ECE 365

Workshop on Reconfigurable Computing

PROJECT: Vending Machine using Nexys-4 Artix-7 FPGA Board

Submitted to,

Dr.SobhitSaxena (23364)

Submitted by,

S.BalaKrishna (11809458)

Gantasala Sandeep(11808691)

CONTEXT

1. ABSTRACT

Abstract –

To design the behavior of the vending machine that we observe in most common public places like the shopping mall, airports, etc... A vending machine is a kind of sequential circuit that helps the user to collect his/her product with the money inserted without any help of a human interface for the transaction. Xilinx Vivado is a simulation as well as synthesis tool that is being used to analyze the vending machine functionality created using the help of Verilog by analyzing it on Nexys-4 Artix-7 FPGA board.

2. INTRODUCTION

a.VENDING MACHINE

A vending machine is a kind of money-operated machine that automatically dispenses goods, such as snacks, beverages, alcohol, cigarettes, and other consumer products. It allows the customer to get what he wants after inserting a certain amount of money or credit into the machine. It is usually seen on college campuses, businesses, schools, factories, and hospitals where food service is required. Recently, vending machines that dispense electronic items such as iPods, headsets, or Digital Cameras were already introduced to the community.

The first modern coin-operated vending machines were introduced in London, England in the early 1880s, dispensing postcards. The machine was invented by Percival Everitt in 1883 and soon became a widespread feature at railway stations and post offices, dispensing envelopes, postcards, and notepaper. The Sweetmeat Automatic Delivery Company was founded in 1887 in England as the first company to deal primarily with the installation and maintenance of vending machines. In 1893, Stollwerck, a German chocolate manufacturer, was selling its chocolate in 15,000 vending machines. It set up separate companies in various territories to manufacture vending machines to sell not just chocolate, but cigarettes, matches, chewing gum, and soap products.

Figure.1. Vending Machine.

b. WORKING

The working principle of the Vending Machine is quite simple It is just a machine that displays all the available products in the Machine through a transparent glass we have a display and some buttons right to it. The User has to select the product and enter the quantity with the help of buttons then insert the money cash into the cash hole and the coin into the hole of the coin when the money inserted is enough for the product it will deliver the products and returns change (if applicable).

Similar to the development of traditional mobile phones into smartphones, vending machines have also progressively, though at a much slower pace, evolved into smart vending machines. Newer technologies at a lower cost of adoption, such as the large digital touch display, internet connectivity, cameras and various types of sensors, more cost-effective embedded computing power, digital signage, various advanced payment systems, and a wide range of identification technology (NFC, RFID, etc.) have contributed to this development. These smart vending machines enable a more interactive user experience and reduce operating costs while improving the efficiency of the vending operations through remote manageability and intelligent back-end analytics. Integrated sensors and cameras also represent a source of such data as customer demographics, purchase trends, and other locality-specific information. It also enables better customer engagement for the brands through interactive multimedia and social media connectivity.

But these days we can even find some of the more advanced technologies involved in such machines where the person can collect the eatery like food, snacks, drinks, etc… within few second without any delay after the money inserted in terms of any mode like liquid cash, with the help of debit cards and online payments, etc…But we can observe that all of these are made with the basic functionality that we have created in this project.

3. MODEL REQUIREMENTS

a. FPGA

Field Programmable Gate Array (FPGA) contains an array of programmable blocks with programmable interconnect. It is used for complex circuits. It can handle larger circuits, in this, there are and/or planes. It provides logic blocks, I/O blocks, and interconnection wires and switches.

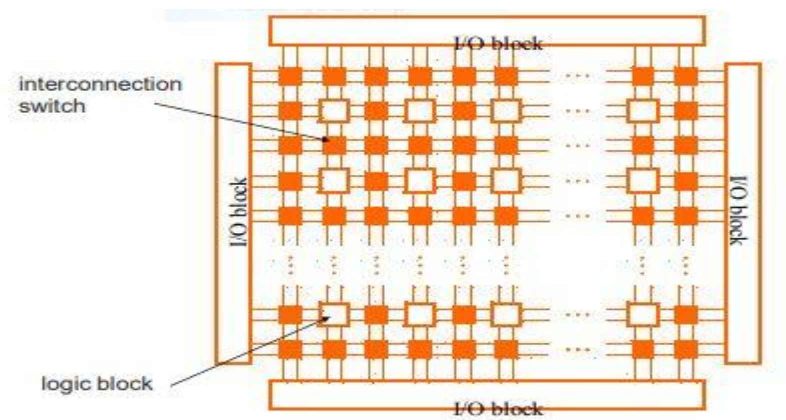


FIGURE.2. Structure of FPGA.

The Logic blocks provide functionality

The interconnection switches allow logic blocks to be connected and the I/O pins.

Logic Block:-

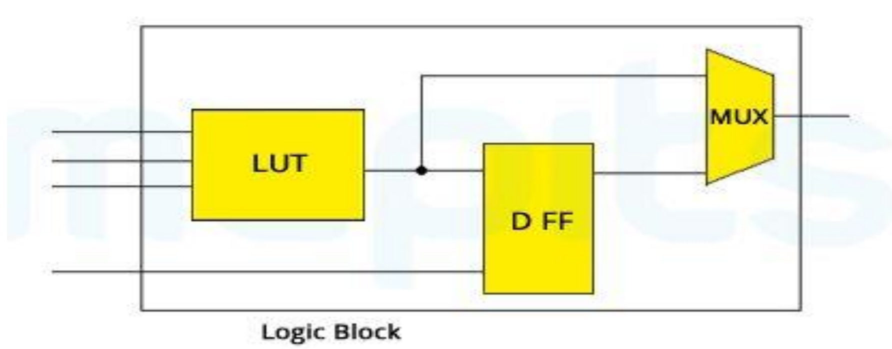


FIGURE.3. Logic Block.

LUT:

Lookup Table (LUT): It is used to implement the combinational logic functions. It contains a small number of inputs and one output. It also contains storage cells that can be loaded with desired values.

Register (DFF): It is used to store the output from Lookup Table

Multiplexer: It is used to select the output from the Lookup Table.

FPGA has no physical layout design and the design ends with a bit-stream used to configure a device.

Programming an FPGA

- ISP method is used
- LUTs contain volatile storage cells
- None of the other PLD technologies are volatile
- FPGA storage cells are loaded via a PROM when power is first applied

Advantages of FPGA:

- Low Development Cost
- Re-Configurability
- It is used for Complex circuits

FPGA families:

- Xilinx
- Actel
- Altera
- Lattice Semiconductors

b. NEXYS-4

The Nexys4 board is a complete, ready-to-use digital circuit development platform based on the latest Artix-7 Field Programmable Gate Array (FPGA) from Xilinx. With its large, high-capacity FPGA (Xilinx part number XC7A100T-1CSG324C), generous external memories, and collection of USB, Ethernet, and other ports, the Nexys4 can host designs ranging from introductory combinational circuits to powerful embedded processors. Several built-in peripherals, including an accelerometer, temperature sensor, MEMs digital microphone, speaker amplifier, and lots of I/O devices allow the Nexys4 to be used for a wide range of designs without needing any other components.

