CS M152A Project 4 Report

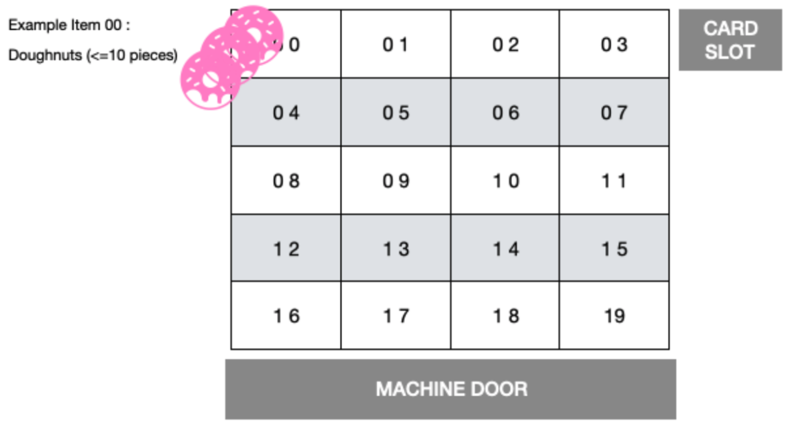
Melody Chen

May 25, 2020

1 Introduction and Requirement

The focus of this lab is for students to learn about how to design finite state machine that matches the specified behavior and to use the Xilinx ISE software to design and test state machines. Finite State Machines (FSMs) are used in many real world systems. An FSM has a finite number of states and can be in one state at a given time. There are two types of FSM machines: Moore Machine and Mealy Machine. A Moore machine's output only depends on which state it is in, but a Mealy machine's output depends on both the current state and input. For this assignment, we're tasked with designing a vending machine with the following characteristics:

1. Vending machine has 20 different snacks for sale. Each snack has two digit code (00 to 19).
2. Each snack is stored in separate slot. There can be up to 10 units of snack stored in 1 slot.
3. A buyer can purchase only 1 item at a time.
4. The machine only accepts payment by card.

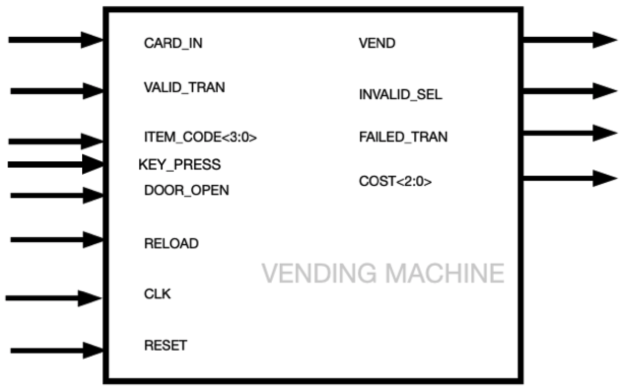


Vending Machine Figure from Manuscript

More details about specific behaviors of our vending machine FSM will be covered in the next sections.

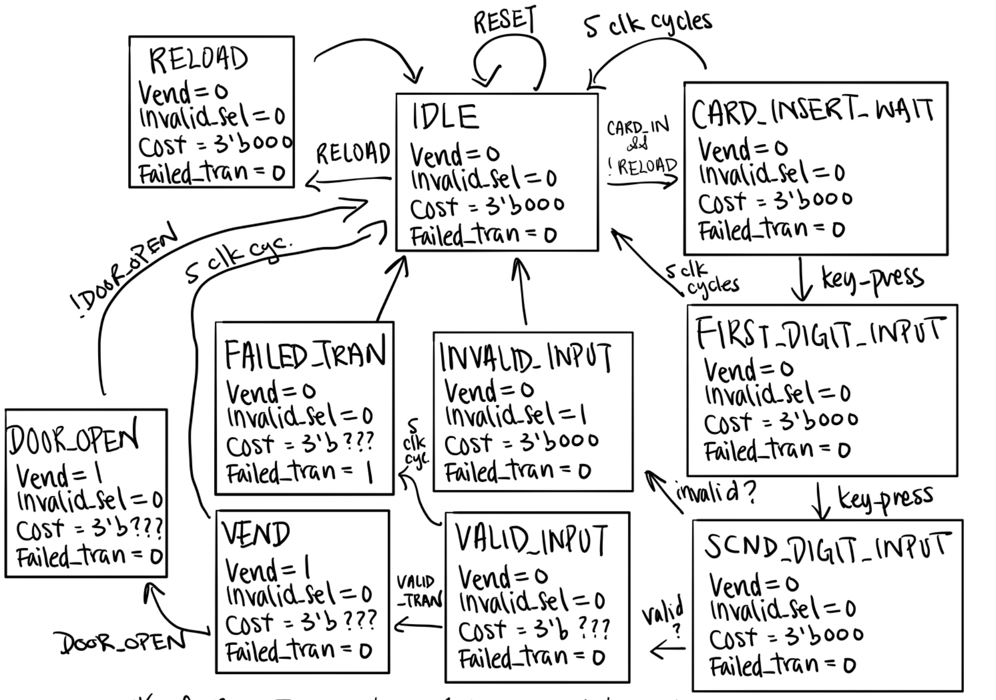
2 Design Description

For the design and implementation of the varying types of FSM, I followed guidance provided in the project specifications. I first consider the inputs provided and what my FSM will need to output. Our Vending Machine FSM will be given the following inputs, and we will need to output specific values for the outputs shown in the diagram below:



Vending Machine I/O from Manuscript

I then designed my FSM with around 10 states that correctly captures the behavior described in the manuscript. My FSM is a Moore machine, meaning that my output depends only on which state I am in, regardless of the inputs.



* if RESET signal is 1 in any state, return to IDLE state. All Counters set to 0. All signal (arrow) above includes "&& !RESET", except RESET arrow, not drawn to avoid repetition.

Moore FSM Design of Vending Machine

Descriptions of states and transitions of above FSM:

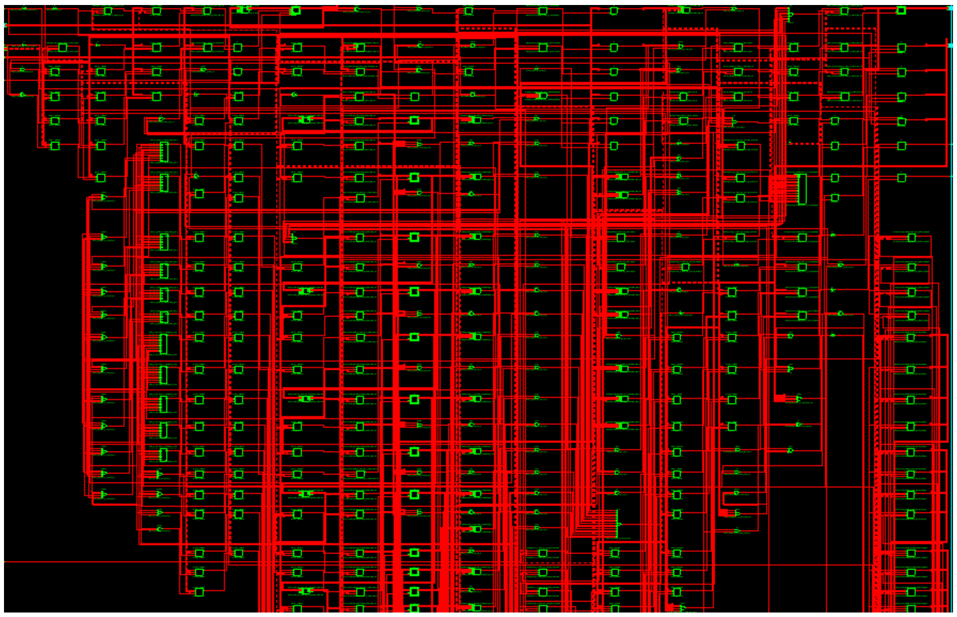
- Once `RESET` signal is 1, we will begin at the IDLE state, where our Vending Machine waits for the `RELOAD` or `CARD_IN` signal to be 1. All outputs should be 0, including count of items inside vending machine. This represents the state where no one is using our vending machine yet.
- Once `RELOAD` signal is 1, we transition to the RELOAD state where all our outputs remain unchanged, but the internal count for each slot in the vending machine should be set to 0. Once counts are set to 0, we directly return to IDLE state.

