INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

IJARIIT

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 7, Issue 4 - V7I4-1160)
Available online at: https://www.ijariit.com

Design and Implementation of a Vending Machine and its performance evaluation using EDA tools

Anand Jonathan Roy
royanandj@gmail.com
SRM Institute of Science and Technology, Chennai, Tamil Nadu

Manish Kumar
manish04.mu@gmail.com
SRM Institute of Science and Technology, Chennai, Tamil Nadu

K. Suganthi
suganthk@srmist.edu.in
SRM Institute of Science and Technology, Chennai, Tamil Nadu

ABSTRACT

In today's world, vending machines have become an extension of the physical and online stores and are evolving as one of the largest consumption modes. Nowadays, you will be able to see them in malls, offices, stations, parks, etc. These machines are rapidly becoming a great part of one's shopping experience. This is the reason why a lot of people are looking forward to investing in them and earning money with this very convenient business opportunity. This project aims to design a common vending machine with an additional feature (reverse vending machine) which enables a consumer to recycle the container/wrapper of the item purchased by them for which appropriate incentives are returned. It offers the person to carry out their transaction via cash/card /online payment modes. Unique QR codes/Barcodes are associated with each product to have information about the product and its cost. The reverse vending machine consists of a scanner that scans the code of the product which is inserted into the machine and completes the transaction accordingly. The vending machine is programmed using Verilog HDL and is simulated on ModelSim-Intel FPGA Edition Software. Chip planning, timing, power, and various other design and performance analysis are carried out using Intel Quartus Prime Software Suite.

Keywords— *Vending Machine, Verilog HDL, Reverse Vending Machine, FPGA, Performance Characterization, Slack, HPWL, Quartus Prime*

1 INTRODUCTION

A vending machine is an automatic machine which is used to provide the users with an assorted selection of products: food items, newspapers, magazines, tickets, tokens etc. A vending machine dispenses a product to the users based on the money inserted and choice of the product. It consists of simple automated systems which aids to automate the entire process. Vending machines aka unstaffed retail shops have developed recently and been prominently altering our shopping styles. Recently there have been many businesses which have opted to use vending machines as an extended business opportunity and have described optimistic reviews. These machines enable businesses to expand to serve customers 24/7 which is great asset for both the businessman and the customer. The machine can be compared to online shopping, and is widely used. With the increasing popularity of vending machines, the level of experience is likely to increase. In this paper we propose to develop a vending machine which will give a user the options to pay via three payment modes, namely cash, card and online modes (UPI). Along with the generic characteristics of a vending machine, we propose to incorporate additional functionality in form of a reverse vending machine, which is a concept of recycling which will help us make the surroundings cleaner. The basic design of the model is based on a Moore type FSM (Finite State Machine) in which each state denotes the inserted amount into the machine. Appropriate signals are designated for various actuators in the code. The code is written in Verilog HDL and simulation is performed on ModelSim. A detailed analysis.

2 LITERATURE SURVEY

Numerous researches have been conducted to design the Vending Machines. There have been articles that talk about the architecture, design, and the various enhancements that have been/or can be incorporated into the design and implementation of a vending machine. Predominantly, we have seen most automatic machines operate with a microcontroller-based system at their core which has proven to work seamlessly, but there is only a level of optimization and customization that can be achieved. After looking at the literature reviews in various research articles, some notable inferences were drawn. The comparisons between microcontroller-based devices and FPGA-based devices (or ASICs) made us conclude that FPGA-based devices will allow us to attain a more dynamic system, where we can design the circuit in a way such that we can optimize every parameter as per the requirements. The FPGA design can be optimized by reviewing and adjusting the architecture to improve slack and other timing constraints. To further improve timing, the placement phase in the FPGA design is looked into. In this paper, we will be implementing the HPWL (Half Perimeter Wire Length) model using the analytical method to improve the placements of the blocks to minimize wire length. There is also a lot of enhancements that have been brought into the generic vending machine, such as, the mobile vending machines which can be tracked via an application to find the nearest from any given location, the option of expanding or reducing the number of items for sale, the deployment of a recycling machine in tandem with the vending machine, the increased number of payment options, etc. Therefore, in this paper, we will design a dynamic vending machine with three payment options, integrate a reverse vending machine. The model will be analysed to check various factors such as space, time, speed, power dissipation, and other parameters using EDA tools.

3 SYSTEM ANALYSIS

3.1 Slack Analysis

Slack of a particular gate is defined as the difference between required arrival time arrival time.

- When the data arrives before the time at which it is required, it results in a positive setup slack. This means that the design is functioning within the specified frequency and it has some more timing margin as well.
- When the data does not arrive before the time at which it is required, it results in a negative setup slack. This means that the design doesn't comply to the constrained frequency and timing, which results into a setup violation.