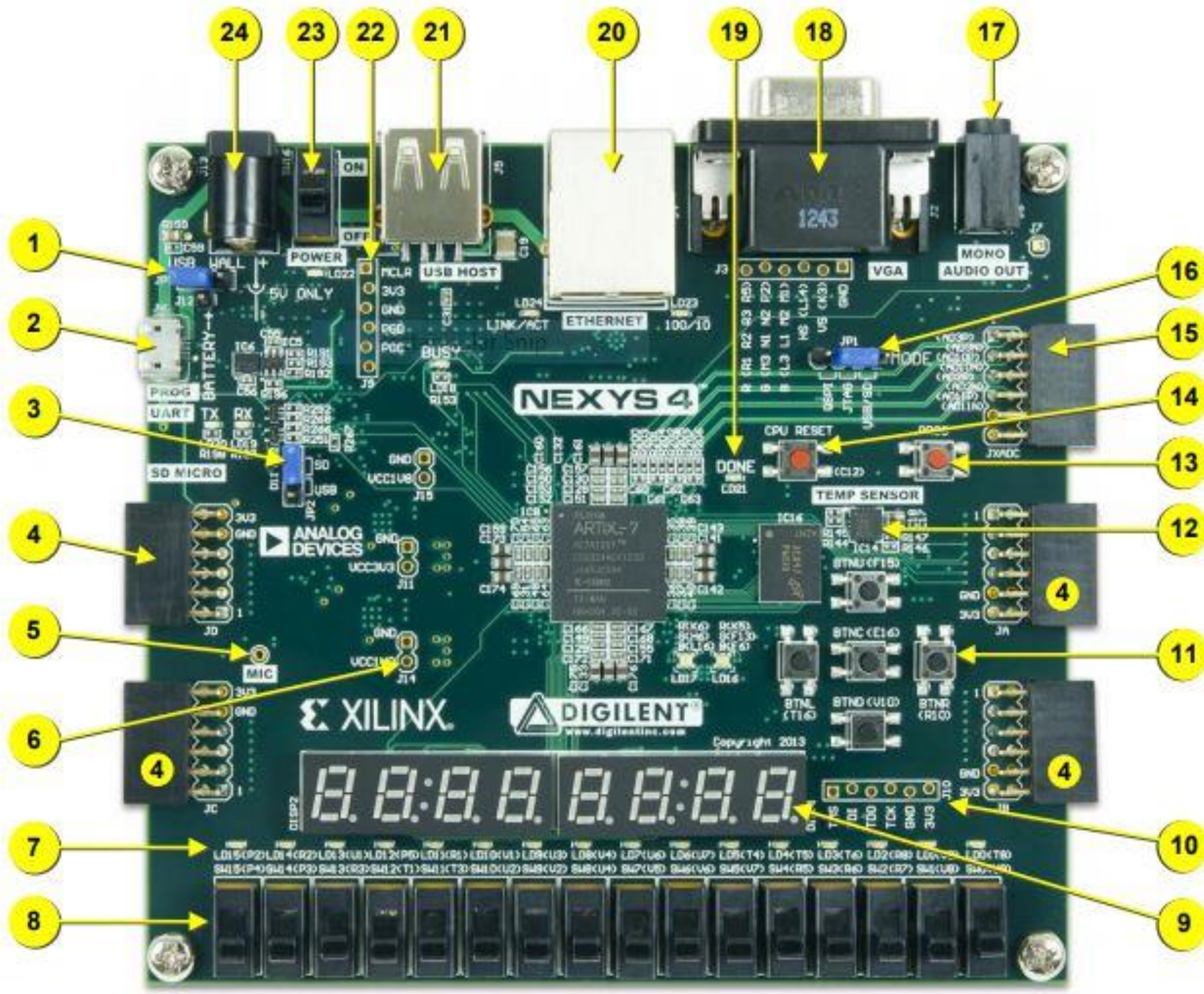


FIGURE.4. Artix-7 NEXYS-4 FPGA Board.

The Nexys4 also offers an improved collection of ports and peripherals, including:

- 16 user switches
- USB-UART Bridge
- 12-bit VGA output
- 3-axis accelerometer
- 16Mbyte CellularRAM
- Pmod for XADC signals
- 16 user LEDs
- Two tri-color LEDs
- PWM audio output
- Temperature sensor
- Serial Flash
- Digilent Adept USB port for programming and data
- Two 4-digit 7-segment displays
- Micro SD card connector
- PDM microphone
- 10/100 Ethernet PHY
- Four Pmod ports
- USB HID Host for mice, keyboards and memory sticks

The Nexys4 is compatible with Xilinx's new high-performance Vivado ® Design Suite as well as the ISE toolset, which includes ChipScope and EDK. Xilinx offers free "Webpack" versions of these toolsets, so designs can be implemented for no additional cost.

FIGURE.5. Artix-7 NEXYS-4 FPGA Board ports.

Callout	Component Description	Callout	Component Description
1	Power select jumper and battery header	13	FPGA configuration reset button
2	Adept USB (JTAG and UART)	14	CPU reset button (for soft cores)
3	External configuration jumper (SD / USB)	15	Analog signal Pmod connector (XADC)
4	Pmod connector(s)	16	Programming mode jumper
5	Microphone	17	Audio connector
6	Power supply test point(s)	18	VGA connector
7	LEDs (16)	19	FPGA programming done LED
8	Slide switches	20	Ethernet connector
9	Eight digit 7-seg display	21	USB host connector
10	JTAG port for (optional) external cable	22	PIC24 programming port (factory use)
11	Five pushbuttons	23	Power switch
12	Temperature sensor	24	Power jack

FIGURE.6. Artix-7 NEXYS-4 FPGA Board Components Description.

Features of Artix-7 100T:

- 15,850 logic slices, each with four 6-input LUTs and 8 flip-flops
- 4,860 Kbits of fast block RAM
- Six clock management tiles, each with a phase-locked loop (PLL)
- 240 DSP slices
- Internal clock speeds exceeding 450MHz
- On-chip analog-to-digital converter (XADC)

Programming methods:

- JTAG Programming (Joint Test Action Group (JTAG))
- Quad-SPI Programming (For programming the non-volatile flash device)
- USB Host and Micro SD programming

Memory:

- 2 memories
- A 128 Mbit cellular RAM (pseudo-static DRAM)
- A 128 Mbit non-volatile serial flash device

Oscillator/clocks:

- A single 100MHz crystal oscillator connected to pin E3

USB-UART Bridge (Serial Port):

- The Nexys4 includes an FTDI FT2232HQ USB-UART bridge (attached to connector J6) that lets you use PC applications to communicate with the board using standard Windows COM port commands.
- Free USB-COM port drivers, available from www.ftdichip.com under the "Virtual Com Port" or VCP heading, convert USB packets to UART/serial port data.
- The connections between the FT2232HQ and the Artix-7 are shown

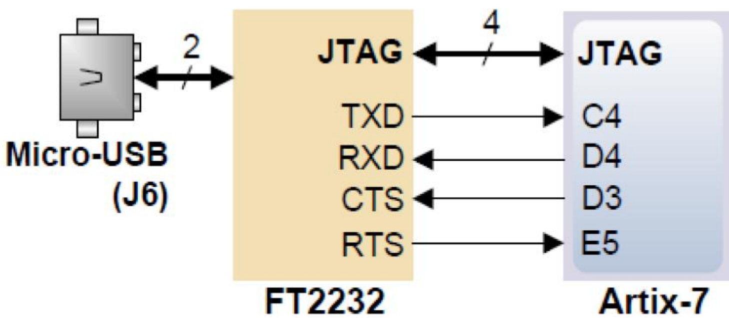


FIGURE.7. Artix-7 NEXYS-4 USB-UART Bridge.

Seven-Segment Display:

- The anodes of the seven LEDs forming each digit are tied together into one “common anode” circuit node, but the LED cathodes remain separate

- These seven cathode signals are available as inputs to the 8-digit display. This signal connection scheme creates a multiplexed display, where the cathode signals are common to all digits but they can only illuminate the segments of the digit whose corresponding anode signal is asserted.

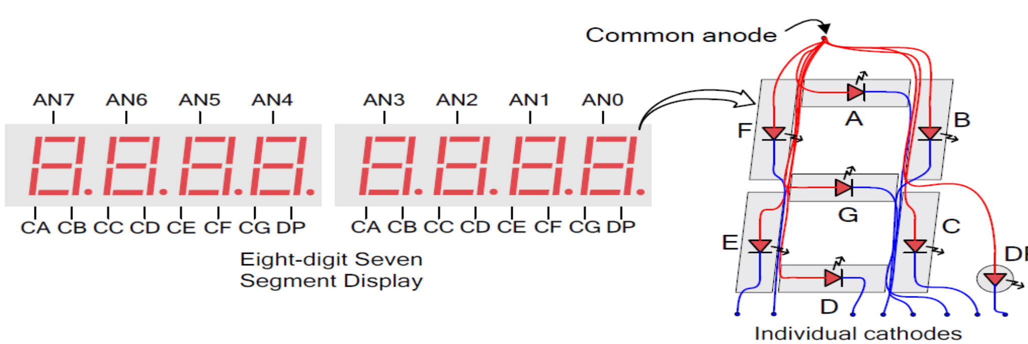


FIGURE.8. 8-Seven Segment Controlling.

- For each of the four digits to appear bright and continuously illuminated, all eight digits should be driven once every 1 to 16ms

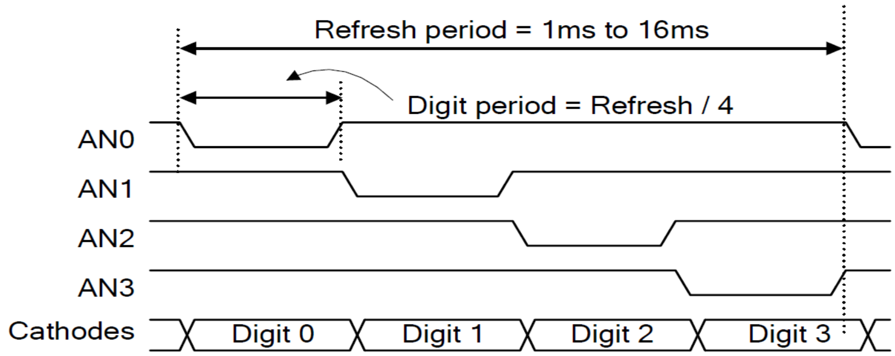


FIGURE.9. 8-Seven Segment Controlling using anode and cathodes.