- From the IDLE state, when `RELOAD` signal is 0, and `CARD_IN` signal is 1, we transition into the CARD_INSERT_WAIT state. Outputs remain all zeroes.
- In CARD_INSERT_WAIT state, we wait for either 5 clock cycles to pass and transition back to the IDLE state or `KEY_PRESS = 1` and transition to FIRST_DIGIT_INPUT state. After 5 cycles has passed, this represents our vending machine not receiving any code entered. When we detect a key press, we want to transition to next state where we prepare to read in a second key press.
- In FIRST_DIGIT_INPUT state, all our outputs remain 0. We check if 5 clock cycles has passed and return to the IDLE state, or we check `KEY_PRESS = 1` and transition to SCND_DIGIT_INPUT state.
- In SCND_DIGIT_INPUT state, all our outputs remain 0. We have detected two different `ITEM_CODE`, so we have check whether the code represents a valid input. If the number falls between 00-19 and the item is still available, we transition to the VALID_INPUT state. Otherwise, we transition to the INVALID_INPUT state.
- In INVALID_INPUT state, we output `INVALID_SEL = 1` and return to IDLE_STATE.
- In VALID_INPUT state, we output the `COST` corresponding to the valid input numbers, and rest of outputs remain 0. After 5 clock cycles, if `VALID_TRAN` remains 0, we transition to the FAILED_TRAN state. If `VALID_TRAN = 1` within 5 cycles, we transition to the VEND state. This signifies the the card payment was accepted.
- In FAILED_TRAN state, we output `FAILED_TRAN = 1`, cost of item selected, and rest of outputs remain 0. We then return to the IDLE state in the next clock cycle.
- In VEND state, we output cost of item and `VEND = 1`. Rest of outputs remain 0. We wait 5 clock cycles for `DOOR_OPEN` to be 1 to transition into the DOOR_OPEN state. If door remains closed for 5 cycles, we return to IDLE state.
- In DOOR_OPEN state, we continue to output cost of item, `VEND = 1`, and rest of outputs are 0. We wait for `DOOR_OPEN = 0` to transition to IDLE state, signifying end of our transaction. We remain in this state until `DOOR_OPEN = 0` or we receive the reset signal.
- Note: If `RESET = 1` at any point in our state diagram, it overrides other signal and arrows, reset all counts in vending machine to zero, and transitions to IDLE.

To implement the above FSM in Verilog, I created the `vending_machine` top module that takes in input listed in the Vending Machine I/O Diagram and outputs the outputs shown in the same diagram. Within my module, I initialized parameters representing all the states I have with a unique 4 bit binary number, two 4-bit registers `current_state` and `next_state`, and 20 4-bit registers corresponding to the count for each item in the vending machine. I have 3 main always block each with a specific task, similar to the sample code provided in the manuscript:

1. Sequential always block to update `current_state` to `next_state` at every `posedge` of the clock or update `current_state` to IDLE state when `RESET = 1`.
2. Combinational always block to decide what `next_state` should be based on the current state and current inputs.
3. Combinational always block to decide what outputs should be based on the current state and values stored in registers from inputs, such as which item is selected.

Aside from the blocks mentioned above, I have two more sequential always block that helps me with counting when 5 cycles has passed once we enter a particular state, such as CARD_INPUT_WAIT state. One always block uses a counter we implemented in previous projects and the other always block is in charge of setting a signal `clk_start` to signify that counter should start counting up.

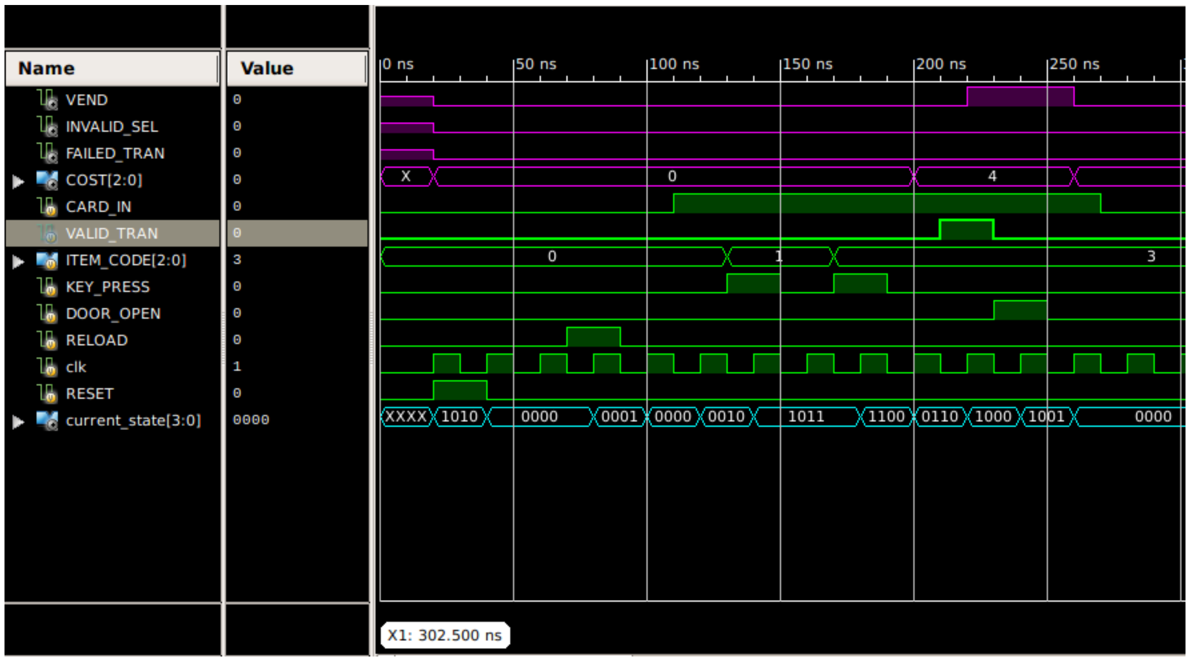

RTL Schematic of Vending Machine FSM

From first look, the RTL schematic above looks filled with registers, muxes, and counters. This makes sense as our FSM is fairly complicated having to keep track of 20 registers for each item in our vending machine on top of keeping track of the combinational logic to calculate the next state and the outputs. My Verilog code results in this schematic being generated as my code has 2 large combinational logic sections and requires many registers to keep track of internal signals to calculate the outputs and the next state.

