

Project Title: FPGA-Based Digital Key Lock System with Configurable Security Code

1. Project Overview

This project involves the design, simulation and implementation of a Digital Key Lock System using VHDL on an Artix-7 FPGA (Basys 3 development board). The system utilizes a 4x4 Matrix Keypad for user input and LED indicators to display the system status (Locked, Unlocked, or Typing).

The core logic is based on a Moore Finite State Machine that manages security validation, user input processing and visual feedback. A key feature of this design is the configurable password capability, by allowing the user to overwrite the default factory code with a custom 4 digit code during runtime.

2. Objectives

a. Primary Objective:

- To design a robust, synchronous digital system that emulates a real-world security access control mechanism.

b. Specific Objectives:

- Implement a modular architecture separating the hardware driver, Keypad Controller, from the control logic.
- Develop a debounce algorithm to filter mechanical noise from the keypad switches.
- Design a secure Finite State Machine capable of validating inputs and managing programmable registers.
- Verify system integrity through exhaustive behavioral simulation on the Test Bench covering edge cases and configuration modes.
- Generate the final Bitstream for physical implementation on the Xilinx Artix-7 architecture.

3. System Architecture & Modules

The design follows a hierarchical structure to ensure modularity and ease of debugging.

a. Top Level Module (Top_Level)

- Function: Acts as the physical wrapper of the system.
- Operation: It instantiates the Keypad_Controller and the FSM_Controller and interconnects them. It maps the internal signals to the physical FPGA ports defined in the constraints file.
- I/O: connects the 100MHz system clock, external Reset button, Keypad Rows/Cols and Status LEDs.

b. Keypad Controller Module (keypad_controller)

- Role: The Hardware Driver.
- Functioning:
 - Scanning: It continuously scans the 4 columns of the keypad at a frequency of 1 kHz.
 - Debouncing: It implements a 20ms delay counter to filter out mechanical bouncing noise on physical buttons.
 - Decoding: Translates the Row/Column coordinates into a 4 bit binary value from 0 to 15.
 - One Shot Pulse: Generates a clean, single cycle key_pressed signal to the FSM whenever a valid key press is detected.

c. FSM Controller Module (fsm_controller)

- Role: The Control Logic (The Brain).
- Topology: Moore Machine (Outputs depend only on the current state for stability).
- Datapath: Features parallel registers to store the "User Input" and the "Master Key".
- States:
 - S_IDLE: System locked (Red LED ON). Waits for input.
 - S_READ_X: Sequential states to capture the 4 digit user entry.
 - S_SET_KEY_X: Configuration mode states to write a new password into the internal registers.
 - S_UNLOCKED: Access granted (Green LED ON). Holds state for 0.5s using a timer.
 - S_FAIL: Access denied (Red LED ON). Holds state for 0.5s using a timer.
- Operation Logic:
 - Default Key: Hardcoded reset value (1-3-5-7).
 - Validation: Uses a parallel comparator logic. The 4th digit is compared in real-time upon entry.
 - Configuration Trigger: Pressing the '*' key in IDLE state transitions the system to programming mode.

4. Simulation & Verification Strategy

The system was verified using a comprehensive Test Bench (Top_Level_tb) in Vivado. The simulation mimics real world timing (ms) rather than just clock cycles in ns.

Test Bench Scenarios

The simulation performs 4 critical tests sequentially over a 2500 ms run time:

- Integrity Test (Default Key):
 - Action: Input 1-3-5-7.
 - Expected Result: System transitions to UNLOCKED (Green LED ON).
 - Outcome: PASSED.
- Configuration Test:
 - Action: Input * followed by 2-4-6-8.
 - Expected Result: System accepts input without unlocking, updates internal registers and returns to IDLE.
 - Outcome: PASSED.
- Security Test (Old Key):
 - Action: Input the old key 1-3-5-7.
 - Expected Result: System detects mismatch against new registers and transitions to FAIL (Red LED ON).
 - Outcome: PASSED.
- Validation Test (New Key):
 - Action: Input the new key 2-4-6-8.
 - Expected Result: System validates the new code and transitions to UNLOCKED.
 - Outcome: PASSED.