Buttons in Nexys 4:

- BTNC
- BTNU
- BTND
- BTNL
- BTNR

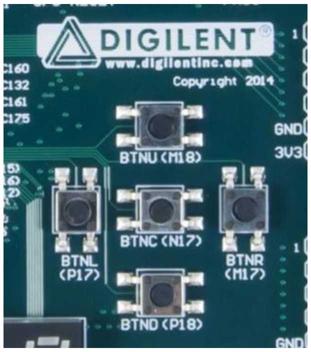


FIGURE.10. Push Buttons in Artix-7 NEXYS-4.

Basic I/O:

- Nexys4 board includes
- two tri-color LEDs
- sixteen slide switches
- six pushbuttons
- sixteen individual LEDs

- an eight-digit seven-segment display

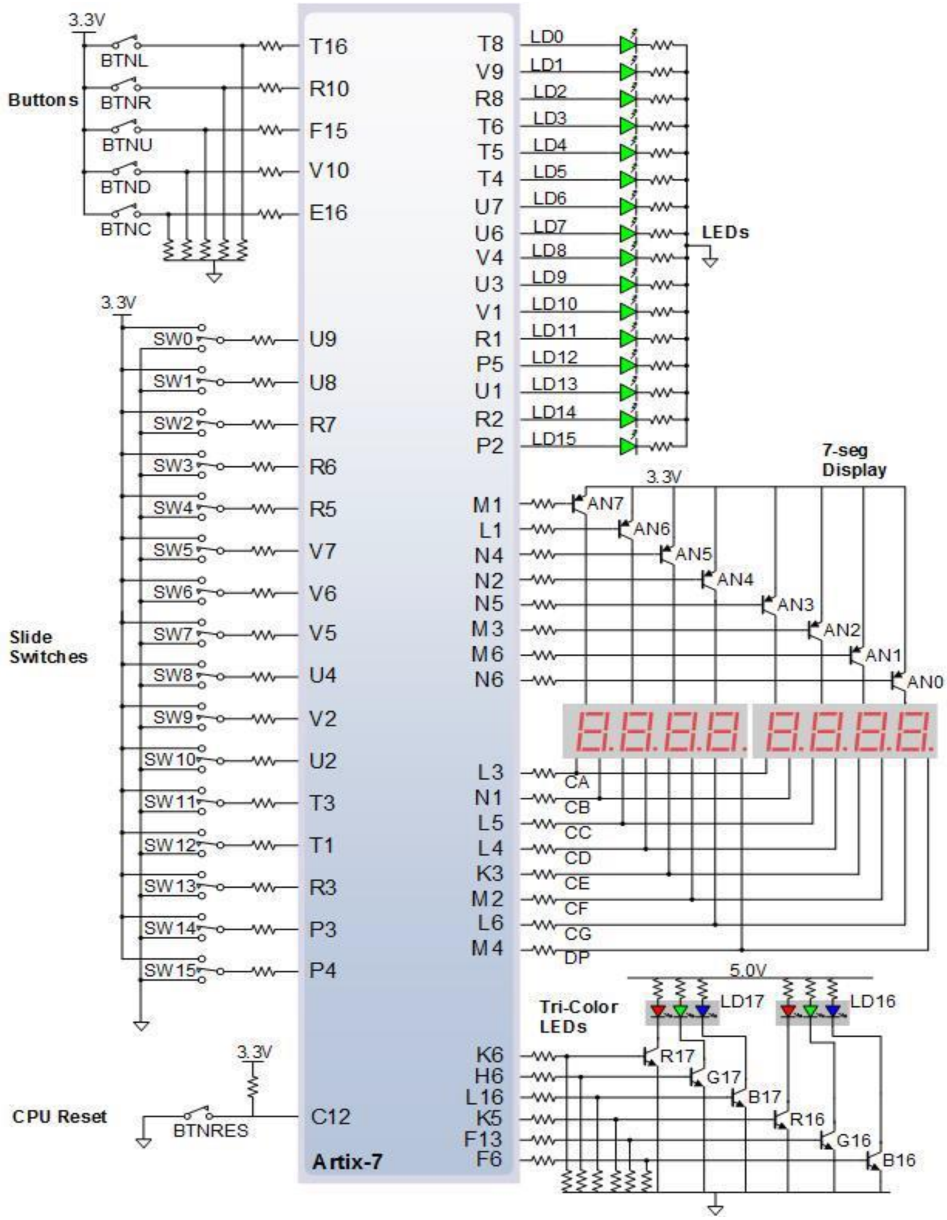


FIGURE.11. Input/Output port of in Artix-7 NEXYS-4.

c. **XILINX VIVADO**

While programming the FPGA board using VIVADO, the following files are required:

- HDL source file
- HDL test vector/bench
- Xilinx Design Constraint (XDC) file
- When programming an FPGA through software such as Xilinx's Vivado, you need to

inform the software what physical pins on the FPGA you plan on using or connecting to about the HDL code that you wrote to describe the behavior of the FPGA.

- Basically, it is a SCRIPT file that describes the physical pins on the FPGA
- We can use a Master XDC file for this purpose

XDC file:

An XDC "package" is a named collection of files that form a unit of versioning, update, and delivery from a producer to a consumer. Each package is embodied as a specially-named directory (and its contents) within a file system

When programming an FPGA through software such as Xilinx's Vivado, you need to inform the software what physical pins on the FPGA you plan on using or connecting to about the HDL code that you wrote to describe the behavior of the FPGA.

Xilinx Vivado:

Vivado Design Suite is a software suite produced by Xilinx for the synthesis and analysis of HDL designs, superseding Xilinx ISE with additional features for system on chip development and high-level synthesis. Vivado represents a ground-up rewrite and re-thinking of the entire design flow (compared to ISE), and has been described by reviewers as "well-conceived, tightly integrated, blazing-fast, scalable, maintainable, and intuitive".

Like the later versions of ISE, Vivado includes the in-built logic simulator ISIM. Vivado also introduces high-level synthesis, with a toolchain that converts C code into programmable logic. Vivado has been described as a "state-of-the-art comprehensive EDA tool with all the latest bells and whistles in terms of the data model, integration, algorithms, and performance

4. MODEL WORKING

a. **SIMPLE MODEL**

Implementation of vending machine using Nexys-4 Artix-7 FPGA board. The vending machine can handle 2 products and we can select a particular product and its the quantity and finally insert the money to get the products delivered and also the change if more money is inserted.

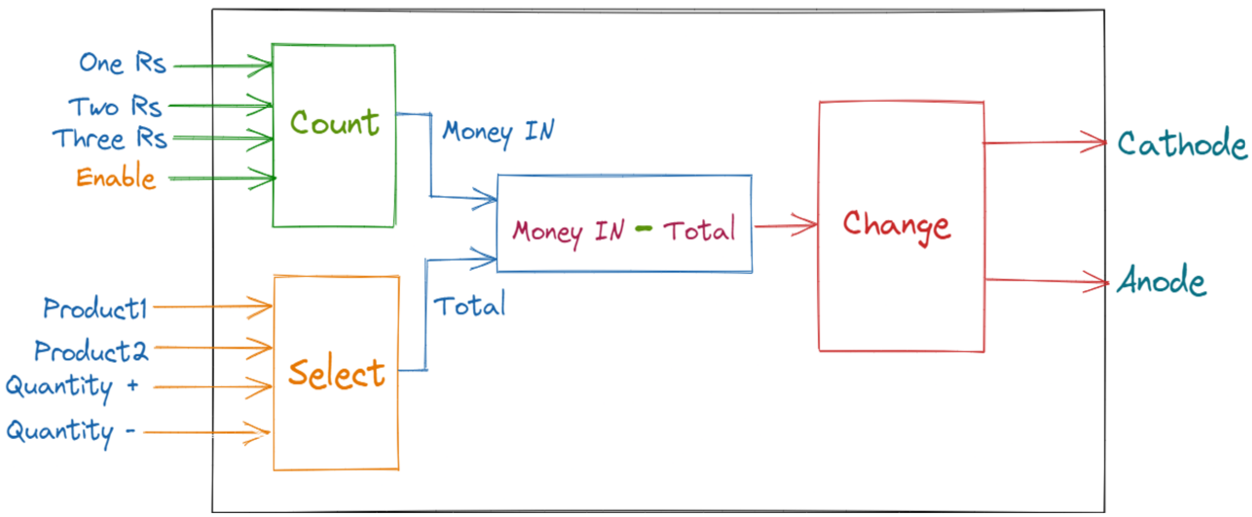


FIGURE.12. Block diagram of simple Vending Machine.

