Lab3 - Finite State Machine Design: Vending Machine

Introduction and Requirements

In this lab, the goal is to design a fintie state machine (FSM) in Verilog HDL that matches the specified behavior. We need to figure out what states should we have and how the machine transitions from one state to another. The significance of this lab is to learn how to apply FSM design to the real problems in life. For this assignment, we are required to design a vending machine.

The detail of the input of output of the module are as below.

INPUTS:

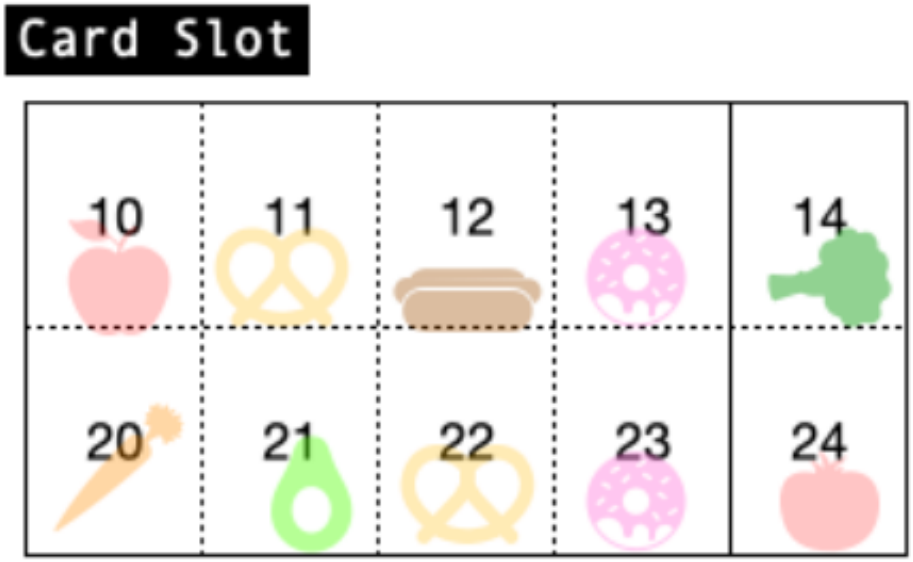
INPUT	SIZE	BEHAVIOR
CLK	1 bit	System clock (T= 10 ns)
RESET	1 bit	Synchronous reset. When high, set all item counters, outputs to 0 and go to the idle state.
RELOAD	1 bit	Reload the machine (set all item counters to 10)
CARD_IN	1 bit	Stays high as long as the card remains inserted.
ITEM_CODE [2:0]	3 bits	Signal to input item code. The 2 digit item code is entered one digit at a time.
KEY_PRESS	1 bit	`ITEM_CODE` is valid for reading when this signal is high.
VALID_TRAN	1bit	If HIGH, the transaction using the card is valid. It can go high any time after item selection is determined to be valid) (card does not need to be inserted when this occurs)

OUTPUTS:

OUTPUT	SIZE	BEHAVIOR
VEND	1 bit	Set to HIGH once the transaction is deemed to be valid. Set to LOW one cycle later.
INVALID_SEL	1 bit	Set to HIGH for 1 cycle if: 1. If 1 digit of ITEM_CODE is entered and there is no 2nd digit after 5 clock cycles 2. If no digit is entered for 5 clock cycles 3. The 2 digit ITEM_CODE is invalid (Ex. 23) 4. The counter for one of the items is 0.
COST [2:0]	3 bits	Set to 000 by default. Set to the cost of an item once item code is entered, and remains at this value until a new transaction begins. (Ex. $5 = 101)
FAILED_TRAN	1 bit	Set to 1 for 1 cycle if `VALID_TRAN` signal does not go high within 5 clock cycles of determining the `ITEM_CODE`

For this assignment, we're tasked with designing a vending machine with the following characteristics:

- Vending machine has 10 different snacks for sale. Each snack has two digit code [10-14, 20-24].
- Each snack is stored in separate slot. There can be up to 10 units of snack stored in 1 slot.
- A buyer can purchase only 1 item at a time.
- The machine only accepts payment by card.

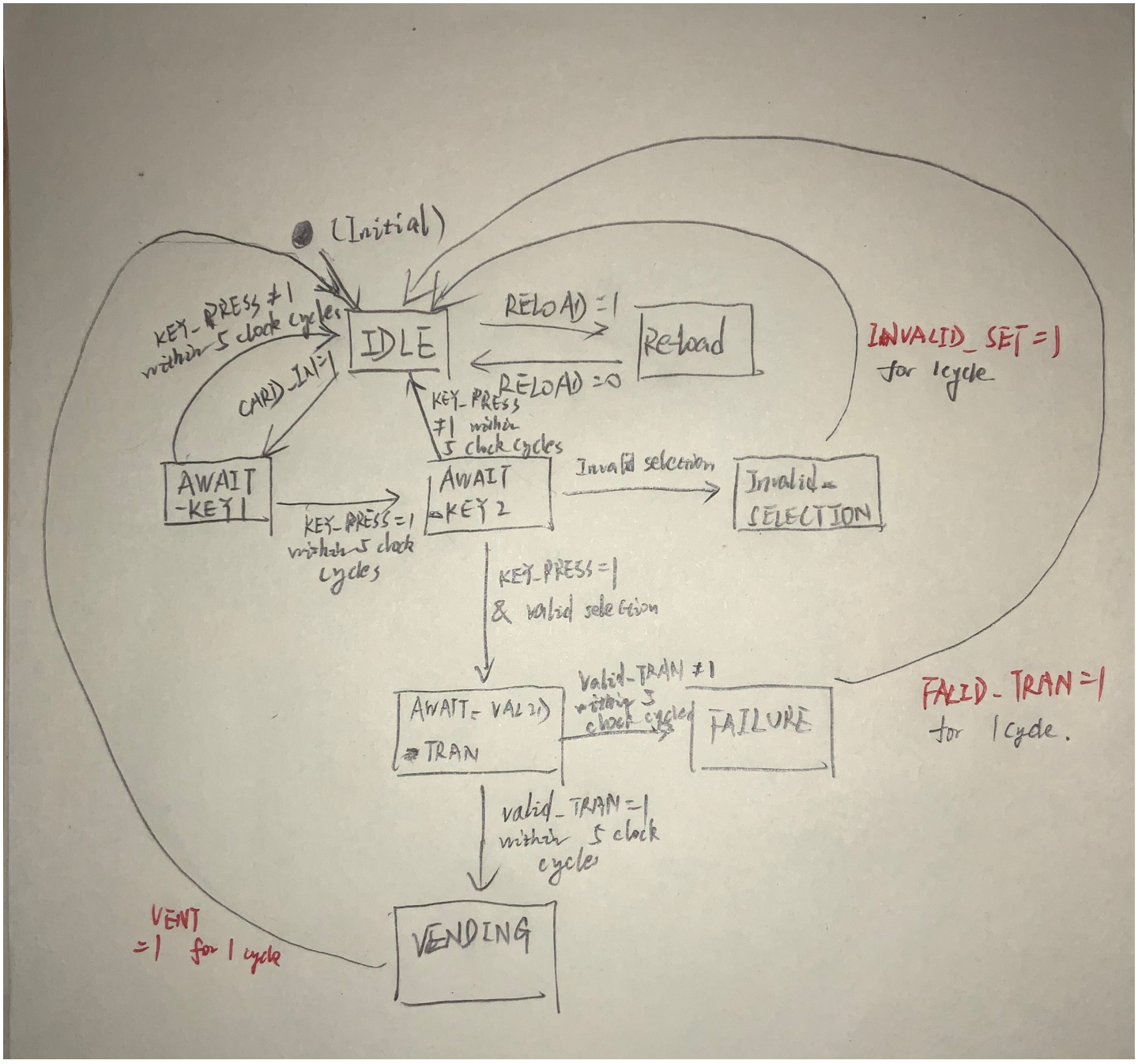


ITEM CODE	COST ($)
10,11,12,13,14	2
20,21,22,23,24	5

Table 1: ITEM CODE vs COST

Design Description (FSM diagram)

Before writing the code, the first step is to draw the FSM diagram to help us to organize the idea to design the module. Basically, I am following the instructioons in the spec. The `IDLE` state is the initial state in the machine, all the other state will return to `IDLE` when the input `reset` =1, and all item counters and outputs are set to 0 in `IDLE` state.



