Digital Lock System with Intruder Detection Using DE10-Lite FPGA and Raspberry Pi

Member (2414202, *sai2414202@iitgoa.ac.in*)

Introduction

This project focuses on designing a Digital Lock System with Intruder Detection that utilizes an FPGA for passcode management and a Raspberry Pi for detecting and recording unauthorized access attempts. The system verifies passcodes, displays feedback on seven-segment displays (SSDs), and captures an image of intruders after multiple failed attempts, ensuring a secure and responsive locking mechanism.

System Overview

A block diagram of the system is given in Fig. 1.

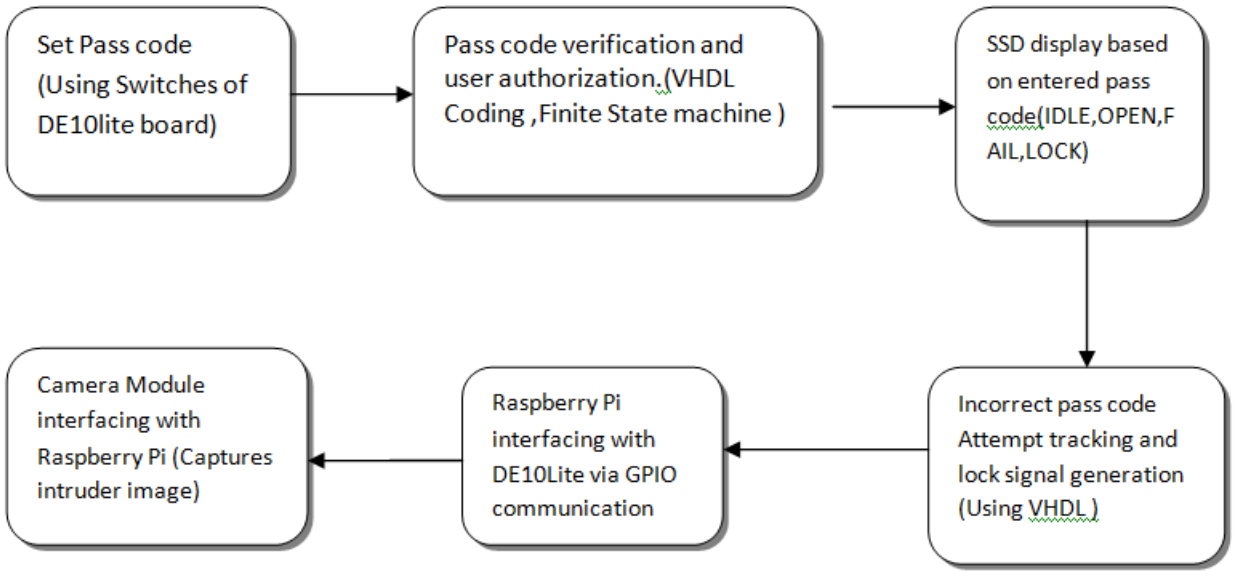

Fig. 1:Block Diagram of Digital Lock System.

- **DE10-Lite FPGA:** The DE10-Lite FPGA handles the core logic of the system, including passcode verification, state transitions, and feedback management. It implements a finite state machine (FSM) with key states such as `IDLE`, `ENTER_PASS`, `CHECK_PASS`, `PASS`, `FAIL`, and `LOCK` to determine whether access is granted or denied. The FPGA also controls the seven-segment displays (SSDs) to provide real-time feedback based on the current state.
- **Raspberry Pi:** The Raspberry Pi is interfaced with the DE10-Lite FPGA via GPIO and serves as a secondary layer of security. Upon receiving a lock signal from the FPGA after multiple failed passcode attempts, the Raspberry Pi captures and stores an image of the intruder. This image is saved locally for future reference.
- **Camera Module:** The camera module is connected to the Raspberry Pi and is triggered during unauthorized attempts. It captures and stores the image of the intruder locally for future reference, enhancing the overall security.
- **Seven-Segment Displays (SSDs):** The SSDs provide visual feedback to the user, displaying the system's current status(ex:IDLE,OPEN,LOCK,ENTER).

