

Vending Machine

Khodur Dbouk, Basil Jajou, Kouder Abbas, Stevan Nissan
Electrical and Computer Engineering Department
School of Engineering and Computer Science
Oakland University, Rochester, MI

kdbouk@oakland.edu, basiljajou@oakland.edu, kouderabbas@oakland.edu, nissan@oakland.edu

Abstract—This report is about small vending machine to dispatch the beverages which will be dispatched only if the user will insert 25c in any possible combination because each beverage in the machine costs \$0.25. The vending machine only takes input as quarters (25c), dimes (10c), and nickels (5c) and it will only return quarters, dimes, and nickels. Each input (25c, 10c, 5c) is a one-clock cycle wide pulse that is synchronous with the clock signal. Inputs will occur “one at a time”. You will not ever have two or more inputs asserted on the same clock cycle. To dispense change, assert each output (return_25c, return_10c, and return_5c) for one clock cycle. The supply of coins for change and beverages is unlimited. The vend signal should only be asserted for one clock cycle when the user has inserted at least \$0.25. This project is done by using VHDL and have been implemented by using FPGA board where switches are used as input money in binary format and LEDs and seven segments are used to show remaining amount/change and dispatched result.

Keywords— FSM, Vending Machine, FPGA, VHDL

I. INTRODUCTION

Vending machine is an automatic machine which provides soft drink, snacks etc. to a user when there is inserted money into the machine. There are also available modern vending machines which are dealing with more products and have the flexibility

to use a credit card as input instead of coins. In our project, we have implemented a vending machine for a soft drink. Where inputs are coins of 25c, 10c or 5c and outputs are a soft drink and remaining amount/change. Before going to furthermore we should know that what finite state machine is?

A. Finite State Machine

Finite state machine (FSM) is actually a mathematical model of computation, this machine can be in one of the states from the total possible states. The present state can be changed according to input from the outside. An FSM can be defined by its states list, initial state and the condition for each state transition. There are two types of FSM

1. Mealy Machine
2. Moore Machine

These both machines have its own pros and cons, which are following:

B. Mealy Machine

In the mealy machine, the output of machine depends upon both the input and present state. The output of the designed machine

will be dependent upon the change in input and present state, which definitely decrease the number of states in design. The block diagram of the mealy machine shown below in figure: 1.

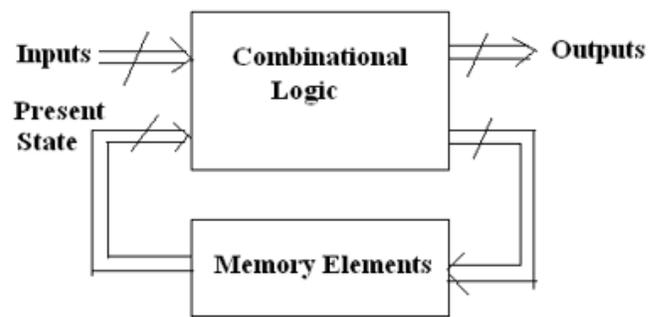


Figure 1: Mealy state machine

C. Moore Machine

In Moore machine, the output of machine depends upon only on the present state. The output of the designed machine will be dependent only upon the change in present state. The output of Moore machine is independent of the change in input, which definitely increases the number of states in design. The block diagram of Moore machine shown below.

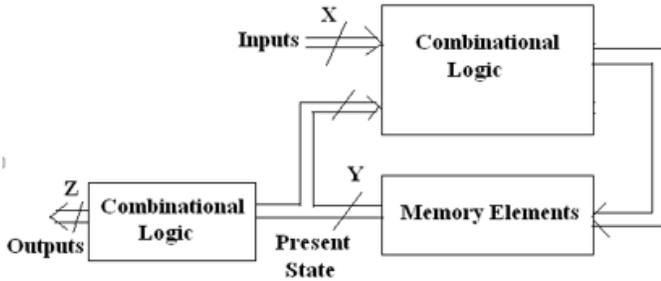


Figure 2: Moore state machine

The vending machines are also implemented through fsm and it can be implemented through moore and mealy machine. As discussed earlier in moore machine we need more states because the output only depends on the current state so we have implemented the vending machine through the mealy machine and it has nine states so we need to represent these states only 4 number of bits.

This report has 4 sections: I- was introduction.

II- will be dealing with methodology. III- will be dealing with results and IV- will be concluding the whole report.

