CS M152A Lab 3 Report

Jonathan Chau

UID: 705-166-732

TA: Boyuan He

February 28, 2021

1. Introduction and Requirements

This lab introduces the use of the finite state machine (FSM) design and how it applies to the Xilinx ISE Environment. Finite state machines are useful for modelling many real-world systems, especially when it comes to behavioral modelling of sequential circuits. The two types of FSM designs include the Moore machine, which depends on the state only; and the Mealy machine, which depends on both the state and the input. For this particular project, we want to design a vending machine that has 20 item slots, each with a capacity of 10 items, that accepts card payment only.

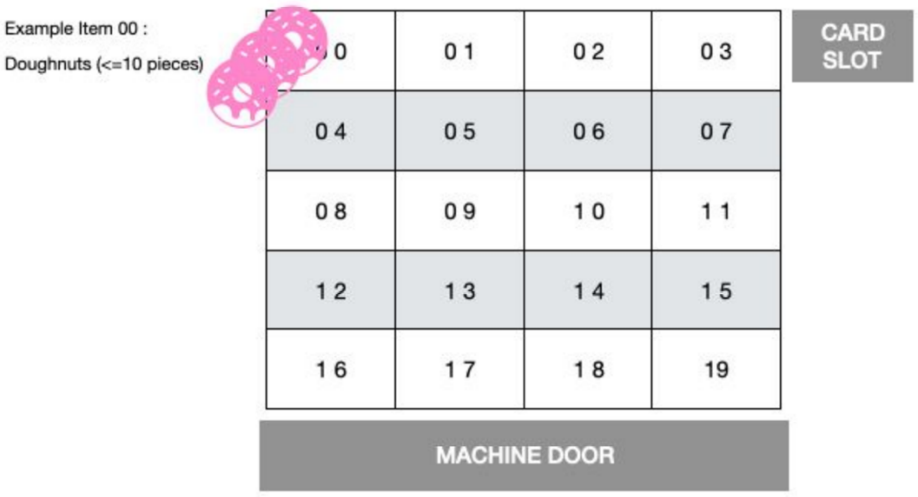

Figure 1. A Vending Machine

2. Design Description

The overall design of the module is based off the input/output model given in the project manual below. To implement this module, I designed an FSM diagram that is consistent with the intended behavior specified in the project manual. The outputs to consider are VEND, INVALID_SEL, FAILED_TRAN, and COST.

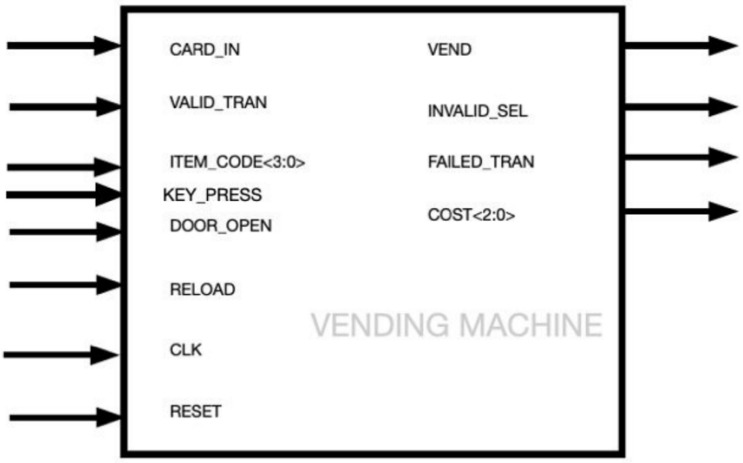

Figure 2. Inputs and Outputs of the Overall Module

2.1 Finite State Machine

According to the project description, a module that has less states tends to have a much cleaner design. As a result, I reduced the number of states in my module to five, in hopes to create a sleeker design. Regardless, the design represents a Moore machine, as the outputs are solely dependent on the state of the machine.

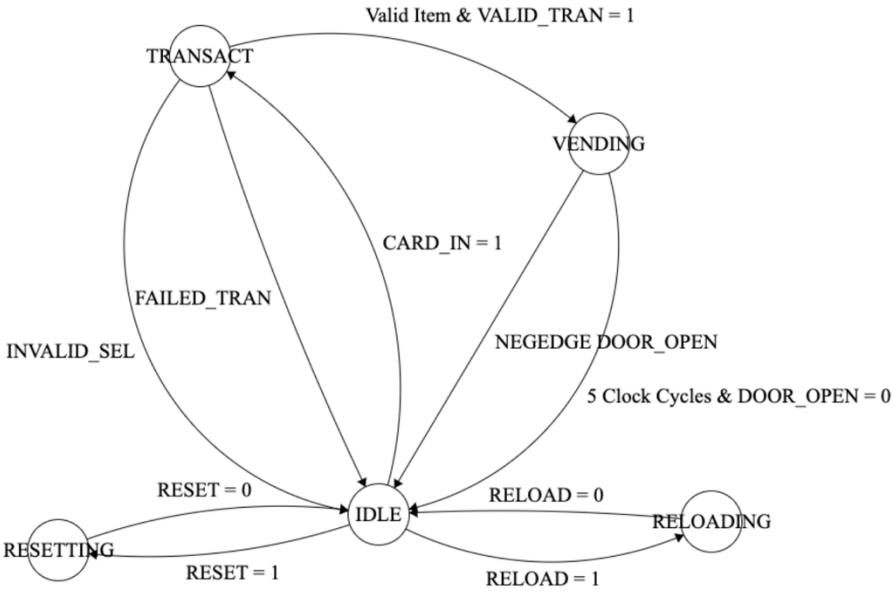

Figure 3. FSM Diagram of the Overall Module

Each of the five states are described as follows:

- **RESETTING:** The vending machine goes to this state anytime the RESET signal is HIGH. All outputs are set to 0, and all item counters are set to 0.