5. Implementation Details

- Target Device: Xilinx Artix-7 (xc7a35tcpg236-1).
- Development Environment: Xilinx Vivado 2025.1 ML Standard.
- Resource Utilization: The design utilizes a minimal footprint of Look Up Tables and Flip Flops, leaving room for future expansion.
- Clocking: Uses the onboard 100MHz oscillator, internally divided for scanning and timing logic.

- Pin Constraints (.xdc): Mapped to the Basys 3 PMOD Header JA (for the keypad) and onboard LEDs/Buttons.

6. Future Improvements

While the current system meets all design requirements, the following improvements are identified for future iterations:

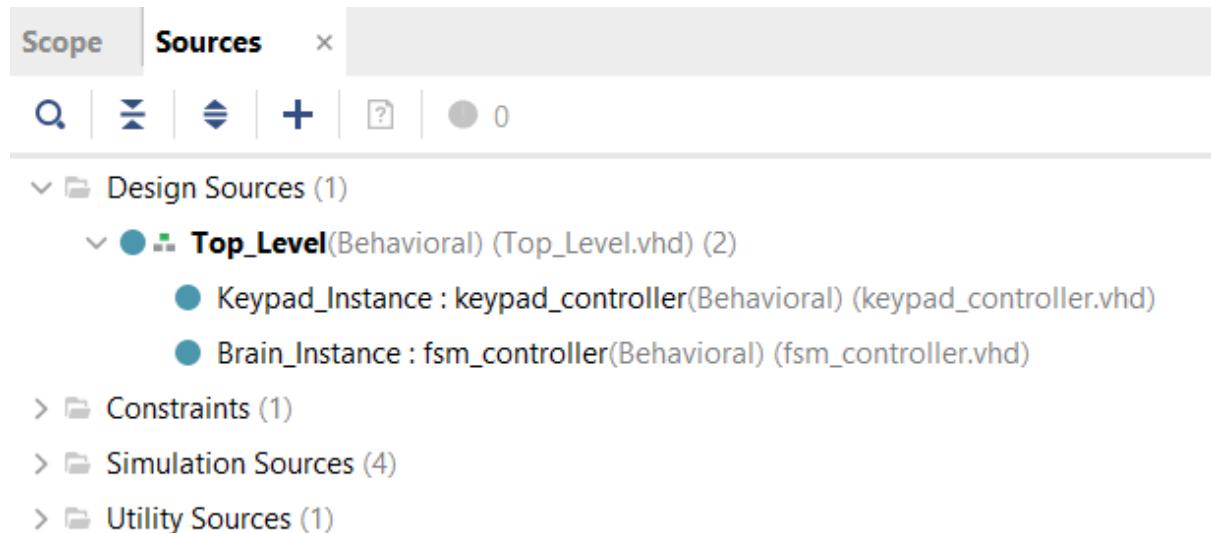
- Non-Volatile Memory: Integration of an SPI Flash Controller to store the password permanently, ensuring it persists after a power cycle.
- Inactivity Timeout: Implementation of a watchdog timer to reset the FSM to IDLE if the user stops typing halfway through a sequence.
- Variable Length Codes: Modifying the datapath to accept passwords between 4 and 8 digits.