This is the Module where all the sub Modules are combined to finish the total Vending Machine which contains mainly three operations first, calculate the total money of the product selection and quantity is done second, similarly we have to calculate the inserted money by the user and subtract them and finally, display all these into seven-segment displays.

- Here we are giving money in the count module, where it can accept 1Rs, 2Rs, and 3Rs as a money-in where it calculates the total money inserted and outputs 7-bit binary.
- And in the select module, Product selection could be done for a particular product, and also the quantity of the product can be incremented or decremented in this module and the final amount will be counted giving output as 7-bit binary.
- Then both the inserted money as well as the total money for the products is sent to a module where a subtraction operation is done to obtain the change value.
- Down we have mentioned examples thus how does this machine works with Items, Quantity, Money, and Change.

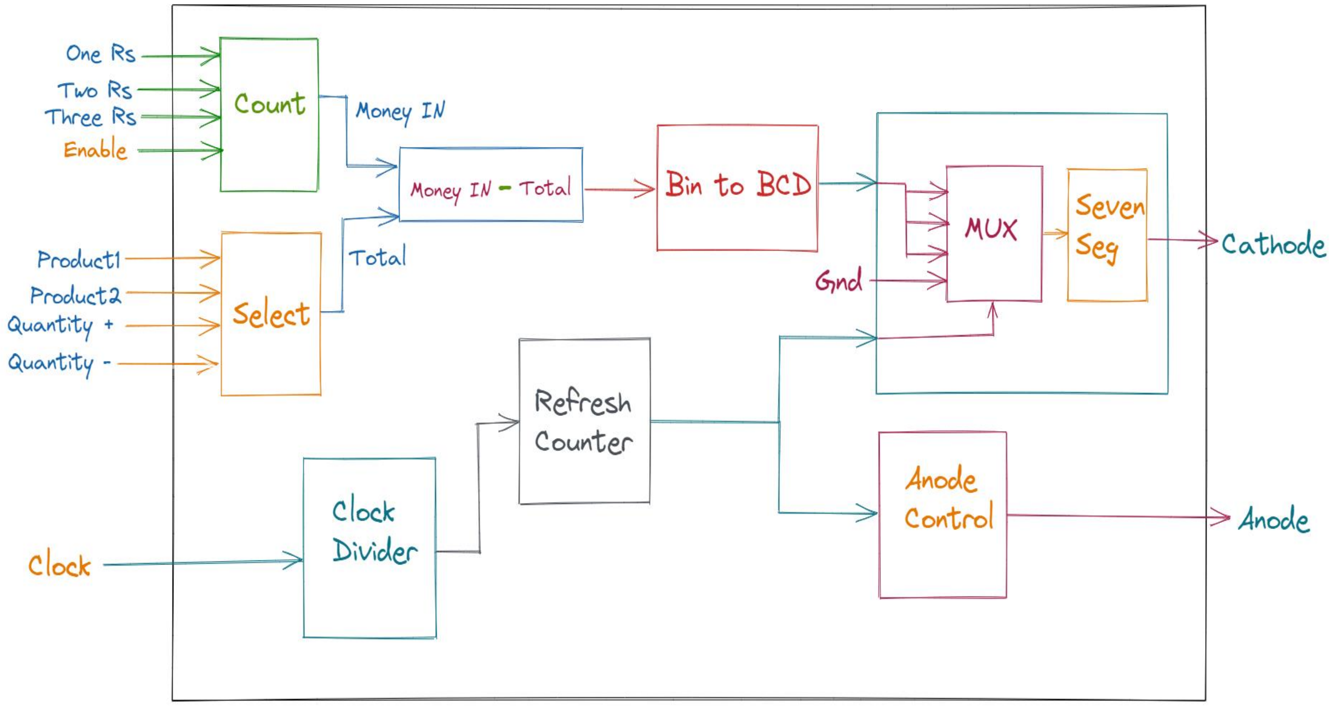


FIGURE.13. Complete Vending Machine Module Block.

b. MODULES WE HAVE USED

1. Vending_machine (Count)
2. Quantity_total(Quantity)
3. bin2bcd (Conversion)
4. Clock Divider
5. sevenseg_control

Vending Machine RTL Daigram:-

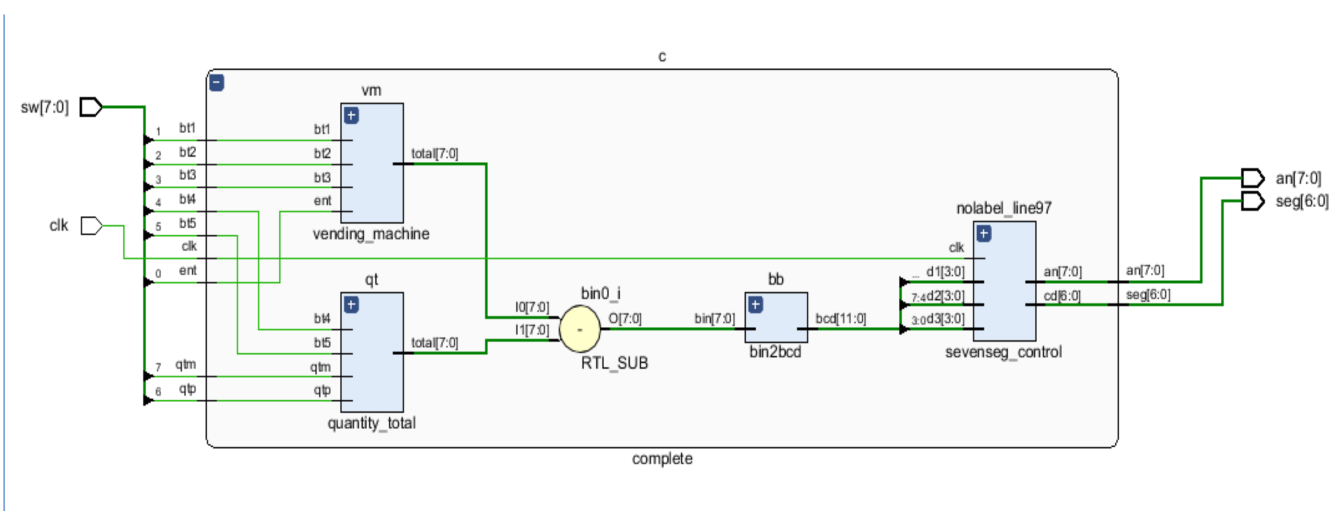


FIGURE.14. Complete Vending Machine.

Explanation Of Modules Used:-

Module Vending Machine (count):-

This Module is used to calculate the money that we have inserted by using buttons and provide total money in a 7-bit binary number.

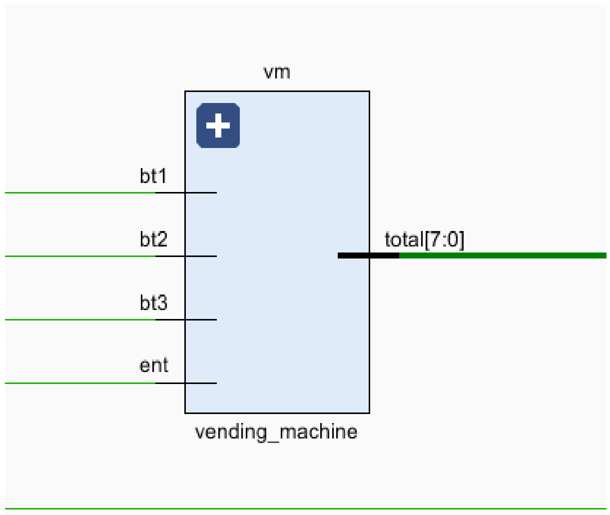


FIGURE.15. Vending Machine Module.

