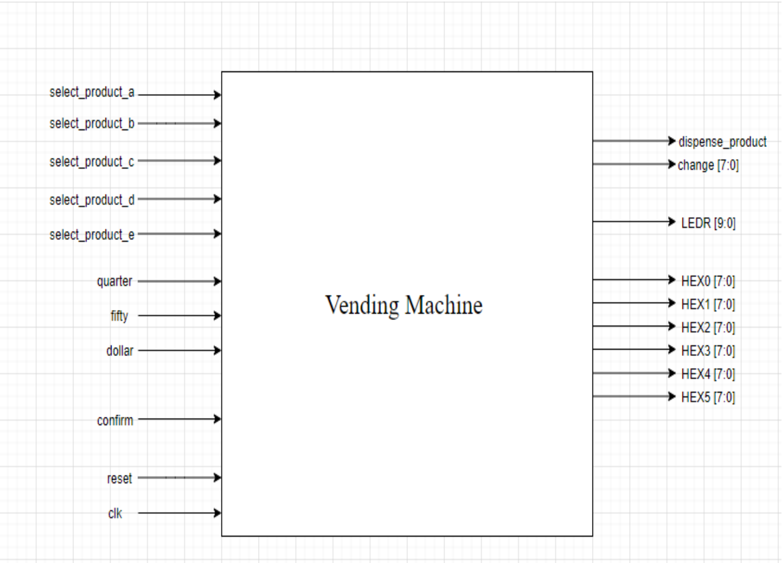


Figure 2. Vending Machine Block Diagram

## Simulation Results

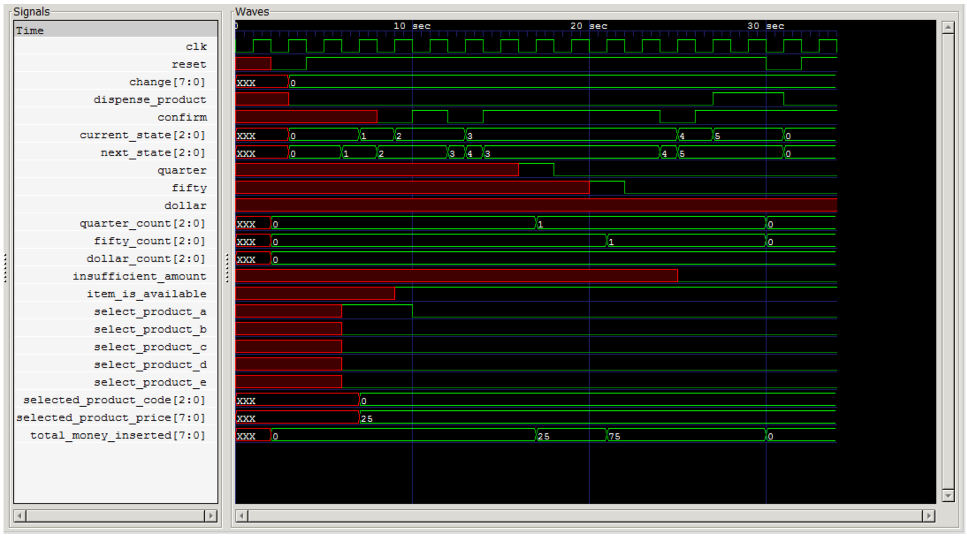
This is the output for the first test case where the user chooses prod1 and inputs the exact amount required (25 cents). The waveforms depict the response of the module to inputs. The following output is from the simulation results using product_a:

```
 ======================Start
simulation================
VCD info: dumpfile Case1.vcd opened for output.
 RESETTING
 quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = x change =   x
 RESETTING
S0 : 7 segment says Hello
S0 : 7 segment says Hello
 quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0

============================Selecting_product=======
==============
S0 : 7 segment says Hello
 Product has been selected
 quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
S1 : 7 segment says 25
 Product price is  25
quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
Confirm button has been pressed!!!
 Product price is  25
quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
S2 : 7 segment says error
S2 : 7 segment says yes
 quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
 quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
S2 : 7 segment says yes
 quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
 Item is available!
 quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
S3 : 7 segment says 000
 quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
 quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
+++++++++++++++++++++PUTTING IN MONEY
NOW+++++++++++++++++++++++
```

```
 Quarter count updated
S3 : 7 segment says 25
 quarter_count = 1 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
++++++++++++++++++++++++++MONEY ADDED
+++++++++++++++++++++++++++
 quarter_count = 1 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
Inserted amount is sufficient!
 quarter_count = 1 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
Inserted amount is sufficient!
 quarter_count = 1 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
Debug: State 6
CONFIRM RECIEVING ITEM
S5 : 7 segment says 0ut=1
 quarter_count = 1 fifty_count = 0 dollar_count = 0
 dispense_product = 1 change =   0
+++++++++++++++++++++CONFIRM BUTTON
PRESSED+++++++++++++++++++++++
Debug: State 6
 quarter_count = 1 fifty_count = 0 dollar_count = 0
 dispense_product = 1 change =   0
S6 : 7 segment says CH 000
Debug: State 7
 THANK YOU
 quarter_count = 1 fifty_count = 0 dollar_count = 0
 dispense_product = 1 change =   0
+++++++++++++++++++++CONFIRM BUTTON
PRESSED+++++++++++++++++++++++
Debug: State 7
 THANK YOU
 quarter_count = 1 fifty_count = 0 dollar_count = 0
 dispense_product = 1 change =   0
S0 : 7 segment says Hello
S0 : 7 segment says Hello
 quarter_count = 1 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
+++++++++++++++++++++CONFIRM BUTTON
PRESSED+++++++++++++++++++++++
===================================Done
simulating==============================
 quarter_count = 1 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
```

