FPGA BASED ELECTRONIC VOTING MACHINE

Abstract

This project focuses on implementing an electronic voting machine (EVM) on a Field-Programmable Gate Array (FPGA) platform. The objective is to create a secure and efficient system for electronic voting, ensuring integrity and confidentiality. The FPGA serves as the hardware platform, enabling real-time processing and robust control. The design incorporates cryptographic techniques to safeguard voting data, and the system aims to provide a user-friendly interface while addressing key challenges in electronic voting systems. The project emphasizes the importance of reliability, security, and accessibility in modernizing the electoral process through FPGA-based EVM implementation.

Introduction:

Electronic voting machines (EVMs) have revolutionized the electoral process by providing a secure, accurate, and efficient means of voting. Unlike traditional paper ballots, EVMs are electronic devices that allow voters to cast their votes by pressing a button or touching a screen. These machines are equipped with advanced technology to ensure the integrity and confidentiality of the voting process. EVMs have been widely adopted around the world due to their ability to minimize errors, reduce the time required to count votes, and enhance the overall transparency of elections.

Designing an Electronic Voting Machine (EVM) on a FieldProgrammable Gate Array (FPGA) offers a flexible and efficient solution for implementing secure and reliable voting systems. FPGA-based EVMs can provide real-time processing, high-speed data transfer, and robust security features. This presentation will explore the key components and advantages of an FPGA-based EVM design, highlighting its potential for enhancing the electoral process

Motivation:

1.Secure voting system: FPGA-based EVMs can provide secure and tamper-resistant voting systems, ensuring the integrity of the voting process.

2.Real time processing: FPGAs allow for real-time processing of votes, enabling quick and efficient tabulation of results.

3.High speed data transfer: FPGAs support high-speed data transfer, which is essential for transmitting voting data securely and quickly.

4.Auditability: FPGAs can be programmed to provide auditability features, allowing for the verification of votes and ensuring transparency in the voting process.

5.Scalability: FPGA-based EVMs can be easily scaled to accommodate different voting needs, from small local elections to large national elections.

Literature Review:

- Describes more about power efficiency. This EVM works for three parties. It has a check input to select a party and then a "select" button to vote. It outputs counts for all three parties. It lacks authentication input. It does not show the voter whether the vote is cast or not. It doesn't have separate access for voters and officers; even voters can see election results.

- EVM works for three parties in another model. It has a voter_enable signal given by the polling officer, which allows voters to cast their votes. It has three seven-segment display outputs with active low enables to show the numbers of votes cast for each party. It also has a Dout output to check the total number of votes cast. It has an opled output corresponding to each party to show voters whether they cast their vote for the intended party. It accepts more than one vote at a time. There is no need to use four different displays when we already have registers to store values; we can select them using another input.

- The Voting Machine is implemented on the FPGA NEXYS-4 board in another model. It works for four candidates. It has a mode input which enables voting when set to 0 and shows the count when set to 1. A vote is considered valid only if the button is pressed for one second. There is no authentication input in this model. It also accepts more than one vote at a time.

Objectives

- **Ballot Casting Design**: Implement FPGA logic to securely handle the voting process.

- **Accuracy and Reliability**: Ensure accurate and reliable vote counting.

- **Security**: Implement robust security measures to prevent tampering and ensure the integrity of the voting system.

- **Transparency**: Provide a transparent and auditable system that allows voters and election authorities to verify the correctness of the voting process.

- **Cost-effectiveness**: Design cost-effective and scalable solutions for electronic voting using FPGAs.

- **Speed and Efficiency**: Speed up the voting process and enable quick and efficient tabulation for easy elections.

Methodology

Xilinx Vivado tool is used for Verilog coding. We have employed the Nexys4 FPGA for this voting machine, which supports up to 255 voters and 4 candidates.