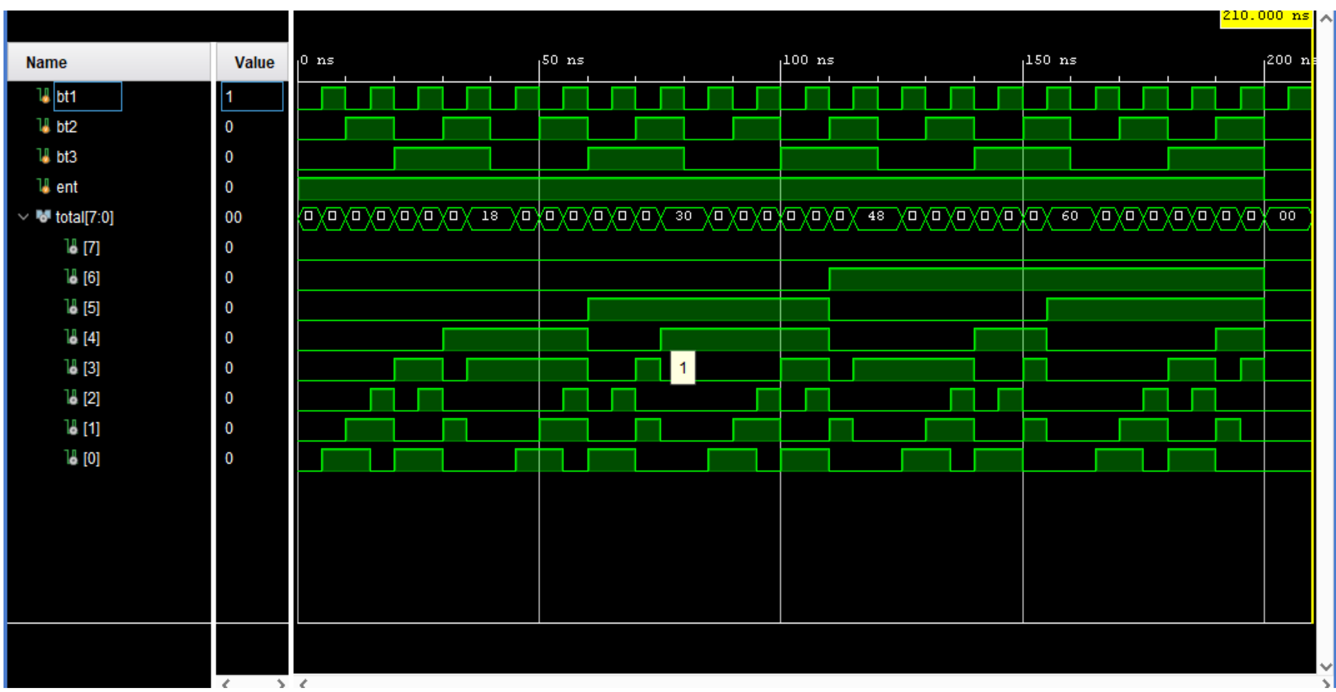


FIGURE.16. Waveform of Vending Machine Module.

Module quantity total(quantity):-

This Module helps with the selection of the product we needed and we can even select the quantity of the selected product it finally provides the total money of the product we have selected along with its quantity in the form of a 7-bit binary number.

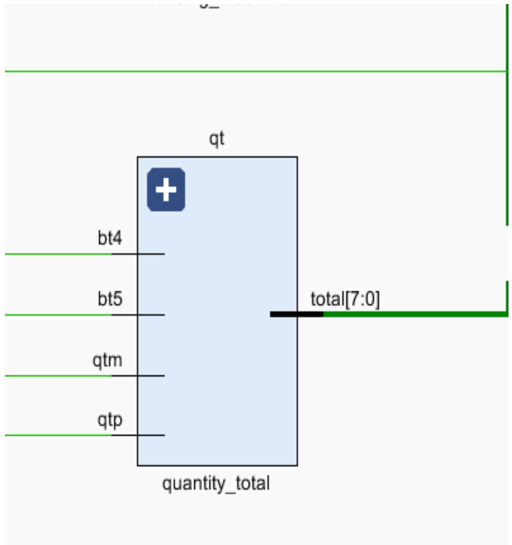


FIGURE.17. Quantity Module.

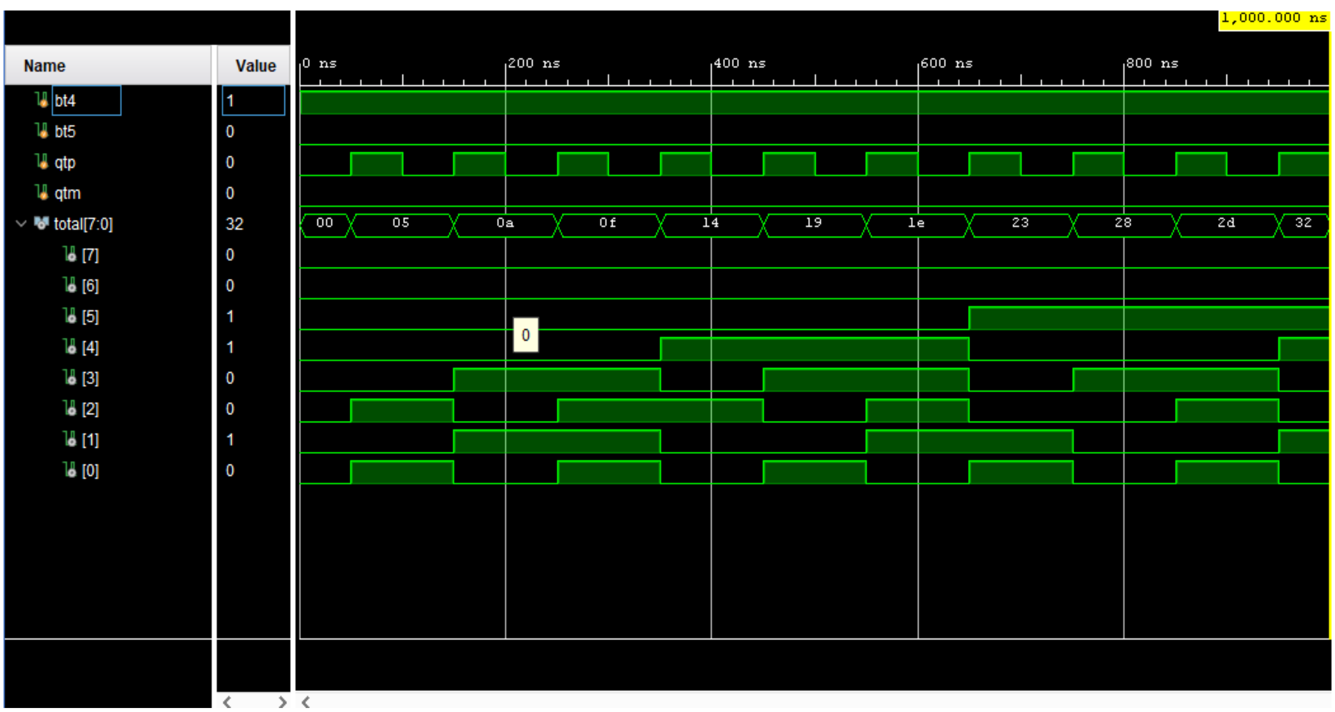


FIGURE.18. The waveform of module quantity total for Product 1(bt4).

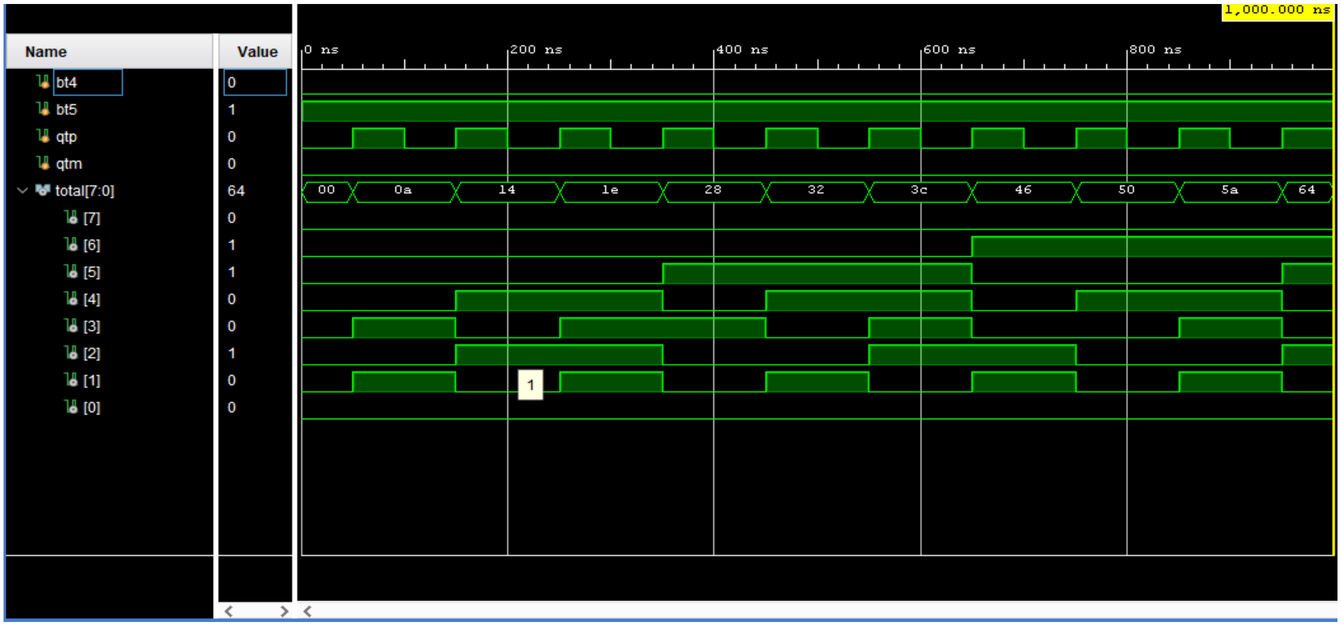


FIGURE.19. The waveform of module quantity total for Product 2(bt5).