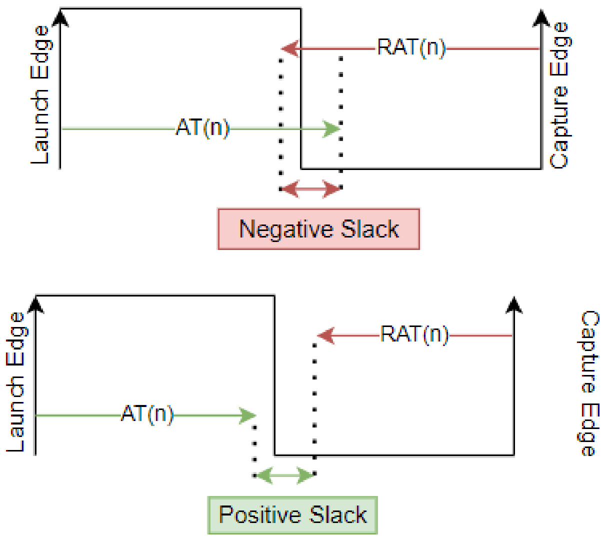


Figure 1: Slack with respect to launch and capture edge of clock cycle

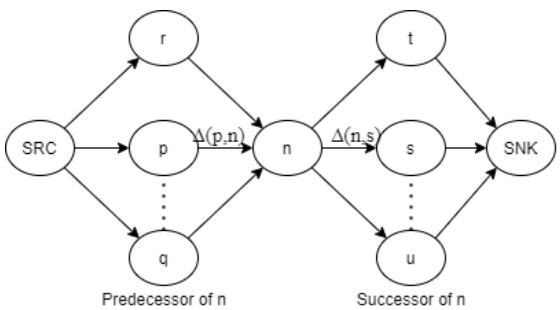


Figure 2: Slack at node 'n'

$$\mathrm{RAT(n)} = \begin{cases} \mathit{Clock\ Period}, & \mathit{if}\ n = SNK \\ \min\left(RAT(s) - \Delta(n,s)\right), & \mathit{otherwise} \end{cases}$$

$$\mathrm{AT(n)} = \begin{cases} 0, & \mathit{if}\ n = SNK \\ \max\left(AT(p) - \Delta(p,n)\right), & \mathit{otherwise} \end{cases}$$

3.2 HPWL Mathematical Modelling

The Half Perimeter Wire Length is calculated using Quadratic Length method, which is used to estimate the wire length. Here the gates positions are assumed in 2-D space (x and y coordinates). Then the distance between the two gates and the gates and connected pads is minimized.

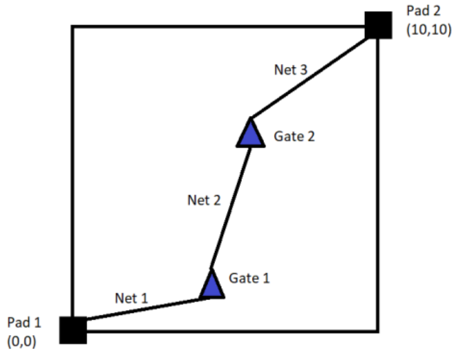


Figure 3: Example of a technology map view

In the above figure there are 2 gates, 2 pads and 3 nets.

Gate 1 = (x_1, y_1)
Gate 2 = (x_2, y_2)
Pad 1 = (0, 0)
Pad 2 = (1, 1)

3.2.1 Basic Calculation Carried out Manually for Calculating HPWL

Now length of Net 1 = $(x_1 - 0)^2 + (y_1 - 0)^2$
Now length of Net 2 = $(x_1 - x_2)^2 + (y_1 - y_2)^2$
Now length of Net 3 = $(x_2 - 1)^2 + (y_2 - 1)^2$

Now, separating x and y terms
$Q(x) = (x_1 - 0)^2 + (x_1 - x_2)^2 + (x_2 - 1)^2$
$Q(y) = (y_1 - 0)^2 + (y_1 - y_2)^2 + (y_2 - 1)^2$

Now minimizing the net length

$$\frac{\partial Q(x)}{\partial x_1} = \frac{\partial}{\partial x_1}(x_1 - 0)^2 + (x_1 - x_2)^2 + (x_2 - 1)^2 \rightarrow \mathbf{4x_1 - 2x_2 = 0}$$

$$\frac{\partial Q(x)}{\partial x_2} = \frac{\partial}{\partial x_2}(x_1 - 0)^2 + (x_1 - x_2)^2 + (x_2 - 1)^2 \rightarrow \mathbf{-2x_1 + 4x_2 = 2}$$

$$\frac{\partial Q(y)}{\partial y_1} = \frac{\partial}{\partial y_1}(y_1 - 0)^2 + (y_1 - y_2)^2 + (y_2 - 1)^2 \rightarrow \mathbf{4y_1 - 2y_2 = 0}$$

$$\frac{\partial Q(y)}{\partial y_2} = \frac{\partial}{\partial y_2}(y_1 - 0)^2 + (y_1 - y_2)^2 + (y_2 - 1)^2 \rightarrow \mathbf{-2y_1 + 4y_2 = 2}$$

Solving these equations,
we get **(x1 = 0.33, y1 = 0.33) & (x2 = 0.66, y2 = 0.66)**

3.2.2 Computational Method

First, we create an auxiliary matrix (A) which contains all the coefficient of the variables used, then we create the *Bx* and *By* matrix.

$$C = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \rightarrow C^* = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$$

C matrix represents which gate is connected to which gate.

$$\mathbf{A} = \begin{bmatrix} \mathbf{2} & \mathbf{-1} \\ \mathbf{-1} & \mathbf{2} \end{bmatrix}$$

A matrix contains all coefficients that was created in the previous method.

$$\boldsymbol{Bx} = \begin{bmatrix} \mathbf{1 * 0} \\ \mathbf{1 * 1} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix} \qquad \boldsymbol{By} = \begin{bmatrix} \mathbf{1 * 0} \\ \mathbf{1 * 1} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix}$$

The Bx and By matrices represent the constant value on RHS.
Solving these matrices, we get the same value.
(x1 = 0.33, y1 = 0.33) & (x2 = 0.66, y2 = 0.66)

4 SYSTEM DESIGN

The design of the model proposed in the paper can be broadly divided into 2 main sections i.e., the general vending machine and the reverse vending machine. The vending machine will majorly accept 3 inputs from a user which are the money, the product required and the mode of payment. The processes that occur in the background will be explained further. The reverse vending machine (RVM) is a concept of recycling which will basically give a user an option to recycle the container in which they have received their product for which appropriate incentives would be given.

