CS M152A Lab 3

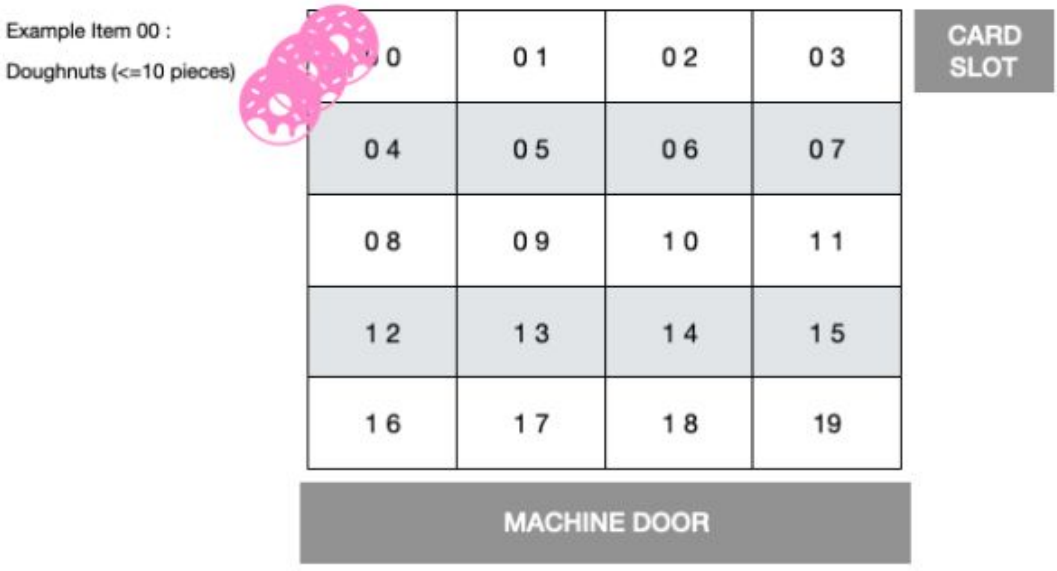
Ryan Riahi

Introduction

The purpose of this laboratory was to design a finite state machine (FSM) that simulates a vending machine. A FSM is a mathematical model of computation of an abstract machine that can be in one of a certain number of finite states at a time. FSMs are a very useful and powerful model of computation that are used in many real world scenarios. Their design allows for easy implementation into real world circuit and logic design. In this specific case, I designed an FSM to simulate the behavior of a vending machine that only accepts card payement and can vend any 1 of 20 items at a time. I then programmed that behavior using the Xilinx ISE software.

Design Description

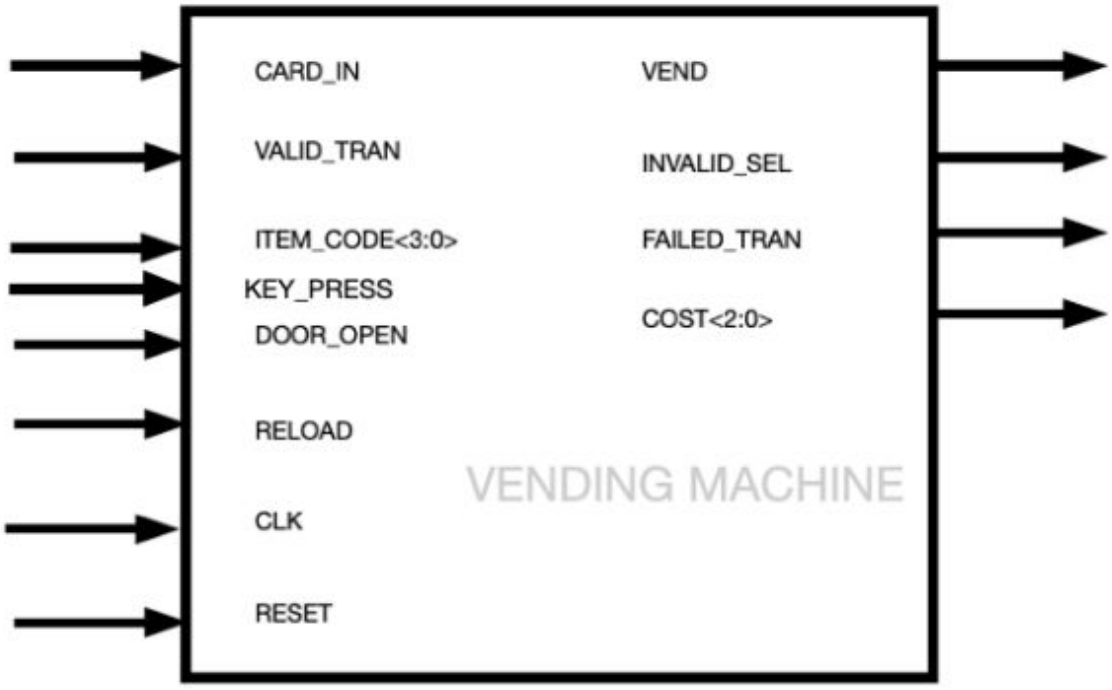
The vending machine itself has 20 items (each with at most 10 pieces), a card slot for credit/debit cards, and a door that is used to grab the purchased item