Details and explanations for the states in the diagram.

1. **IDLE** (Initial State) [4'b0001]
    - All the outputs are set to 0.
    - If the `CARD_IN` signal to flip to 1, initiate a new transaction (transit to AWAIT_KEY1 state), or the `RELOAD`

signal flip to 1 to reload the item (transit to RELOADING state).

2. **RELOADING**[4'b0001]
    - All the snack counters are set to 10, i.e. the machine is fully re-loaded
    - The state can only be reached from the IDLE state
    - A new transaction cannot begin when the machine is in the state of RELOADING
3. **AWAIT_KEY1**[4'b0010]:
    - Reached from IDLE state if the CARD_IN is set to 1 (credit card inserted).
    - If KEY_PRESS goes to 1, transist to the `AWAIT_KEY2` state
    - Otherwise if within 5 clock cycles, the KEY_PRESS signal does not go to high, return to the initial state `IDEL`.
4. **AWAIT_KEY2** [4'b0100]: State to check if the selection is valid.
    - If selection is valid (i.e. the code is a number between 10-14 or 20-24 and there are a non-zero number of items corresponding to that code left in the machine), display the $ amount of the selection on `COST[2:0]`, and wait for the `VALID_TRAN` signal (represents a valid/invalid transaction signal from the bank):
        - But the `VALID_TRAN` signal does not go high within 5 clock cycles, the transaction failed. Set the 'FAILED_TRAN' bit to high for 1 cycle, and transist to the `IDLE` state.
        - If the `VALID_TRAN` signal goes to high within the time limit, the state transists to `VENDING` state.
    - If the selection is invalid, set the '`INVALID_SEL`' bit to high for 1 cycle and go to the `IDLE` state
5. **INVALID_SELECTION** [4'b0100]
    - If selection is invalid (input item code out of the range), moved from `AWAIT_KEY_2`.
    - Set the '`INVALID_SET`' to high for 1 cycle and return to IDLE state.
6. **AWAIT_VALID_TRAN** [4'b0101]
    - The state is reached from `AWAIT_KEY2` if the selection is valid (INVALID_SET does not goes to high).
    - Waiting for the `VALID_TRAN` signal.
        - If the signal goes to high within 5 clock cycles, the state transist to `VENDING` state.
        - Otherwise, if the signal does not flip within 5 clock cycles, '`FAILED_TRAN`' bit is set to high for 1 cycle, and return to the `IDLE` state.
7. **FAILURE**[4'b0110]
    - A state that reached from `AWAIT_VALID_TRAN` if the `VALID_TRAN` signal does not goes to high within time limit.
    - Set `FAILED_TRAN` bit to high for 1 cycle, return to `IDLE` state.
8. **VENDING**[4'b0111]
    - Decrement counter of the corresponding item by one (i.e., the item is dispensed)
    - Set `VEND` bit to 1 for 1 cycle
    - Return to IDLE state.

For the coding, I didn't use the template, I only use one always block. I put the updates of the current_state, next_state, and output values, all the logic operation in this block. I assume the KEY_PRESS signal goes high for 1 clock cycle, and the return to low bit. As the way it performs in the real life.

In addition, I set up a time_cnt variable as timer counter to set up the time limit, which is 5 clock cycles.

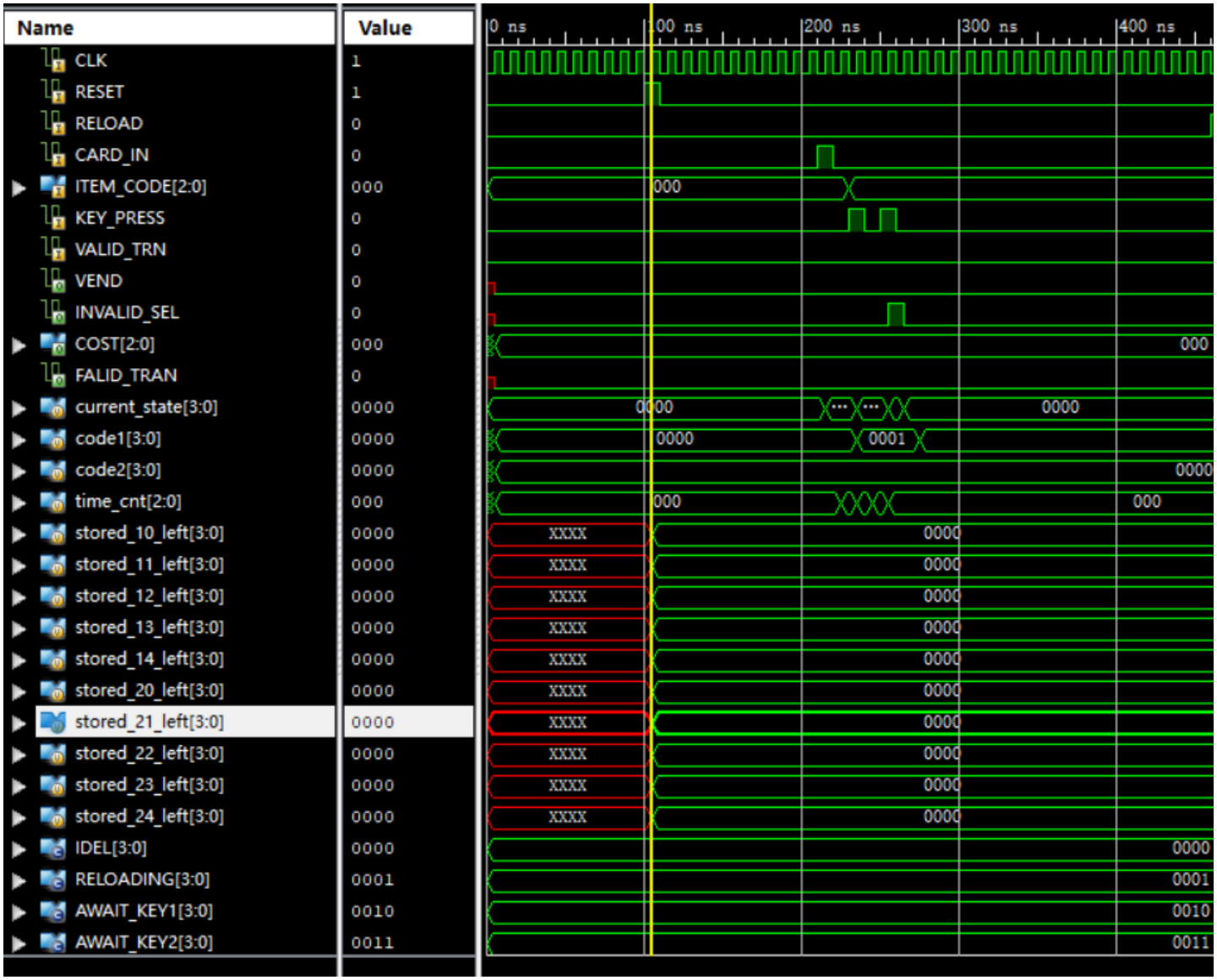
Simulation Documentation

To test the correctness of my code, check if the state transit properly, I created the test bench file and some test cases inside. The below screenshot of the waveforms are basically showing if the variables hold the correct values when they are in the specific states, and the relation between states. And in the specific condition (e.g., no snacks or vend successfully), whether the program outputs properly or not.

