International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 7 July 2023, pp: 158-167 www.ijaem.net ISSN: 2395-5252

Design and Implementation of ATM Machine controller using VHDL.

Aasavari Wani [#1], Sakshi Patil [#2], Shravani Kadam[#3], Siddhi More[#4], Rajveer Sastri[#5].

Department of Electronics of Telecommunication, VPKBIET Baramati, Savitraibai Phule Pune University, Maharashtra, India

Date of Submission: 25-06-2023 Date of Acceptance: 07-07-2023

ABSTRACT-
The Automated Teller Machine (ATM) has become an integral part of modern banking systems, providing convenient and secure access to financial services. This research paper presents the design and implementation of an ATM machine controller using the VHDL (VHSIC Hardware Description Language) programming language.
The study focuses on developing a robust and efficient control system for managing various functionalities of an ATM machine, including user authentication, transaction processing, cash dispensing.
The VHDL language is utilized to describe the behavior and structure of the ATM machine controller. The design process involves decomposing the system into modules and defining their interconnections. Finite State Machine (FSM) modelling is employed to represent the different states and transitions of the ATM controller, ensuring proper sequencing of operations and error handling. The implementation of the ATM machine controller involves synthesizing the VHDL code into a hardware description, which can be realized on an FPGA (Field-Programmable Gate Array) Simulation and testing techniques are employed to verify the functionality and correctness of the controller, including scenario-based testing and corner-case analysis.
The findings of this research paper contribute to the field of embedded systems and digital design by providing a comprehensive approach to designing and implementing an ATM machine controller in VHDL. The developed controller can serve as a basis for further enhancements and optimizations in ATM systems, ensuring reliable and secure financial transactions for users.
Overall, this research paper demonstrates the feasibility and effectiveness of using VHDL for developing complex control systems like ATM machine controllers. The presented design and implementation methodology can be applied to other similar applications in the domain of embedded systems, fostering innovation and advancements in the field. It consists of the implementation of the code using FPGA Kit Basys 3

Keyword: FSM; VHDL; FPGA

I. INTRODUCTION-

The Automated Teller Machine (ATM) has revolutionized the banking industry by providing customers with convenient and secure access to financial services. Behind the user-friendly interface lies a complex control system that manages various operations, including user authentication, transaction processing, cash dispensing, and receipt generation. This research paper focuses on the design and implementation of an ATM machine controller using the VHDL (VHSIC Hardware Description Language) programming language.

The design of an efficient and reliable ATM machine controller is crucial for ensuring seamless user experiences and maintaining the integrity of financial transactions. The controller acts as the central component that orchestrates the interactions between the user, the ATM hardware, and the bank's backend system.

The VHDL programming language offers a powerful and structured approach to describe and simulate digital systems. By utilizing VHDL, we can model the behavior and structure of the ATM machine controller, providing a foundation for efficient development and testing. VHDL's ability to capture complex hardware interactions and its support for high-level abstractions make it an ideal choice for implementing the intricate functionality of an ATM machine controller.

This research paper aims to provide a comprehensive understanding of the design and implementation process of an ATM machine

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 7 July 2023, pp: 158-167 www.ijaem.net ISSN: 2395-5252

controller using VHDL. We will explore the various modules and their interconnections within the controller, focusing on key functionalities such as user authentication, transaction processing, cash management, and error handling. The challenges and considerations encountered during the development of the ATM machine controller will be discussed. Security measures, such as encryption and secure communication protocols, will be addressed to ensure the confidentiality and integrity of sensitive user information. Concurrency handling techniques will be explored to manage multiple users accessing the ATM concurrently. Fault tolerance mechanisms will also be examined to ensure the system can recover from errors or failures gracefully.To validate the effectiveness and correctness of the implemented ATM machine controller, comprehensive testing techniques will be employed. Simulation environments will be set up to mimic various user scenarios and edge cases, allowing us to assess the controller's behavior and performance under different conditions. We will evaluate metrics such as transaction throughput, response time, and resource utilization to gauge the efficiency and effectiveness of the ATM machine controller.