The item costs of the machine correspond to the following table:

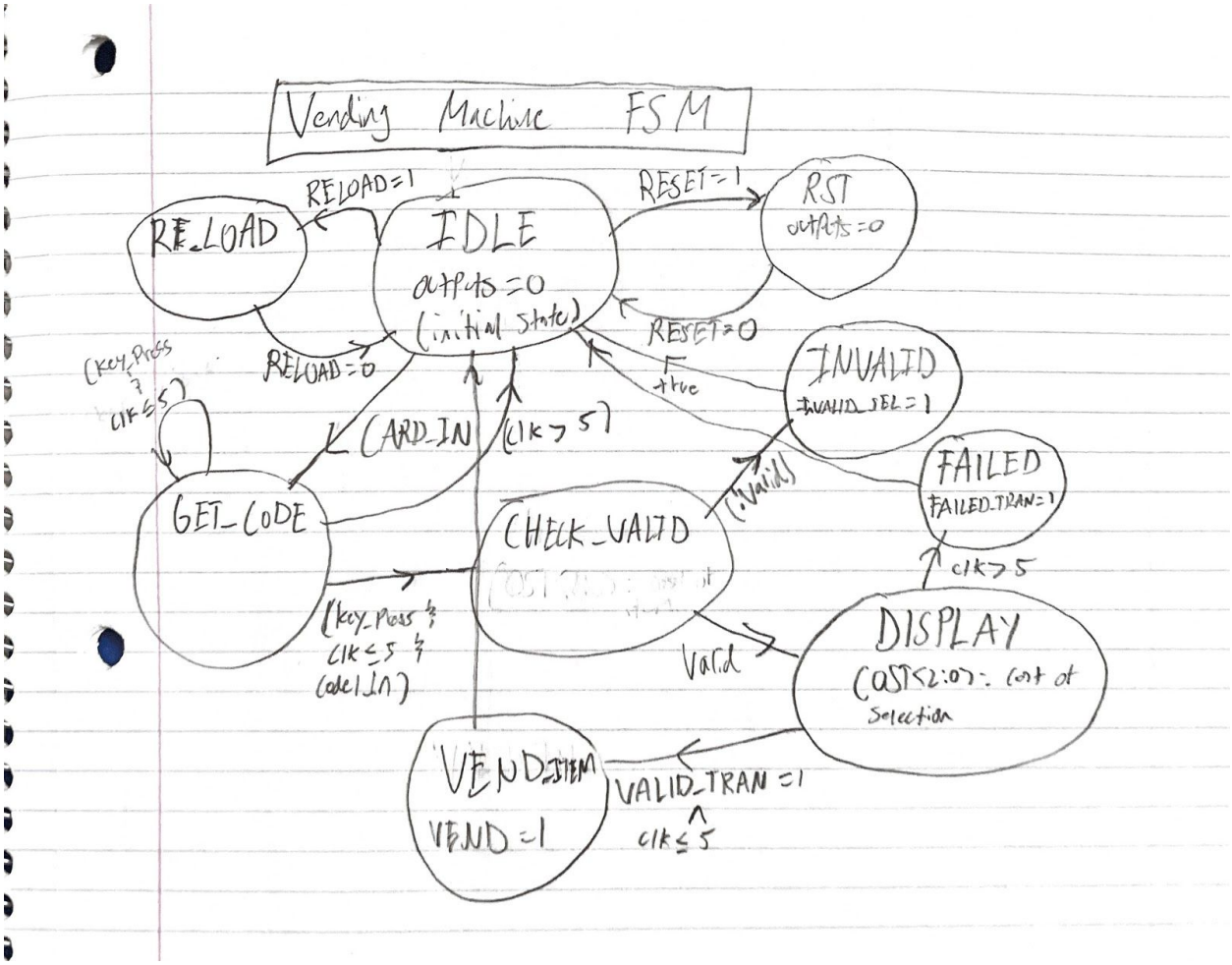
ITEM CODE	COST ($)	ITEM CODE	COST ($)
00, 01,02,03	1	12,13,14,15	4
04,05,06,07	2	16,17	5
08,09,10,11	3	18,19	6