3 Simulation Documentation

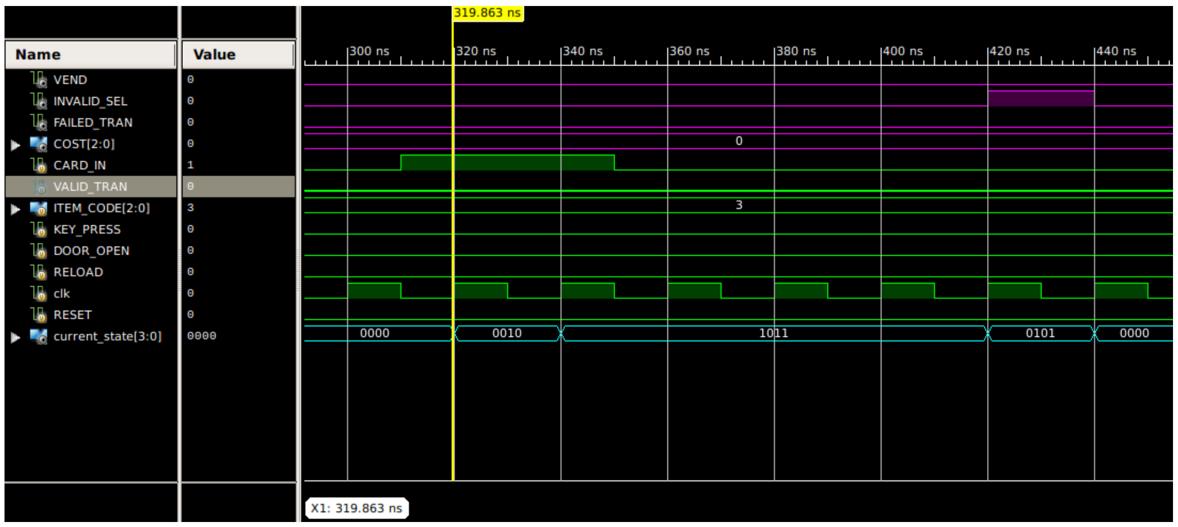
To test whether the FSM I implemented works as the manuscript described, I created my testbench that tests not only the successful runs of ordering an item from the vending machine, but all of the possible special cases, such as no key is pressed or an item has ran out. To simulate behavior of someone interacting with the vending machine, in my testbench, I created a system clock with T=10ns, and at each negative edge I change a particular input to observe how my FSM will deal with the change in input. Below are all the different cases I tested and their respective waveform.

1. **Successful Purchase of Valid In-Stock Item**
   This case represents a successful purchase from the vending machine machine where there were no errors all the way until vending machine door closes. The last light blue wave in the diagram below is an extra waveform I chose to display as output in order to show transition from one state to another. It is the binary representation of which state we're in. The way I simulated this behavior in the test bench is by first setting `RESET` as high to initialize all outputs to 0 and transition to the IDLE state. I then set `RELOAD = 1` in order to fill up the vending machine with stocks(shown at 70ns). Now to simulate someone using the machine, I set input `CARD_IN = 1` (shown at 110ns) and we're now in state `4'b0010` CARD_INSERT_WAIT state. So far all output remains 0 as no mistakes has occurred and nothing has been inputted yet. We then set `KEY_PRESS` to high twice in the next 3 cycles in order to simulating pressing keys 1 and 3 into the machine(130ns-200ns). At 200ns, we transition into state `4'b0110` VALID_INPUT as 13 corresponds to a valid in stock item in our machine. We correctly see that output `COST = 4` at 200ns and beyond, since from the manuscript cost of item 13 is 4. We then set `VALID_TRAN = 1` to represent the acceptance of our payment and once this signal is process we see that output `VEND = 1`. At 230ns, we set `DOOR_OPEN = 1` for one cycle to simulate item coming out of the machine. Once `DOOR_OPEN = 0` at 250ns, we see that `COST = 0` and `VEND = 0` as we have returned to IDLE state, which is the correct behavior.



Simulation Waveform for Case 1

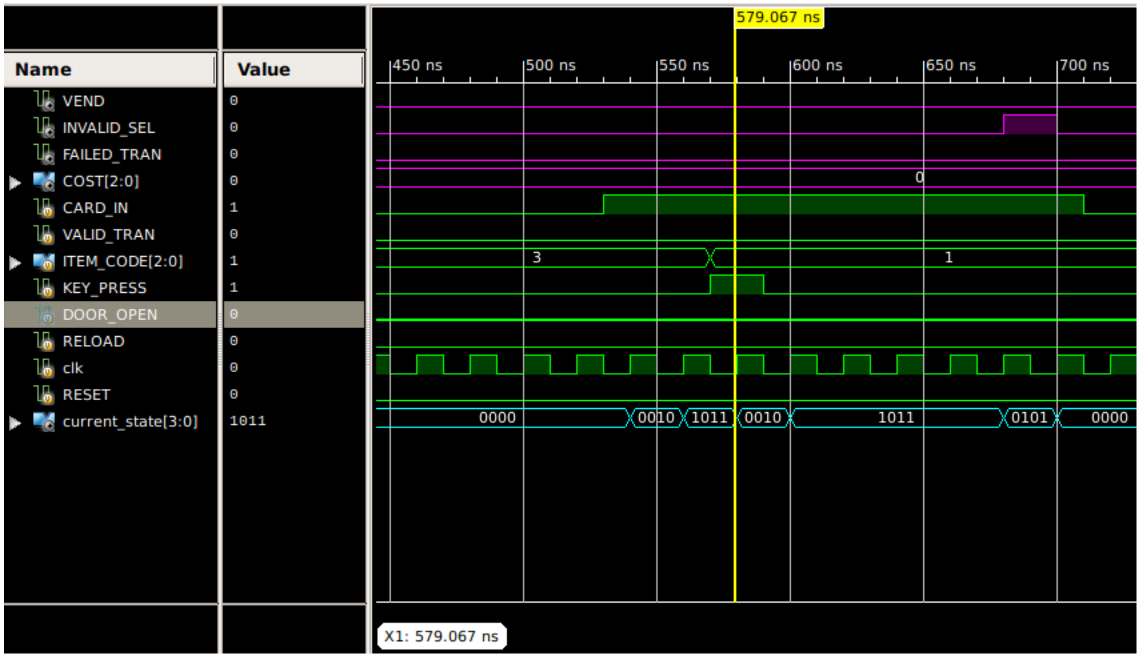
2. **Unsuccessful Purchase, No Key Press Detected**
This case represents an unsuccessful purchase where we return to IDLE state as after 5 clock cycles, no key press is detected. To simulate this case, we simply set `CARD_IN = 1` to transition into state `4'b0010` CARD_INSERT_WAIT and observe what happens after 5 clock cycles. At 320ns, we enter CARD_INSERT_WAIT, where all outputs are 0, and at 420ns, we see `INVALID_SEL = 1` for one cycle. One cycle is 20ns, thus our FSM correctly waits for 5 cycles before entering `4'b0101` INVALID_INPUT state, and outputting `INVALID_SEL = 1`. After one cycle, we return to IDLE state where all output is 0. Thus, our FSM behaves as described by the manuscript in this edge case.



Simulation Waveform for Case 2

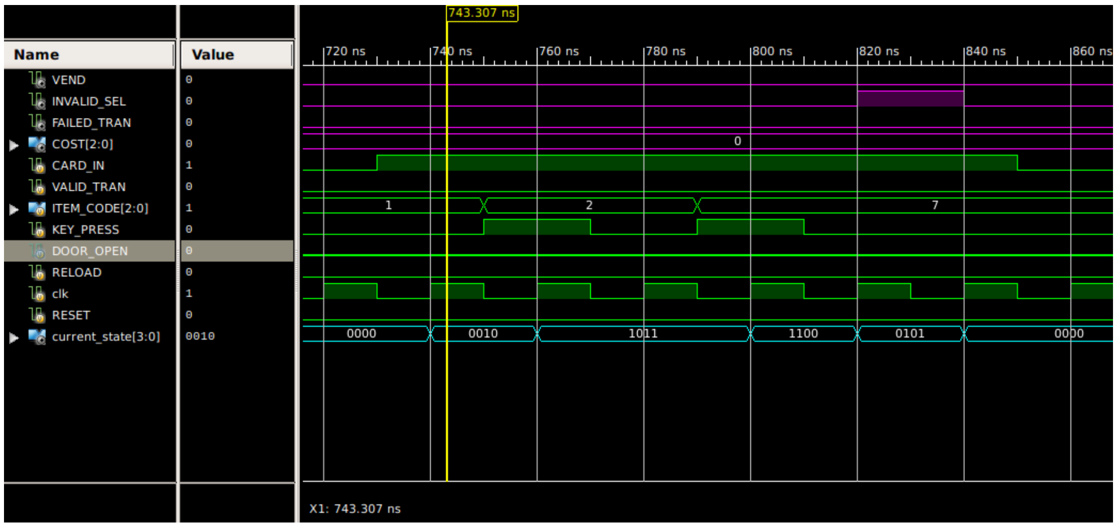
3. **Unsuccessful Purchase, Only 1 Key Press Detected**
This case represents an unsuccessful purchase where we return to IDLE state as we detect 1 key press, but after 5 cycles, we do not detect a second key press. To simulate this case, we simply set `CARD_IN = 1` to transition into state `4'b0010` CARD_INSERT_WAIT, we then set `KEY_PRESS = 1` for one cycle,