In summary, this research paper aims to explore the design and implementation of an ATM machine controller using VHDL. By addressing key functionalities, challenges, and testing techniques, we aim to provide a comprehensive understanding of developing robust and efficient control systems for ATM machines.

1.1Operation of ATM machine controller -

The operation of the ATM machine controller using VHDL involves implementing the various functionalities and processes required for the successful functioning of an ATM machine. In this section of the research paper, we will outline the operation of the ATM machine controller in detail, focusing on the VHDL implementation aspects.

User Authentication:

1. The VHDL implementation of user authentication involves interfacing with the card reader module to extract the card information. The card data is then processed and validated against the bank's database using VHDL code. This includes checking the card's validity, verifying the entered PIN, and validating the account status. VHDL components can be designed to handle card reading, data extraction, and database communication to accomplish this task.

Transaction Selection:

2. Once the user is authenticated, the VHDL implementation allows the user to select a transaction from the menu options displayed on the user interface. This involves monitoring the keypad module for user input and updating the transaction selection variable accordingly. VHDL code can be written to handle keypad input, debounce signals, and update the transaction selection value.

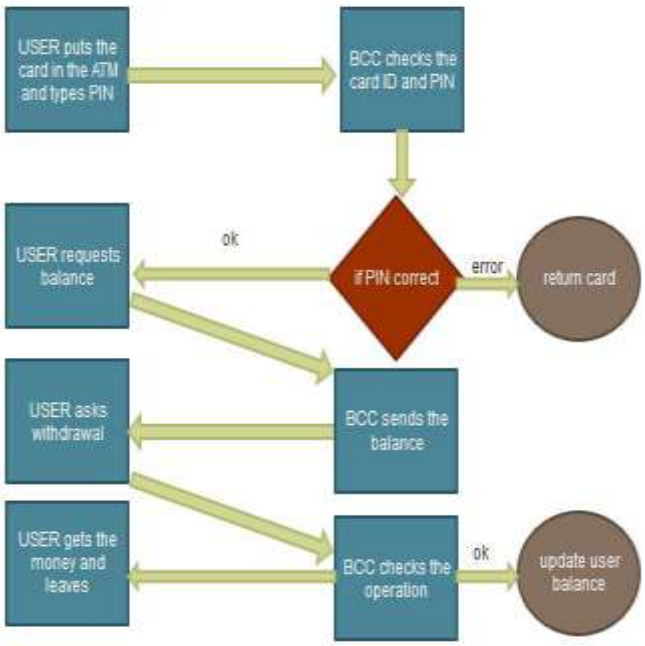


Fig.1) Flow chart of atm machine controller

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 7 July 2023, pp: 158-167 www.ijaem.net ISSN: 2395-5252

Transaction Processing:

0. After the user selects a transaction, the VHDL implementation initiates the processing of the chosen transaction. This involves communication with the backend system using VHDL code to send transaction details, such as transaction type, amount, and recipient account information. VHDL components can be designed to establish communication protocols with the backend system and handle data transmission and reception.

Cash Dispensing:

0. For withdrawal transactions, the VHDL implementation coordinates with the cash dispensing module to release the requested amount of cash to the user. VHDL code can be written to manage the cash inventory, update the available cash balance, and control the dispensing mechanism. It ensures that the requested amount is within the available cash limit and triggers appropriate signals to release the cash.

Concurrent Operations:

5. The VHDL implementation of the ATM machine controller should be capable of handling multiple users concurrently. VHDL code can be designed to handle concurrent transactions using techniques such as state machines, finite state machines, or concurrent processes. It ensures that each user's transaction remains isolated and secure, and manages the resources efficiently to prevent conflicts or inconsistencies.

By implementing the operation of the ATM machine controller using VHDL, this research paper aims to explore the design considerations, challenges, and implementation techniques involved in developing a robust and efficient ATM machine controller. The VHDL code for each operation will be discussed in detail, along with the integration of various modules and components to achieve a fully functional ATM machine controller.