- **IDLE:** All outputs are set to zero. There are two possible ways to change states. The first is when the RELOAD signal is set to HIGH, where the module will transition to the RELOAD state. The other method is when CARD_IN is 1, and RELOAD is LOW, where the module will transition to the TRANSACT state.

- **RELOADING:** All of the item counters are restocked to 10. When the RELOAD signal is LOW, the module will transition back to the IDLE state. In this state, all outputs are set to zero.

- **TRANSACT:** This state consists of multiple parts, such as entering the item code, signaling when the transaction is valid, and when the transaction fails and goes back to the IDLE state. Here, the outputs are dependent based on the sub-state of this machine. When transitioned from the IDLE state, go to the NO_INPUT sub-state.

    - **NO_INPUTS:** All outputs are set to zero. If it detects a positive edge on KEY_PRESS, it goes to the ONE_INPUT sub-state. Otherwise, if there is no KEY_PRESS within five clock cycles (50ns), then it goes to the INVALID state.

- **ONE_INPUT:** All outputs are set zero. If it detects a positive edge on KEY_PRESS, two things can happen. If the item code is within the valid range (00-19), and that item is still in stock, transition to the VALIDATION state. Otherwise, transition to the INVALID state. If there is no KEY_PRESS within five clock cycles (50ns), then it goes to the INVALID state.

- **VALIDATION:** Set COST to the cost of the selected item. Set all other outputs to zero. If VALID_TRAN is set to HIGH, go to the VENDING state. Otherwise, if VALID_TRAN isn't set at HIGH for five clock cycles (50ns), transition to the FAILED sub-state.

- **INVALID:** Set INVALID_SEL to 1. Set all other outputs to 0. After one clock cycle, go to the IDLE state.

- **FAILED:** Set FAILED_TRAN to 1. Set COST to the cost of the item selected. Set all other outputs to 0. After one clock cycle, go to the IDLE state.

- For convenience, an FSM diagram of the TRANSACT state is shown in detail below.