Figure 4 shows the basic flow diagram of the entire model, where a user has to select one mode of operation i.e., VM or RVM. If VM (Vending machine) is selected, the user can start inserting money which will be counted by the machine. At any point, the user can select a product which they wish to purchase. The cost of the selected product will be compared against the amount of money inserted, basis which the product will be delivered along with due balance or an error message stating insufficient funds will be flashed and the user can continue to add more money or cancel the transaction.

If RVM (Reverse vending machine) is selected, the user can insert the item to be recycled. Once inserted, the machine will determine the value of the product based on the product information. Once the value of the item is determined, appropriate incentives can be provided as per the user's choice of mode.

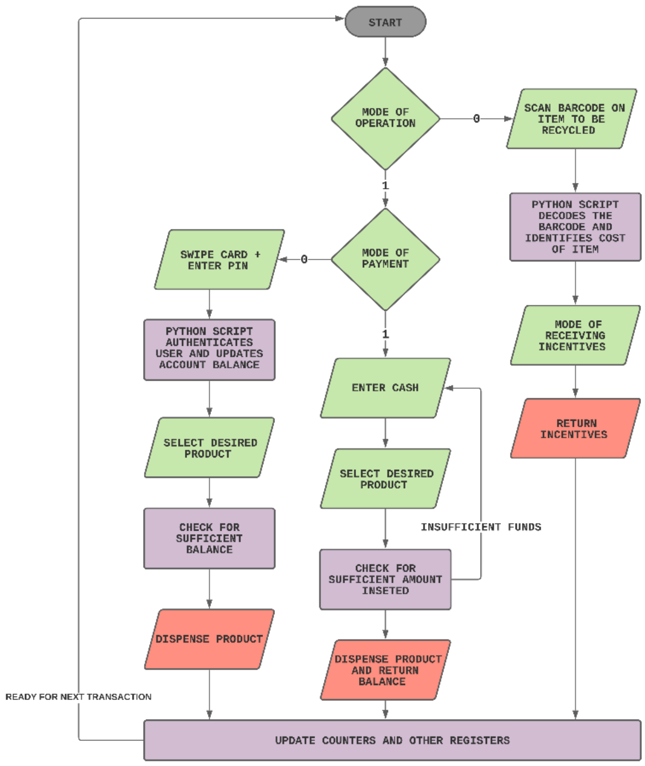


Figure 4: Flow diagram of the basic workflow of the model

4.1 Vending Machine (VM)

In the vending machine module, this model proposes to have three payment modes namely cash, card and UPI. Figure 4 also shows the various payment modes in this model. The first mode is the cash mode where the user has to insert cash/ coins and select a product, post which the machine will vend the product based on whether a sufficient amount of money is inserted or not. The second mode is the card mode where the user will have to swipe their card and then select the product of their choice; if there is enough balance in their card, the product will be delivered or else the transaction will be cancelled. The UPI payment mode is where the user will scan the code which will be unique for each product and make the payment.

4.1.1 Cash Payment Module

A currency counting device is required to count the denominations inserted in it. It takes in each bill individually, and recognizes the number of times an internal beam of light is obstructed to determine the bill and provide a total count. Some currency counters also have the capability to distinguish genuine bills from forgeries. The cash module is designed using an FSM (Finite State Machine). FSMs are at the core of most digital system design. Its basic operation is to store an arrangement of unique states and transition between them depending on the inputs and the state of the machine. They can be classified into two types: Moore - where the output of the state machine is purely dependent on the state variables; and Mealy - where the output is dependent on the current state variable as well as the input values. This characteristic of the mealy machine enables us to make this model scalable for products of different costs as required by the vendor.

In this model we have designed an FSM in which each state represents the total inserted amount at any time during the transaction. Initially the when no money is inserted, the FSM is at state 0; when money is inserted into the machine (i.e., denominations of 5, 10, 20, 50) there are various states where 'n' in state n denotes the amount of money inserted at that point. The output of the FSM depends upon the current state as well as the input (selected product/cancel transaction button). Figure 5 is an FSM which depicts the working of the cash module.

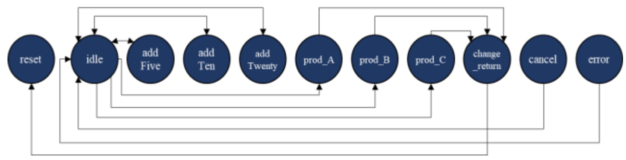


Figure 5: FSM for CASH Module of the vending machine

4.1.2 Card Payment Module

A card reading machine is an input device that reads data from credit or debit cards holding information embedded in it. The information includes details of the cardholder.