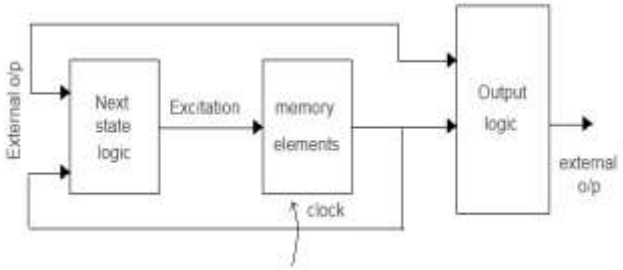
1.2 FSM (Finite State Machine) –

FSMs can be classified into two main types: Mealy machines and Moore machines. Here's some information about each type and how they relate to the code:

Mealy Machine: The outputs of a Mealy machine depend on both the current state and the inputs.

In terms of the code, a Mealy machine would typically use combinational logic to determine the output based on the current state and input values.

Moore Machine: The outputs of a Moore machine depend only on the current state.



In the code, a Moore machine would typically use the current state as an input to combinational logic to determine the output values.

In the context of the provided code, it is not explicitly stated whether it represents a Mealy or Moore machine. If the outputs depend on both the state and inputs, it is likely a Mealy machine. If the outputs depend only on the state, it is likely a Moore machine.

II. BASYS 3

It appears that the code is written for the Basys 3 FPGA board. The Basys 3 is a development board designed by Digilent and is based on the Xilinx Artix-7 FPGA. Here are some key features of the Basys 3 board:

FPGA: The board features the Xilinx Artix-7 FPGA with 33,280 logic cells and 1,800 Kbits of block RAM.

Clock: The board has a 100 MHz clock oscillator.

User Inputs: The code defines several user inputs for the board, including userinput (3-bit vector), pb0 (push button 0), and pb1 (push button 1). These inputs are used to interact with the ATM system.

Display: The Basys 3 board has four 7-segment LED displays for displaying information. The code uses the LED_out signal to control the cathode patterns of the 7-segment displays.

Memory: The code uses an array RAM to store user account information. The address variable is used to access the memory locations.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 7 July 2023, pp: 158-167 www.ijaem.net ISSN: 2395-5252

State Machine: The code implements a state machine to control the behavior of the ATM system. The state machine transitions between different states based on user inputs and timers.
Timing: The code uses various timers (timer_4sec, timer_3sec, timer_2sec, timer_1min) to keep track of time in different states of the ATM system.

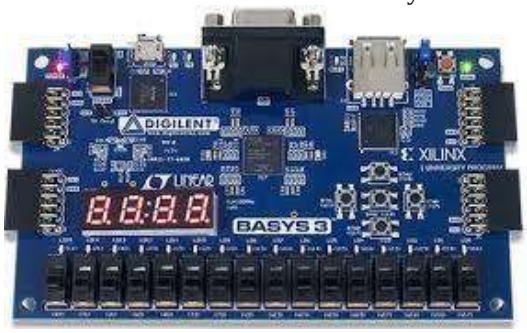


Fig.2) Basys 3 kit

Overall, the code seems to implement a basic ATM system using the Basys 3 board, where users can enter an account number, passcode, and perform deposit or withdrawal operations.

III. FPGA-

FPGA stands for Field-Programmable Gate Array. It is a type of integrated circuit that can be configured and reprogrammed to perform various digital logic functions. Here is some short information about FPGA in the context of the provided code for the Basys 3 FPGA board:

FPGA Basics: An FPGA consists of an array of configurable logic blocks (CLBs) interconnected by programmable interconnects. These logic blocks can be programmed to implement different digital logic functions, allowing for flexible and customizable circuit designs.
Basys 3 FPGA Board: The Basys 3 is a development board that features a Xilinx Artix-7 FPGA. It provides various hardware components such as push buttons, LED displays, switches, and memory to interact with and control the FPGA.
Hardware Description Language (HDL): The code for the Basys 3 FPGA board is typically written in a hardware description language like Verilog or VHDL. These languages allow designers to describe the desired hardware behavior and structure.
RTL Design: The code for an FPGA typically follows a Register Transfer Level (RTL) design approach. RTL describes the flow of data between registers, specifying the timing and control signals for each operation.
Synthesis and Place-and-Route: Once the code is written, it needs to be synthesized and mapped to the specific FPGA device. Synthesis converts the high-level RTL code into a gate-level representation. Place-and-route determines the physical location of each logic element on the FPGA and configures the interconnections accordingly.