7. Conclusion

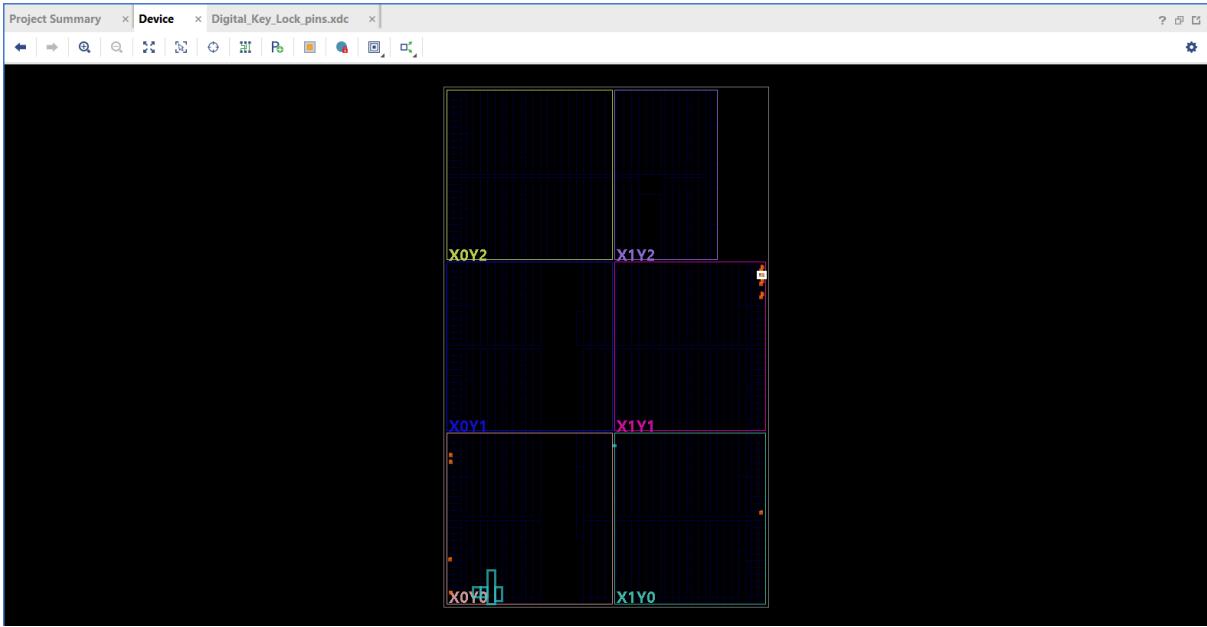
The "Digital Key Lock" project was successfully designed, simulated and implemented. The modular design approach proved effective for debugging and feature expansion. The addition of the configurable password feature elevates the project complexity, demonstrating a solid understanding of synchronous logic, register manipulation and finite state machine design. The successful generation of the bitstream confirms the design is valid for physical hardware deployment.

8. Images

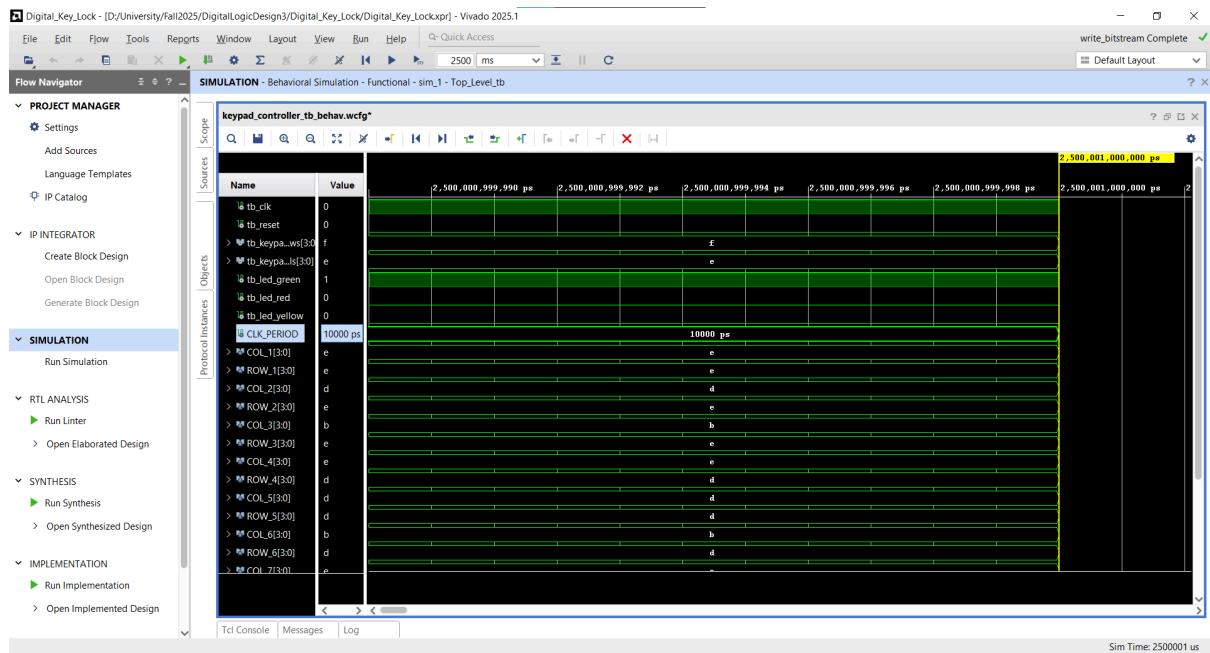
- Module Hierarchy and Structural Design