1. **Reset, IDLE**

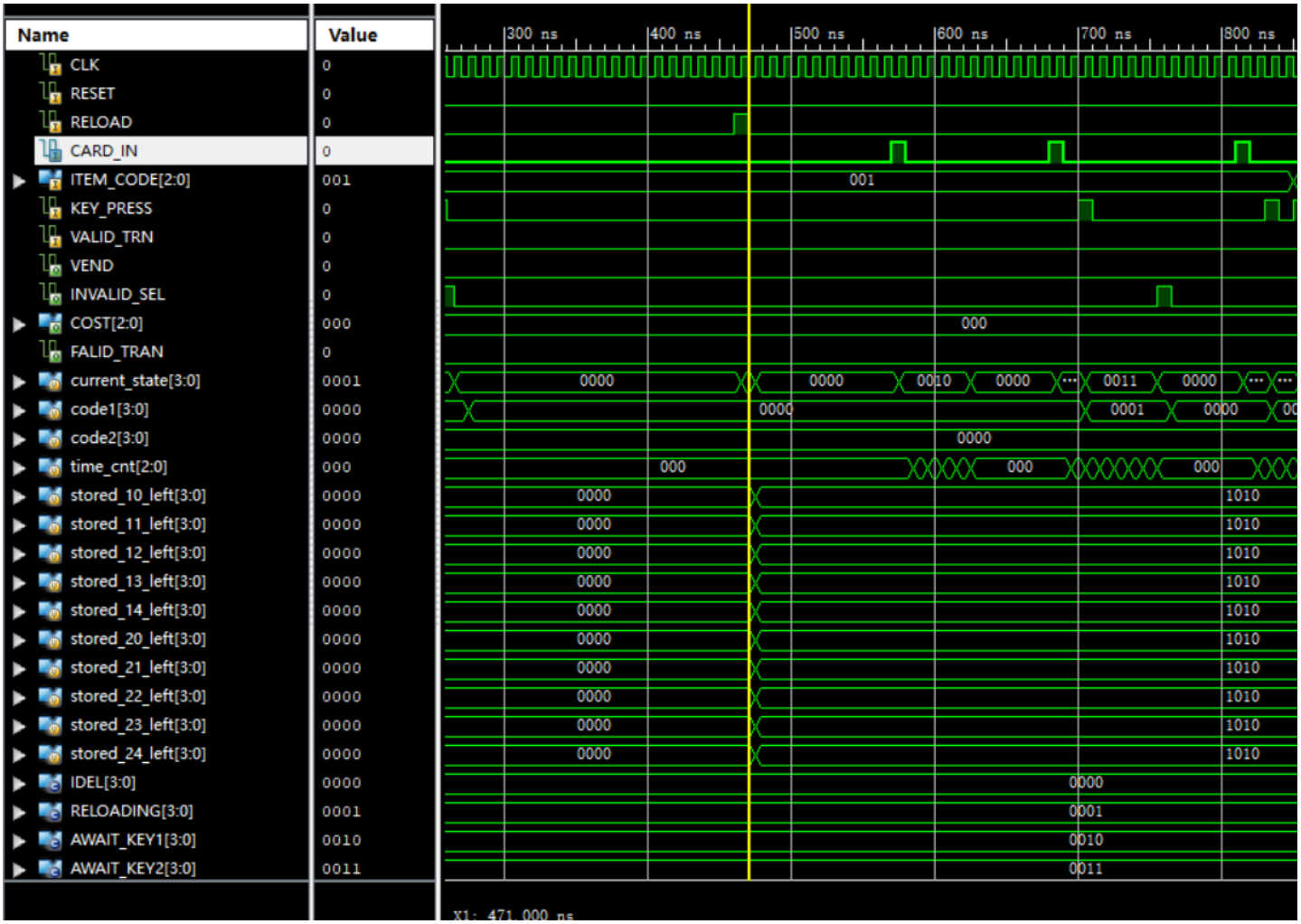
First, check the initial state, IDLE. Through the wavefor, when RESET is 1, we can see that outputs VEND, INVALID_SEL, COST[2:0], FAILED_TRAN hold the value 0, which is correct.

And at the same time, current_state is at 4b'0000 and so as well as IDLE.



2. **Re-load State**

When the input RELOAD is being set to 1, all the snack counters are set to 10 (4b'1010). This action can be indicated in the below waveform screenshot. In addtion, current_state goes to [4b'0001] RELOADING state as expected in the diagram.

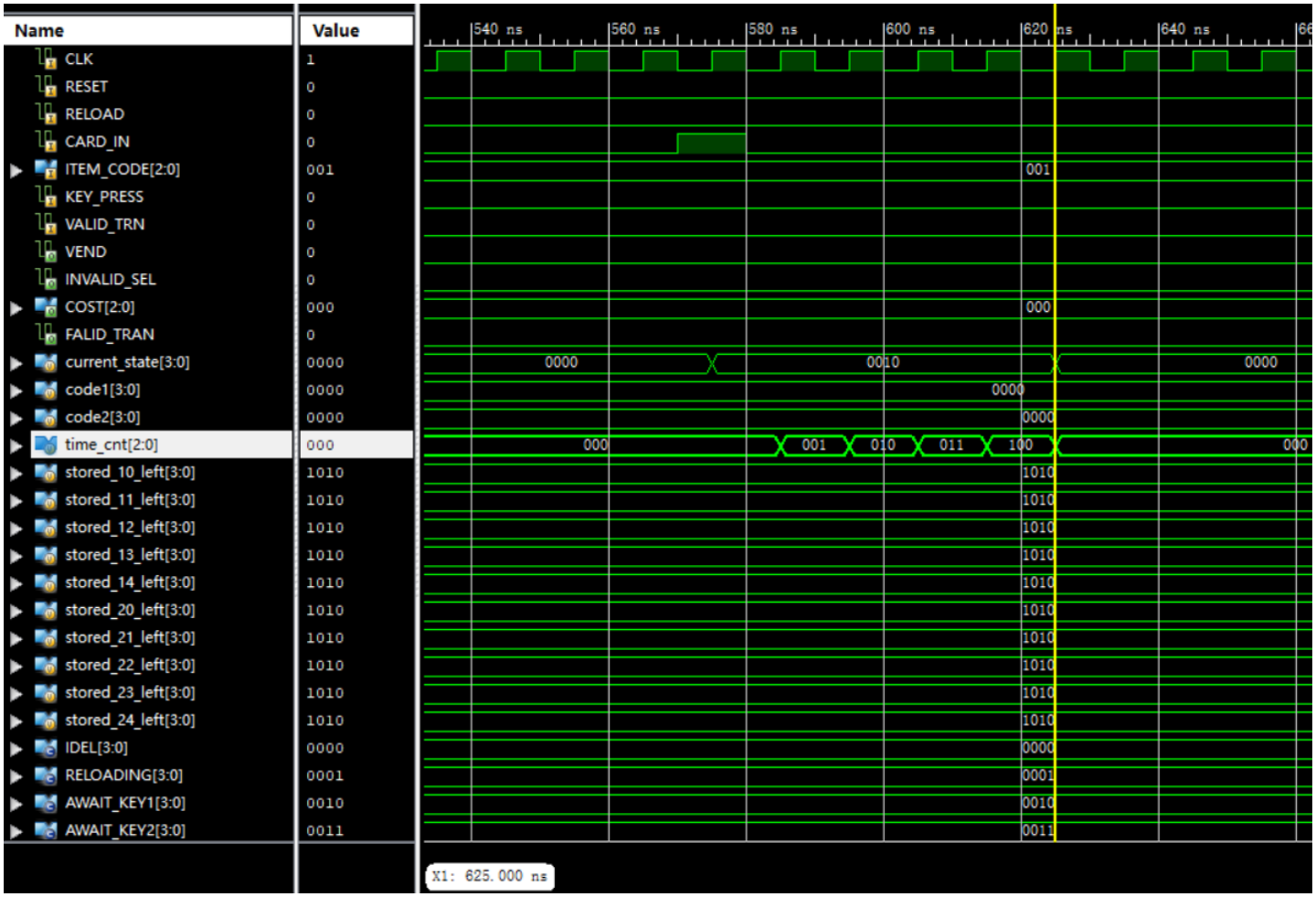


3. **No keypress within 5 clock cycle after insert card**

This test case is to test the specific condition when the card is inserted but no any actions following.

From the below image, we can conclude that when the CARD_IN is set to 1 (credit card insert), the current_state goes to AWAIT_KEY_1 [0010], the timer counter `time_cnt[2:0]` start counting from 0 to 4. And then if the user does not press any item code (i.e., KEY_PRESS = 0) within the 5 clock cycles, the `time_cnt` is reset to 0 and the current_state returns to initial state IDLE [0000].

This result is as expected.