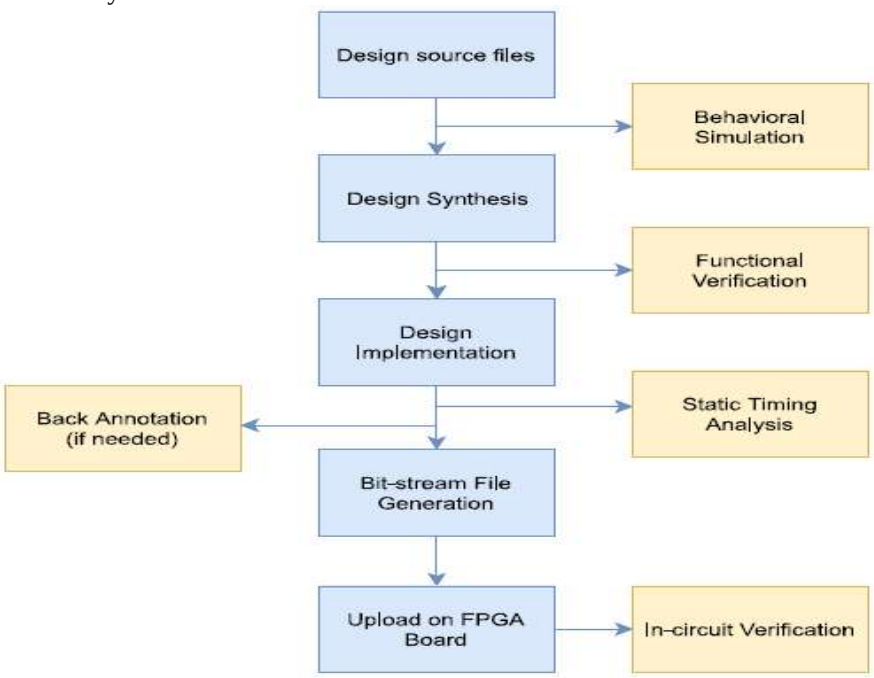


Fig.3) Flow of FPGA

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 7 July 2023, pp: 158-167 www.ijaem.net ISSN: 2395-5252

Configuration: After synthesis and place-and-route, the final configuration file is generated. This file contains the programming information required to configure the FPGA with the desired circuit design. It specifies the connections, logic elements, and memory configuration.

Bitstream: The configuration file is converted into a bitstream, which is a binary file that contains the configuration information for the FPGA. This bitstream is loaded onto the Basys 3 FPGA board to configure the FPGA and implement the desired circuit.

Reconfigurability: One of the key advantages of FPGA is its reconfigurability. The FPGA can be reprogrammed multiple times to implement different circuits or modify the existing design without changing the underlying hardware.

Parallel Processing: FPGAs excel at parallel processing due to their ability to implement multiple logic functions simultaneously. This makes them suitable for applications that require high-performance computing, signal processing, and real-time operations.

Application-Specific Designs: FPGAs are commonly used in various applications such as digital signal processing, telecommunications, aerospace, industrial automation, image and video processing, and prototyping of custom integrated circuits.

In summary, FPGA is a flexible and reprogrammable integrated circuit that allows for the implementation of custom digital logic designs. The Basys 3 FPGA board provides a platform for developing and testing FPGA-based designs, including the code for the provided ATM system.

IV. DESIGN METHODOLOGY -

To design this code, a systematic approach known as a design methodology is typically followed. Here is a general design methodology that could be used for this code:

Requirements Analysis: Clearly understand the requirements and specifications of the ATM system. Identify the desired functionalities, inputs, outputs, and constraints.

High-Level Design: Create a high-level design for the ATM system. Identify the major components and their interactions. Determine the overall architecture and data flow.