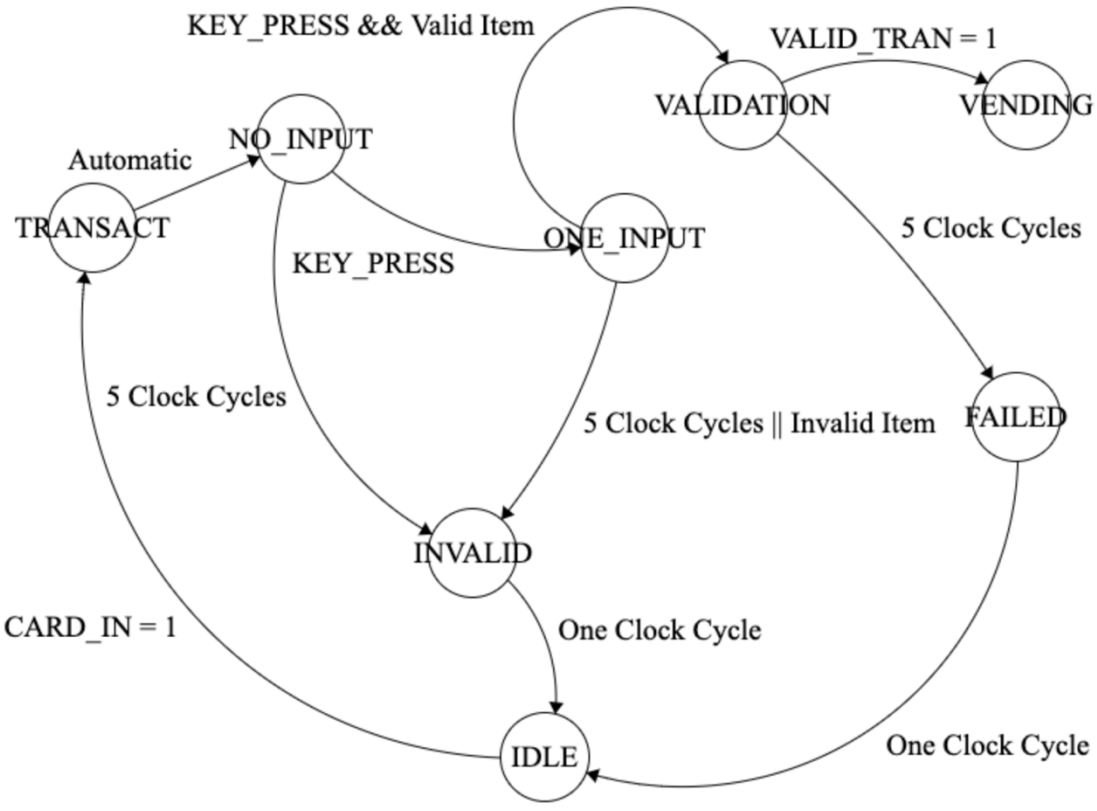

Figure 4. FSM Diagram of the TRANSACT state in Detail

- **VENDING:** In this state, the COST is set to the cost of the corresponding item to vend, and VEND is set to HIGH. The machine can go back to the IDLE state anytime it detects a negative edge on DOOR_OPEN. Otherwise, if DOOR_OPEN isn't set to HIGH within five clock cycles (50ns), the machine automatically goes back to the IDLE state. Do not go to the IDLE state anytime DOOR_OPEN is HIGH.

To implement my module, I first divided it into three submodules, one that checks the states, one that processes the key presses, and one that keeps track of the items in the vending machine. Each submodule consists of one always sequential block and one always combinational block, and I implemented them based on the logic that combinational blocks execute before the sequential blocks.

- The first module has a combinational always block that assigns the next state of the vending machine. The sequential always block pushes the next state to the current state.

- The second submodule has a combinational always block that reads the item code. The sequential always block tracks the transaction timer and the vending timer.

- The final submodule has a combinational always block to check whether the inputted item code is in range and in stock. The sequential always block processes the reload, transact, and vending states.

Once I implemented the submodules, I tested each of them to ensure that they are producing correct behavior. Afterwards, I tested the overall module, only to find several bugs. To resolve these bugs, I combined all three submodules into one submodule, where the intermediate wires now act as registers so that their values transfer across multiple parts.

2.2 RTL Schematic

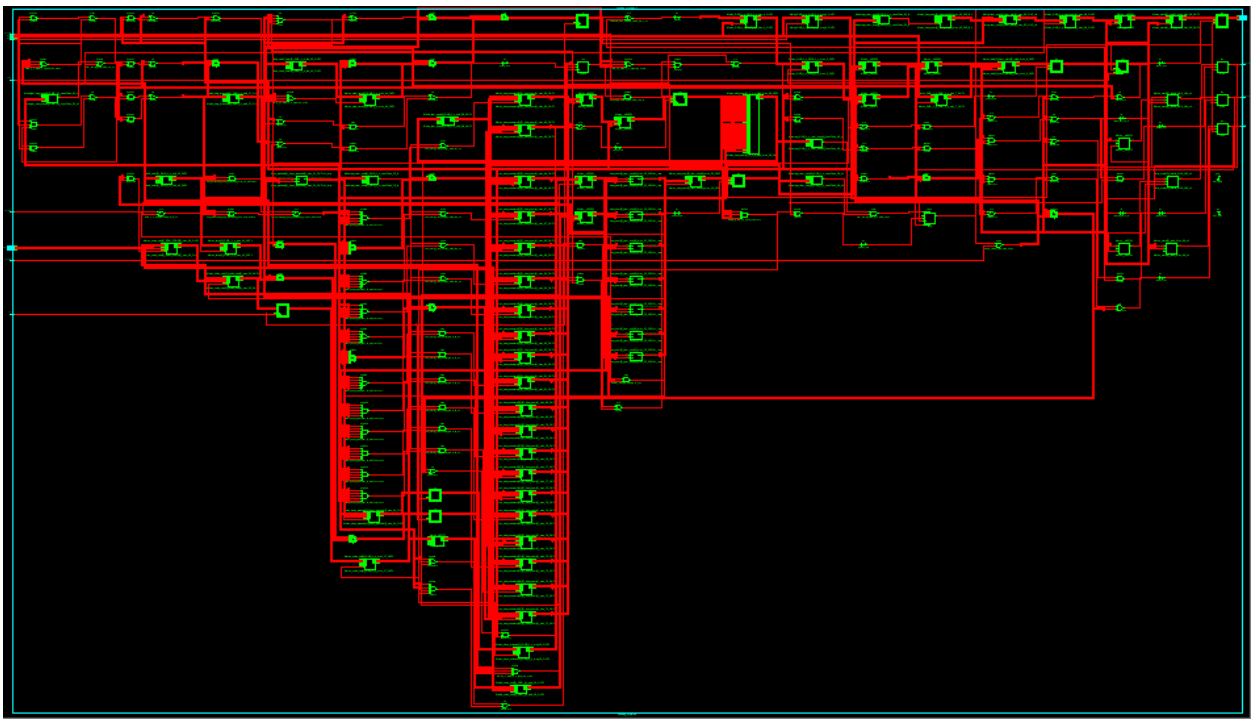

Figure 5. RTL Schematic of the Vending Machine Module

The RTL schematic might seem like a complicated mess. However, a big factor that contributed to this design is the amount of item counters I had to keep track of, since this vending machine has twenty items. This resulted in a column of registers that are displayed in the center left of this schematic. The diagram is filled with multiplexers to select the state and counters to keep track of two different timers. Each of these parts work together to produce the desired output of the vending machine module.