and observe what happens after 5 clock cycles. At 580ns, we detected and registered the first key press, all outputs remain 0 and we begin our counter. At 680ns, we see `INVALID_SEL = 1` for one cycle. One cycle is 20ns, thus our FSM correctly waits for 5 cycles before entering `4'b0101` INVALID_INPUT state, and outputting `INVALID_SEL = 1`. After one cycle, we return to IDLE state where all output is 0. Thus, our FSM behaves as described by the manuscript in this edge case.



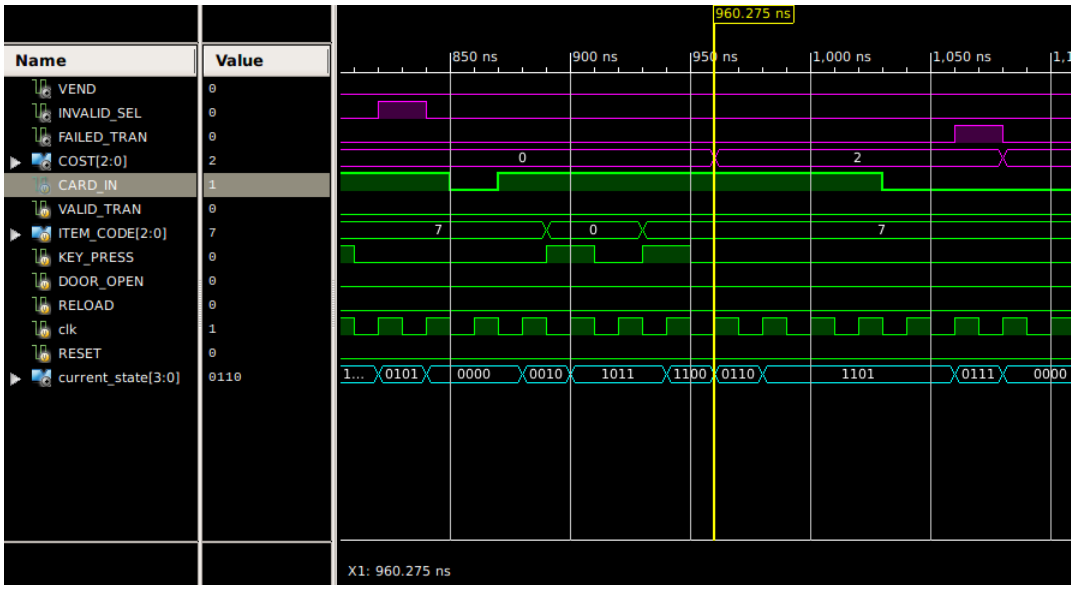
Simulation Waveform for Case 3

4. **Unsuccessful Purchase, Out of Range Item Number Entered**
   This case represents an unsuccessful purchase where two key press are detected, but the number they represent is out of range, so we want `INVALID_SEL = 1`. To simulate this case, we simply set `CARD_IN = 1` to transition into state `4'b0010` CARD_INSERT_WAIT, we then set `KEY_PRESS = 1` for one cycle, and then set `KEY_PRESS = 1` for one cycle. The two keys we entered are 2, 7, representing an item that doesnt exist in our vending machine. At 760ns, we see that 2 is read by our FSM, and then at 800ns, 7 is read by our FSM. One cycle after 7 is read, we see that at 820ns, `INVALID_SEL = 1` for one cycle. This is the correct behavior as 27 is an invalid input.



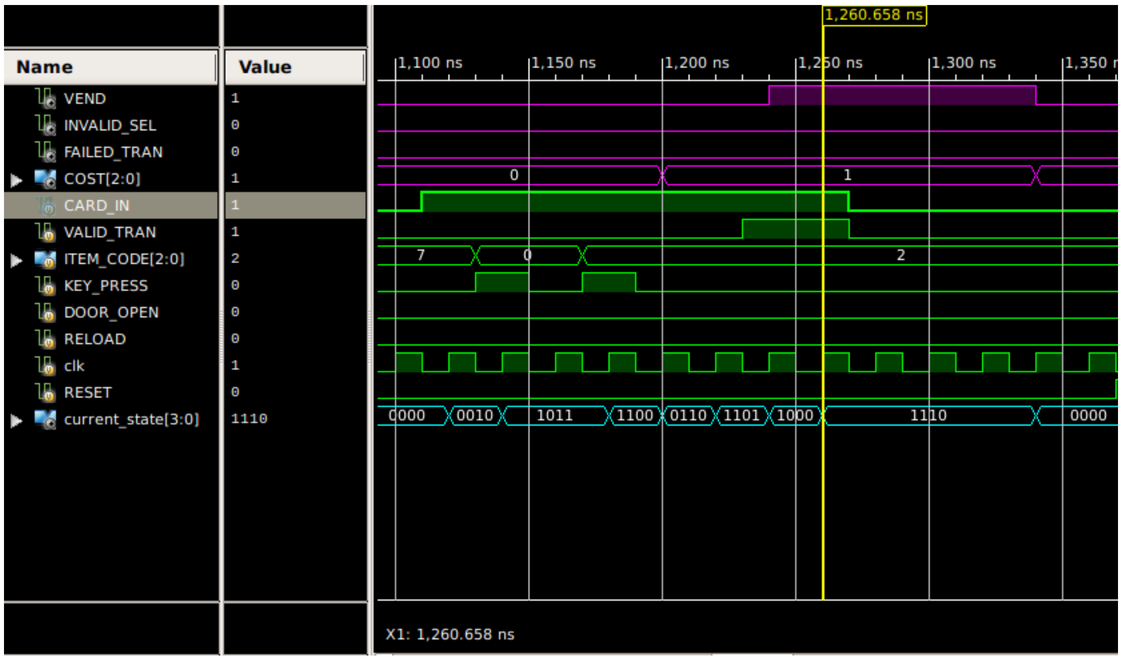
Simulation Waveform for Case 4

5. **Unsuccessful Purchase, Invalid Transaction**
This case represents an unsuccessful purchase where valid keys are inputted and the item requested has stock, but our card transaction has failed. To simulate this case, we set `CARD_IN = 1` to transition into state `4'b0010` CARD_INSERT_WAIT, we then set `KEY_PRESS = 1` for one cycle, and then set `KEY_PRESS = 1` for one cycle. From diagram below, we can see that we entered the key 0 and then the key 7, which is a valid item with stock as the machine was reloaded previously. At 960ns, we enter state `4'b0110` VALID_INPUT and wait for 5 cycles to pass. We see that output `COST` is correctly 2. At 1060ns, we see correctly observe `FAILED_TRAN = 1`.