Module bin2bcd(conversion):-

This module helps with the conversion of the input binary number which is of 7 bits into a 12-bit BCD number we are making this conversion because we can't display the direct the binary numbers into seven segments so we knew that the only possible way is to have BCD number of 12-bit which are made for seven-segment displays.

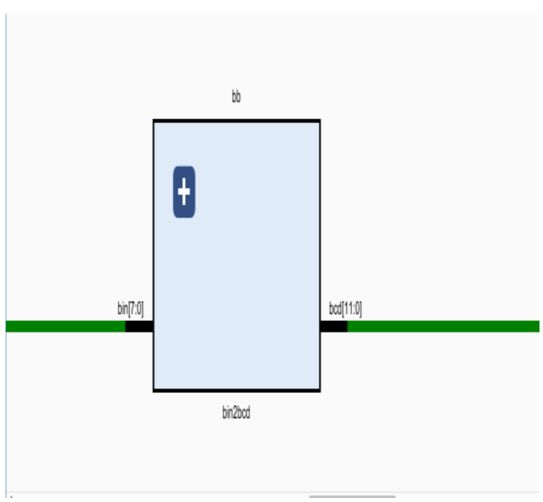


FIGURE.20. 7-bit Binary to 12-bit BCD.

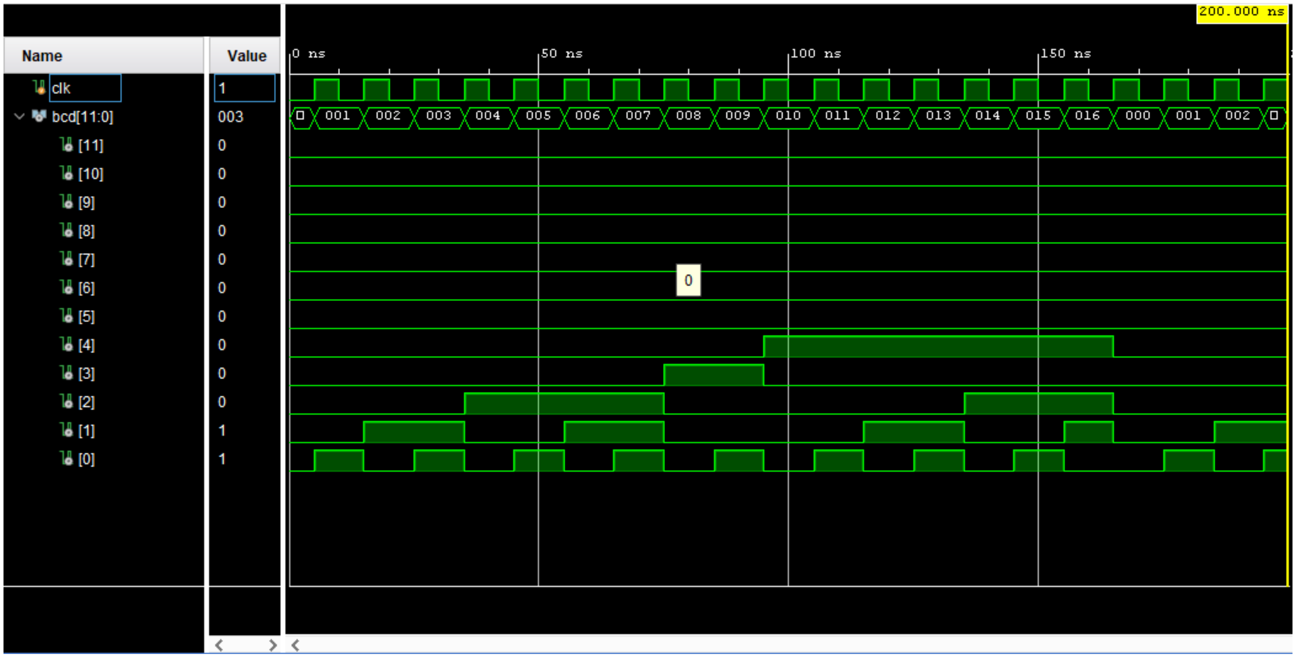


FIGURE.21. The waveform of Binary to BCD Conversion module.

Module Clock Divider:-

This module helps with the clock signal division part where we can convert the high-frequency clock to the required frequency clock because in this Artix-7 NEXYS -4 board we have a default clock of 100 MHz when which is very fast we can observe the change going on like in seven-segment display etc...

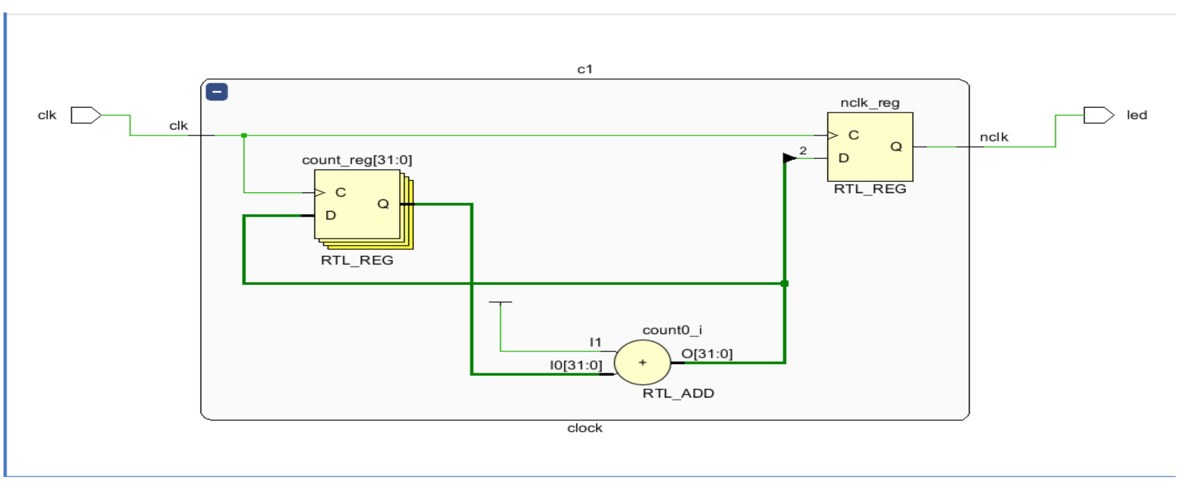


FIGURE.22. Clock Divider Module.