Implementation Details

- **VHDL Coding and State Machine Logic:** The core functionality of the system is implemented using VHDL, where a finite state machine (FSM) handles the passcode verification and system behavior. The FSM transitions

through the following states:

* `IDLE`: The system is waiting for user input.
* `ENTER_PASS`: The user enters the passcode.
* `CHECK_PASS`: The system verifies the entered passcode.
* `PASS`: Access is granted if the passcode is correct.
* `FAIL`: The system records a failed attempt and increments the failure count.
* `LOCK`: After three failed attempts, the system locks and triggers the Raspberry Pi to capture an intruder's image.

The VHDL logic controls state transitions based on the passcode entered and failed attempt count. It ensures that the system remains in the `LOCK` state after three failed attempts, requiring a reset to resume normal operation. Additionally, the state machine controls the output to the SSDs, providing feedback to the user at each stage.

– **Seven-Segment Displays (SSDs):** The seven-segment displays are controlled by the FPGA to show real-time feedback based on the current system state. The system displays the following:

* `OPEN`: For a successful passcode entry.
* `FAIL`: For an incorrect passcode.
* `LOCK`: When the system locks due to multiple failed attempts.
* `ENTR`: When the system is in the `ENTER_PASS` state.

– **Raspberry Pi Integration:** The Raspberry Pi is interfaced with the DE10-Lite FPGA via GPIO pins. Upon receiving a lock signal from the FPGA after three failed attempts, the Raspberry Pi activates the camera module to capture an intruder's image. The Raspberry Pi communicates with the FPGA through specific GPIO pins for synchronization and control.

– **Camera Module:** The camera module is connected to the Raspberry Pi and is triggered through GPIO pins when the system detects multiple failed passcode attempts. It captures and stores images of the intruder, which are saved locally on the Raspberry Pi for future reference.

– **Reset Mechanism:** The reset mechanism clears the failed attempts counter and returns the system to the `IDLE` state, enabling the user to enter a new passcode. The reset function is controlled via the FPGA and uses GPIO pins for communication with the Raspberry Pi.

Results

Figures 1 to 5 show the FSM's performance, including the idle state, passcode entry, successful unlocking, and failed attempt scenarios.

Conclusion

This project demonstrates the feasibility and effectiveness of integrating FPGA-based state machines with Raspberry Pi for implementing a robust digital lock system. By combining hardware control with software interfacing, the system provides reliable passcode verification and intruder detection capabilities. Future improvements could include adding wireless connectivity for remote monitoring and using advanced authentication methods such as biometrics.

References

[1] A. Kumar,P.S.Singh, and R. Verma,"Design and implementation of a combinational lock state system using VHDL," *ResearchGate*, 2022. https://www.researchgate.net/publication/362751143$_{D}esign_{a}nd_{i}mplementation_{o}f_{a}combina$

[2] M. C. Chen, C. H. Lai, and J. H. Wu, "A new VHDL implementation of a digital lock system," *IEEE Xplore*, 2020. https://ieeexplore.ieee.org/document/9242688/

[3] J. S. Lee, S. W. Park, and J. H. Kim, "Smart Lock System using FPGA and VHDL," *IEEE Xplore*, 2017. https://ieeexplore.ieee.org/abstract/document/8079785