State Machine Design: Design the state machine that represents the behavior of the ATM system. Identify the different states, transitions between states, and the actions to be performed in each state.

Input/Output Design: Define the input and output interfaces for the ATM system. Determine how the Basys 3 board's components such as push buttons, LED displays, and memory will be utilized.

Memory Design: Determine the memory requirements for storing user account information. Design the memory organization and access methods.

Testing and Debugging: Test the code thoroughly to ensure it meets the specified requirements. Verify the functionality of each component and the overall system behavior. Debug any issues that arise during testing.

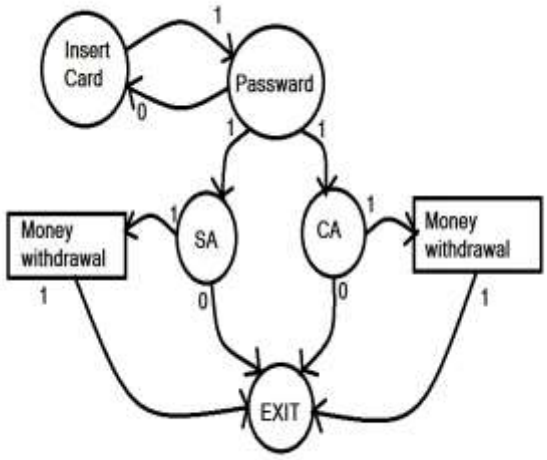


Fig.4) State diagram of atm machine controller

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 7 July 2023, pp: 158-167 www.ijaem.net ISSN: 2395-5252

Coding: Implement the code based on the high-level design. Write the necessary code to handle user inputs, control the state machine, manage memory, and interact with the Basys 3 board's components.
Optimization and Refinement: Optimize the code for performance and resource utilization. Refactor the code if necessary to improve readability, maintainability, and efficiency.
Validation and Verification: Validate the ATM system by performing extensive testing with various scenarios and edge cases. Verify that it meets all the specified requirements and behaves correctly.