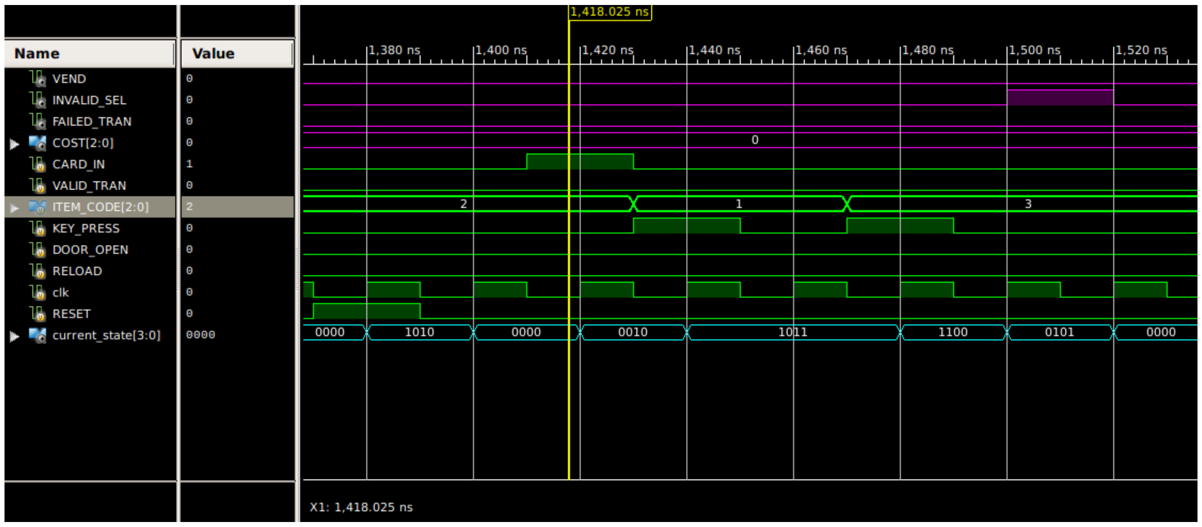
Simulation Waveform for Case 5

6. **Unsuccessful Purchase, Door Does Not Open**
This case represents an unsuccessful purchase where valid keys are inputted, and the transaction is valid, but the door of vending machine does not open. To simulate this case, we set `CARD_IN = 1` to transition into state `4'b0010` CARD_INSERT_WAIT, we then set `KEY_PRESS = 1` for one cycle, and then set `KEY_PRESS = 1` for one cycle. From diagram below, we can see that we entered the key 0 and then the key 2, which is a valid item with stock as the machine was reloaded previously. Thus, we see that output `COST` is correctly 2. At 1240ns, the `VALID_TRAN` signal is detected by our FSM, so output `VEND = 1`. From 1240ns on, we are in the state where we're waiting for either door to open or 5 cycles to pass. At 1340ns, 5 cycles has passed and door did not open, so we set all output to 0 and return to IDLE state.



Simulation Waveform for Case 6

7. **Unsuccessful Purchase, Valid Input, No Stock**
   This case represents an unsuccessful purchase where valid keys are inputted, but because we have no more stock of the requested item left, we set `INVALID_SEL = 1` and return to IDLE state. To simulate this case, we have to first make sure there is no more stock of a particular item. To do this, we set input `RESET = 1` for one cycle to make sure all counts of our item is 0. We then set `CARD_IN = 1` to transition into state `4'b0010` CARD_INSERT_WAIT, we then set `KEY_PRESS = 1` for one cycle, and then set `KEY_PRESS = 1` for one cycle. From diagram below, we can see that we entered the key 1 and then the key 3, which is a valid item number as it is within 00-19. But since our vending machine is out of stock, we can see that at 1500ns, `INVALID_SEL = 1` for one cycle after second key is entered. This is the correct expected behavior. All other outputs remain correctly at 0.



Simulation Waveform for Case 7

8. **Case Where Signals Change**

This case occurs throughout the waveform diagrams above as often the `CARD_IN` signal stays high throughout transaction or goes low in the middle of a transaction. We can see that we correctly handle this case as regardless of how long `CARD_IN` signal stays high, our module treats it correctly. If card is taken out earlier, it is okay, as our vending machine remembers the card info. So no odd behavior will happen when signals change arbitrarily.

One bug I found during simulation for this project was that my counter that keeps track of whether 5 cycle has passed was incorrect, as it continues to count up even in cases where the counter is not needed. I fixed this by adding an additional internal register to signal when count should be incremented.

4 Synthesis and Implementation Report

Sections of Synthesis and Implementation Report is attached at end of report. From the 'Design Summary' part of the synthesis report, we can conclude the different components that my implementation of the Vending Machine FSM uses. For example, we can see that we indeed do use a lot of registers from the synthesis report and we can also see the different clock signals needed for our module. From implementation report, we can see that there is no errors when trying to implement our module, but there are a few warnings due to truncation of numbers into registers. The warnings can be ignored as in my code I made sure that truncation will not occur for those registers.

5 Conclusion

In this project, I designed and implemented the Vending Machine FSM according to the behavior in the project manuscript. I designed the Vending Machine based on a Moore Machine, that has similar structure to the turnstile FSM example in the manuscript. To correctly implement this, I had to first draw the FSM diagram to allow me to clearly see what output should be in each state and how to transition to a different state. One of the major difficulty I encountered in this assignment was trying to incorporate a counter in order to keep track of whether 5 cycles has passed. I dealt with this by experimenting my code on a smaller, simpler module in order to see how to correctly use different signal within my module to signal counter to start counting and to detect output of the counter.