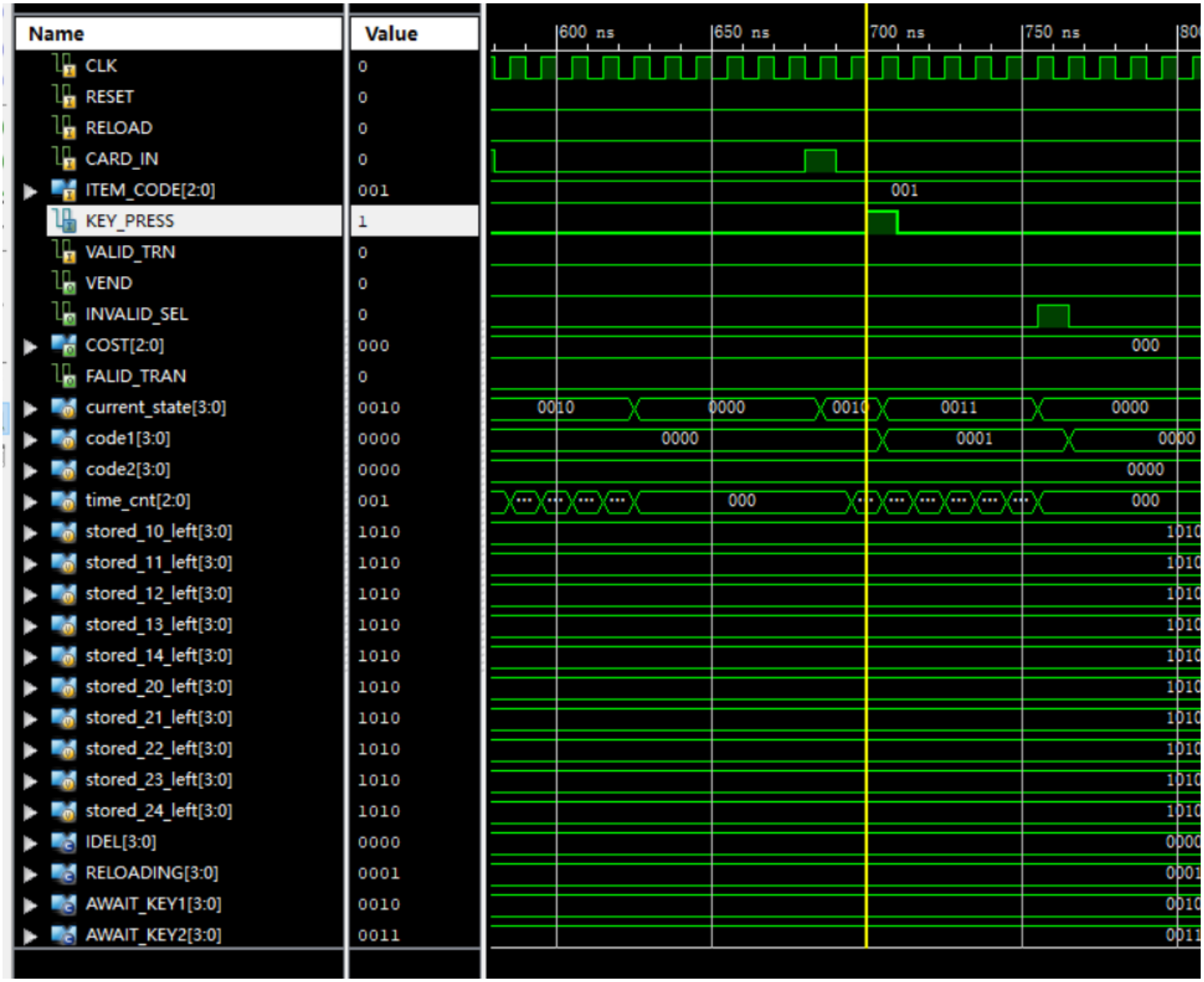


4. **No keypress within 5 clock cycle after enter the first digit code**

If the user have entered the first item digit code, he/she is suppose to enter the second digit code within 5 clock cycles.

Through the below waveforms iamge, it indicates that when the KEY_PRESS is set to 1 and the current state is in AWAIT_KEY1, the state goes to AWAIT_KEY2 [0011]. Then the time_cnt counter start counting from 0 to 4. If the KEY_PRESS does not flip to 1, and `time_cnt` reset to 0 once it reach the 5 clock cycle time limit, the current_state return to initial IDLE state [0000].

This result is as expected.



5. **Invalid Selection**

The items digit code is range from 10-14, 20-24. Any input values outside this range should be consider as invalid.

In my test case, I set a selection value 15 as an invalid input to the vending machine. The first digit is valid, the KEY_PRESS signal goes to high then to low, state move from IDLE to AWAIT_KEY1[0010]. And the second digit 5 enter and it is invalid. At the beginning KEY_PRESS is set to 1 and the state move from AWAIT_KEY1 to AWAIT_KEY2 [0011], but soon it move to INVALID_SEL[0100] and the 'INVALID_SEL' bit is set to high for 1 cycle. Finally return to the initial IDLE state.

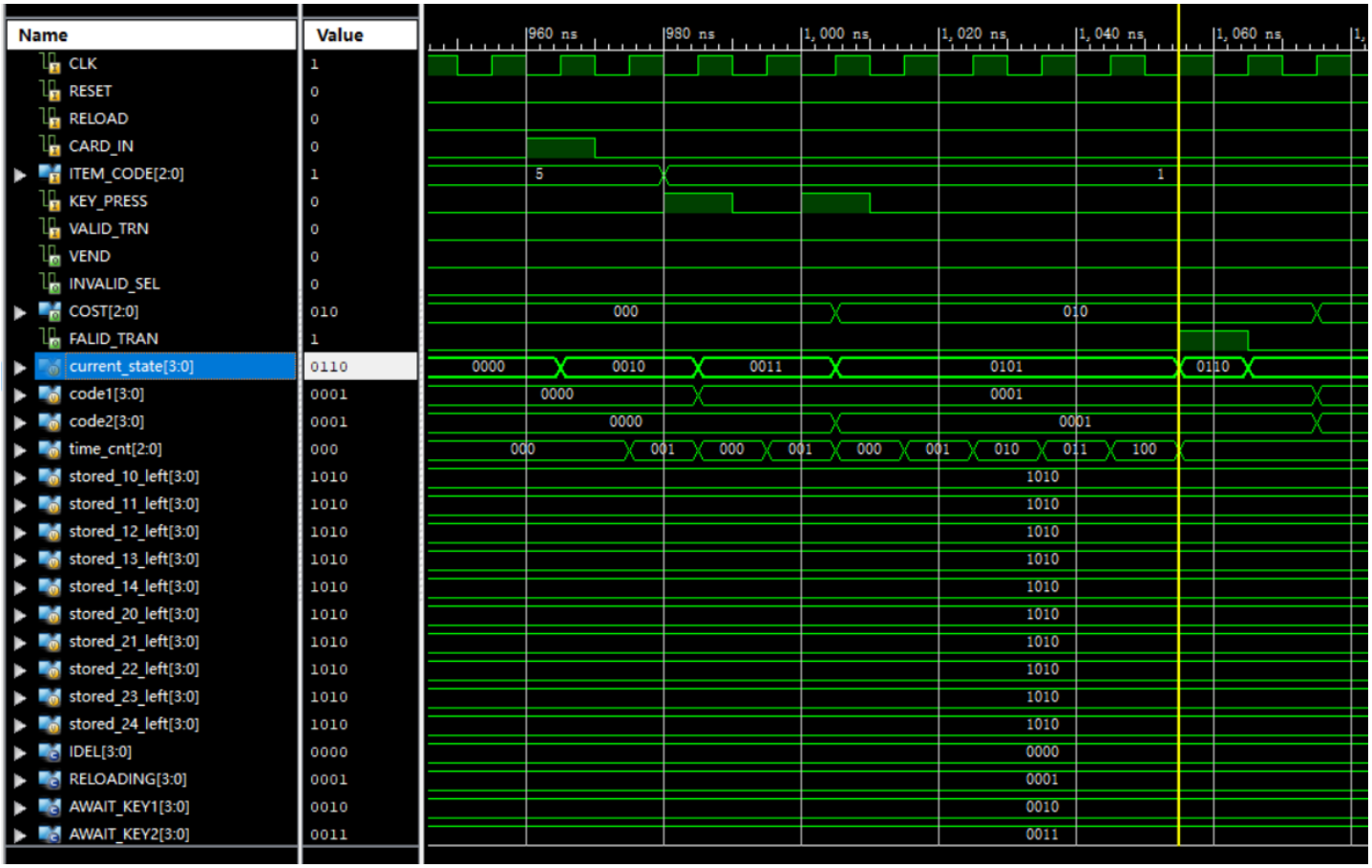
The result is as expected, the test case is passed.