## Waveform



## Conclusion

In conclusion, a 7 state vending machine has been designed using only verilog code, and the DE10-Lite FPGA. A description, state machine, schematic, and resulting outputs have been given, to prove the design of the working vending machine. The project was written over around 800 lines of code. However, a different attempt was made as well, where it was around 1800 lines of code. The secondary attempt was much more detailed and designed to make sure it was rock solid code with animations one the seven segment display.

## Reference

De10-Lite Manual (attached)

However, due to register issues, it is though that the lite version was not able to modify itself after the code, and was therefore scrapped to be made much simpler. Overall, the project was a very rich in new information, such as FPGAs and Quartus. It has taught the group a lot and we really enjoyed the ride. But sad we couldn't use the big code.

## Future Work

For future work, the group believes that there should be more team effort in distributing the work correctly with more communication.

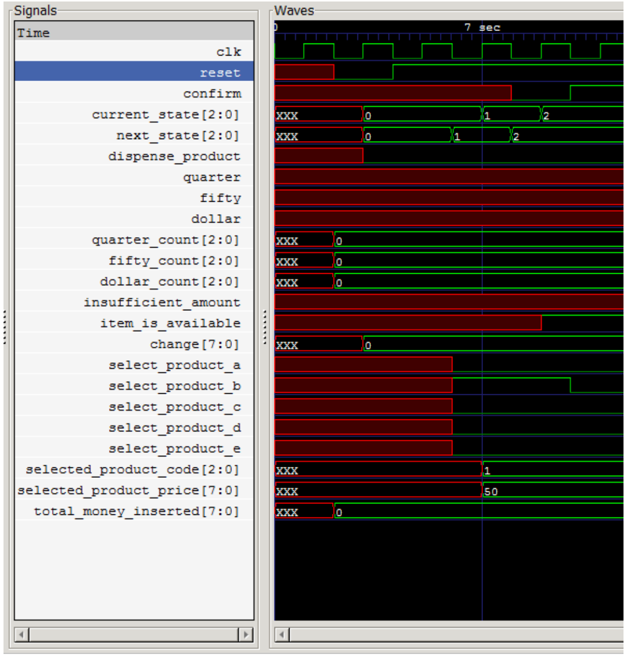
## Contributions

| Name | Contributions |
|---|---|
| Johan | Architecture, Report, Tasks |
| Talal | Architecture, design, implementation |
| Naures | Architecture, Test bench, implementations |

TEST CASE 2:

TEST CASE 2: Now we are testing the case where user overpays for the product and change is expected. Looking at the wavefroms we can see that quarter was inserted, while the selected product is product b (select_product_b is high and product b price is 50). As we can see, the change of 50-25=25 cents is being given in state 6.