Card readers can 'read' the card in one of the following ways:

1. By reading the magnetic stripe on the back of the card combined with the buyer's signature or confidential PIN code.
2. By reading the chip when inserted into the reader, accompanied by the PIN code.
3. By contactless payment, which often does not require a PIN code if it is under a certain specified amount.

4.1.3 UPI Payment Module

Unified Payment Interface (UPI) allows instant transfer of money from one account to a another. A payment can be made through the UPI application on a smartphone only. Transactions can be carried out 24x7.

Money transfer has become a lot more convenient because of UPI. The users need not remember the receiver's account number, account type, IFSC, and bank name. The money can be transferred knowing their mobile phone number registered with the bank account or UPI ID. As most banks support UPI payments these days, most people would like to have this payment facility in the machine as well.

4.2 Reverse Vending Machine

The reverse vending machine gives the user two options i.e., cash or card to receive the incentives. Once the mode is selected, the user can insert the item to be recycled which is then scanned by the barcode scanner which will help the machine determine the type of the inserted item. For identification of the item, a barcode scanner is used which can read barcodes, decode the data contained in them and send the data to a computing device.

4.2.1 Item Classification In RVM

Two libraries are imported in the python code for capturing and decoding the data embedded in the barcode, they are (i) pyzbar and (ii) OpenCV.

(i) Pyzbar: Pyzbar is a python library (of the latest version), that reads 1D barcodes and quick response codes using the zbar library (originally made for python 2). It is an open-source software suite that is used for reading codes from any image/video source.

(ii) OpenCV: OpenCV is a cross-platform library using which one can develop real-time computer vision applications. It mainly focuses on image processing, video capture, and analysis including features like face detection and object detection.

5 SIMULATION RESULTS

5.1 ModelSIM

The entire model is stimulated on ModelSim software. The images shown below depict the behavior of the various ports in the vending machine with the help of signals. There is a clock for the model which synchronizes the operations. The green colored ports are the input ports, red colored ports are output ports and the white colored ports are internal registers used for logic implementation. In this section, the simulation results of the different operating modes are described.

5.1.1 Cash Module

The flowchart below (Figure 6) shows a scenario where a user firstly selects a mode of operation followed by a mode of payment. Here in this case, *op_mode* is 1, which implies that the vending mode is selected, similarly, *pm_mode* is also 1, which implies that the cash mode is selected. Using the input *button*, an input of ₹20 is made (which is 00010 on the button). After inserting ₹20, the product *001* is selected using register *prod* which costs ₹15, hence a balance of ₹5 is due. The balance is returned which is indicated on register *di5*. The other registers, such as counters *di5*, *di10*, *di20*, *p_count_0*, *p_count_1*, *p_count_2* is updated simultaneously and *balance_due*, *amount_inserted* are reset to initial values. All of the above discussed transaction can be seen in the ModelSIM wave graph from Figure 7.

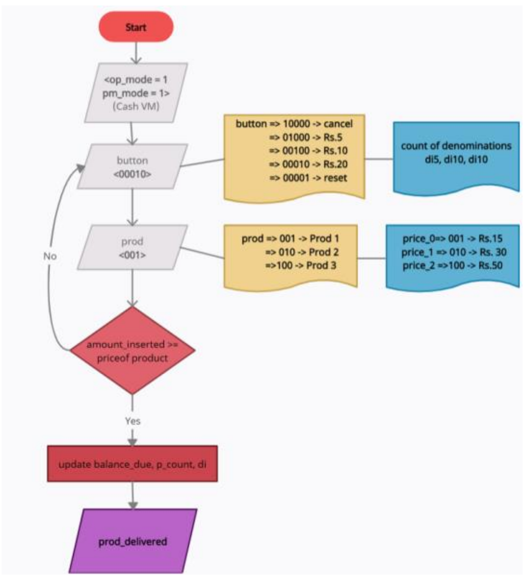


Figure 6: Flow chart showing a scenario in the cash vending machine module

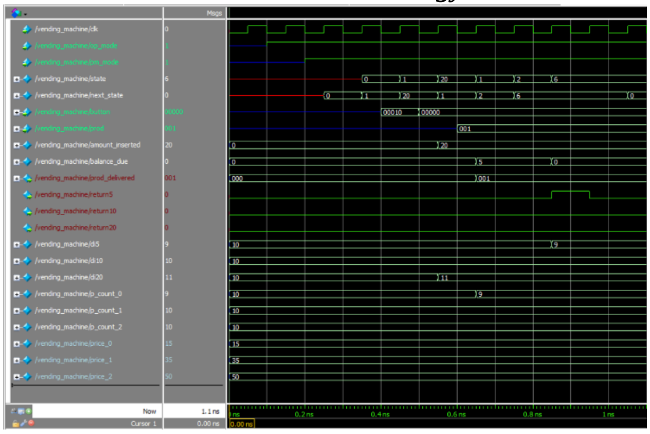


Figure 7: ModelSIM wave graph for the above discussed scenario along with its transcript (cash)

5.1.2 Card Module

The flowchart below (Figure 8) shows a scenario where a user firstly selects a mode of operation followed by a mode of payment. Here in this case, op_mode or operation mode is 1, which implies that the vending mode is selected, similarly, pm_mode or payment mode is 0, which implies that the card mode is selected. Using the input card_scan, scans the card. Then the user authenticates the transaction by entering the password. The product 100 is selected using register prod which costs ₹50. The user confirms the order using the input enter. Once the transaction is complete the card balance is updated and the product is delivered. The other registers, such as counters p_count_0, p_count_1, p_count_2 is updated simultaneously. All of the above discussed transaction can be seen in the ModelSIM wave graph from Figure 9.