6. **Invalid Transaction**

If the selection is valid. Then the vending machine is suppose to have the valid VALID_TRAN high signal (represesnt a vald transaction signal from the bank) to complete the transaction. If the signal is low, then mean the reuqest transaction is invalid.

Through the below image, when selection is valid, the state transist form AWAIT_KEY2[0011] to AWAIT_VALID_TRAN[0101]. And the VALID_TRIN signal does not filp to 1 within the 5 clock cycles (`time_cnt` reaches its time limit), then the state goes to FAILURE [0110], which means the transaction faile. The 'FAILED_TRAN' bit is set to high for 1 cycle. And finally the state returns to the initial IDLE state.

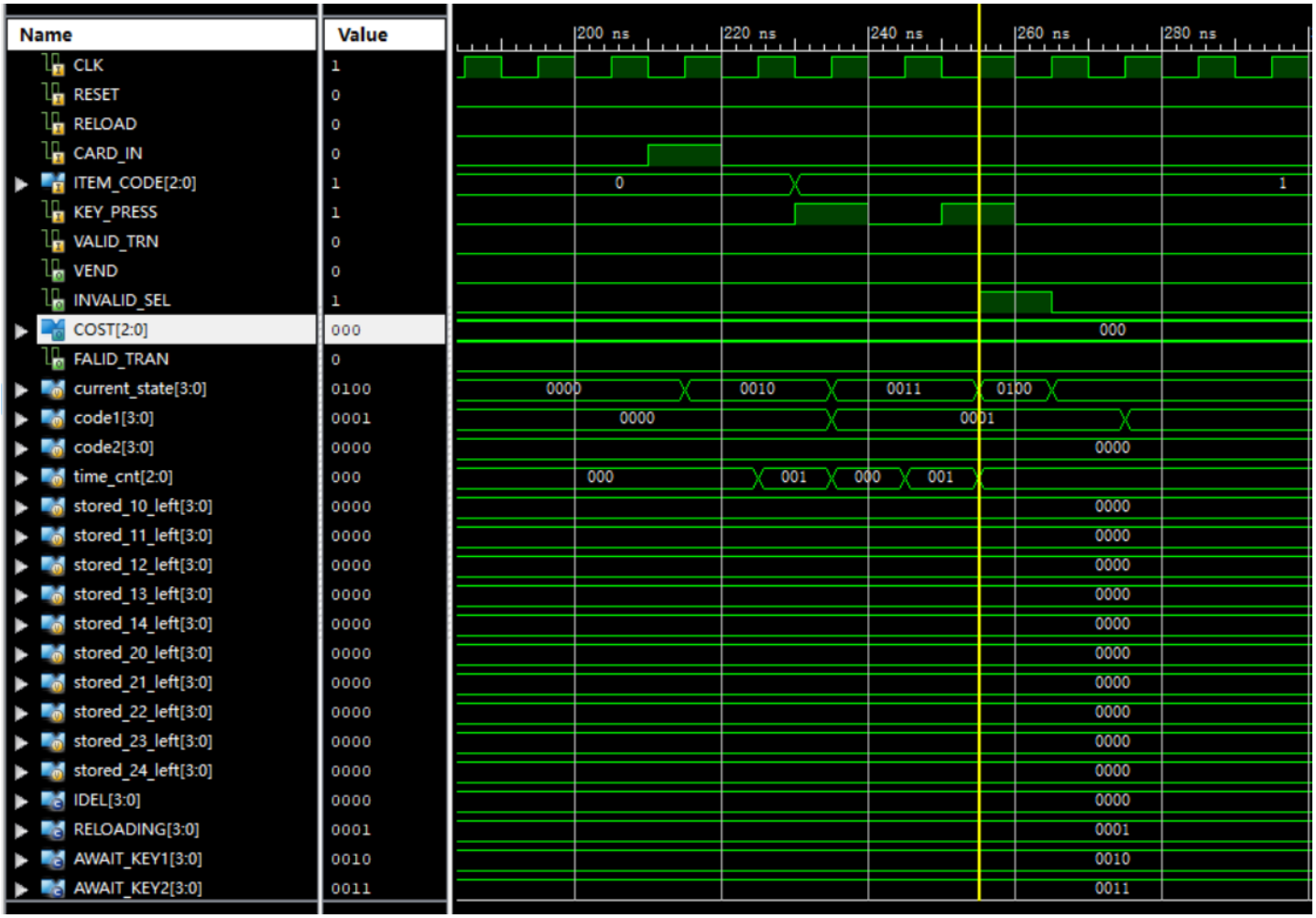
The result is as expected, the test case is passed.



7. **Empty, no snacks**

When there are no snacks in the machine (i.e., all the counter value represent storing each snack is 0), obviously any purchasing attemption would be considered as invalid. The INVALID_SEL bit would be set to high. And the state should be in INVALID_SELECTION [4b'0100].

The result is as expected, the testcase is passed.

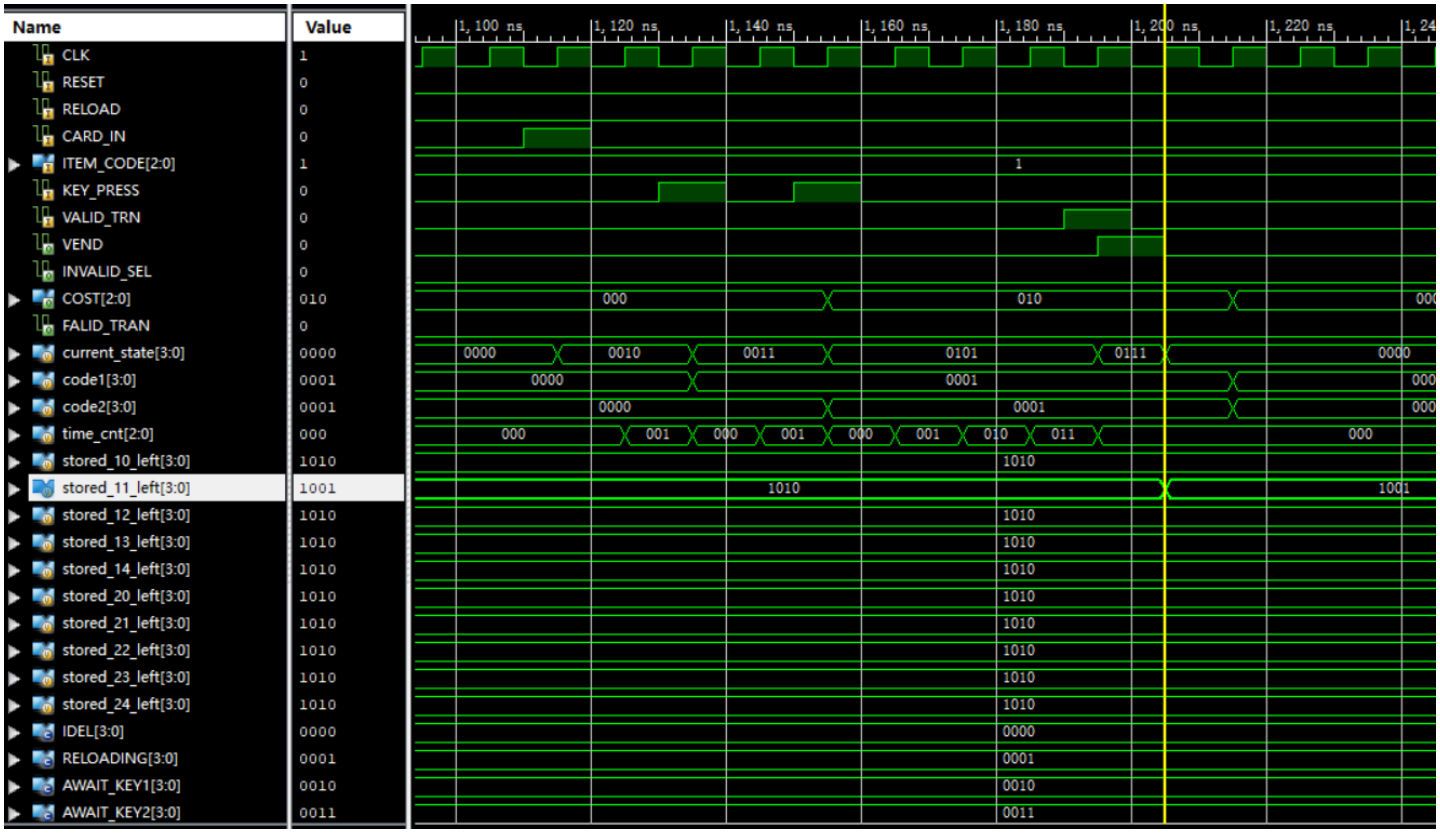


8. **Vend successfully**

This final test case is to test if the program could successfully vend snack. In this case, the VALID_TRN should be 1 and INVALID_SEL is 0. i.e., check if the state VENDING works properly.