3. Simulation Documentation

I used the Xilinx ISE environment to verify and test the vending machine module for expected behavior, as well as any interesting cases. By the project manual, each clock cycle lasts 10ns. The inputs for this module can happen at any time and the outputs are processed the next time the module detects a positive edge on the CLK signal.

3.1 Verification and Design Tasks

3.1.1 Successful Purchase

This case tests for expected behavior of the vending machine module. The test begins by setting RESET to 1 immediately to initialize the outputs and item counters. After that, I set RESET to 0 to transition to the IDLE state. At 30ns, I set RELOAD to 1 to increase all item counters to 10, and transition back to the IDLE state. At 50ns, I set CARD_IN to 1 to go to the TRANSACT state. Here, I immediately pressed 1 to the module, still at 50ns. I then set KEY_PRESS to 0 at 60ns. At 80ns, I pressed the key again, but with the ITEM_CODE set to 7. The item number should read 17, which is between 0 and 19. Since item 17 has 10 items, the selection succeeds which means the cost output shows 5, the cost of item 17. Vending can happen as soon as the cost displays a nonzero number, as long as I set VALID_TRAN to 1. However, I set VALID_TRAN to 1 back at 20ns, which means the transaction succeeds and immediately goes to the VENDING state, where VEND is set to 1 at 80ns. At the VENDING state, I set DOOR_OPEN to 1 at 90ns and set DOOR_OPEN to 0 at 140ns. Since a negative edge of DOOR_OPEN is detected at 140ns, VEND is set to 0 and goes back to the IDLE state, where all four outputs are set to 0. This completes a successful transaction.

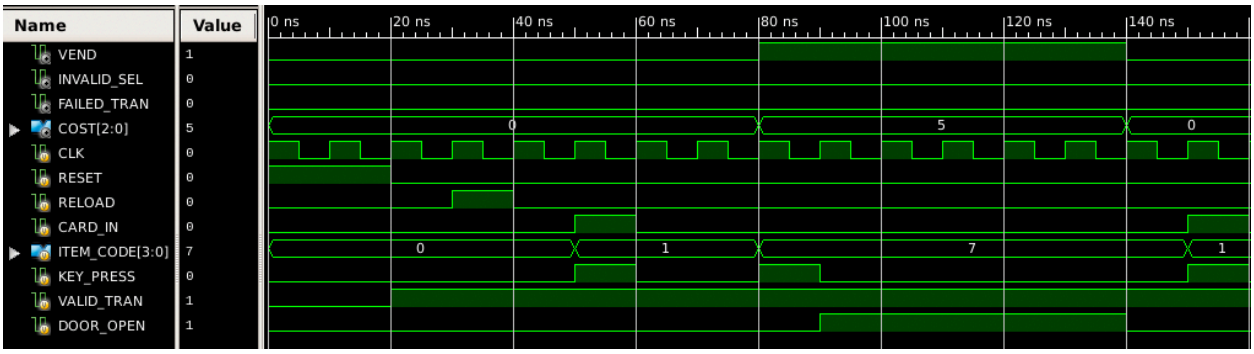

Figure 6. Simulation Waveform of a Successful Purchase

3.1.2 Out of Items

This case tests a transaction where all items are used up. The input process repeats the transaction from test 1 eleven more times, making 12 repetitions total. The last two transaction iterations should result in an INVALID_SEL, since item 17 is out of stock. To easily visualize this, we count ten bumps on VEND followed by two bumps of INVALID_SEL.

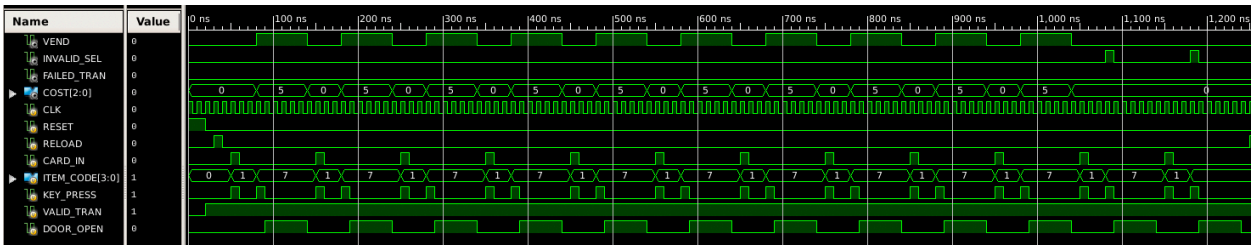

Figure 7. Simulation Waveform of Item Out of Stok after 10 Transactions

Going into more detail, the moment the second key press is inputted at 1080ns, the output INVALID_SEL should be set to 1. That is because while 17 is in the acceptable range, it has ran out of stock, so the INVALID_SEL is set to HIGH before returning to IDLE state.