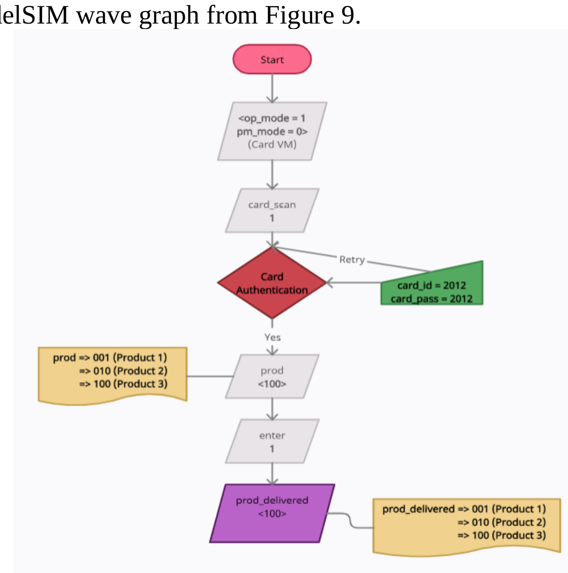


Figure 8: Flow chart showing a scenario in the card vending machine module

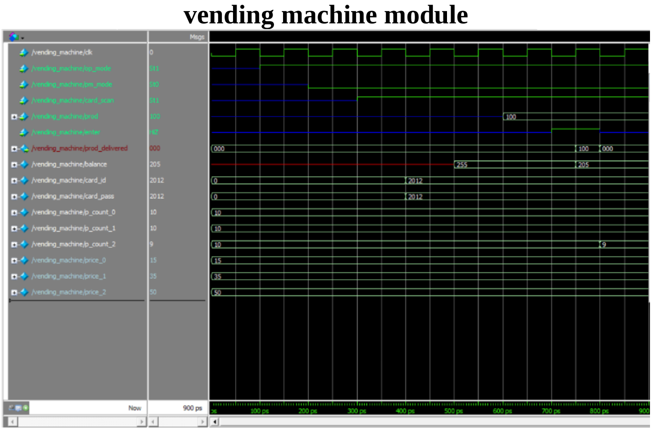


Figure 9: ModelSIM wave graph for the above discussed scenario along with its transcript (card)

5.1.3 Reverse Vending Machine

The flowchart below (Figure 10) shows a scenario where a user firstly selects a mode of operation. Here in this case, op_mode or operation mode is 0, which implies that the reverse vending machine mode is selected. Using the input card, the user selects the mode by which they want to receive rewards. Then the user scans the bar code on the product by which the type of the item is determined. The confirms the transaction and the value of the item is returned. The other registers, such as counters count_0, count_1, count_2 is updated simultaneously. All of the above discussed transaction can be seen in the ModelSIM wave graph from Figure 11.

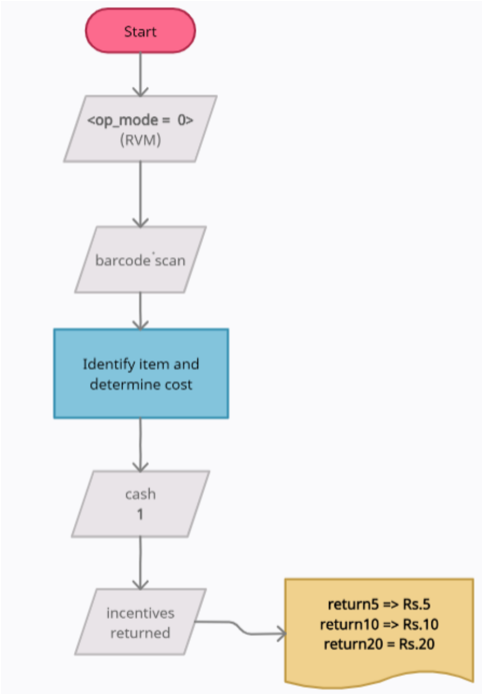


Figure 10: Flow chart showing a scenario in the reverse vending machine module

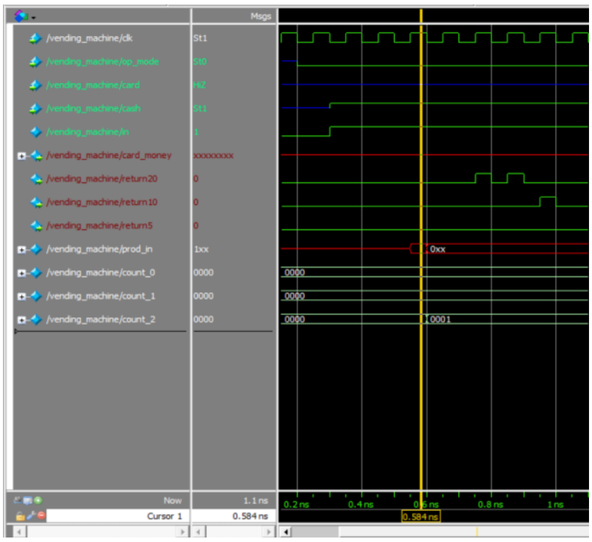


Figure 11: ModelSIM wave graph for the above discussed scenario along with its transcript (RVM)