The below waveforms screenshot indicates that the transition of current_state from IDLE [0000]-> AWAIT_KEY_1 [0010]-> AWAIT_KEY_2 [0011] -> AWAIT_VALID_TRAN [0101] -> VENDING [0111] and finally return to initial state IDLE[0000], with the corresponding output bits set. Through the image, VENT is set to 1, and stored_11_left is 4'b1001 (9 left) as expected .

The result is as expected, the testcase is passed.



Schematics

The below image is the RTL generated from ISE. It indicate the I/O design of the module I designed.

Through the image, we know the I/O of the module `vending_machine` in my design is exactly the same as shown in Figure 4 in the spec.

Design Summary

Design Summary Report

vending_machine Project Status (05/19/2021 - 22:57:25)			
Project File:	project_3.xise	**Parser Errors:**	No Errors
Module Name:	vending_machine	**Implementation State:**	Placed and Routed
Target Device:	xc6slx16-3csg324	• **Errors:**	No Errors
Product Version:	ISE 14.7	• **Warnings:**	2 Warnings (0 new)
Design Goal:	Balanced	• **Routing Results:**	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	• **Timing Constraints:**	All Constraints Met
Environment:	System Settings	• **Final Timing Score:**	0 (Timing Report)

Device Utilization Summary [-]				
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	60	18,224	1%	
Number used as Flip Flops	60			
Number used as Latches	0			
Number used as Latch-thrus	0			
Number used as AND/OR logics	0			
Number of Slice LUTs	88	9,112	1%	
Number used as logic	88	9,112	1%	
Number using O6 output only	58			
Number using O5 output only	0			
Number using O5 and O6	30			
Number used as ROM	0			
Number used as Memory	0	2,176	0%	
Number of occupied Slices	27	2,278	1%	
Number of MUXCYs used	0	4,556	0%	
Number of LUT Flip Flop pairs used	89			

- Design Overview
  - Summary
  - IOB Properties
  - Module Level Utilization
  - Timing Constraints
  - Pinout Report
  - Clock Report
  - Static Timing
- Errors and Warnings
  - Parser Messages
  - Synthesis Messages
  - Translation Messages
  - Map Messages
  - Place and Route Messages
  - Timing Messages
  - Bitgen Messages
  - All Implementation Messa...
- Detailed Reports
  - Synthesis Report
  - Translation Report
  - Map Report

Design Properties
- [ ] Enable Message Filtering

Optional Design Summary Contents
- [ ] Show Clock Report
- [ ] Show Failing Constraints
- [ ] Show Warnings
- [ ] Show Errors

Device Utilization Summary [-]				
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	60	18,224	1%	
Number used as Flip Flops	60			
Number used as Latches	0			
Number used as Latch-thrus	0			
Number used as AND/OR logics	0			
Number of Slice LUTs	88	9,112	1%	
Number used as logic	88	9,112	1%	
Number using O6 output only	58			
Number using O5 output only	0			
Number using O5 and O6	30			
Number used as ROM	0			
Number used as Memory	0	2,176	0%	
Number of occupied Slices	27	2,278	1%	
Number of MUXCYs used	0	4,556	0%	
Number of LUT Flip Flop pairs used	89			
Number with an unused Flip Flop	49	89	55%	
Number with an unused LUT	1	89	1%	
Number of fully used LUT-FF pairs	39	89	43%	
Number of unique control sets	15			
Number of slice register sites lost to control set restrictions	60	18,224	1%	
Number of bonded IOBs	15	232	6%	
Number of RAMB16BWERs	0	32	0%	
Number of RAMB8BWERs	0	64	0%	

Design Summary

Implementation (Map) 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 23 18:01:13 2021

Design Summary
--------------
Number of errors:      0
Number of warnings:    0
Slice Logic Utilization:
  Number of Slice Registers:                   60 out of  18,224    1%
    Number used as Flip Flops:                 60
    Number used as Latches:                     0
    Number used as Latch-thrus:                 0
    Number used as AND/OR logics:               0
  Number of Slice LUTs:                        88 out of   9,112    1%
    Number used as logic:                      88 out of   9,112    1%
      Number using O6 output only:             58
      Number using O5 output only:              0
      Number using O5 and O6:                  30
      Number used as ROM:                       0
    Number used as Memory:                      0 out of   2,176    0%

Slice Logic Distribution:
  Number of occupied Slices:                   27 out of   2,278    1%
  Number of MUXCYs used:                        0 out of   4,556    0%
  Number of LUT Flip Flop pairs used:          89
    Number with an unused Flip Flop:           49 out of      89   55%
    Number with an unused LUT:                  1 out of      89    1%
    Number of fully used LUT-FF pairs:         39 out of      89   43%
    Number of unique control sets:             15
    Number of slice register sites lost
      to control set restrictions:             60 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:                        15 out of     232    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:                   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:                4.10

Peak Memory Usage:  761 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
---------------------------------

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

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 |              |          |          |
| FALID_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            |
      |          |      |              |          |          |
| KEY_PRESS                        | IOB              | INPUT     | LVCMOS25            |
      |          |      |              |          |          |
| RELOAD                           | IOB              | INPUT     | LVCMOS25            |
      |          |      |              |          |          |
| RESET                            | IOB              | INPUT     | LVCMOS25            |
      |          |      |              |          |          |
| VALID_TRN                        | 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.

```

Synthesis Report

```
Release 14.7 - xst P.20131013 (lin64)
Copyright (c) 1995-2013 Xilinx, Inc.  All rights reserved.
-->
Parameter TMPDIR set to xst/projnav.tmp


Total REAL time to Xst completion: 0.00 secs
Total CPU time to Xst completion: 0.03 secs

-->
Parameter xsthdpdir set to xst


Total REAL time to Xst completion: 0.00 secs
Total CPU time to Xst completion: 0.03 secs

```

```
-->
Reading design: vending_machine.prj

TABLE OF CONTENTS
  1) Synthesis Options Summary
  2) HDL Parsing
  3) HDL Elaboration
  4) HDL Synthesis
       4.1) HDL Synthesis Report
  5) Advanced HDL Synthesis
       5.1) Advanced HDL Synthesis Report
  6) Low Level Synthesis
  7) Partition Report
  8) Design Summary
       8.1) Primitive and Black Box Usage
       8.2) Device utilization summary
       8.3) Partition Resource Summary
       8.4) Timing Report
            8.4.1) Clock Information
            8.4.2) Asynchronous Control Signals Information
            8.4.3) Timing Summary
            8.4.4) Timing Details
            8.4.5) Cross Clock Domains Report


=========================================================================
*                     Synthesis Options Summary                         *
=========================================================================
---- Source Parameters
Input File Name                    : "vending_machine.prj"
Ignore Synthesis Constraint File   : NO

---- Target Parameters
Output File Name                   : "vending_machine"
Output Format                      : NGC
Target Device                      : xc6slx16-3-csg324

---- Source Options
Top Module Name                    : vending_machine
Automatic FSM Extraction           : YES
FSM Encoding Algorithm             : Auto
Safe Implementation                : No
FSM Style                          : LUT
RAM Extraction                     : Yes
RAM Style                          : Auto
ROM Extraction                     : Yes
Shift Register Extraction          : YES
ROM Style                          : Auto
```