II. METHODOLOGY

We will make sure to include all the states necessary to collect the coins and vend a beverage as soon as the total amount of change inserted is greater than or equal to \$0.25 and always dispense the proper amount of change. for example, if the user inserts 3 dimes, our machine will vend 1 drink and then dispense 1 nickel in change or If the user inserts 4 nickels and then a quarter, our machine will vend 1 drink and then dispense 2 dimes in change. The block diagram of the complete vending machine is given below

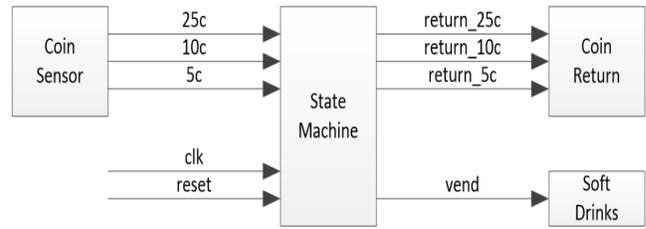


Figure 3: Block level diagram of vending machine for soft drink

To include all the possible combinations, we have to make a state diagram, as the input coin will only be 5, 10 and 25 so at every state check that which combination will be true. Based on that combination it will decide whether it goes to the next state or in the same state because if the input will be of the other than the mentioned, it does not change the state. The state diagram of this system is shown below in figure:4.

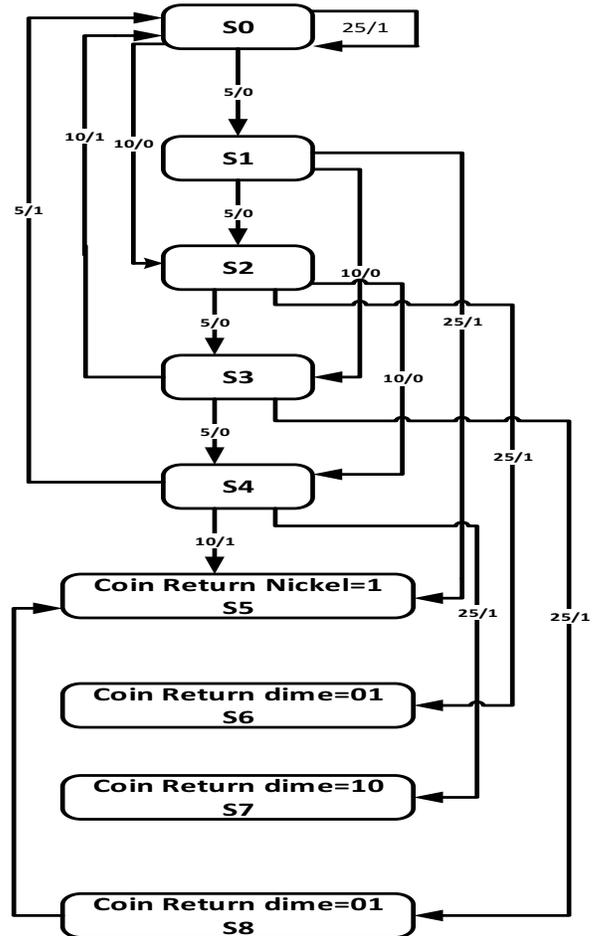


Figure 4: State diagram of vending machine

III. IMPLEMENTATION

Implementation of the vending machine was a great challenge for us because there were many things which were needed to be integrated; although

we have covered all these things in our lectures but there were still such things which were still unsettling us. To implement this machine, we have used state register and Non-clock process.

A. Synchronous Process for State Machine

This state machine is implemented through the process in VHDL and it triggers only whenever the rising edge of the clock occurs and that's why this process is called as synchronous process and it resets the whole vending machine when reset input will be high. At each rising edge of the clock present state of the machine will be changed to next state.

B. Asynchronous Process for Next State Logic

The state diagram mentioned in the figure has to be implemented in this process, in the first state the vending output (soft drink) will be zero and in the same way, coin returns output is also zero. Now when the state is S0 if the input coin will be 5 it will change the state to S1. If the input coin will be 25, it will remain in the same state but the vending output will go high because the user has put the exact amount to dispatch the soft drink, no change will be given to the user this time. If the user will give 10c as an input then it will go to the S2. This process will be the same in every state, it checks at every possible combination of the input and changes the state according to the flow chart. When the user enters the excess amount then the remaining amount will be given back to the user like if we have to return the 5c, then coin return nickel output will be high which shows the 5c coin.

C. Clock Divider Process

As we know that the internal clock is generated from a crystal oscillator which is about 100MHz. Switching at 100 MHz will is not possible to see for us and system will change its states in just Nano seconds so we need to divide this 100 MHz to 1 Hz of frequency to run the system on slower speed. In VHDL code there is made a process to divide this fast clock.

D. Switches, LEDs, and Seven-Segment Display

In order to maximize the usefulness of the circuit, is important to make its design user friendly. A series of five switches are used to input the desired amount of coins from the user. Switches 0-4 are used

to take an input from the user i.e. the 5c, 10c, 25c. The switches values are only loaded into the input coin when the rising edge of the clock occurs. The reset button is wired to all process and the non-clock process whenever its value is high it resets everything. Finally, LEDs are illuminated when the vending output will be high or whenever the change will be returned to the user, meanwhile the same will appear on the seven-segment also.