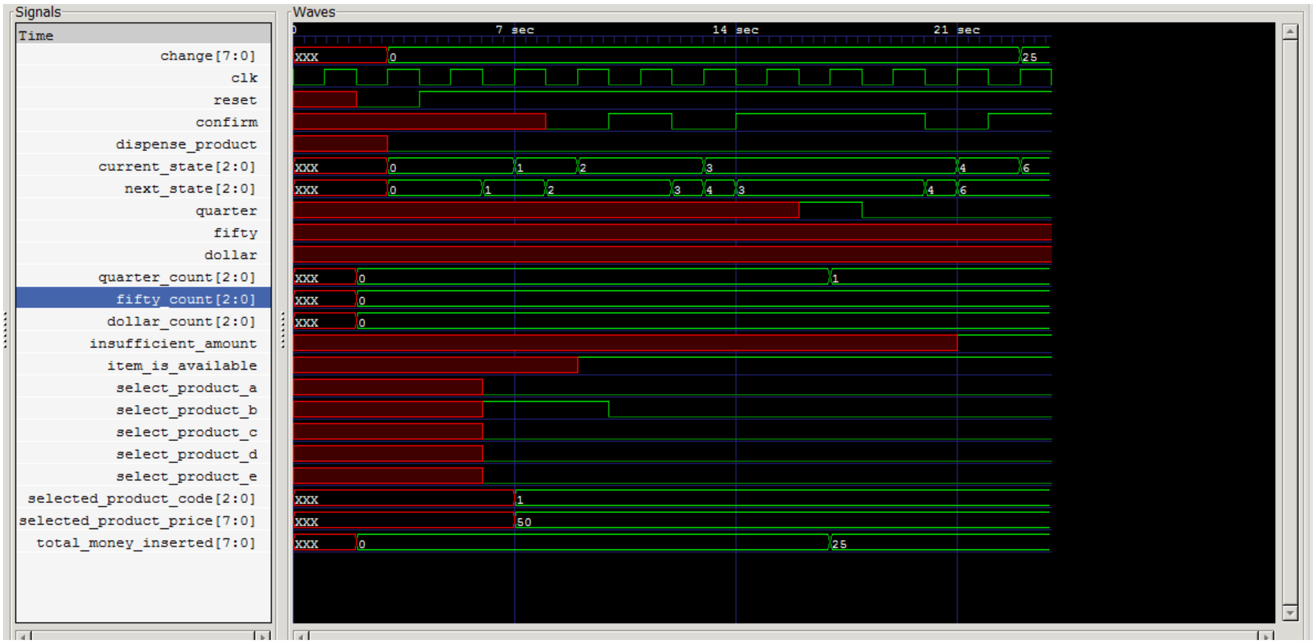
```
 ======================Start
simulation================
VCD info: dumpfile Case2.vcd opened for output.
 quarter_count = x fifty_count = x dollar_count = x
 dispense_product = x change =   x
 RESETTING
 quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = x change =   x
 RESETTING
S0 : 7 segment says Hello
S0 : 7 segment says Hello
 quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
==============================Selecting_product=
=====================
S0 : 7 segment says Hello
 Product has been selected
 quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
S1 : 7 segment says 25
 Product price is  25
 quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
Confirm button has been pressed!!!
 Product price is  25
 quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
S2 : 7 segment says error
S2 : 7 segment says yes
 quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
S2 : 7 segment says yes
 quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
 Item is available!
 quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
S3 : 7 segment says 000
 quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
++++++++++++++++++++PUTTING IN MONEY
NOW+++++++++++++++++++++++
 quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
+++++++++++++++++++++++++MONEY ADDED
quarter ++++++++++++++++++++++++++
 Quarter count updated
S3 : 7 segment says 25
 quarter_count = 1 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
+++++++++++++++++++++++++MONEY ADDED fifty
++++++++++++++++++++++++++
 Fifty count updated
S3 : 7 segment says 75
 quarter_count = 1 fifty_count = 1 dollar_count = 0
 dispense_product = 0 change =   0
++++++++++++++++++++CONFIRM BUTTON
PRESSED++++++++++++++++++++++
 quarter_count = 1 fifty_count = 1 dollar_count = 0
 dispense_product = 0 change =   0
Inserted amount is sufficient!
 quarter_count = 1 fifty_count = 1 dollar_count = 0
 dispense_product = 0 change =   0
Inserted amount is sufficient!
 quarter_count = 1 fifty_count = 1 dollar_count = 0
 dispense_product = 0 change =   0
Debug: State 6
CONFIRM RECIEVING ITEM
S5 : 7 segment says 0ut=1
 quarter_count = 1 fifty_count = 1 dollar_count = 0
 dispense_product = 1 change =   0
 RESETTING
Debug: State 6
CONFIRM RECIEVING ITEM
S5 : 7 segment says 0ut=1
 quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = 1 change =   0
 RESETTING
S0 : 7 segment says Hello
S0 : 7 segment says Hello
 quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
==================================Done
simulating==============================
```