```
Resource Sharing                    : YES
Asynchronous To Synchronous         : NO
Shift Register Minimum Size         : 2
Use DSP Block                       : Auto
Automatic Register Balancing        : No

---- Target Options
LUT Combining                       : Auto
Reduce Control Sets                 : Auto
Add IO Buffers                      : YES
Global Maximum Fanout               : 100000
Add Generic Clock Buffer(BUFG)      : 16
Register Duplication                : YES
Optimize Instantiated Primitives    : NO
Use Clock Enable                    : Auto
Use Synchronous Set                 : Auto
Use Synchronous Reset               : Auto
Pack IO Registers into IOBs         : Auto
Equivalent register Removal         : YES

---- General Options
Optimization Goal                   : Speed
Optimization Effort                 : 1
Power Reduction                     : NO
Keep Hierarchy                      : No
Netlist Hierarchy                   : As_Optimized
RTL Output                          : Yes
Global Optimization                 : AllClockNets
Read Cores                          : YES
Write Timing Constraints            : NO
Cross Clock Analysis                : NO
Hierarchy Separator                 : /
Bus Delimiter                       : <>
Case Specifier                      : Maintain
Slice Utilization Ratio             : 100
BRAM Utilization Ratio              : 100
DSP48 Utilization Ratio             : 100
Auto BRAM Packing                   : NO
Slice Utilization Ratio Delta       : 5

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


=========================================================================
*                          HDL Parsing                                  *
=========================================================================
Analyzing Verilog file "/home/ise/152A/vending_machine.v" into library work
Parsing module <vending_machine>.
```

```
113
114  =========================================================================
115  *                           HDL Elaboration                             *
116  =========================================================================
117
118  Elaborating module <vending_machine>.
119
120  =========================================================================
121  *                           HDL Synthesis                               *
122  =========================================================================
123
124  Synthesizing Unit <vending_machine>.
125      Related source file is "/home/ise/152A/vending_machine.v".
126          IDEL = 4'b0000
127          RELOADING = 4'b0001
128          AWAIT_KEY1 = 4'b0010
129          AWAIT_KEY2 = 4'b0011
130          INVALID_SELECTION = 4'b0100
131          AWAIT_VALID_TRAN = 4'b0101
132          FAILURE = 4'b0110
133          VENDING = 4'b0111
134      Found 4-bit register for signal <stored_11_left>.
135      Found 4-bit register for signal <stored_12_left>.
136      Found 4-bit register for signal <stored_13_left>.
137      Found 4-bit register for signal <stored_14_left>.
138      Found 4-bit register for signal <stored_20_left>.
139      Found 4-bit register for signal <stored_21_left>.
140      Found 4-bit register for signal <stored_22_left>.
141      Found 4-bit register for signal <stored_23_left>.
142      Found 4-bit register for signal <stored_24_left>.
143      Found 4-bit register for signal <code1>.
144      Found 4-bit register for signal <code2>.
145      Found 1-bit register for signal <VEND>.
146      Found 1-bit register for signal <INVALID_SEL>.
147      Found 3-bit register for signal <COST>.
148      Found 1-bit register for signal <FALID_TRAN>.
149      Found 3-bit register for signal <time_cnt>.
150      Found 4-bit register for signal <current_state>.
151      Found 4-bit register for signal <stored_10_left>.
152      Found finite state machine <FSM_0> for signal <current_state>.
153      -----------------------------------------------------------------------
154      | States             | 8                                              |
155      | Transitions        | 120                                            |
156      | Inputs             | 22                                             |
157      | Outputs            | 11                                             |
158      | Clock              | CLK (rising_edge)                              |
159      | Reset              | RESET (positive)                               |
160      | Reset type         | synchronous                                    |
```

```
161       | Reset State        | 0000                                          |
162       | Encoding           | auto                                          |
163       | Implementation     | LUT                                           |
164       ----------------------------------------------------------------------
165       Found 4-bit subtractor for signal <stored_10_left[3]_GND_1_o_sub_91_OUT> created at line
      260.
166       Found 4-bit subtractor for signal <stored_11_left[3]_GND_1_o_sub_95_OUT> created at line
      264.
167       Found 4-bit subtractor for signal <stored_12_left[3]_GND_1_o_sub_99_OUT> created at line
      268.
168       Found 4-bit subtractor for signal <stored_13_left[3]_GND_1_o_sub_103_OUT> created at
      line 272.
169       Found 4-bit subtractor for signal <stored_14_left[3]_GND_1_o_sub_107_OUT> created at
      line 276.
170       Found 4-bit subtractor for signal <stored_20_left[3]_GND_1_o_sub_111_OUT> created at
      line 280.
171       Found 4-bit subtractor for signal <stored_21_left[3]_GND_1_o_sub_115_OUT> created at
      line 284.
172       Found 4-bit subtractor for signal <stored_22_left[3]_GND_1_o_sub_119_OUT> created at
      line 288.
173       Found 4-bit subtractor for signal <stored_23_left[3]_GND_1_o_sub_123_OUT> created at
      line 292.
174       Found 4-bit subtractor for signal <stored_24_left[3]_GND_1_o_sub_127_OUT> created at
      line 296.
175       Found 3-bit adder for signal <time_cnt[2]_GND_1_o_add_4_OUT> created at line 140.
176       Found 4-bit comparator greater for signal <GND_1_o_stored_10_left[3]_LessThan_15_o>
      created at line 154
177       Found 4-bit comparator greater for signal <GND_1_o_stored_11_left[3]_LessThan_18_o>
      created at line 160
178       Found 4-bit comparator greater for signal <GND_1_o_stored_12_left[3]_LessThan_21_o>
      created at line 166
179       Found 4-bit comparator greater for signal <GND_1_o_stored_13_left[3]_LessThan_24_o>
      created at line 172
180       Found 4-bit comparator greater for signal <GND_1_o_stored_14_left[3]_LessThan_27_o>
      created at line 178
181       Found 4-bit comparator greater for signal <GND_1_o_stored_20_left[3]_LessThan_30_o>
      created at line 184
182       Found 4-bit comparator greater for signal <GND_1_o_stored_21_left[3]_LessThan_33_o>
      created at line 190
183       Found 4-bit comparator greater for signal <GND_1_o_stored_22_left[3]_LessThan_36_o>
      created at line 196
184       Found 4-bit comparator greater for signal <GND_1_o_stored_23_left[3]_LessThan_39_o>
      created at line 202
185       Found 4-bit comparator greater for signal <GND_1_o_stored_24_left[3]_LessThan_42_o>
      created at line 208
186       Summary:
187     inferred  11 Adder/Subtractor(s).
188     inferred  57 D-type flip-flop(s).
```

```
    inferred  10 Comparator(s).
    inferred  38 Multiplexer(s).
    inferred   1 Finite State Machine(s).
Unit <vending_machine> synthesized.

=========================================================================
HDL Synthesis Report

Macro Statistics
# Adders/Subtractors                                   : 11
 3-bit adder                                           : 1
 4-bit subtractor                                      : 10
# Registers                                            : 17
 1-bit register                                        : 3
 3-bit register                                        : 2
 4-bit register                                        : 12
# Comparators                                          : 10
 4-bit comparator greater                              : 10
# Multiplexers                                         : 38
 1-bit 2-to-1 multiplexer                              : 15
 3-bit 2-to-1 multiplexer                              : 11
 4-bit 2-to-1 multiplexer                              : 12
# FSMs                                                 : 1

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

=========================================================================
*                       Advanced HDL Synthesis                          *
=========================================================================


Synthesizing (advanced) Unit <vending_machine>.
The following registers are absorbed into counter <stored_11_left>: 1 register on signal
<stored_11_left>.
The following registers are absorbed into counter <stored_12_left>: 1 register on signal
<stored_12_left>.
The following registers are absorbed into counter <stored_13_left>: 1 register on signal
<stored_13_left>.
The following registers are absorbed into counter <stored_14_left>: 1 register on signal
<stored_14_left>.
The following registers are absorbed into counter <stored_20_left>: 1 register on signal
<stored_20_left>.
The following registers are absorbed into counter <stored_21_left>: 1 register on signal
<stored_21_left>.
The following registers are absorbed into counter <stored_22_left>: 1 register on signal
<stored_22_left>.
The following registers are absorbed into counter <stored_23_left>: 1 register on signal
<stored_23_left>.
```

```
The following registers are absorbed into counter <stored_24_left>: 1 register on signal <stored_24_left>.
The following registers are absorbed into counter <stored_10_left>: 1 register on signal <stored_10_left>.
Unit <vending_machine> synthesized (advanced).