IV. EXPERIMENTAL SETUP

The software that was used to create this vending machine was Xilinx. The hardware that was used to implement the design was the Nexus 4 DDR board. The specific inputs and outputs of the board can be seen in the table-1, which includes: two seven-segment displays, a switch to reset time, five switches to input coin, and LEDs to notify the time vending output and the coin return change. One seven-segment display is used to display the vending output like when dispatched the drink it will display the "d" at seven segment. The second seven-segment will be used to display the change like for 5c it will display the "5" and for 10c it will display the "A" at seven segment. The design's functionality was tested by loading onto the Nexus board. RTL top level diagram is shown in figure:5.

Table 1: Input/output description

Inputs /Outputs	FPGA resources	Description
Coin_input(4 downto 0)	Coin_input[4]= SW4 Coin_input[3]= SW3 Coin_input[2]= SW2 Coin_input[1]= SW1 Coin_input[0]= SW0	This is coin input which is 5 bit and implemented using switches of the FPGA board. We have only three type of coin in our vending machine 5c, 10c and 25c so these 5, 10 and 25 will be inputs but in form of their binary equivalent. E.g. 5 will be inputted as 00101.
vending_drink	vending_drink=LED15	This LED show that a soft drink is been vended.
return_coin_	return_coin_nickel=LED14	This LED will display return a coin nickel

Enter button	BTNC N17	Change states and add inputs
sevensegment_display_out(6 downto 0)	<p>sevensegment_display_out[6] = CG sevensegment_display_out[5] = CF sevensegment_display_out[4] = CE sevensegment_display_out[3] = CD sevensegment_display_out[2] = CC sevensegment_display_out[1] = CB sevensegment_display_out[0] = CA</p> <p>This is used to on-off the corresponding seven segments from the array of 8 seven segments. These seven segment are common anode.</p>	<p>There are total 8 seven segments in our FPGA we are using only one for to show dispatching and another one to show remaining amount/change. The output of the machine is a soft drink which will be dispatched when input money $\geq 25c$ and this output is displayed in our case on a seven segment of the FPGA. The output will be 'd' on the seven-segment display which will indicate that soft drink is been dispatched.</p>
return_coin_dime(1 downto 0)	<p>return_coin_dime[1]=LED13 return_coin_dime[0]=LED12</p>	<p>These are again the output which will show that how many coin dime of 10c should be returned to make the proper change, for example, total input of the system was 45c then the output will be 2 dimes of 10c coins to return 20c back.</p>
RESET	RESET= SW15	<p>This is reset button which will reset the vending machine to its initial state to accept a new when its value is one and will have no effect when it will be 0.</p>
CLK	CLK=E3 which is connected to 100MHZ crystal oscillator	<p>This will provide a clock signal to run the whole system properly.</p>

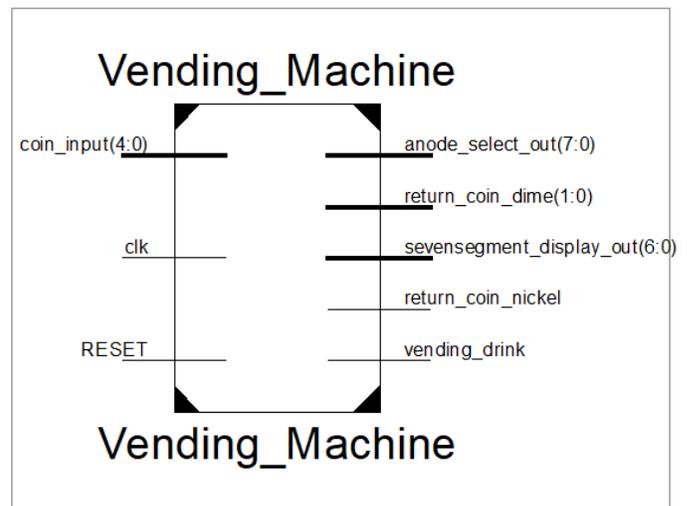


Figure 5: I/O detail of vending machine

Pin detail and hardware resources used of FPGA are also shown in below figure:6.