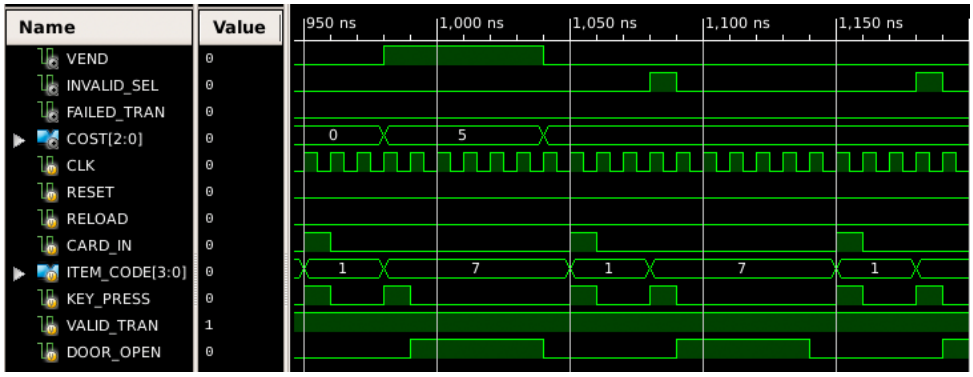

Figure 8. Simulation Waveform Closeup of an Item Out of Stock

3.1.3 No Key Press

This test checks the case where no digit is pressed five cycles after CARD_IN is set to HIGH. At 1270ns, CARD_IN is set to 1, beginning a transaction. Since we see no key presses for the next five clock cycles, an INVALID_SEL rises at the sixth clock cycle at 1320ns, before going back to the IDLE state at 1330ns.

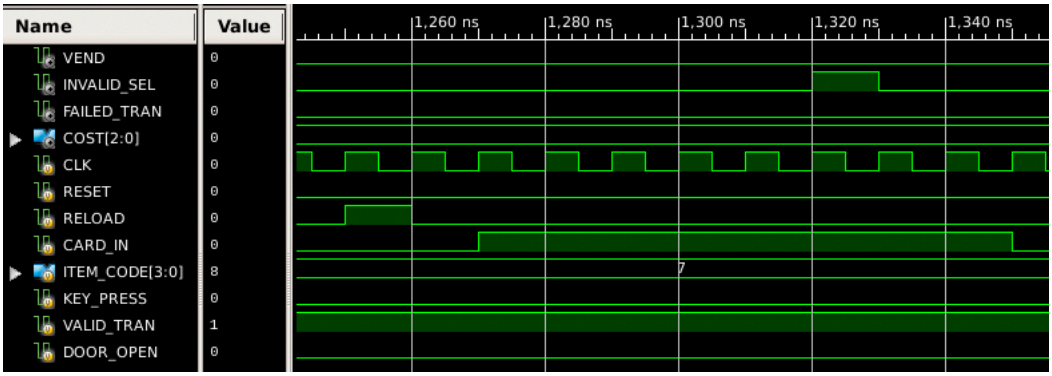

Figure 9. Simulation Waveform of a Transaction with No Key Presses

3.1.4 One Key Press

In truth, the manual mentions that if CARD_IN is still at 1, the next transaction should begin immediately, so at 1330ns, the module is at the TRANSACT state, just that it is processing a new transaction. To simulate one press, I set KEY_PRESS to 1 and ITEM_CODE to 1 at 1360ns, then KEY_PRESS to 0 at 1370ns. After five cycles the INVALID_SEL is set to 1 at 1410ns, before going to the IDLE state at 1420ns. Notice that I set ITEM_CODE to 9 at 1380ns while KEY_PRESS = 0. That number didn't get read, so the INVALID_SEL rises to 1 after five clock cycles, or 50ns.

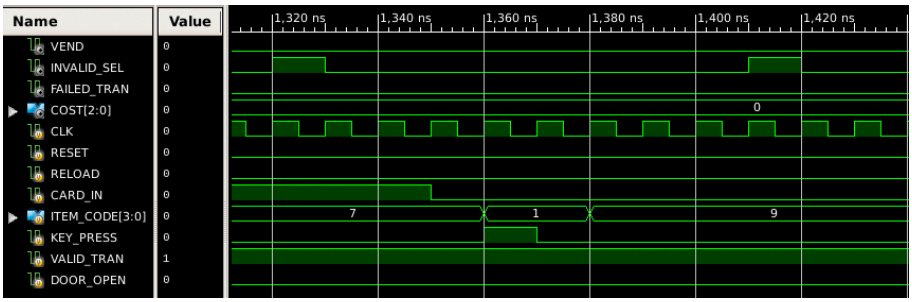

Figure 10. Simulation Waveform of a Transaction with One Key Press

3.1.5 Key Press on IDLE

At the IDLE state, I raised the KEY_PRESS signal to 1 at 1450ns and 1480ns. Since the module is not in the TRANSACT state, nothing happens to the outputs INVALID_SEL and COST and as they remain at 0. This completes the KEY_PRESS while in IDLE state check.