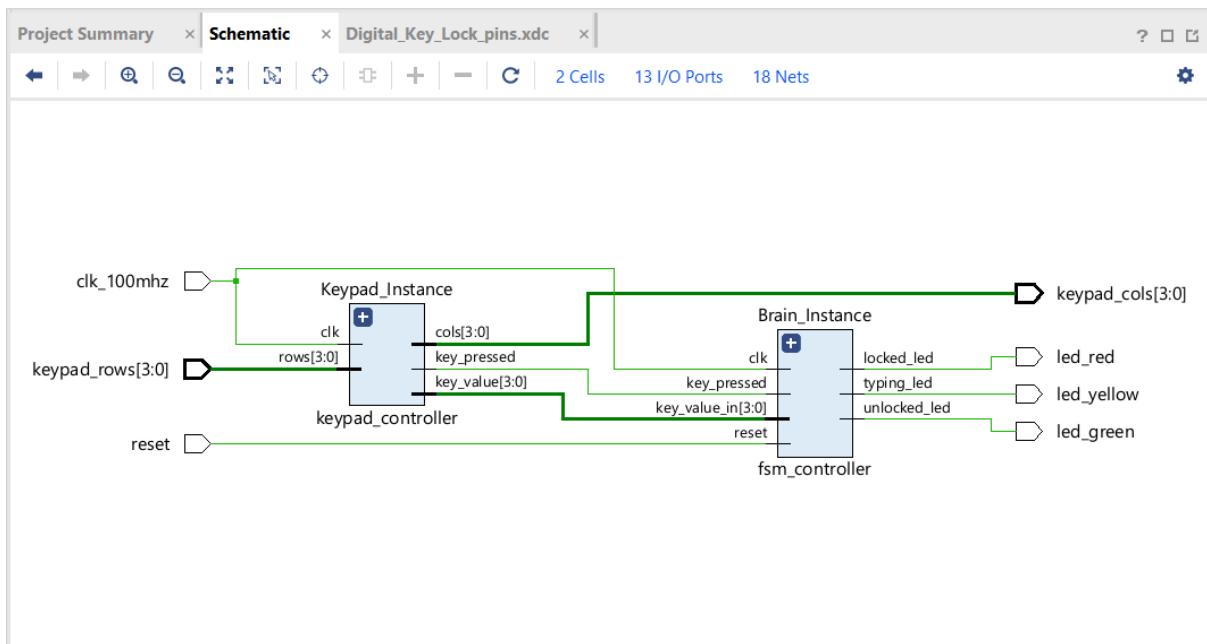
● Chip View



● Waveform detail