The I/O of the machine is as follows:



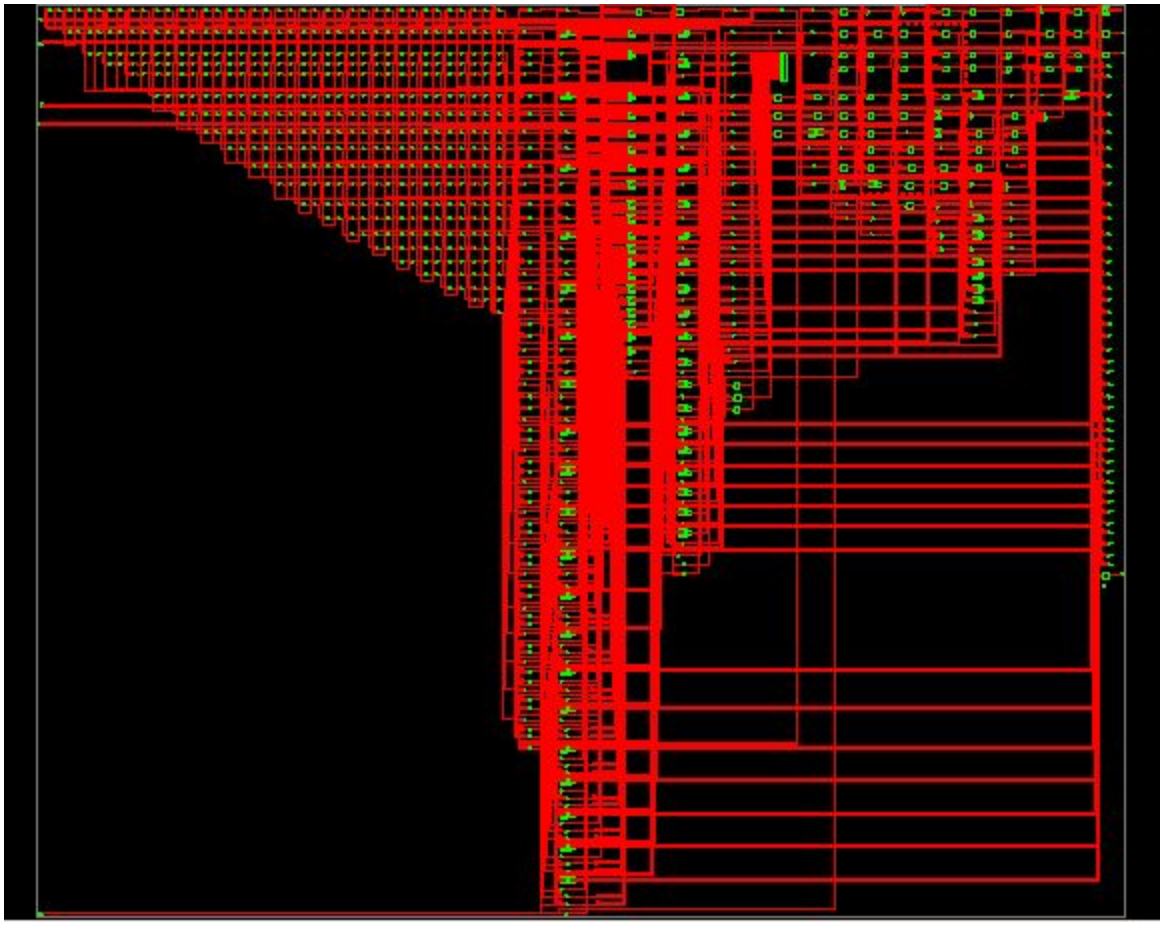
Depending on which inputs go high, the machine knows that certain events have transpired. For example, if CARD_IN goes high, then we know a card was inputted. And the machine will set certain outputs high for other events. It will set COST equal to the cost of the requested item.