6 Reports

6.1 Synthesis Report

```
=========================================================================
*                            Design Summary                             *
=========================================================================

Top Level Output File Name         : vending_machine.ngc

Primitive and Black Box Usage:
------------------------------
# BELS                             : 226
#      INV                         : 18
#      LUT2                        : 5
#      LUT3                        : 54
#      LUT4                        : 37
#      LUT5                        : 31
#      LUT6                        : 70
#      MUXF7                       : 11
# FlipFlops/Latches                : 89
#      FDR                         : 37
#      FDRE                        : 35
#      FDS                         : 4
#      LD                          : 13
# Clock Buffers                    : 1
#      BUFGP                       : 1
# IO Buffers                       : 15
#      IBUF                        : 9
#      OBUF                        : 6

Device utilization summary:
---------------------------

Selected Device : 6slx16csg324-3


Slice Logic Utilization:
 Number of Slice Registers:              83  out of  18224     0%
 Number of Slice LUTs:                  215  out of   9112     2%
    Number used as Logic:               215  out of   9112     2%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:    220
   Number with an unused Flip Flop:     137  out of    220    62%
   Number with an unused LUT:             5  out of    220     2%
   Number of fully used LUT-FF pairs:    78  out of    220    35%
   Number of unique control sets:        27

IO Utilization:
 Number of IOs:                          16
 Number of bonded IOBs:                  16  out of    232     6%
    IOB Flip Flops/Latches:               6

Specific Feature Utilization:
 Number of BUFG/BUFGCTRLs:                1  out of     16     6%

---------------------------
```

```
Partition Resource Summary:
---------------------------

  No Partitions were found in this design.

---------------------------


=========================================================================
Timing Report

NOTE: THESE TIMING NUMBERS ARE ONLY A SYNTHESIS ESTIMATE.
      FOR ACCURATE TIMING INFORMATION PLEASE REFER TO THE TRACE REPORT
      GENERATED AFTER PLACE-and-ROUTE.

Clock Information:
------------------
-------------------------------------------------------------------------------------+-------------------------+--
Clock Signal                                                                         | Clock buffer(FF name)   | L
-------------------------------------------------------------------------------------+-------------------------+--
_n1087(_n10871:O)                                                                    | NONE(*)(INVALID_SEL)    | 1
_n1084(_n10841:O)                                                                    | NONE(*)(FAILED_TRAN)    | 1
_n1079(_n10791:O)                                                                    | NONE(*)(VEND)           | 1
CLK                                                                                  | BUFGP                   | 7
current_state[3]_GND_6_o_Mux_472_o(Mmux_current_state[3]_GND_6_o_Mux_472_o11:O)      | NONE(*)(code1_2)        | 3
current_state[3]_GND_2_o_Mux_464_o(Mmux_current_state[3]_GND_2_o_Mux_464_o11:O)      | NONE(*)(code2_2)        | 3
current_state[3]_GND_10_o_Mux_480_o(Mmux_current_state[3]_GND_10_o_Mux_480_o1:O)     | NONE(*)(code1inputted)  | 1
current_state[3]_PWR_14_o_Select_533_o(Mmux_current_state[3]_PWR_14_o_Select_533_o11:O)| NONE(*)(COST_2)         | 3
-------------------------------------------------------------------------------------+-------------------------+--
(*) These 7 clock signal(s) are generated by combinatorial logic,
and XST is not able to identify which are the primary clock signals.
Please use the CLOCK_SIGNAL constraint to specify the clock signal(s) generated by combinatorial logic.
INFO:Xst:2169 - HDL ADVISOR - Some clock signals were not automatically buffered by XST with BUFG/BUFR resources. P

Asynchronous Control Signals Information:
----------------------------------------
No asynchronous control signals found in this design

Timing Summary:
---------------
Speed Grade: -3

   Minimum period: 4.511ns (Maximum Frequency: 221.683MHz)
   Minimum input arrival time before clock: 5.316ns
   Maximum output required time after clock: 3.648ns
   Maximum combinational path delay: No path found

Timing Details:
---------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default period analysis for Clock 'CLK'
  Clock period: 4.511ns (frequency: 221.683MHz)
  Total number of paths / destination ports: 1267 / 179
-------------------------------------------------------------------------
Delay:               4.511ns (Levels of Logic = 3)
  Source:            count_2 (FF)
  Destination:       count_0 (FF)
  Source Clock:      CLK rising
  Destination Clock: CLK rising

  Data Path: count_2 to count_0
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FDRE:C->Q             4   0.447   0.912  count_2 (count_2)
     LUT3:I0->O           11   0.205   0.883  GND_1_o_GND_1_o_equal_375_o<2>1 (GND_1_o_GND_1_o_equal_375_o)
     LUT6:I5->O            1   0.205   0.684  Mmux_count_up22 (Mmux_count_up21)
     LUT6:I4->O            3   0.203   0.650  Mmux_count_up25 (count_up)
     FDRE:CE                   0.322          count_0
    ----------------------------------------
    Total                      4.511ns (1.382ns logic, 3.129ns route)
                                       (30.6% logic, 69.4% route)

=========================================================================
Timing constraint: Default OFFSET IN BEFORE for Clock 'current_state[3]_GND_6_o_Mux_472_o'
  Total number of paths / destination ports: 3 / 3
-------------------------------------------------------------------------
Offset:              1.875ns (Levels of Logic = 1)
  Source:            ITEM_CODE<2> (PAD)
  Destination:       code1_2 (LATCH)
  Destination Clock: current_state[3]_GND_6_o_Mux_472_o falling

  Data Path: ITEM_CODE<2> to code1_2
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             2   1.222   0.616  ITEM_CODE_2_IBUF (ITEM_CODE_2_IBUF)
     LD:D                      0.037          code1_2
    ----------------------------------------
    Total                      1.875ns (1.259ns logic, 0.616ns route)
                                       (67.1% logic, 32.9% route)

=========================================================================
Timing constraint: Default OFFSET IN BEFORE for Clock 'current_state[3]_GND_2_o_Mux_464_o'
  Total number of paths / destination ports: 6 / 3
-------------------------------------------------------------------------
Offset:              2.449ns (Levels of Logic = 2)
  Source:            KEY_PRESS (PAD)
  Destination:       code2_2 (LATCH)
  Destination Clock: current_state[3]_GND_2_o_Mux_464_o falling

  Data Path: KEY_PRESS to code2_2
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O            11   1.222   0.987  KEY_PRESS_IBUF (KEY_PRESS_IBUF)
     LUT4:I2->O            1   0.203   0.000  current_state[3]_code2[2]_Mux_465_o1 (current_state[3]_code2[2]_Mux_4
     LD:D                      0.037          code2_2
    ----------------------------------------
    Total                      2.449ns (1.462ns logic, 0.987ns route)
                                       (59.7% logic, 40.3% route)

=========================================================================
Timing constraint: Default OFFSET IN BEFORE for Clock 'current_state[3]_PWR_14_o_Select_533_o'
  Total number of paths / destination ports: 3 / 3
-------------------------------------------------------------------------
```

```
Offset:              3.491ns (Levels of Logic = 3)
  Source:            RELOAD (PAD)
  Destination:       COST_2 (LATCH)
  Destination Clock: current_state[3]_PWR_14_o_Select_533_o falling

  Data Path: RELOAD to COST_2
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             6   1.222   0.745  RELOAD_IBUF (RELOAD_IBUF)
     LUT5:I4->O            5   0.205   1.079  _n10871 (_n1087)
     LUT6:I0->O            1   0.203   0.000  Mmux_current_state[3]_GND_1_o_Select_536_o11 (current_state[3]_GND_1_
     LD:D                      0.037          COST_0
    ----------------------------------------
    Total                      3.491ns (1.667ns logic, 1.824ns route)