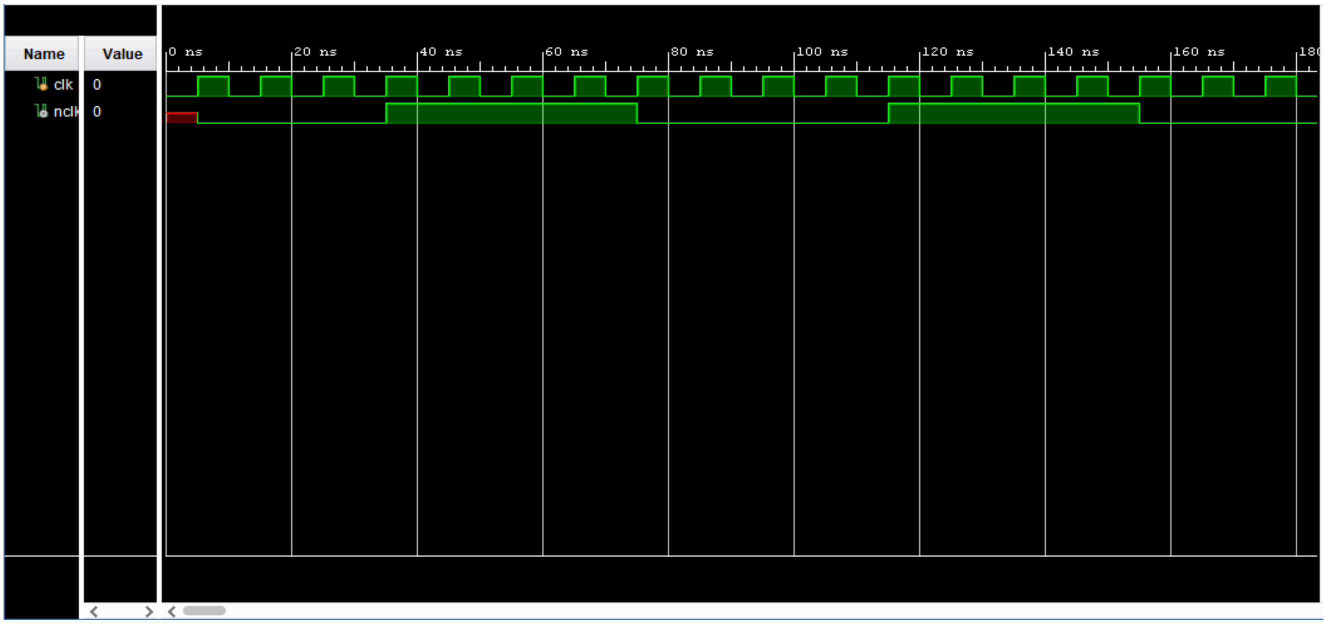


FIGURE.23. The waveform of Clock Divider Module.

Module Seven Segment Control:-

A seven-segment display is a form of an electronic display device for displaying decimal numerals that is an alternative to the more complex dot matrix displays. Seven-segment displays are widely used in digital clocks, electronic meters, basic calculators, and other electronic devices that display numerical information. But it is not that simple to display the values in seven segments especially the Artix-7 NEXYS -4 board because it has the eight seven-segment displays but for accessing them we have only 8-bit anode control and a common 7-bit cathode that is connected to all of them we can change a particular seven-segment value at a time if that happens then the value of all the seven segments will be changed at once to over this we will introduce a refresh counter module which helps to display all the context at their respective seven segments at every change happens the working principle of this module is simple it has the input as clock signal which then connected to a counter of 3 bit which has values from 000 to all the way to 111 according to the clock signal positive edge before that the default clock is very high so we will first divide the clock signal frequency and send to the refresh counter.

The output of the refresh counter is sent to a multiplexer which has the input as the digits in the form of their respective BCD number format to be displayed in their respective seven segments and the which is controlled by the select line given by the refresh counter to send the input to the output which has to be displayed.

Similarly, we also send the output from refresh counter to the anode control which basically active the particular seven segment to be activated.

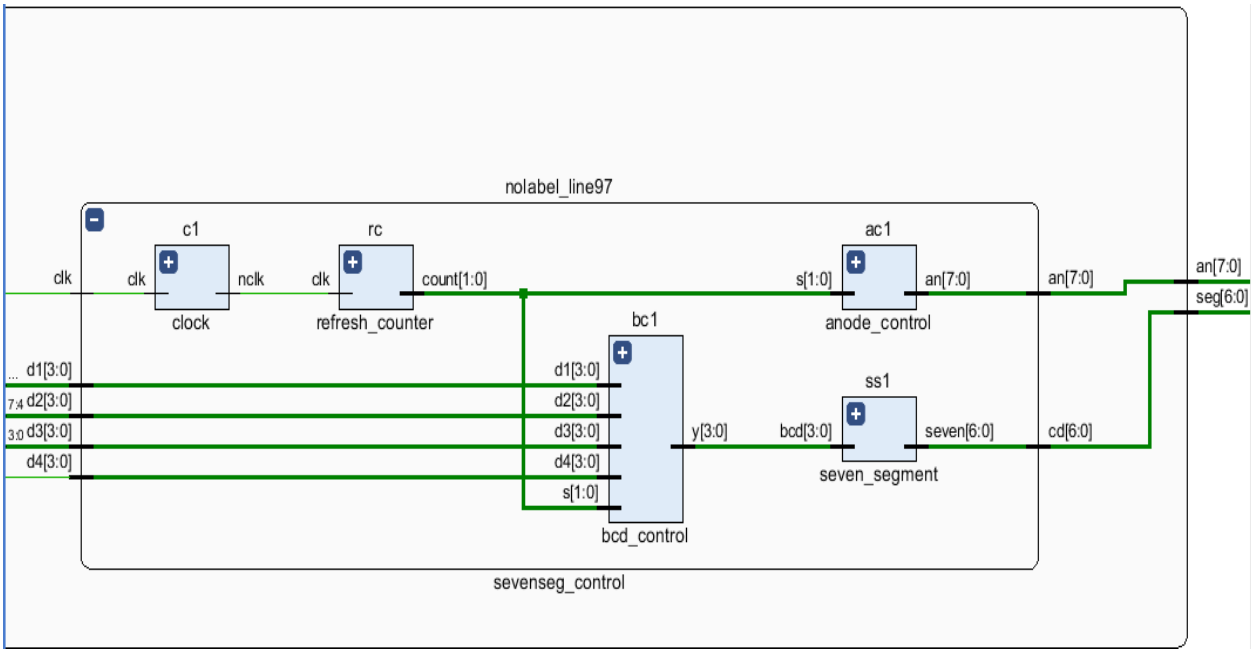


FIGURE.24. Seven Segment Control Module.

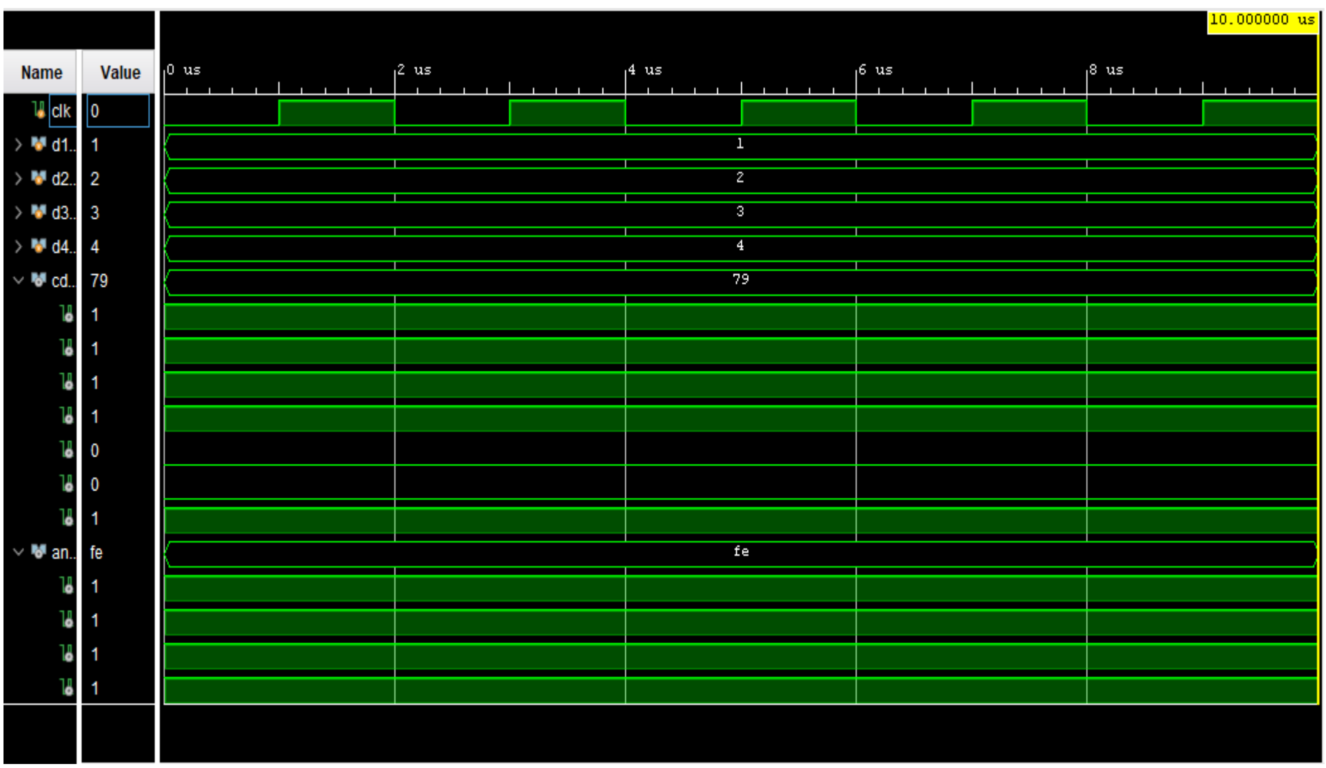


FIGURE.25. The waveform of Seven Segment Control.