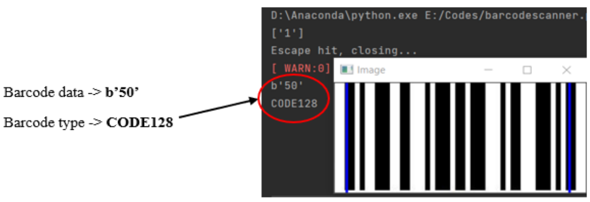


Figure 12: Barcode decoding in the reverse vending machine module

Figure 12 shows the python code decoding the scanned barcode. The encircled part of the snapshot shows the barcode data and type.

5.2 Quartus Prime

Flow Summary indicates whether the design exceeds the available device resources and also shows the overall logic utilization. A more detailed report can be viewed under the Resource Section in the Fitter section of the Compilation Report.

Flow Summary

<<Filter>>

Flow Status	Successful - Mon Mar 22 19:13:15 2021
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Lite Edition
Revision Name	vending_machine
Top-level Entity Name	vending_machine
Family	MAX 10
Device	10M08DAF484C8GES
Timing Models	Preliminary
Total logic elements	384 / 8,064 (5 %)
Total registers	70
Total pins	32 / 250 (13 %)
Total virtual pins	0
Total memory bits	0 / 387,072 (0 %)
Embedded Multiplier 9-bit elements	0 / 48 (0 %)
Total PLLs	0 / 2 (0 %)
UFM blocks	0 / 1 (0 %)
ADC blocks	0 / 1 (0 %)

Figure 13: Flow Summary

Slow 1200mV 85C Model Fmax Summary

<<Filter>>

	Fmax	Restricted Fmax	Clock Name
1	154.56 MHz	154.56 MHz	clk

Figure 14: Frequency Summary

Slow 1200mV 85C Model Setup Summary

<<Filter>>

	Clock	Slack	End Point TNS
1	clk	-5.470	-281.022

Figure 15: Slack Summary

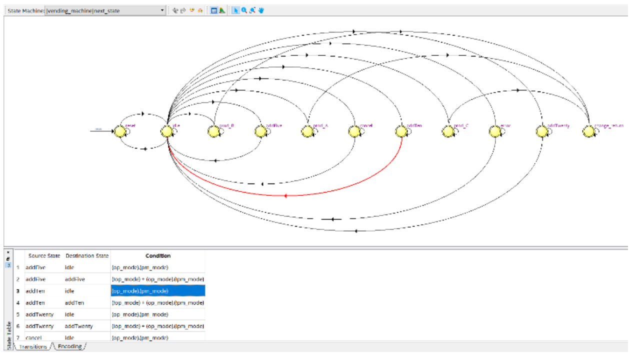


Figure 16: State machine (next_state)

5.2.1 Timing Analysis

- Slack is given as the difference between data required time and data arrival time.
- A positive setup slack implies that the design is functioning at the specified frequency and it has some more margin as well.
- A negative setup slack implies that the design doesn't achieve the constrained frequency and timing. This is called as setup violation.

Once the slack is analyzed, Timing Analyzer is used where clock constraints along with input and output delay are set. Initially the clock period was set at 10 ns which resulted in a negative slack (-5.041) (Figure 15, Figure 17 and Figure 18). Then clock period was set to 12.5 ns (80 MHz) which resulted in a positive slack (+1.568) (refer Figure 19 to 21).

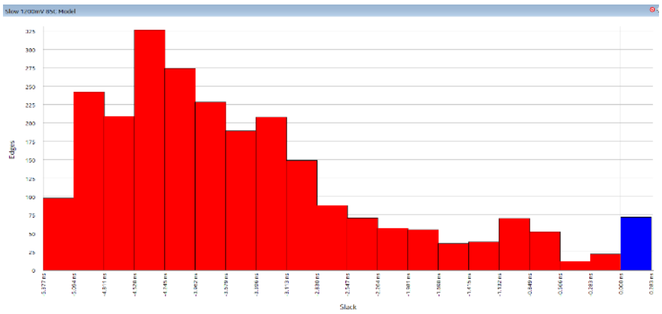


Figure 17: Slack Histogram (Clock Period →10ns)

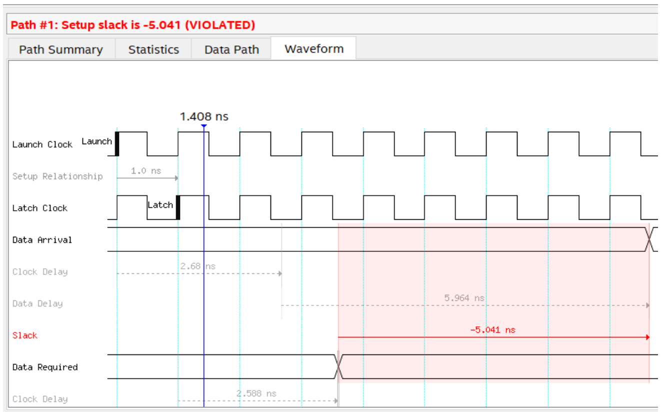


Figure 18: Timing Report (Setup Slack)