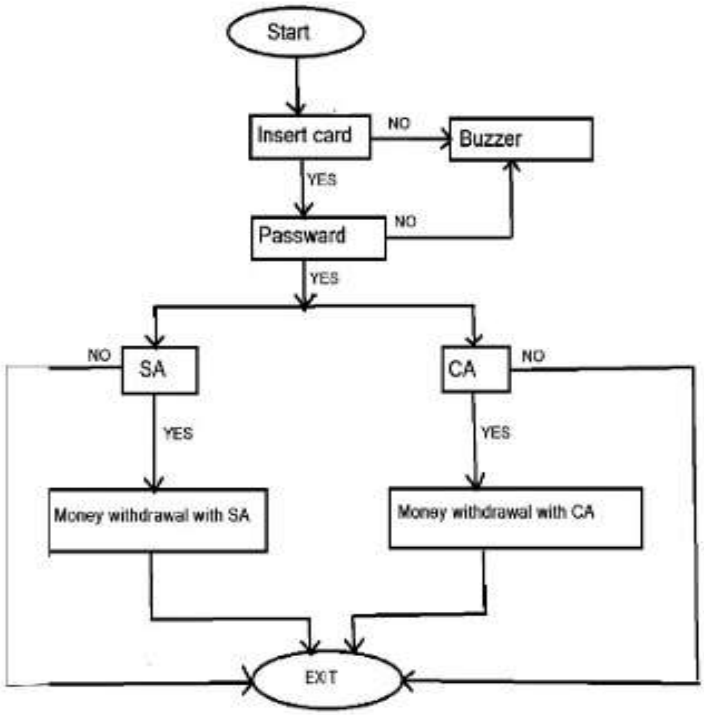


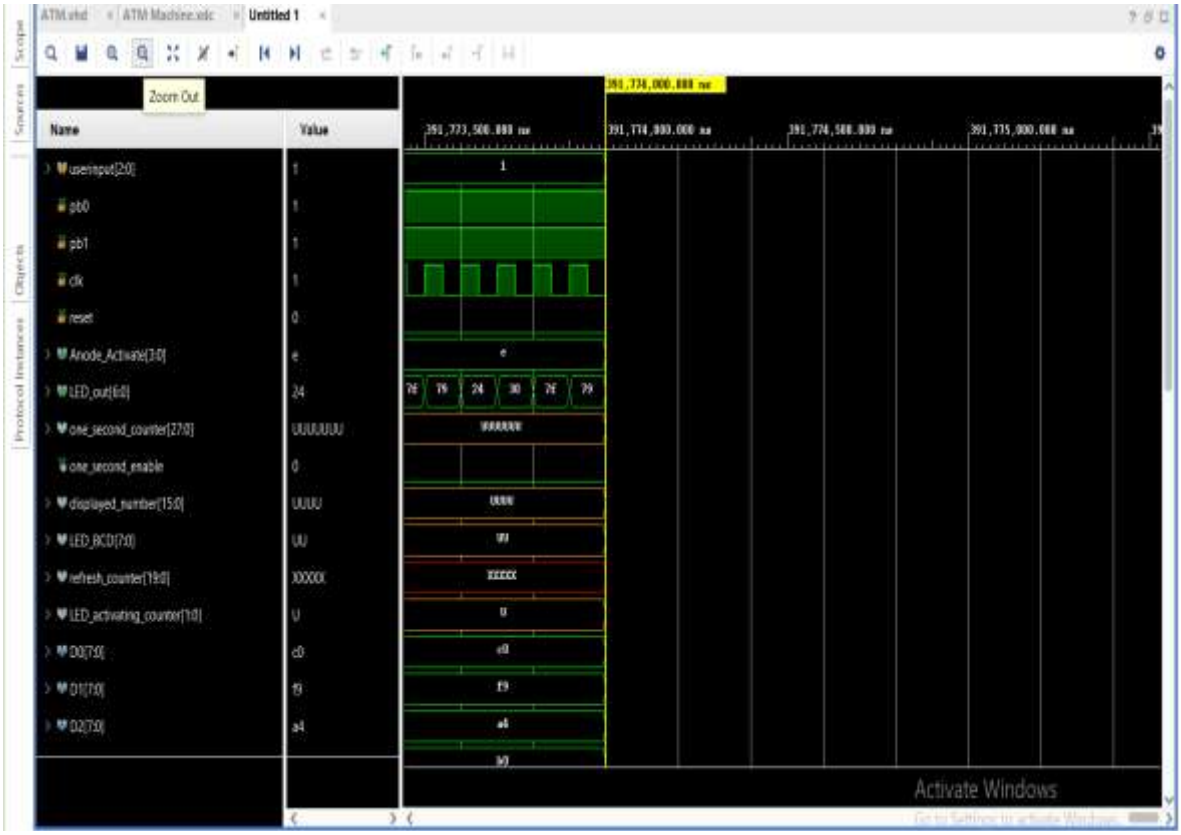
Fig.5) Block diagram of atm machine

Integration: Integrate the code with the Basys 3 board, ensuring proper hardware and software interactions. Test the system as a whole to verify its functionality.
Following this design methodology helps ensure a structured and systematic approach to designing the code for the Basys 3-based ATM system. It promotes modularity, maintainability, and reliability in the final implementation.

V. SIMULATION RESULTS

The state diagram is implemented on vivado software and we get output as this simulation. User input is the data or information that is entered into a system by the user it is used to determine what output the system will produce.
In the ATM entity, the input signals are userinput, pb0,pb1,clk and reset .

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 7 July 2023, pp: 158-167 www.ijaem.net ISSN: 2395-5252



The output signals are Anode_Activate and LED_output

As we have given the inputs as pb0 and pb1 are input signal that represent users selection of particular operation. The values of pb0 and pb1 depend on specific operation that user has selected. Here are the values of pb0 and pb1 for some common operation:

Withdrawal : pb0 = 0 , pb1 = 1

Deposite : pb0 =0 , pb1 = 1

Balance inquiry: pb0= 1 , pb = 1

Transfer : pb0 = 1 , pb = 1

The reset input is a signal that is sent to the ATM to reset its state and prepare it for new

Transcation. The clk is a digital signal that is used to synchronize the operation of ATM entity. The values of clk signal depend on specific implementation of ATM entity .