Complete Vending Machine:-

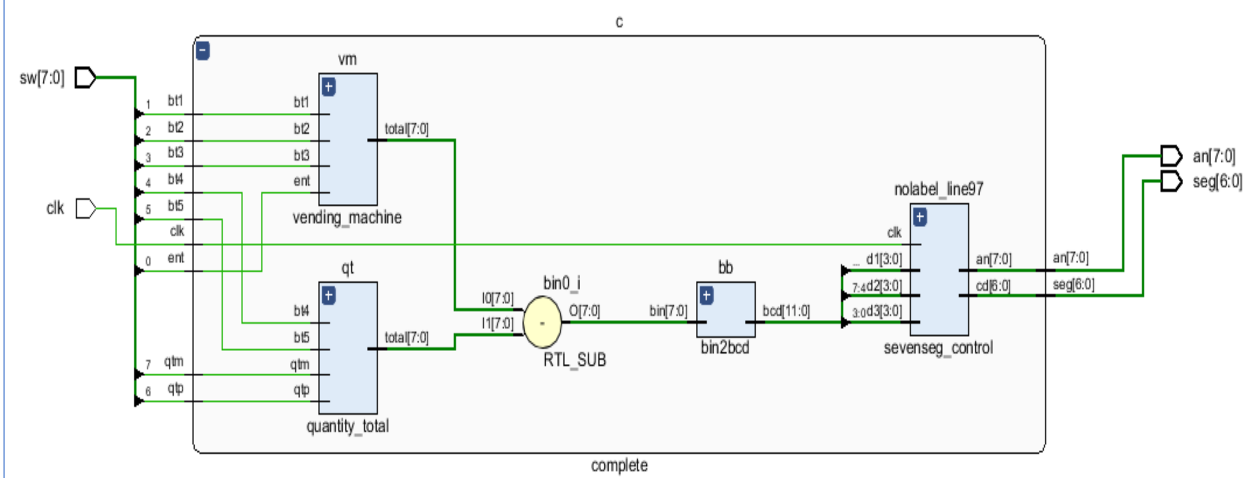


FIGURE.25. Complete Vending Machine Module.

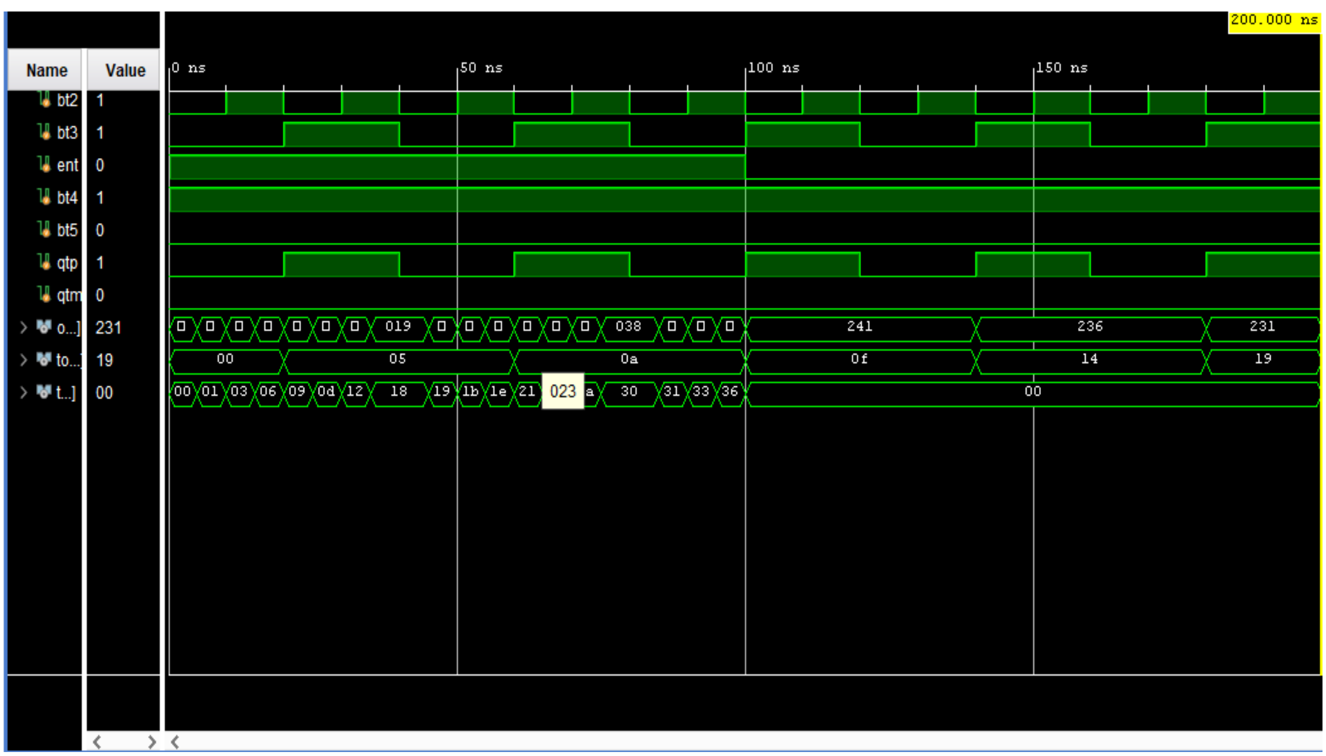


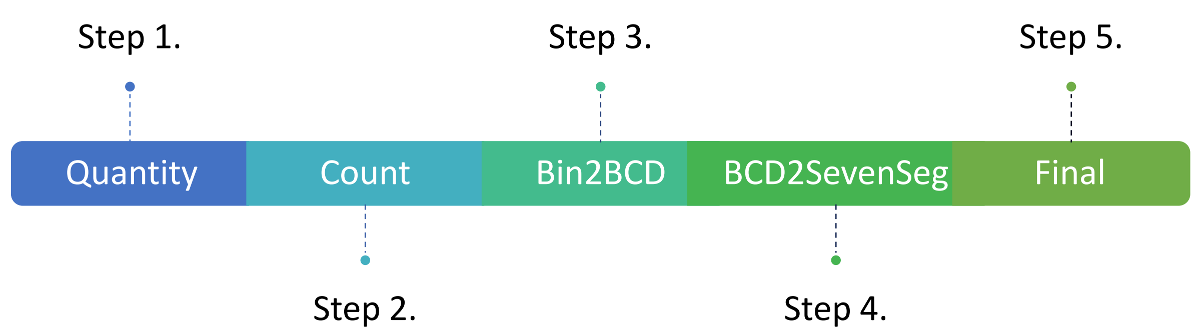
FIGURE.26. The waveform of Example for the Model.

5.RESULTS AND DISCUSSION

	Product	Quantity	Money	Change
Item 0	{0,0}	0	6	6
Item 1	{1,0}	1	9	4
Item 2	{1,0}	1	13	4
Item 1,2	{1,1}	1	18	5

TABLE.1. Example of the Operations.

Working Process of Vending Machine:-



Above Steps Working Processes

1. The selection of products and their quantity will be done here.
2. The Count of the Money Inserted is done here.
3. The Change in Binary will be converted to BCD.
4. Change in form of BCD will send to the seven segments.
5. Display the change in the seven seg.

From the results of the final completed vending machine module, we can clearly have obtained the results as per the table mentioned above. As we can confirm that the model is working fine through the waveforms

6. CONCLUSION

The greatest benefit of these vending machines is that they are very versatile and convenient. When you find yourself in the middle of closed shops, with a hungry stomach, a vending machine can come in handy.

When you talk about vending machines, costly items always come with them, making this one of its major disadvantages. Compared to items being sold in other stores, those items that are on vending machines are way on the high-priced spectrum.

In the Waveform of the final module, we can observe the results as per our requirements which indicates the working of the model in the future we can also implement some functionality that can help in upgrading the model according to the moving time.

In this, the model made us quite unique because most of the models were made to build the vending machine functionality topic using the help of finite state machine unlike us which makes our model simple and unique for the vending machine functionality.