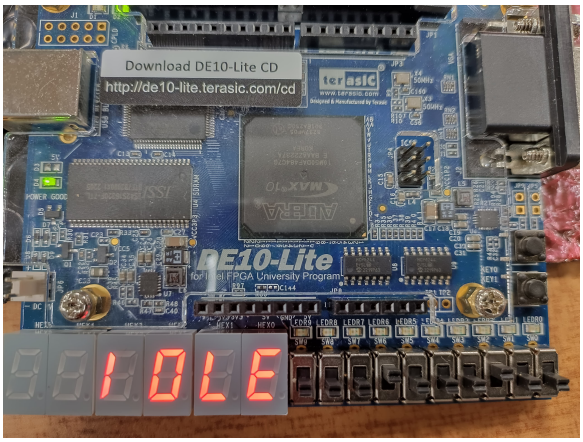

Figure 1: Image1

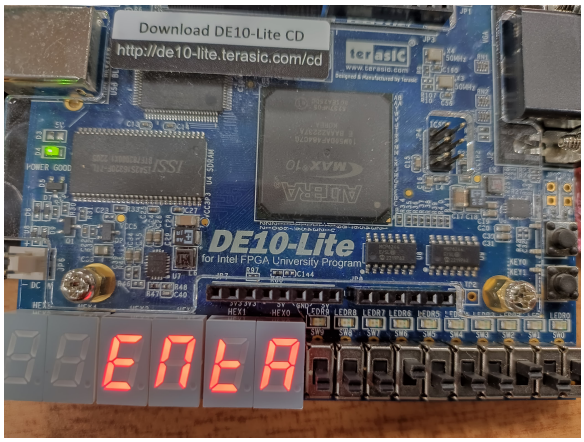

Figure 2: Image2

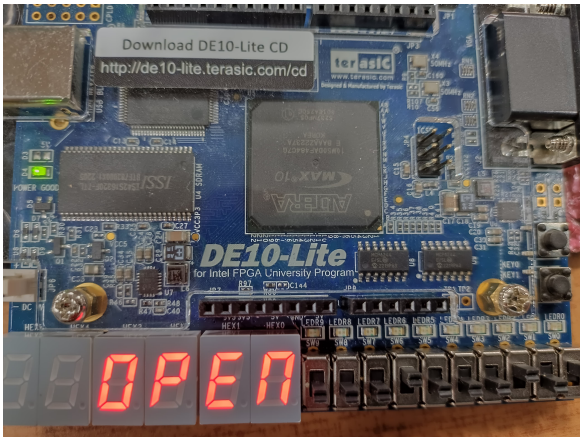

Figure 3: Image3

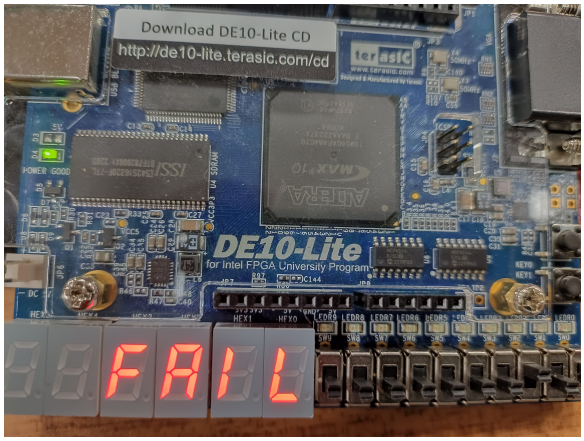

Figure 4: Image4

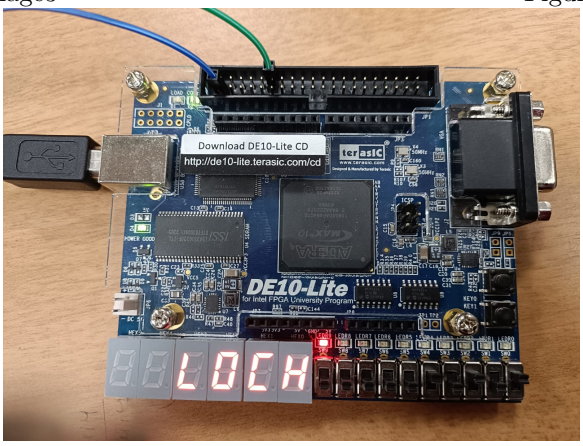

Figure 5: Image5