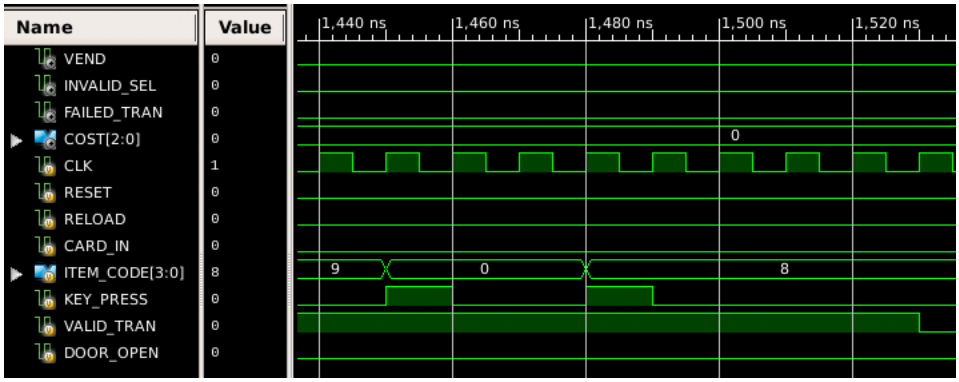

Figure 11. Simulation Waveform of Two Key Presses while in IDLE state

3.1.6 Door Does Not Open

The test simulates a successful transaction, but the door does not open. At 1540ns, I set CARD_IN to 1 and KEY_PRESS to 1 to read the first item code, which is 0. The second input is read at 1570ns where ITEM_CODE is set to 7. This validates the transaction and COST is set to 2, the cost of item 7. At the same time, I raised VALID_TRAN to 1 to immediately signify a successful transaction where the module transitions to the VENDING state. There, VEND is set to 1. After five clock cycles, or 50ns, VEND is set to 0 at 1620ns and the module transitions back to the IDLE state, where COST is also set to 0.

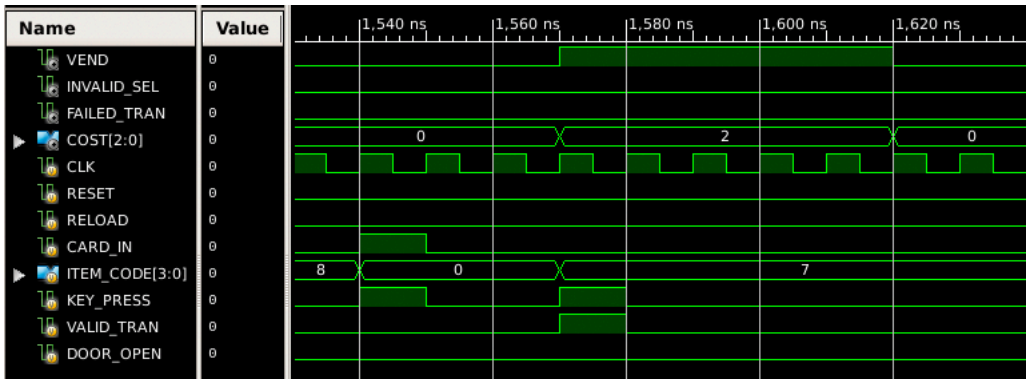

Figure 12. Simulation Waveform of a Successful Transaction, but Door does not Open

3.1.7 VALID_TRAN Not HIGH

This test simulates a failed transaction where the transaction is validated, but VALID_TRAN is not set to HIGH after five clock cycles. At 1660ns, I set CARD_IN to 1 and KEY_PRESS to 1 to read the first item code, which is 0. The second input is read at 1690ns where ITEM_CODE is set to 8. This validates the transaction and COST is set to 3, the cost of item 8. In the waveform below, I did not set VALID_TRAN to 1 at all, so the FAILED_TRAN output is set to 1 at 1740ns, five clock cycles after the second KEY_PRESS. At 1750ns, the module transitions back to the IDLE state, setting FAILED_TRAN and COST to 0.

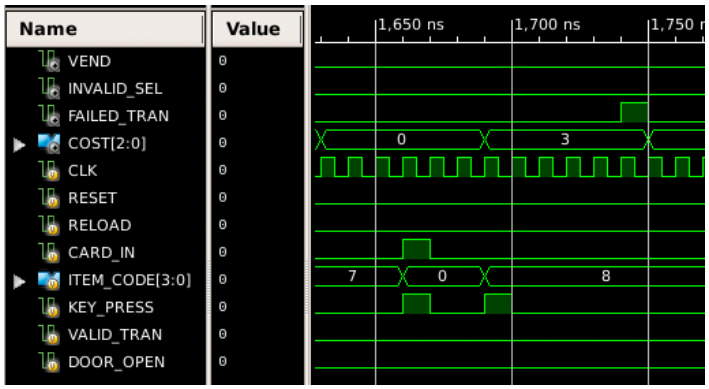

Figure 13. Simulation Waveform of a Failed Transaction

3.1.8 Item Number Out of Range

The final test simulates what happens is the item entered is not one of the numbers between 00-19. At 1780ns, I set CARD_IN to 1 and KEY_PRESS to 1 to read the first item code, which is 2. The second input is read at 1810ns where ITEM_CODE is set to 0. The number read is 20, but there is no such thing as item 20 in this vending machine, so INVALID_SEL is set to 1 immediately after the second KEY_PRESS, also at 1810ns, before returning to the IDLE state one cycle later. Notice that since INVALID_SEL is set to 1, the COST remains at 0, which further signifies an invalid transaction.