                                       (47.8% logic, 52.2% route)

=========================================================================
Timing constraint: Default OFFSET IN BEFORE for Clock 'CLK'
  Total number of paths / destination ports: 66 / 19
-------------------------------------------------------------------------
Offset:              5.316ns (Levels of Logic = 4)
  Source:            KEY_PRESS (PAD)
  Destination:       count_0 (FF)
  Destination Clock: CLK rising

  Data Path: KEY_PRESS to count_0
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O            11   1.222   0.883  KEY_PRESS_IBUF (KEY_PRESS_IBUF)
     LUT2:I1->O            1   0.205   0.944  Mmux_count_up21 (Mmux_count_up2)
     LUT6:I0->O            1   0.203   0.684  Mmux_count_up22 (Mmux_count_up21)
     LUT6:I4->O            3   0.203   0.650  Mmux_count_up25 (count_up)
     FDRE:CE                   0.322          count_0
    ----------------------------------------
    Total                      5.316ns (2.155ns logic, 3.161ns route)
                                       (40.5% logic, 59.5% route)

=========================================================================
Timing constraint: Default OFFSET OUT AFTER for Clock 'current_state[3]_PWR_14_o_Select_533_o'
  Total number of paths / destination ports: 3 / 3
-------------------------------------------------------------------------
Offset:              3.648ns (Levels of Logic = 1)
  Source:            COST_2 (LATCH)
  Destination:       COST<2> (PAD)
  Source Clock:      current_state[3]_PWR_14_o_Select_533_o falling

  Data Path: COST_2 to COST<2>
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     LD:G->Q               1   0.498   0.579  COST_2 (COST_2)
     OBUF:I->O                 2.571          COST_2_OBUF (COST<2>)
    ----------------------------------------
    Total                      3.648ns (3.069ns logic, 0.579ns route)
                                       (84.1% logic, 15.9% route)

=========================================================================
Timing constraint: Default OFFSET OUT AFTER for Clock '_n1079'
  Total number of paths / destination ports: 1 / 1
-------------------------------------------------------------------------
Offset:              3.648ns (Levels of Logic = 1)
  Source:            VEND (LATCH)
  Destination:       VEND (PAD)
  Source Clock:      _n1079 falling

  Data Path: VEND to VEND
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     LD:G->Q               1   0.498   0.579  VEND (VEND_OBUF)
     OBUF:I->O                 2.571          VEND_OBUF (VEND)
    ----------------------------------------
    Total                      3.648ns (3.069ns logic, 0.579ns route)
                                       (84.1% logic, 15.9% route)

=========================================================================
Timing constraint: Default OFFSET OUT AFTER for Clock '_n1087'
  Total number of paths / destination ports: 1 / 1
-------------------------------------------------------------------------
Offset:              3.648ns (Levels of Logic = 1)
  Source:            INVALID_SEL (LATCH)
  Destination:       INVALID_SEL (PAD)
  Source Clock:      _n1087 falling

  Data Path: INVALID_SEL to INVALID_SEL
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     LD:G->Q               1   0.498   0.579  INVALID_SEL (INVALID_SEL_OBUF)
     OBUF:I->O                 2.571          INVALID_SEL_OBUF (INVALID_SEL)
    ----------------------------------------
    Total                      3.648ns (3.069ns logic, 0.579ns route)
                                       (84.1% logic, 15.9% route)

=========================================================================
Timing constraint: Default OFFSET OUT AFTER for Clock '_n1084'
  Total number of paths / destination ports: 1 / 1
-------------------------------------------------------------------------
Offset:              3.648ns (Levels of Logic = 1)
  Source:            FAILED_TRAN (LATCH)
  Destination:       FAILED_TRAN (PAD)
  Source Clock:      _n1084 falling

  Data Path: FAILED_TRAN to FAILED_TRAN
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     LD:G->Q               1   0.498   0.579  FAILED_TRAN (FAILED_TRAN_OBUF)
     OBUF:I->O                 2.571          FAILED_TRAN_OBUF (FAILED_TRAN)
    ----------------------------------------
    Total                      3.648ns (3.069ns logic, 0.579ns route)
                                       (84.1% logic, 15.9% route)

=========================================================================

Cross Clock Domains Report:
```

```
-------------------------

Clock to Setup on destination clock CLK
-----------------------------------+---------+---------+---------+---------+
                                   | Src:Rise| Src:Fall| Src:Rise| Src:Fall|
Source Clock                       |Dest:Rise|Dest:Rise|Dest:Fall|Dest:Fall|
-----------------------------------+---------+---------+---------+---------+
CLK                                |    4.511|         |         |         |
current_state[3]_GND_10_o_Mux_480_o|         |    4.668|         |         |
current_state[3]_GND_2_o_Mux_464_o |         |    4.873|         |         |
current_state[3]_GND_6_o_Mux_472_o |         |    4.456|         |         |
-----------------------------------+---------+---------+---------+---------+

Clock to Setup on destination clock _n1079
---------------+---------+---------+---------+---------+
               | Src:Rise| Src:Fall| Src:Rise| Src:Fall|
Source Clock   |Dest:Rise|Dest:Rise|Dest:Fall|Dest:Fall|
---------------+---------+---------+---------+---------+
CLK            |         |         |    2.531|         |
---------------+---------+---------+---------+---------+

Clock to Setup on destination clock _n1084
---------------+---------+---------+---------+---------+
               | Src:Rise| Src:Fall| Src:Rise| Src:Fall|
Source Clock   |Dest:Rise|Dest:Rise|Dest:Fall|Dest:Fall|
---------------+---------+---------+---------+---------+
CLK            |         |         |    2.531|         |
---------------+---------+---------+---------+---------+

Clock to Setup on destination clock _n1087
---------------+---------+---------+---------+---------+
               | Src:Rise| Src:Fall| Src:Rise| Src:Fall|
Source Clock   |Dest:Rise|Dest:Rise|Dest:Fall|Dest:Fall|
---------------+---------+---------+---------+---------+
CLK            |         |         |    2.548|         |
---------------+---------+---------+---------+---------+

Clock to Setup on destination clock current_state[3]_GND_10_o_Mux_480_o
---------------+---------+---------+---------+---------+
               | Src:Rise| Src:Fall| Src:Rise| Src:Fall|
Source Clock   |Dest:Rise|Dest:Rise|Dest:Fall|Dest:Fall|
---------------+---------+---------+---------+---------+
CLK            |         |         |    2.531|         |
---------------+---------+---------+---------+---------+

Clock to Setup on destination clock current_state[3]_GND_2_o_Mux_464_o
-----------------------------------+---------+---------+---------+---------+
                                   | Src:Rise| Src:Fall| Src:Rise| Src:Fall|
Source Clock                       |Dest:Rise|Dest:Rise|Dest:Fall|Dest:Fall|
-----------------------------------+---------+---------+---------+---------+
CLK                                |         |         |    2.531|         |
current_state[3]_GND_10_o_Mux_480_o|         |         |    1.597|         |
-----------------------------------+---------+---------+---------+---------+

Clock to Setup on destination clock current_state[3]_PWR_14_o_Select_533_o
----------------------------------+---------+---------+---------+---------+
                                  | Src:Rise| Src:Fall| Src:Rise| Src:Fall|
Source Clock                      |Dest:Rise|Dest:Rise|Dest:Fall|Dest:Fall|
----------------------------------+---------+---------+---------+---------+
CLK                               |         |         |    3.813|         |
current_state[3]_GND_2_o_Mux_464_o|         |         |    2.402|         |
current_state[3]_GND_6_o_Mux_472_o|         |         |    2.445|         |
----------------------------------+---------+---------+---------+---------+

=========================================================================


Total REAL time to Xst completion: 6.00 secs
Total CPU time to Xst completion: 5.67 secs

-->


Total memory usage is 396832 kilobytes

Number of errors   :    0 (   0 filtered)
Number of warnings :   39 (   0 filtered)
Number of infos    :    1 (   0 filtered)
```