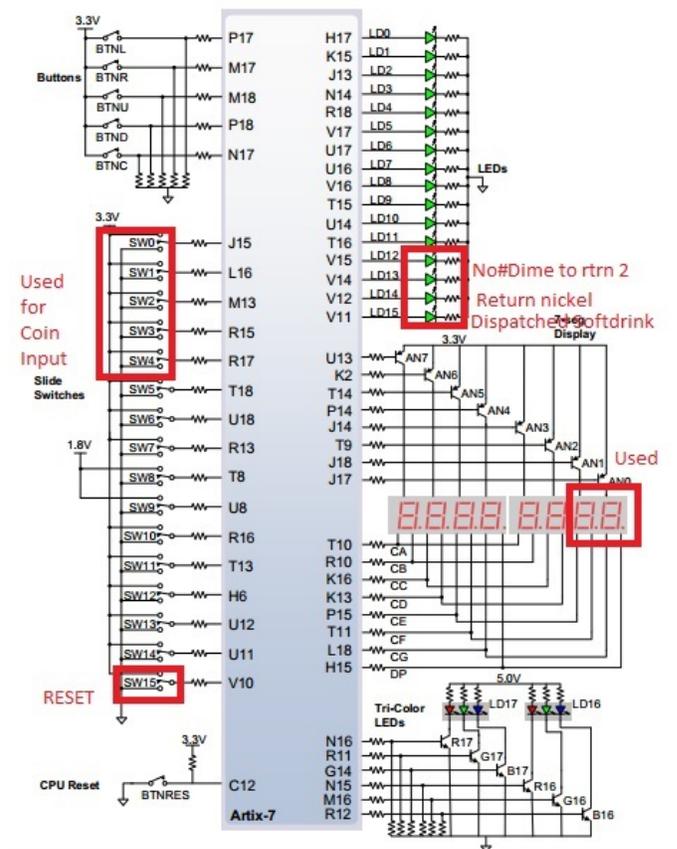


Figure 6: Pin detail and used resources of FPGA

V. RESULTS

The design successfully dispatches the drink as an output and change of the amount to the user. The design notifies the amount through led and seven-segment display and resets when the reset button is pressed. All inputs and outputs function as desired.

Attached below is a screen shot of the different combinations of the input coin.

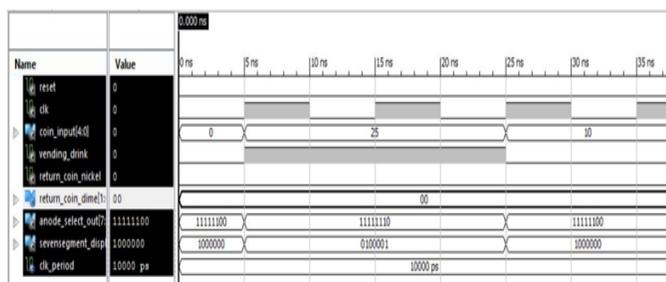


Figure 7: Simulation result when user input is 25c exactly

In figure:7 as the input coin will be at the start is zero and then when the first rising edge of the clock occurs its value is also updated and that is 25c, the other side the vending output will also be one at the same time because the user inserts right amount at the first attempt. The coin return outputs are zero because no change has to be returned for this amount. The output also showed on the seven-segment, “d” will be displayed on the first seven-segment.

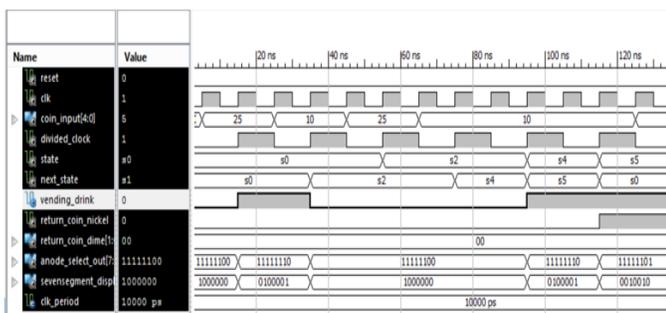


Figure 8: When user input 30c

In figure:8 when the user inputs three times ten then again, the output will be high but this time one can see that the vending machine will return the change also like the coin return nickel output is high.

VI. CONCLUSION

With FSM state machine the task of creating a vending machine was achieved. This was no simple feat, as to include all the combinations within minimum states, proved quite the challenge. Improvements to the design that could be made include the coin sensor used in the input side which sense the right amount and the output will also be the drink and return of the amount will be the coins.

VII. REFERENCES

- [1] Daniel Llamocca. “VHDL Coding for FPGAs”. <http://www.secs.oakland.edu/~llamocca/VHDLforFPGAs.html>
- [2] Chu, Pong P. "FPGA PROTOTYPING BY VHDL EXAMPLES." (n.d.): n. pag. Ebooksclub . JOHN WILEY & SONS, INC., 2008. Web. 27 Nov 2017
- [3] “Reference Manual,” Nexys 4 DDR Resource Center, 23 Nov - 2017. [Online]. Available at: <https://reference.digilentinc.com/reference/programmable-logic/nexys-4-ddr/start?redirect=1>