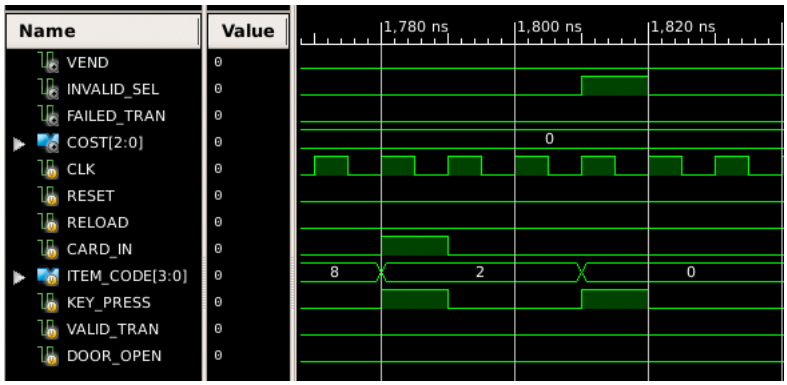

Figure 14. Simulation Waveform of an Item Number that is Out of Range

4. Synthesis and Implementation Map Report

The synthesis and implementation report are attached at the end of this paper. I made sure that my Verilog code is fully synthesizable by showing no errors and no warnings in the synthesis and implementation report. I noticed that if I implemented my combinational code with an always @(*) block, I would generate several warnings regarding the one-bit latch, so I used an always @(posedge CLK) instead to get rid of the warnings.

5. Conclusion

In this lab, I learned how to differentiate between a Moore and Mealy machine, and designed my finite state machine diagram using this knowledge. I originally simplified my process by dividing the module into submodules and later combined it all into one module to keep the outputs consistent. This was because before consulting the TA, I had a bug where the vending machine module would progress one clock cycle after the intended input, but I wanted it to process the input right away. After fixing this bug, I continued building the vending machine until the code is fully synthesizable and it could pass all of the interesting cases mentioned above.

6. Synthesis Report

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

Top Level Output File Name         : vending_machine.ngc

Primitive and Black Box Usage:
------------------------------
# BELS                             : 208
#      GND                         : 1
#      LUT2                        : 4
#      LUT3                        : 6
#      LUT4                        : 11
#      LUT5                        : 83
#      LUT6                        : 92
#      MUXCY                       : 3
#      MUXF7                       : 4
#      XORCY                       : 4
# FlipFlops/Latches                : 118
#      FD                          : 17
#      FDE                         : 97
#      FDR                         : 4
# Clock Buffers                    : 1
#      BUFGP                       : 1
# IO Buffers                       : 16
#      IBUF                        : 10
#      OBUF                        : 6

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

Selected Device : 6slx16csg324-3


Slice Logic Utilization:
 Number of Slice Registers:             118  out of  18224     0%
 Number of Slice LUTs:                  196  out of   9112     2%
    Number used as Logic:               196  out of   9112     2%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:    202
   Number with an unused Flip Flop:      84  out of    202    41%
   Number with an unused LUT:             6  out of    202     2%
   Number of fully used LUT-FF pairs:   112  out of    202    55%
   Number of unique control sets:         6

IO Utilization:
 Number of IOs:                          17
 Number of bonded IOBs:                  17  out of    232     7%

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)  | Load  |
-----------------------------------+------------------------+-------+
CLK                                | BUFGP                  | 118   |
-----------------------------------+------------------------+-------+

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

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

   Minimum period: 5.664ns (Maximum Frequency: 176.543MHz)
   Minimum input arrival time before clock: 4.713ns
   Maximum output required time after clock: 3.820ns
   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: 5.664ns (frequency: 176.543MHz)
  Total number of paths / destination ports: 8913 / 215
-------------------------------------------------------------------------
Delay:               5.664ns (Levels of Logic = 5)
  Source:            item_counters_0_32 (FF)
  Destination:       item_counters_0_3 (FF)
  Source Clock:      CLK rising
  Destination Clock: CLK rising

  Data Path: item_counters_0_32 to item_counters_0_3
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FDE:C->Q              2   0.447   0.981  item_counters_0_32 (item_counters_0_32)
     LUT6:I0->O            1   0.203   0.827  mux_91 (mux_91)
     LUT6:I2->O            1   0.203   0.000  mux_4 (mux_4)
     MUXF7:I0->O          45   0.131   1.477  mux_2_f7
(Msub_item_num[4]_GND_1_o_sub_69_OUT_cy<0>)
     LUT5:I4->O           11   0.205   0.883  Mmux__n04031101 (Mmux__n0403110)
     LUT5:I4->O            1   0.205   0.000  Mmux__n040311 (_n0403<10>)
     FDE:D                     0.102          item_counters_0_70
```

```
    ----------------------------------------
    Total                      5.664ns (1.496ns logic, 4.168ns route)
                                       (26.4% logic, 73.6% route)