=========================================================================
Advanced HDL Synthesis Report

Macro Statistics
# Adders/Subtractors                                   : 1
 3-bit adder                                           : 1
# Counters                                             : 10
 4-bit down counter                                    : 10
# Registers                                            : 17
 Flip-Flops                                            : 17
# Comparators                                          : 10
 4-bit comparator greater                              : 10
# Multiplexers                                         : 28
 1-bit 2-to-1 multiplexer                              : 15
 3-bit 2-to-1 multiplexer                              : 11
 4-bit 2-to-1 multiplexer                              : 2
# FSMs                                                 : 1

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

=========================================================================
*                        Low Level Synthesis                            *
=========================================================================
Analyzing FSM <MFsm> for best encoding.
Optimizing FSM <FSM_0> on signal <current_state[1:4]> with user encoding.
-------------------
 State | Encoding
-------------------
 0000  | 0000
 0001  | 0001
 0010  | 0010
 0011  | 0011
 0100  | 0100
 0101  | 0101
 0111  | 0111
 0110  | 0110
-------------------
WARNING:Xst:1710 - FF/Latch <code1_3> (without init value) has a constant value of 0 in block <vending_machine>. This FF/Latch will be trimmed during the optimization process.
WARNING:Xst:1895 - Due to other FF/Latch trimming, FF/Latch <code2_3> (without init value) has a constant value of 0 in block <vending_machine>. This FF/Latch will be trimmed during the optimization process.
```

```
INFO:Xst:2261 - The FF/Latch <COST_0> in Unit <vending_machine> is equivalent to the following FF/Latch, which will be removed : <COST_2>

Optimizing unit <vending_machine> ...

Mapping all equations...
Building and optimizing final netlist ...
Found area constraint ratio of 100 (+ 5) on block vending_machine, actual ratio is 1.
FlipFlop current_state_FSM_FFd2 has been replicated 2 time(s)
FlipFlop current_state_FSM_FFd3 has been replicated 1 time(s)

Final Macro Processing ...

=========================================================================
Final Register Report

Macro Statistics
# Registers                                          : 60
 Flip-Flops                                          : 60

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

=========================================================================
*                           Partition Report                            *
=========================================================================

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

  No Partitions were found in this design.

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

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

Top Level Output File Name         : vending_machine.ngc

Primitive and Black Box Usage:
------------------------------
# BELS                             : 122
#      INV                         : 1
#      LUT2                        : 18
#      LUT3                        : 18
#      LUT4                        : 28
#      LUT5                        : 27
#      LUT6                        : 26
```

```
#      MUXF7                       : 4
# FlipFlops/Latches                : 60
#      FDR                         : 11
#      FDRE                        : 49
# Clock Buffers                    : 1
#      BUFGP                       : 1
# IO Buffers                       : 14
#      IBUF                        : 8
#      OBUF                        : 6

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

Selected Device : 6slx16csg324-3


Slice Logic Utilization:
 Number of Slice Registers:             60  out of  18224     0%
 Number of Slice LUTs:                 118  out of   9112     1%
    Number used as Logic:              118  out of   9112     1%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:   121
   Number with an unused Flip Flop:     61  out of    121    50%
   Number with an unused LUT:            3  out of    121     2%
   Number of fully used LUT-FF pairs:   57  out of    121    47%
   Number of unique control sets:       15

IO Utilization:
 Number of IOs:                         15
 Number of bonded IOBs:                 15  out of    232     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)  | Load  |
-----------------------------------+------------------------+-------+
CLK                                | BUFGP                  | 60    |
-----------------------------------+------------------------+-------+

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

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

   Minimum period: 4.558ns (Maximum Frequency: 219.373MHz)
   Minimum input arrival time before clock: 5.706ns
   Maximum output required time after clock: 3.732ns
   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: 4.558ns (frequency: 219.373MHz)
  Total number of paths / destination ports: 1358 / 112
-------------------------------------------------------------------------
Delay:               4.558ns (Levels of Logic = 4)
  Source:            code1_2 (FF)
  Destination:       INVALID_SEL (FF)
  Source Clock:      CLK rising
  Destination Clock: CLK rising

  Data Path: code1_2 to INVALID_SEL
                                Gate     Net
     Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
     ----------------------------------------  ------------
      FDRE:C->Q             5   0.447   0.943  code1_2 (code1_2)
      LUT4:I1->O            6   0.205   0.745  code1[3]_GND_1_o_AND_12_o11
(code1[3]_GND_1_o_AND_12_o1)
      LUT4:I3->O            1   0.205   0.924  code1[3]_GND_1_o_AND_12_o1
(code1[3]_GND_1_o_AND_12_o)
```

```
     LUT6:I1->O            1   0.203   0.580  current_state[3]_GND_1_o_Select_196_o<3>1
(current_state[3]_GND_1_o_Select_196_o<3>)
     LUT6:I5->O            1   0.205   0.000  current_state[3]_GND_1_o_Select_196_o<3>3
(current_state[3]_GND_1_o_Select_196_o)
     FDR:D                     0.102          INVALID_SEL
    ----------------------------------------
    Total                      4.558ns (1.367ns logic, 3.191ns route)
                                       (30.0% logic, 70.0% route)

==========================================================================
Timing constraint: Default OFFSET IN BEFORE for Clock 'CLK'
  Total number of paths / destination ports: 216 / 82
--------------------------------------------------------------------------
Offset:              5.706ns (Levels of Logic = 5)
  Source:            ITEM_CODE<1> (PAD)
  Destination:       INVALID_SEL (FF)
  Destination Clock: CLK rising

  Data Path: ITEM_CODE<1> to INVALID_SEL
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O            11   1.222   1.111  ITEM_CODE_1_IBUF (ITEM_CODE_1_IBUF)
     LUT3:I0->O            1   0.205   0.924  Mmux__n1172211 (Mmux__n1172211)
     LUT5:I0->O            1   0.203   0.684  Mmux__n1172212 (Mmux__n1172212)
     LUT6:I4->O            6   0.203   0.849  Mmux__n1172214 (Mmux__n117221)
     LUT6:I4->O            1   0.203   0.000  current_state[3]_GND_1_o_Select_196_o<3>3
(current_state[3]_GND_1_o_Select_196_o)
     FDR:D                     0.102          INVALID_SEL
    ----------------------------------------
    Total                      5.706ns (2.138ns logic, 3.568ns route)
                                       (37.5% logic, 62.5% route)

==========================================================================
Timing constraint: Default OFFSET OUT AFTER for Clock 'CLK'
  Total number of paths / destination ports: 6 / 6
--------------------------------------------------------------------------
Offset:              3.732ns (Levels of Logic = 1)
  Source:            COST_0 (FF)
  Destination:       COST<2> (PAD)
  Source Clock:      CLK rising

  Data Path: COST_0 to COST<2>
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FDR:C->Q              5   0.447   0.714  COST_0 (COST_0)
     OBUF:I->O                 2.571          COST_2_OBUF (COST<2>)
```

```
    ----------------------------------------
    Total                     3.732ns (3.018ns logic, 0.714ns route)
                                      (80.9% logic, 19.1% 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            |    4.558|         |         |         |
---------------+---------+---------+---------+---------+

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


Total REAL time to Xst completion: 6.00 secs
Total CPU time to Xst completion: 5.04 secs

-->


Total memory usage is 484704 kilobytes

Number of errors   :    0 (   0 filtered)
Number of warnings :    2 (   0 filtered)
Number of infos    :    1 (   0 filtered)
```

And from the design summary report, I could see the Macro Statistics section about the number of differnt logic gate in my design. And I can check how my program performs through the report.

Conclusion

This project is more harder than the last two projects. The module design is complex. How to design a FSM and apply to real life system is challange but it is also useful.

In this project, I designed and implemented the Vending Machine, Final State Machine according to the behavior in the project manuscript. To correctly implement this, I first draw the FSM diagram to help me to have the clear idea that what output should be in each state and how to transit to a different state.

One of the major difficulty I encountered in this lab was 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.