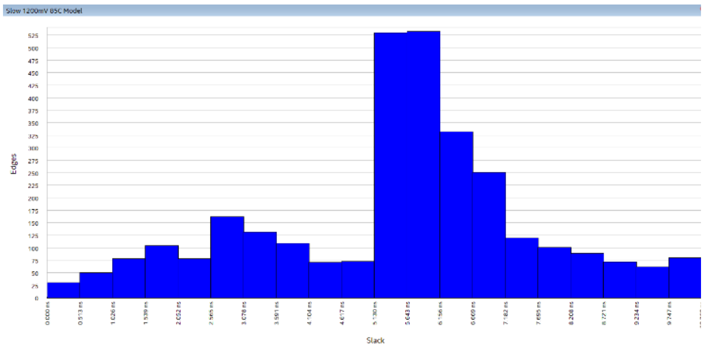


Figure 19: Slack Histogram (Clk Period → 12.5 ns)

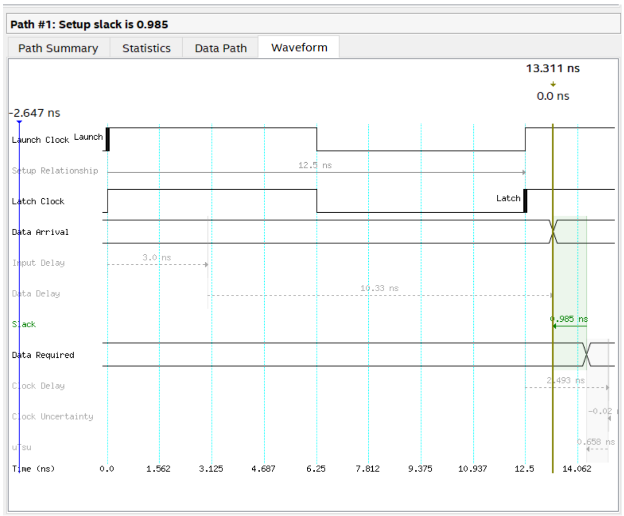


Figure 20: Timing Report (Setup Slack)

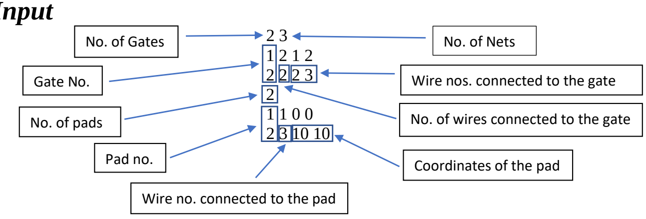
Slow 1200mV 85C Model			
	Clock	Slack	End Point TNS
1	clk1	1.568	0.000

Figure 21: Final Setup Summary

5.3 Half Perimeter Wire Length (HPWL)

In VLSI design, placement of gates plays a vital role. Here the gates are placed such that the wires/nets used are minimized. An analytical placer solves the placement problem where the objective function is half-perimeter wirelength and typical constraints are non-overlapping of cells, routability, power, delay, congestion, etc.

Input



Output

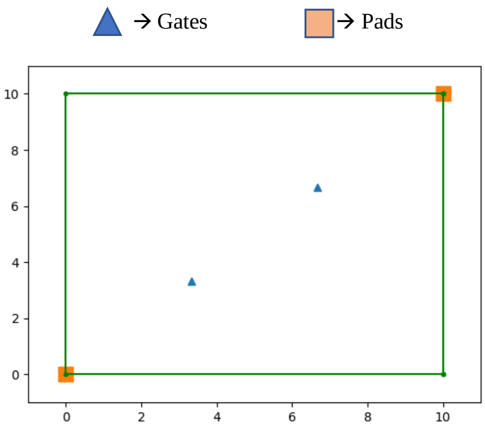


Figure 22: Placed Gates on IC

6 RESULT

The proposed vending machine system is designed by compiling synthesizable codes for each specific functionality. Each of the modules are simulated on ModelSIM to check for correct operation as required and synthesized on Quartus Prime for detailed analysis.

Various design analysis were carried out on Quartus Prime such as timing analysis, where the slack was studied in detail, the chip planner was used to understand the logic utilization in the FPGA board (MAX 10 FPGA Dev Kit) and the pin planner showed the pin assignments.

Design constraints (clock, input and output delays) were added taking into account realistic delays in order to have a desirable operating frequency and positive slack (1.568ns as compared to -5.470 ns initially) (Fig.15 and 21).

The code was optimized to various settings (such as balanced mode, aggressive performance mode, aggressive power mode, etc.). When set to operate in the balanced mode, the slack obtained was 0.985ns and power utilization was 52.62mW, when set to an aggressive performance mode the slack of the design improved to 1.568 ns and, when set to aggressive power mode the power utilization improved to 52.06mW.

Mode	No optimization	Balanced	Performance (Aggressive)	Power (Aggressive)
Slack (in ns)	-5.470	+0.985	+1.820	+1.608
Total Thermal Power Dissipation (in mW)	-	52.62	-	52.06

Table 1: Slack and Power Comparisons

The placement method, Half Perimeter Wire Length (HPWL) was implemented using python where the code will return the recommended placement for given inputs (Figure 22).