Synthesis Report:

A digital diagram of BASYS 3 is a schematic representation of BASYS 3 board, which is complete, ready to use digital circuit development platform based on latest Artix- 7^tm field programmable gate array (FPGA) from Xilinx the diagram shows various ports and peripherals that are available on the board, such as switches, LED's , buttons , 7 segment display, vga , usb , pmod connectors , etc. the diagram also shows power supply circuit and FPGA configuration circuit. The digital digram of BYSYS 3 can help us to understand hoe to connect use the board for different design and application.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 7 July 2023, pp: 158-167 www.ijaem.net ISSN: 2395-5252

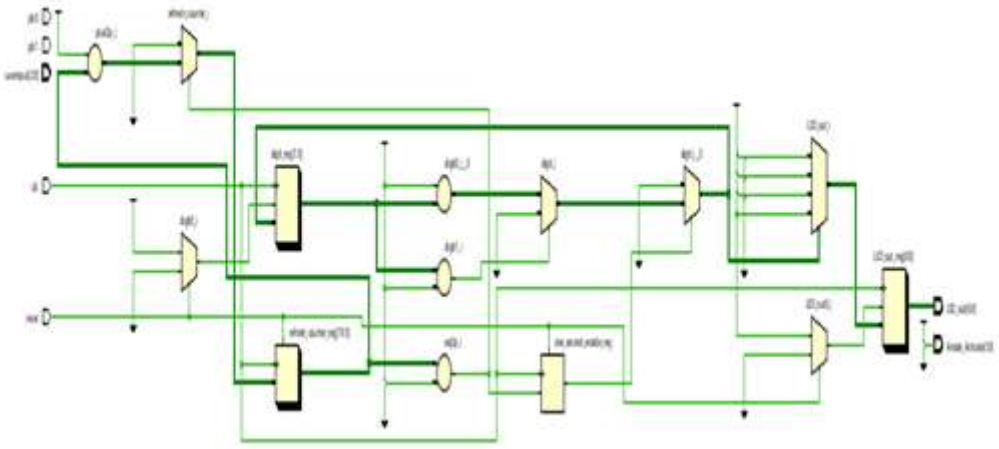
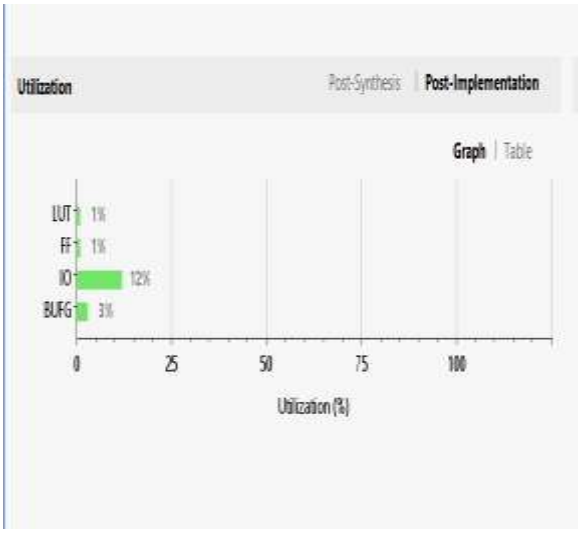


Fig.6) RTL design

Utilization:

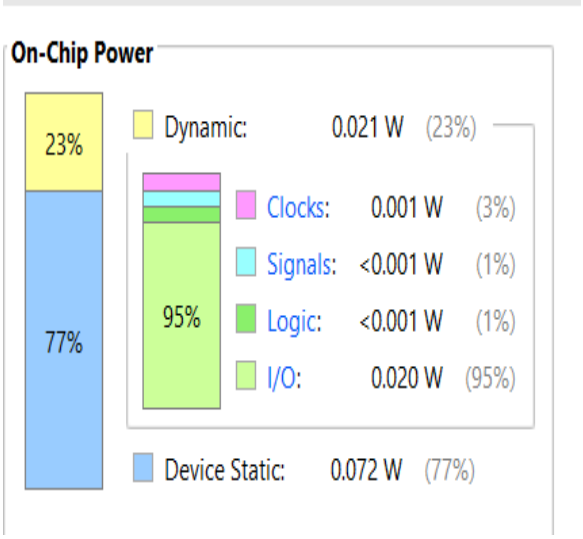
Figure shows graph of utilization of code. In which IO is used most that is 12 %, and other LUT, FF and BUFG are used 1 %, 1%, 3% respectively.