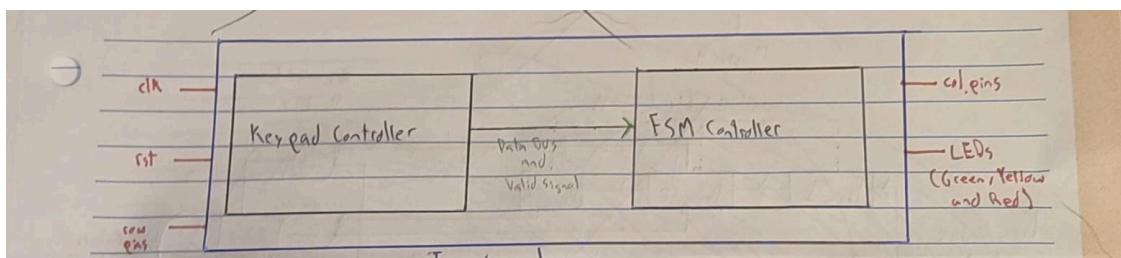
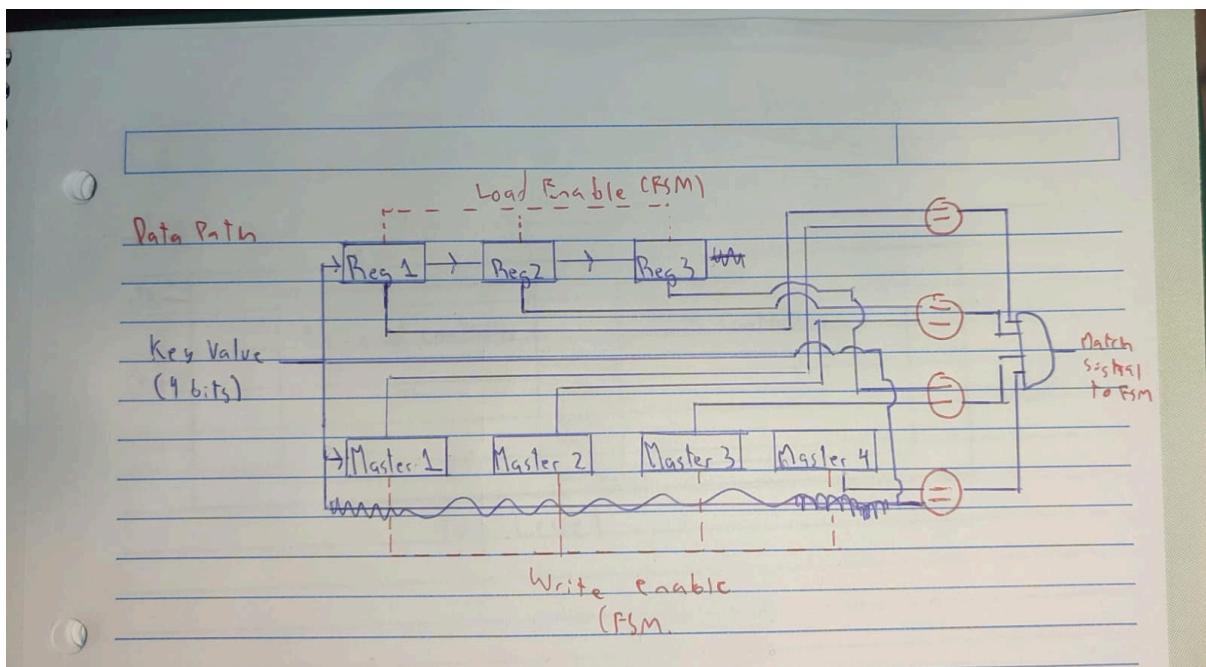
- Schematic