Inputs: Push buttons are assigned for each candidate, alongside an authentication input. Additionally, there is a "mode" input which toggles between enabling voting (when reset) and displaying the vote count (when set).
Outputs: The system features 4 LEDs to indicate which button each voter has pressed, one LED to indicate whether their vote has been cast, and an 8-bit vote count output showing the number of votes for the selected candidate.

Our main module is `voting_mc`, which accepts inputs and includes instantiations of a validation module. This module checks the validity of the vote based on the time the button is pressed and authentication status. If the vote for a candidate is valid and there are no valid votes for other candidates, the count for that specific candidate is incremented by 1. Subsequently, it outputs a signal (`person_voted`) to indicate that the voter has cast their vote, prompting the polling officer to lower the authentication signal (or perform another action to prevent double voting). This functionality is active when the mode is set to 0.

When the mode is set to 1, the system displays the number of votes for the selected candidate if the user is authenticated and has pressed the button for at least 1 second.

Results:

LEDs corresponding to candidates light up when a button is pressed for 10 ns and authentication is high. Vote is counted, and "person_voted" goes high only when a single valid vote is recorded. After the mode input is set high, counting stops, and `vote_count` is displayed based on the selected candidate.

fig1

Four push buttons are assigned for each candidate's inputs. The first slide switch is used for reset, the second for mode, and the third for authentication. The leftmost LEDs represent `led1`, `led2`, `led3`, `led4`, corresponding to each candidate, indicating when their respective button has been pressed for more than 1 second. The middle LED glows if a valid vote is counted (i.e., when a valid vote is recorded).

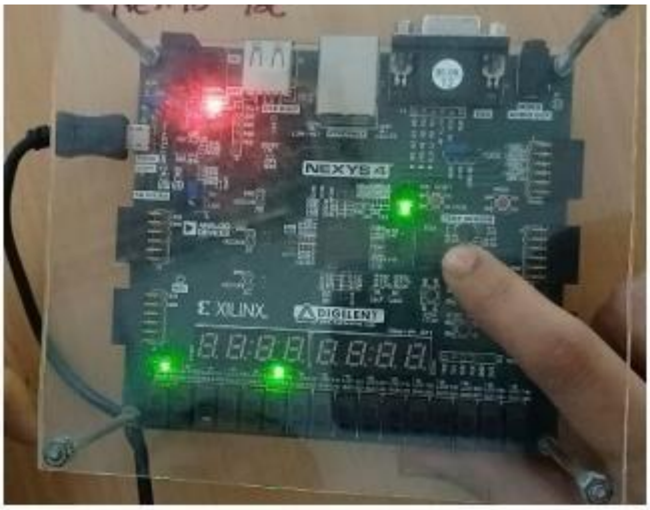

fig2

In this configuration, reset is set to 0, mode is set to 0, and authentication is set to Therefore, when the button corresponding to the first candidate is pressed for more than a second, and no other button is pressed simultaneously, `led1` and the `person_voted` LED light up.

fig3

When mode is set to 1 and authentication is also set to 1, pressing the button corresponding to the first candidate displays the number of votes cast for that specific candidate using the 8 rightmost LEDs.

Conclusion

The `voting_mc` module represents a versatile Verilog implementation of a voting system designed to manage and count votes for up to four candidates. It offers comprehensive functionality including vote validation, mode selection for voting or displaying counts, LED indication of voting actions, and confirmation of cast votes. This module provides a robust framework suitable for implementing diverse types of voting systems in hardware, ensuring reliability and accuracy in electoral processes. Its flexibility and capability make it wellsuited for a broad spectrum of applications, from local elections to more complex voting scenarios requiring secure and efficient electronic voting solutions.

Future scope

Accessibility Features: Consider adding features to make the voting machine more accessible to a wider range of voters, such as support for different languages or interfaces for voters with disabilities.

Data Analytics: Explore the use of data analytics to analyze voting patterns and trends, which could provide valuable insights for election organizers.

Voter Verification: Implement additional methods for verifying voters' identities, such as biometric authentication or two-factor authentication.