Signals
Waves
Time
0
7 sec
clk
reset
confirm
current_state[2:0]
next_state[2:0]
dispense_product
quarter
fifty
dollar
quarter_count[2:0]
fifty_count[2:0]
dollar_count[2:0]
insufficient_amount
item_is_available
change[7:0]
select_product_a
select_product_b
select_product_c
select_product_d
select_product_e
selected_product_code[2:0]
selected_product_price[7:0]
total_money_inserted[7:0]
xxx
0
1
2
50

TEST CASE 3

This time we are testing the case when user tries to press the confirm button without paying the required amount of money for the chosen product.In the waveforms this can be seen as quarter is being given, while the chosen product is product b, which costs 50 cents. This time, insufficient_amount flag is set and the money is given back as a change after the user presses confirm button while underpaying.

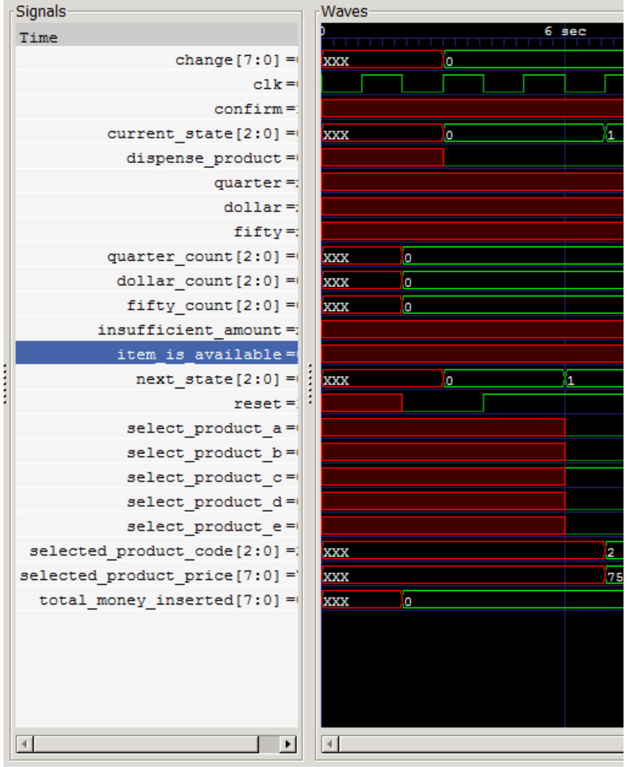
```
 =====================Start
simulation================
VCD info: dumpfile Case3.vcd opened for output.
 quarter_count = x fifty_count = x dollar_count = x
 dispense_product = x change =   x
 RESETTING
 quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = x change =   x
 RESETTING
S0 : 7 segment says Hello
S0 : 7 segment says Hello
 quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
===========================Selecting_product=
===================
S0 : 7 segment says Hello
 Product has been selected
 quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
S1 : 7 segment says 50
 Product price is  50
 quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
Confirm button has been pressed!!!
 Product price is  50
 quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
S2 : 7 segment says error
S2 : 7 segment says yes
 quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
S2 : 7 segment says yes
 quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
 Item is available!
 quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
S3 : 7 segment says 000
 quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
++++++++++++++++++++Trying to get the product
underpaying++++++++++++++++++++++
++++++++++++++++++++++PUTTING IN MONEY
NOW++++++++++++++++++++++++
 quarter_count = 0 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
+++++++++++++++++++++++++++MONEY ADDED
quarter ++++++++++++++++++++++++++++
 Quarter count updated
S3 : 7 segment says 25
 quarter_count = 1 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
++++++++++++++++++++++CONFIRM BUTTON
PRESSED++++++++++++++++++++++++
 quarter_count = 1 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
S3 : 7 segment says Error
 quarter_count = 1 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
 quarter_count = 1 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =   0
S6 : 7 segment says CH 000
Debug: State 7
SORRY, YOUR MONEY BACK IS  25
S6 : 7 segment says CH 25
 quarter_count = 1 fifty_count = 0 dollar_count = 0
 dispense_product = 0 change =  25
==================================Done
simulating==============================
```