POWER:

In the below digram the power uses table and graph is given that tells us about how much power is used all the components like here on chip power is used that is 0.093 W.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 7 July 2023, pp: 158-167 www.ijaem.net ISSN: 2395-5252



In this total on chip power is 0.093W.
Here design power budget is not specified and the power budget margin is null, junction temperature is 25.5-degree C. Thermal margin is 59.5-degree C (11.8 W) Effective u JA is 5.0C/W
Power supplied to off-chip devices: 0W. The Confidence level is Medium.

VI. CONCLUSION

In conclusion, the implementation of an ATM machine controller using VHDL provides an efficient and reliable solution for managing the functionality of an ATM system. By utilising VHDL, we can design and simulate the various components of the controller, including the user interface, transaction processing, cash dispensing, and security mechanisms.

The ATM machine controller developed in VHDL allows for modular and scalable design, making it easier to integrate new features or adapt to changing requirements. VHDL's hardware description capabilities enable us to capture the complex behaviour and interactions within the ATM system accurately.

Through the development process, we gained insights into the design considerations and challenges involved in building an ATM machine controller. VHDL's ability to model concurrent processes and simulate their interactions facilitated the verification and testing of the controller's functionality, ensuring its correctness and reliability.

Overall, the use of VHDL in implementing an ATM machine controller offers numerous advantages, including flexibility, modularity, scalability, and efficient simulation capabilities. It serves as a foundation for creating robust and secure ATM systems, meeting the demanding requirements of modern banking and financial services.

REFERENCES

[1]. Yingxu Wang and Yanan Zhang, "The Formal Design Model of an Automatic Teller Machine (ATM)" University of Calgary, Canada, International Journal of Software Science and Computational Intelligence, 2(1), 102-131, January-March 2010.
[2]. Avenet Avenue, user's guide, Xilinx® Spartan™-3 Development Kit.
[3]. Stanley MAZOR and Patricia LINGSTRAAT, "A guide to VHDL (2nd Edition)", copyright 1993, Kluwer Achedamic Publishers, pp no: 1-1 to 7-16.
[4]. http://www.fpga.com.cn/hdl/training/Vhdl_Golden_Reference_Guide.pdf
[5]. Peter J. Ashenden and Jim Lewis, "The Designer's Guide to VHDL (3rd Edition)" copyright 2008, Morgan KJaufmann Publication, pp no:207-225.
[6]. Pong P. Chu "RTL Hardware Design Using VHDL, Coding For Efficiency, portability and Scalability", Willy Interscience a john wilay & sons, inc., Publication, pp no:23-160.
[7]. Hardware Implementation of Watchdog Timer for Application in ATM Machine Using Verilog and FPGA Iqbalur Rahman Rokon, Toufiq Rahman, Md. Murtoza Ali Quader, and Mukit Alam,International Conference on Electronics, Biomedical Engineering and its Applications (ICEBEA'2012) Jan. 7-8, 2012 Dubai

AUTHORS:
[1] First Author – Manali Dhar, M.Tech (Microelectronics & VLSI Design), Dr. Sudhir Chandra Sur Degree Engineering College, Assistant Professor, Dept. of ECE, JIS Group and madhuja4u@gmail.com.
[2] Second Author – Debolina Roy, Final Year B.Tech Student (Electronics & Communication Engineering), Dr. Sudhir Chandra Sur Degree Engineering College, JIS Group .
[3] Third Author – Tamosha Saha, Final Year B.Tech Student (Electronics & Communication Engineering), Dr. Sudhir Chandra Sur Degree Engineering College, JIS Group.