• FINAL SIMULATION RESULTS (SUCCESS)

```
Tcl Console x Messages Log ? - _ |  
|  
# } else {  
#   send_msg_id Add_Wave-1 WARNING "No top level signals found. Simulator will start without a wave window. If you want to open a wave window go to 'File->New Waveform"  
# }  
# }  
# run 1000ns  
Note: --- Starting Full System Simulation (v2.0) ---  
Time: 0 ps Iteration: 0 Process: /Top_Level_tb/stimulus_process File: D:/University/Fall12025/DigitalLogicDesign3/Digital_Key_Lock/Digital_Key_Lock.srcs/sim_1/new/Top_Level_tb.v  
INFO: [USP-XSim-96] XSim completed. Design snapshot 'Top_Level_tb_behav' loaded.  
INFO: [USP-XSim-97] XSim simulation ran for 1000ns  
launch_simulation: Time (s): cpu = 00:00:13 ; elapsed = 00:00:17 . Memory (MB): peak = 1645.926 ; gain = 24.332  
run 2500 ms  
Note: Reset done. System in S_IDLE. Default key is 1-3-5-7.  
Time: 1100 ns Iteration: 0 Process: /Top_Level_tb/stimulus_process File: D:/University/Fall12025/DigitalLogicDesign3/Digital_Key_Lock/Digital_Key_Lock.srcs/sim_1/new/Top_Level_tb.v  
Note: --- Test 1: Trying Default Passcode (1-3-5-7) ---  
Time: 1100 ns Iteration: 0 Process: /Top_Level_tb/stimulus_process File: D:/University/Fall12025/DigitalLogicDesign3/Digital_Key_Lock/Digital_Key_Lock.srcs/sim_1/new/Top_Level_tb.v  
Note: SUCCESS (Test 1): Default key worked! Green LED is ON.  
Time: 285599995 ns Iteration: 0 Process: /Top_Level_tb/stimulus_process File: D:/University/Fall12025/DigitalLogicDesign3/Digital_Key_Lock/Digital_Key_Lock.srcs/sim_1/new/Top_Level_tb.v  
Note: --- Test 2: Setting New Passcode to 2-4-6-8 ---  
Time: 885599995 ns Iteration: 0 Process: /Top_Level_tb/stimulus_process File: D:/University/Fall12025/DigitalLogicDesign3/Digital_Key_Lock/Digital_Key_Lock.srcs/sim_1/new/Top_Level_tb.v  
Note: Pressed "2". FSM should be in S_SET_KEY_1.  
Time: 957499995 ns Iteration: 0 Process: /Top_Level_tb/stimulus_process File: D:/University/Fall12025/DigitalLogicDesign3/Digital_Key_Lock/Digital_Key_Lock.srcs/sim_1/new/Top_Level_tb.v  
Note: New key 2-4-6-8 is set. FSM should be back in S_IDLE.  
Time: 1242499995 ns Iteration: 0 Process: /Top_Level_tb/stimulus_process File: D:/University/Fall12025/DigitalLogicDesign3/Digital_Key_Lock/Digital_Key_Lock.srcs/sim_1/new/Top_Level_tb.v  
Note: --- Test 3: Trying OLD Passcode (1-3-5-7) ---  
Time: 1242599995 ns Iteration: 0 Process: /Top_Level_tb/stimulus_process File: D:/University/Fall12025/DigitalLogicDesign3/Digital_Key_Lock/Digital_Key_Lock.srcs/sim_1/new/Top_Level_tb.v  
Note: SUCCESS (Test 3): Old key correctly FAILED! Red LED is ON.  
Time: 152599995 ns Iteration: 0 Process: /Top_Level_tb/stimulus_process File: D:/University/Fall12025/DigitalLogicDesign3/Digital_Key_Lock/Digital_Key_Lock.srcs/sim_1/new/Top_Level_tb.v  
Note: --- Test 4: Trying NEW Passcode (2-4-6-8) ---  
Time: 212599995 ns Iteration: 0 Process: /Top_Level_tb/stimulus_process File: D:/University/Fall12025/DigitalLogicDesign3/Digital_Key_Lock/Digital_Key_Lock.srcs/sim_1/new/Top_Level_tb.v  
Note: SUCCESS (Test 4): New key worked! Green LED is ON.  
Time: 241059995 ns Iteration: 0 Process: /Top_Level_tb/stimulus_process File: D:/University/Fall12025/DigitalLogicDesign3/Digital_Key_Lock/Digital_Key_Lock.srcs/sim_1/new/Top_Level_tb.v  
run: Time (s): cpu = 00:00:58 ; elapsed = 00:05:59 . Memory (MB): peak = 3303.145 ; gain = 1657.219  
|
```

- Data path, Top Level Architecture and state diagram



Top Level
Separated in case of a keypad change, the FSM won't have to get changed.

Centre FSM Controller

4 Internal Registers instead of

constants to allow rewriting
during the execution time