Based off of these specifications, I have designed the following FSM for the vending machine:



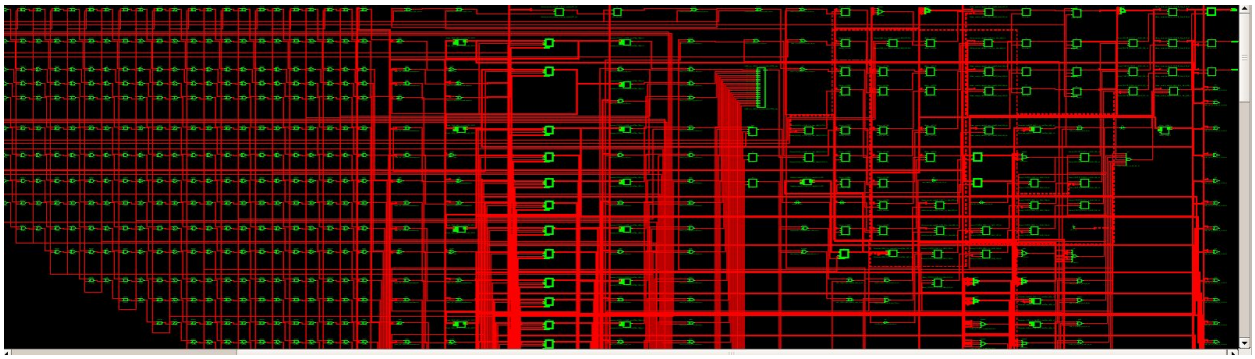
The FSM has 9 states:

- It starts in the IDLE state, this is the default state. If RELOAD or RESET is set high, it will go to the RE_LOAD or RST states respectively. If CARD_IN is set to high, then the FSM will go to the GET_CODE state.
- In the RESET/ RELOAD state, the machine either sets all item stocks to 0 or reloads them back to 10, then goes back to the IDLE state
- In the GET_CODE state, the machine for 2 key_press values to go high and read in two different digits for the item code. If this takes too long, the FSM will go back to the IDLE state. Otherwise, it will go the CHECK_VALID.
- In CHECK_VALID, the FSM will see if the inputted item code is a valid one that can be dispensed, if it is not it will go to the INVALID state, set INVALID_SEL to 1, and the go to the IDLE state. Otherwise, it will go to the DISPLAY state.
- In the DISPLAY state, the cost of the item is displayed and the FSM waits for the VALID_TRAN signal to go high. If it does not, the FSM will go the the FAILED state, set FAILED_TRAN to 1, and then return to IDLE. Otherwise it will go to the VEND_ITEM state.
- In the VEND_ITEM state, the item is vended, VEND is set to 1, and then the machine returns back to its original IDLE state.



This was the schematic that came out for my vending machine. There are lots of mutexes, registers, and counters in this, which makes sense given how much the FSM has to keep track of.

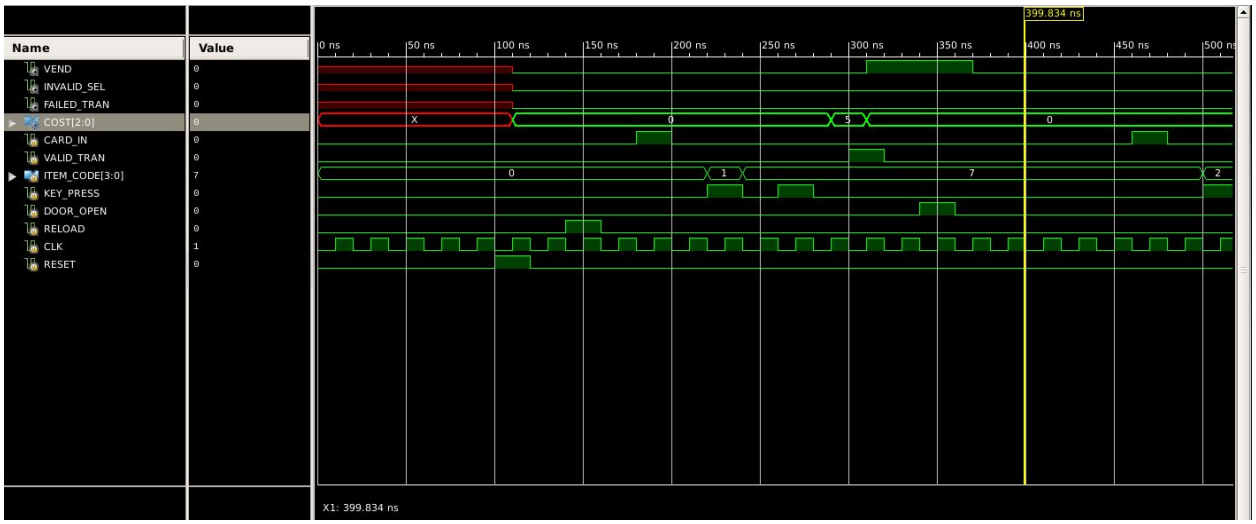
A closer look:



Simulation Documentation

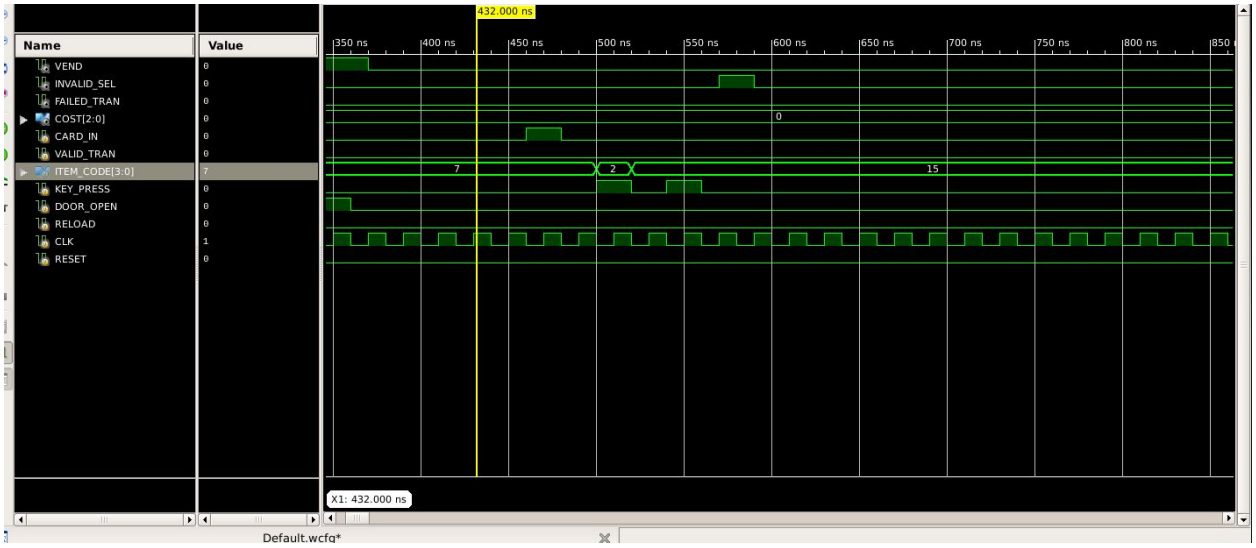
In order to ensure that my vending machine was working properly, I had to test it on all sorts of test cases to ensure that it works in all situations.

Succesful vending of item 17:



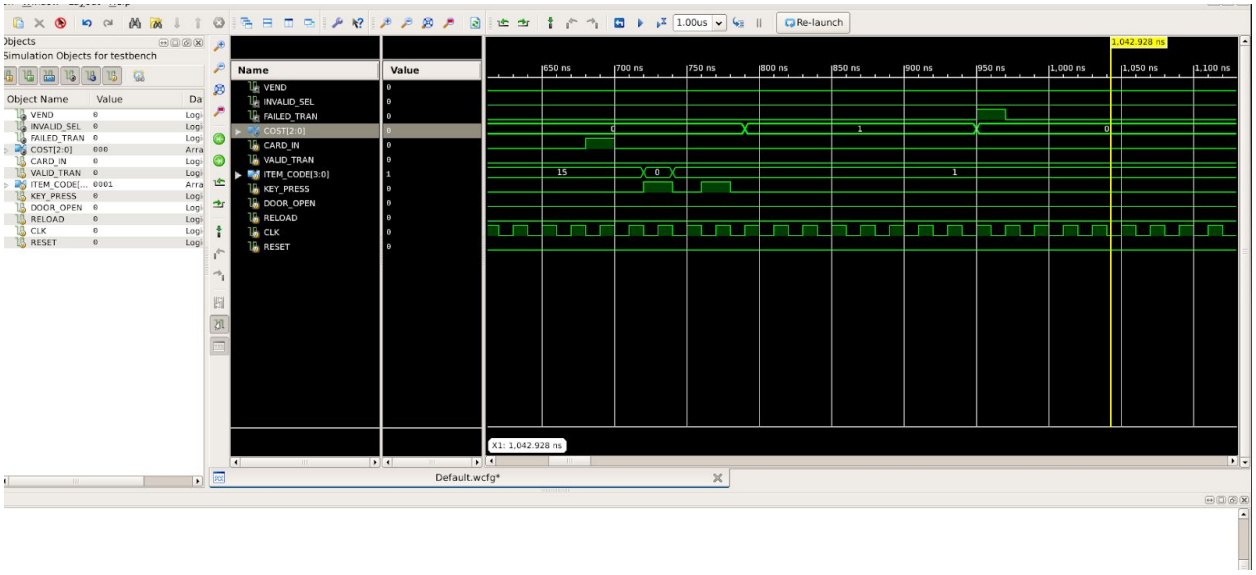
As can be seen in the simulation, everything works smoothly, the cost of 5 is displayed, and the item is successfully vended

Invalid selection of item “215”:



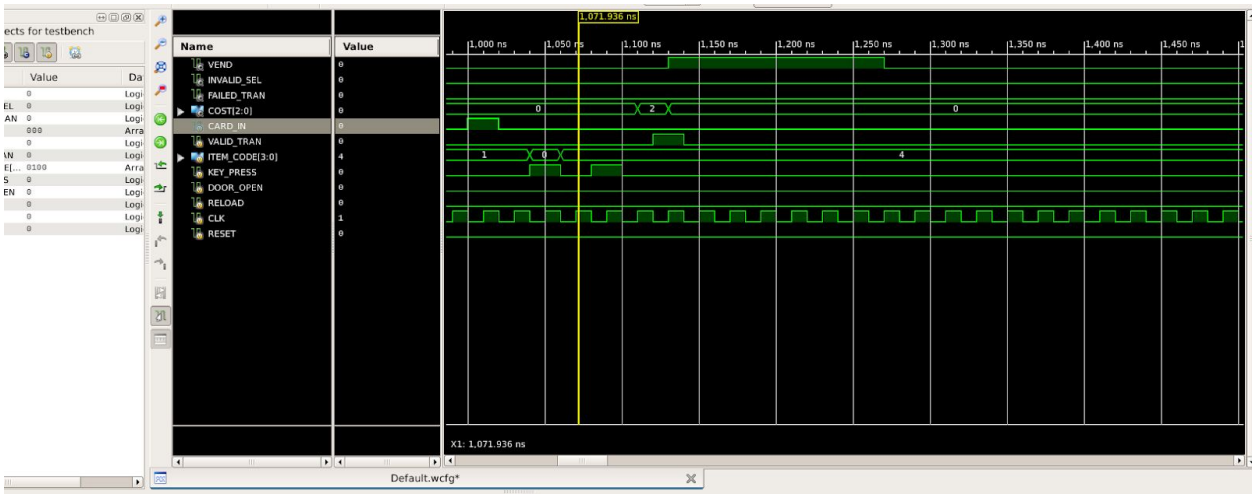
In this simulation, we can see that the code ‘2’ and then ‘15’ are inputted, and as a result the INVALID_SEL bit is set to high.

Credit card not accepted:



In this simulation we can see that everything works smoothly until the credit card is never accepted (i.e. the VALID_TRAN signal never goes high) and thus, after 5 clock cycles of waiting, the machine sets FAILED_TRAN to high.

Successful vending of item 4, except door does not open:



As can be seen, item 4 is put it and the cost value of 2 is properly displayed and then VEND goes high. The door is never opened though, which is not a problem because after 5 clock cycles VEND goes low and the machine returns to IDLE state.

Conclusion

In conclusion, this lab was very informative and interesting. I designed and implemented a Moore FSM for a vending machine. Overall I felt the process was much more straightforward than a lot of the other projects. It was easy for me to write the code

for the vending machine once I had designed the FSM. I am happy to have completed the project and have it working properly!