6.2 Implementation Report

```
Release 14.7 Map P.20131013 (lin64)
Xilinx Mapping Report File for Design 'vending_machine'

Design Information
------------------
Command Line   : map -intstyle ise -p xc6slx16-csg324-3 -w -logic_opt off -ol
high -t 1 -xt 0 -register_duplication off -r 4 -global_opt off -mt off -ir off
-pr off -lc off -power off -o vending_machine_map.ncd vending_machine.ngd
vending_machine.pcf
Target Device  : xc6slx16
Target Package : csg324
Target Speed   : -3
Mapper Version : spartan6 -- $Revision: 1.55 $
Mapped Date    : Sun May 24 23:44:11 2020

Design Summary
--------------
Number of errors:      0
Number of warnings:    7
Slice Logic Utilization:
  Number of Slice Registers:                    83 out of  18,224    1%
    Number used as Flip Flops:                  76
    Number used as Latches:                      7
    Number used as Latch-thrus:                  0
    Number used as AND/OR logics:                0
  Number of Slice LUTs:                        156 out of   9,112    1%
    Number used as logic:                      156 out of   9,112    1%
      Number using O6 output only:             129
      Number using O5 output only:               0
      Number using O5 and O6:                   27
      Number used as ROM:                        0
    Number used as Memory:                       0 out of   2,176    0%

Slice Logic Distribution:
  Number of occupied Slices:                    55 out of   2,278    2%
  Number of MUXCYs used:                         0 out of   4,556    0%
  Number of LUT Flip Flop pairs used:          164
    Number with an unused Flip Flop:            82 out of     164   50%
    Number with an unused LUT:                   8 out of     164    4%
    Number of fully used LUT-FF pairs:          74 out of     164   45%
    Number of unique control sets:              23
    Number of slice register sites lost
      to control set restrictions:             125 out of  18,224    1%

  A LUT Flip Flop pair for this architecture represents one LUT paired with
  one Flip Flop within a slice.  A control set is a unique combination of
  clock, reset, set, and enable signals for a registered element.
  The Slice Logic Distribution report is not meaningful if the design is
  over-mapped for a non-slice resource or if Placement fails.

IO Utilization:
  Number of bonded IOBs:                        16 out of     232    6%
    IOB Latches:                                 6

Specific Feature Utilization:
  Number of RAMB16BWERs:                         0 out of      32    0%
  Number of RAMB8BWERs:                          0 out of      64    0%
  Number of BUFIO2/BUFIO2_2CLKs:                0 out of      32    0%
  Number of BUFIO2FB/BUFIO2FB_2CLKs:            0 out of      32    0%
  Number of BUFG/BUFGMUXs:                      1 out of      16    6%
    Number used as BUFGs:                       1
    Number used as BUFGMUX:                     0
  Number of DCM/DCM_CLKGENs:                    0 out of       4    0%
  Number of ILOGIC2/ISERDES2s:                  0 out of     248    0%
  Number of IODELAY2/IODRP2/IODRP2_MCBs:        0 out of     248    0%
  Number of OLOGIC2/OSERDES2s:                  6 out of     248    2%
    Number used as OLOGIC2s:                    6
    Number used as OSERDES2s:                   0
  Number of BSCANs:                             0 out of       4    0%
  Number of BUFHs:                              0 out of     128    0%
  Number of BUFPLLs:                            0 out of       8    0%
  Number of BUFPLL_MCBs:                        0 out of       4    0%
  Number of DSP48A1s:                           0 out of      32    0%
  Number of ICAPs:                              0 out of       1    0%
  Number of MCBs:                               0 out of       2    0%
  Number of PCILOGICSEs:                        0 out of       2    0%
  Number of PLL_ADVs:                           0 out of       2    0%
  Number of PMVs:                               0 out of       1    0%
  Number of STARTUPs:                           0 out of       1    0%
  Number of SUSPEND_SYNCs:                      0 out of       1    0%

Average Fanout of Non-Clock Nets:               4.56

Peak Memory Usage:  676 MB
Total REAL time to MAP completion:  10 secs
Total CPU time to MAP completion:   7 secs

Table of Contents
-----------------
Section 1 - Errors
Section 2 - Warnings
Section 3 - Informational
Section 4 - Removed Logic Summary
Section 5 - Removed Logic
Section 6 - IOB Properties
Section 7 - RPMs
Section 8 - Guide Report
Section 9 - Area Group and Partition Summary
Section 10 - Timing Report
Section 11 - Configuration String Information
Section 12 - Control Set Information
Section 13 - Utilization by Hierarchy

Section 1 - Errors
------------------

Section 2 - Warnings
--------------------
WARNING:PhysDesignRules:372 - Gated clock. Clock net
   current_state[3]_GND_6_o_Mux_472_o is sourced by a combinatorial pin. This is
   not good design practice. Use the CE pin to control the loading of data into
   the flip-flop.
WARNING:PhysDesignRules:372 - Gated clock. Clock net _n1087 is sourced by a
   combinatorial pin. This is not good design practice. Use the CE pin to
   control the loading of data into the flip-flop.
WARNING:PhysDesignRules:372 - Gated clock. Clock net
```

```
   current_state[3]_PWR_14_o_Select_533_o is sourced by a combinatorial pin.
   This is not good design practice. Use the CE pin to control the loading of
   data into the flip-flop.
WARNING:PhysDesignRules:372 - Gated clock. Clock net
   current_state[3]_GND_10_o_Mux_480_o is sourced by a combinatorial pin. This
   is not good design practice. Use the CE pin to control the loading of data
   into the flip-flop.
WARNING:PhysDesignRules:372 - Gated clock. Clock net
   current_state[3]_GND_2_o_Mux_464_o is sourced by a combinatorial pin. This is
   not good design practice. Use the CE pin to control the loading of data into
   the flip-flop.
WARNING:PhysDesignRules:372 - Gated clock. Clock net _n1084 is sourced by a
   combinatorial pin. This is not good design practice. Use the CE pin to
   control the loading of data into the flip-flop.
WARNING:PhysDesignRules:372 - Gated clock. Clock net _n1079 is sourced by a
   combinatorial pin. This is not good design practice. Use the CE pin to
   control the loading of data into the flip-flop.

Section 3 - Informational
-------------------------
INFO:MapLib:562 - No environment variables are currently set.
INFO:LIT:244 - All of the single ended outputs in this design are using slew
   rate limited output drivers. The delay on speed critical single ended outputs
   can be dramatically reduced by designating them as fast outputs.
INFO:Pack:1716 - Initializing temperature to 85.000 Celsius. (default - Range:
   0.000 to 85.000 Celsius)
INFO:Pack:1720 - Initializing voltage to 1.140 Volts. (default - Range: 1.140 to
   1.260 Volts)
INFO:Map:215 - The Interim Design Summary has been generated in the MAP Report
   (.mrp).
INFO:Pack:1650 - Map created a placed design.

Section 4 - Removed Logic Summary
---------------------------------

Section 5 - Removed Logic
-------------------------

To enable printing of redundant blocks removed and signals merged, set the
detailed map report option and rerun map.

Section 6 - IOB Properties
--------------------------

+---------------------------------------------------------------------------------------------------------------
| IOB Name                           | Type             | Direction | IO Standard          | Diff  | Drive    | Sle
|                                    |                  |           |                      | Term  | Strength | Rat
+---------------------------------------------------------------------------------------------------------------
| CARD_IN                            | IOB              | INPUT     | LVCMOS25             |       |          |
| CLK                                | IOB              | INPUT     | LVCMOS25             |       |          |
| COST<0>                            | IOB              | OUTPUT    | LVCMOS25             |       | 12       | SLO
| COST<1>                            | IOB              | OUTPUT    | LVCMOS25             |       | 12       | SLO
| COST<2>                            | IOB              | OUTPUT    | LVCMOS25             |       | 12       | SLO
| DOOR_OPEN                          | IOB              | INPUT     | LVCMOS25             |       |          |
| FAILED_TRAN                        | IOB              | OUTPUT    | LVCMOS25             |       | 12       | SLO
| INVALID_SEL                        | IOB              | OUTPUT    | LVCMOS25             |       | 12       | SLO
| ITEM_CODE<0>                       | IOB              | INPUT     | LVCMOS25             |       |          |
| ITEM_CODE<1>                       | IOB              | INPUT     | LVCMOS25             |       |          |
| ITEM_CODE<2>                       | IOB              | INPUT     | LVCMOS25             |       |          |
| KEY_PRESS                          | IOB              | INPUT     | LVCMOS25             |        |          |
| RELOAD                             | IOB              | INPUT     | LVCMOS25             |        |          |
| RESET                              | IOB              | INPUT     | LVCMOS25             |        |          |
| VALID_TRAN                         | IOB              | INPUT     | LVCMOS25             |        |          |
| VEND                               | IOB              | OUTPUT    | LVCMOS25             |        | 12       | SLC
+-------------------------------------------------------------------------------------------------------------------

Section 7 - RPMs
----------------

Section 8 - Guide Report
------------------------
Guide not run on this design.

Section 9 - Area Group and Partition Summary
--------------------------------------------

Partition Implementation Status
-------------------------------

  No Partitions were found in this design.

-------------------------------

Area Group Information
----------------------

  No area groups were found in this design.

----------------------

Section 10 - Timing Report
--------------------------
A logic-level (pre-route) timing report can be generated by using Xilinx static
timing analysis tools, Timing Analyzer (GUI) or TRCE (command line), with the
mapped NCD and PCF files. Please note that this timing report will be generated
using estimated delay information. For accurate numbers, please generate a
timing report with the post Place and Route NCD file.

For more information about the Timing Analyzer, consult the Xilinx Timing
Analyzer Reference Manual; for more information about TRCE, consult the Xilinx
Command Line Tools User Guide "TRACE" chapter.

Section 11 - Configuration String Details
-----------------------------------------
Use the "-detail" map option to print out Configuration Strings

Section 12 - Control Set Information
------------------------------------
Use the "-detail" map option to print out Control Set Information.

Section 13 - Utilization by Hierarchy
-------------------------------------
Use the "-detail" map option to print out the Utilization by Hierarchy section.
```