Signals
Waves
Time
0
7 sec
14 sec
21 sec
change[7:0]
clk
reset
confirm
dispense_product
current_state[2:0]
next_state[2:0]
quarter
fifty
dollar
quarter_count[2:0]
fifty_count[2:0]
dollar_count[2:0]
insufficient_amount
item_is_available
select_product_a
select_product_b
select_product_c
select_product_d
select_product_e
selected_product_code[2:0]
selected_product_price[7:0]
total_money_inserted[7:0]
XXX
0
25
1
2
3
4
6
50

TEST CASE 4

Now, we check the case when user tries to buy a product that is out of stock. In this case for testing purposes product C is unavailable and when the user tries to press confirm, he is not allowed to proceed further and is being sent back to the beginning with an error message.

```
 ======================Start
simulation================
VCD info: dumpfile Case4.vcd opened for
output.
 quarter_count = x fifty_count = x dollar_count
= x
 dispense_product = x change =   x
 RESETTING
 quarter_count = 0 fifty_count = 0 dollar_count
= 0
 dispense_product = x change =   x
 RESETTING
S0 : 7 segment says Hello
S0 : 7 segment says Hello
 quarter_count = 0 fifty_count = 0 dollar_count
= 0
 dispense_product = 0 change =   0
============================Selecting
_product=====================
S0 : 7 segment says Hello
 Product has been selected
 quarter_count = 0 fifty_count = 0 dollar_count
= 0
 dispense_product = 0 change =   0
S1 : 7 segment says 75
 Product price is  75
 quarter_count = 0 fifty_count = 0 dollar_count
= 0
 dispense_product = 0 change =   0
+++++++++++++++++++CONFIRM BUTTON
PRESSED+++++++++++++++++++++
 Product price is  75
 quarter_count = 0 fifty_count = 0 dollar_count
= 0
 dispense_product = 0 change =   0
S2 : 7 segment says error
S2 : 7 segment says error
 quarter_count = 0 fifty_count = 0 dollar_count
= 0
 dispense_product = 0 change =   0
S2 : 7 segment says error
 quarter_count = 0 fifty_count = 0 dollar_count
= 0
 dispense_product = 0 change =   0
S0 : 7 segment says Hello
 quarter_count = 0 fifty_count = 0 dollar_count
= 0
 dispense_product = 0 change =   0
+++++++++++++++++++CONFIRM BUTTON
PRESSED+++++++++++++++++++++
 quarter_count = 0 fifty_count = 0 dollar_count
= 0
 dispense_product = 0 change =   0
 quarter_count = 0 fifty_count = 0 dollar_count
= 0
 dispense_product = 0 change =   0
+++++++++++++++++++CONFIRM BUTTON
PRESSED+++++++++++++++++++++
 quarter_count = 0 fifty_count = 0 dollar_count
= 0
 dispense_product = 0 change =   0
 quarter_count = 0 fifty_count = 0 dollar_count
= 0
 dispense_product = 0 change =   0
================================Do
ne
simulating===========================
=
```


Signals
Waves
Time
0
6 sec
change[7:0] =
clk =
confirm =
current_state[2:0] =
dispense_product =
quarter =
dollar =
fifty =
quarter_count[2:0] =
dollar_count[2:0] =
fifty_count[2:0] =
insufficient_amount =
item_is_available =
next_state[2:0] =
reset =
select_product_a =
select_product_b =
select_product_c =
select_product_d =
select_product_e =
selected_product_code[2:0] =
selected_product_price[7:0] =
total_money_inserted[7:0] =
xxx
0
1
2
75