7 CONCLUSION AND FUTURE SCOPE

The system so designed is dynamic and can be personalized for the retailer meaning, they can customize the type, number, and cost of the product as required. The overall performance metrics were tested using the EDA tools and optimized to achieve the required performance characteristics.

We were able to improve the overall slack of the system by 7.038ns to achieve a positive slack and minimized the wire length using Half Perimeter Wire Length placement technique to improve timing further. The system can further be developed into ASICs (Application Specific Integrated Circuits) with the design specifications obtained.

Future enhancements of this design can provide additional features and increased efficiency.

- The system can further be developed into ASICs (Application Specific Integrated Circuits) with the design specifications obtained.
- A UPI payment mode can be integrated for a complete contact-less experience. The UPI option is available on most card readers nowadays and is a feasible option.
- An additional verification module can be incorporated in the RVM for better security.
- A mobile application can be integrated with the machine to give insights to the owner of the machine.

8 REFERENCES

[1] L. Shen, C. Qiu, X. Wu, C. Han and L. Hu, "Design of removable vending machine and research on the key implementation technology," in The Journal of Engineering, vol. 2019, no. 13, pp. 402-405, 1 2019, doi: 10.1049/joe.2018.9021.

[2] S. M. Trimberger, "Three Ages of FPGAs: A Retrospective on the First Thirty Years of FPGA Technology," in Proceedings of the IEEE, vol. 103, no. 3, pp. 318-331, March 2015, doi: 10.1109/JPROC.2015.2392104.

[3] R. Fung, V. Betz and W. Chow, "Slack Allocation and Routing to Improve FPGA Timing While Repairing Short-Path Violations," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 27, no. 4, pp. 686-697, April 2008, doi: 10.1109/TCAD.2008.917585.

[4] T. Jia, Y. Wei, R. Joseph and J. Gu, "An Adaptive Clock Scheme Exploiting Instruction-Based Dynamic Timing Slack for a GPGPU Architecture," in IEEE Journal of Solid-State Circuits, vol. 55, no. 8, pp. 2259-2269, Aug. 2020, doi: 10.1109/JSSC.2020.2979451.

[5] M. Xu, G. Grewal, S. Areibi, C. Obimbo and D. Banerji, "Near-linear wirelength estimation for FPGA placement," in Canadian Journal of Electrical and Computer Engineering, vol. 34, no. 3, pp. 125-132, Summer 2009, doi: 10.1109/CJECE.2009.5443860.

[6] V.V.S.Vijay krishna, A. Monisha, Sk.Sadulla, J. Prathiba 4th ICCCNT – 2013 "Design and Implementation of an automatic Beverages Vending Machine and its performance evaluation using Xilinx ISE and Cadence" IEEE – 31661

[7] P.Pradeepa, T.Sudhalavanya, K.Suganthi, N.Suganthi, M. Menagadevi IJAET/Vol. IV/ Issue I/Jan.-March., 2013/51-53 "Design And Implementation Of Vending Machine Using Verilog Hdl" International Journal of Advanced Engineering Technology E-ISSN 0976-3945

[8] Jagandeep Kaur1, Shruti Karkra May 2016 "Sailing through the Silicon Maze: FPGA versus ASIC" IJCSMC, Vol. 5, Issue. 5

[9] R. F. Molanes, J. J. Rodríguez-Andina and J. Fariña, "Performance Characterization and Design Guidelines for Efficient Processor–FPGA Communication in Cyclone V FPSoCs," in IEEE Transactions on Industrial Electronics, vol. 65, no. 5, pp. 4368-4377, May 2018, doi: 10.1109/TIE.2017.2766581.

[10] H. Zhang, D. Li, Y. Ji, H. Zhou, W. Wu and K. Liu, "Toward New Retail: A Benchmark Dataset for Smart Unmanned Vending Machines," in IEEE Transactions on Industrial Informatics, vol. 16, no. 12, pp. 7722-7731, Dec. 2020, doi: 10.1109/TII.2019.2954956.

[11] Quartus Prime Standard Edition Handbook

[12] Poonam V. Bhonde, Asst. Prof. Shweta Thakur Volume 6 Issue IV, April 2018 "Review on Design Simulation of Smart Vending Machine Using FPGA" International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887

[13] P.Pradeepa, T.Sudhalavanya, K.Suganthi, N.Suganthi, M. Menagadevi IJAET/Vol. IV/ Issue I/Jan.-March., 2013/51-53 "Design And Implementation Of Vending Machine Using Verilog Hdl" International Journal of Advanced Engineering Technology E-ISSN 0976-3945

[14] V.V.S.Vijay krishna, A. Monisha, Sk.Sadulla, J. Prathiba 4th ICCCNT – 2013 "Design and Implementation of an automatic Beverages Vending Machine and its performance evaluation using Xilinx ISE and Cadence" IEEE – 31661

[15] Maofic Farhan Karin, Khandaker Sharif Noor, Hasan U. Zaman December 2016 "Hardware Based Design and Implementation of a Bottle Recycling Machine using FPGA" IEEE Conference on Systems, Process and Control (ICSPC 2016)

[16] Jagandeep Kaur1, Shruti Karkra May 2016 "Sailing through the Silicon Maze: FPGA versus ASIC" IJCSMC, Vol. 5, Issue. 5.