==========================================================================
Timing constraint: Default OFFSET IN BEFORE for Clock 'CLK'
  Total number of paths / destination ports: 153 / 123
--------------------------------------------------------------------------
Offset:              4.713ns (Levels of Logic = 4)
  Source:            ITEM_CODE<2> (PAD)
  Destination:       item_num_5 (FF)
  Destination Clock: CLK rising

  Data Path: ITEM_CODE<2> to item_num_5
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             6   1.222   1.089  ITEM_CODE_2_IBUF (ITEM_CODE_2_IBUF)
     LUT5:I0->O            3   0.203   0.879  Maddsub_n0177_Madd_lut<2>1
(Maddsub_n0177_Madd_lut<2>)
     LUT5:I2->O            1   0.205   0.808  Maddsub_n0177_Madd_cy<4>11_SW0 (N23)
     LUT6:I3->O            1   0.205   0.000  Mmux_item_num[6]_item_num[6]_mux_35_OUT61
(item_num[6]_item_num[6]_mux_35_OUT<5>)
     FDE:D                      0.102          item_num_5
    ----------------------------------------
    Total                      4.713ns (1.937ns logic, 2.776ns route)
                                       (41.1% logic, 58.9% route)

==========================================================================
Timing constraint: Default OFFSET OUT AFTER for Clock 'CLK'
  Total number of paths / destination ports: 6 / 6
--------------------------------------------------------------------------
Offset:              3.820ns (Levels of Logic = 1)
  Source:            vend (FF)
  Destination:       VEND (PAD)
  Source Clock:      CLK rising

  Data Path: vend to VEND
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FD:C->Q               8   0.447   0.802  vend (vend)
     OBUF:I->O                 2.571          VEND_OBUF (VEND)
    ----------------------------------------
    Total                      3.820ns (3.018ns logic, 0.802ns route)
                                       (79.0% logic, 21.0% 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            |    5.664|         |         |         |
---------------+---------+---------+---------+---------+
```

```
================================================================================


Total REAL time to Xst completion: 4.00 secs
Total CPU time to Xst completion: 3.88 secs

-->


Total memory usage is 484556 kilobytes

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

7. 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 Feb 28 17:05:54 2021

Design Summary
--------------
Number of errors:      0
Number of warnings:    0
Slice Logic Utilization:
  Number of Slice Registers:                  118 out of  18,224    1%
    Number used as Flip Flops:                118
    Number used as Latches:                     0
    Number used as Latch-thrus:                 0
    Number used as AND/OR logics:               0
  Number of Slice LUTs:                       188 out of   9,112    2%
    Number used as logic:                     188 out of   9,112    2%
      Number using O6 output only:            179
      Number using O5 output only:              0
      Number using O5 and O6:                   9
      Number used as ROM:                       0
    Number used as Memory:                      0 out of   2,176    0%

Slice Logic Distribution:
  Number of occupied Slices:                   54 out of   2,278    2%
  Number of MUXCYs used:                        4 out of   4,556    1%
  Number of LUT Flip Flop pairs used:         192
    Number with an unused Flip Flop:           75 out of     192   39%
    Number with an unused LUT:                  4 out of     192    2%
    Number of fully used LUT-FF pairs:        113 out of     192   58%
    Number of unique control sets:              6
    Number of slice register sites lost
      to control set restrictions:             26 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:                       17 out of     232    7%

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:                   0 out of     248    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:                5.36

Peak Memory Usage:  763 MB
Total REAL time to MAP completion:  5 secs
Total CPU time to MAP completion:   5 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
--------------------

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
---------------------------------
   1 block(s) optimized away

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

Optimized Block(s):
TYPE           BLOCK
GND            XST_GND

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

+----------------------------------------------------------------------------------------------------
-----------------------------------------------------+
| IOB Name                           | Type             | Direction | IO Standard          | Diff
| Drive    | Slew | Reg (s)      | Resistor | IOB      |
|                                    |                  |           |                      | Term
| Strength | Rate |              |          | Delay    |
+----------------------------------------------------------------------------------------------------
-----------------------------------------------------+
| CARD_IN                            | IOB              | INPUT     | LVCMOS25             |
|          |      |              |          |          |
| CLK                                | IOB              | INPUT     | LVCMOS25             |
|          |      |              |          |          |
| COST<0>                            | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |          |
| COST<1>                            | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |          |
| COST<2>                            | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |          |
| DOOR_OPEN                          | IOB              | INPUT     | LVCMOS25             |
|          |      |              |          |          |
| FAILED_TRAN                        | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |          |
| INVALID_SEL                        | IOB              | OUTPUT    | LVCMOS25             |
| 12       | SLOW |              |          |          |
| ITEM_CODE<0>                       | IOB              | INPUT     | LVCMOS25             |
|          |      |              |          |          |
| ITEM_CODE<1>                       | IOB              | INPUT     | LVCMOS25             |
|          |      |              |          |          |
| ITEM_CODE<2>                       | IOB              | INPUT     | LVCMOS25             |
|          |      |              |          |          |
| ITEM_CODE<3>                       | 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       | SLOW |              |          |          |
```

```
+-----------------------------------------------------------------------------------------------------------
-----------------------------------------------------------+

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.
```