Smart Vault Controller using Basys3 FPGA Development Board

ANU College of Engineering, Computing and Cybernetics
Australian National University
Canberra ACT 0200 Australia
www.anu.edu.au

+61 2 6125 5254

Student 1 ID	u7114996		
Student 1 Name	Brady Edwards		
Student 2 ID	u7310067		
Student 2 Name	Aedan Jaeson		
Course Code	ENGN4213/6213		
Course Name	Digital Systems and Microprocessors		
Assignment Item	☒ FPGA Project Report		
Word Count		Due Date	22 April 2024
Date Submitted	21 April 2024	Extension Granted	

I declare that this work:

☒ upholds the principles of academic integrity, as defined in the University Academic Integrity Rule;

☒ is original, except where collaboration (for example group work) has been authorised in writing by the course convener in the course outline and/or Wattle site;

☒ is produced for the purposes of this assessment task and has not been submitted for assessment in any other context, except where authorised in writing by the course convener;

☒ gives appropriate acknowledgement of the ideas, scholarship and intellectual property of others insofar as these have been used;

☒ in no part involves copying, cheating, collusion, fabrication, plagiarism or recycling.

Initials
For group assignments, each student must initial.

B.E
A.J

CONTRIBUTION STATEMENT

We, the undersigned members of ENGN4213/6213 FPGA Project Group No: 27, hereby state our main individual contributions.

- Climate Controller
- Parts of TOP module
- SSD Manager
- Door Access Controller (minus morse code features)
- Parts of the report

Signature 1 with name:

Brady Edwards

Date: 21/04/2024

- Parts of TOP module
- Door Access Controller (incl. morse code submodule)
- Modified Seven Segment Decoder
- Parts of the report

Signature 2 with name:

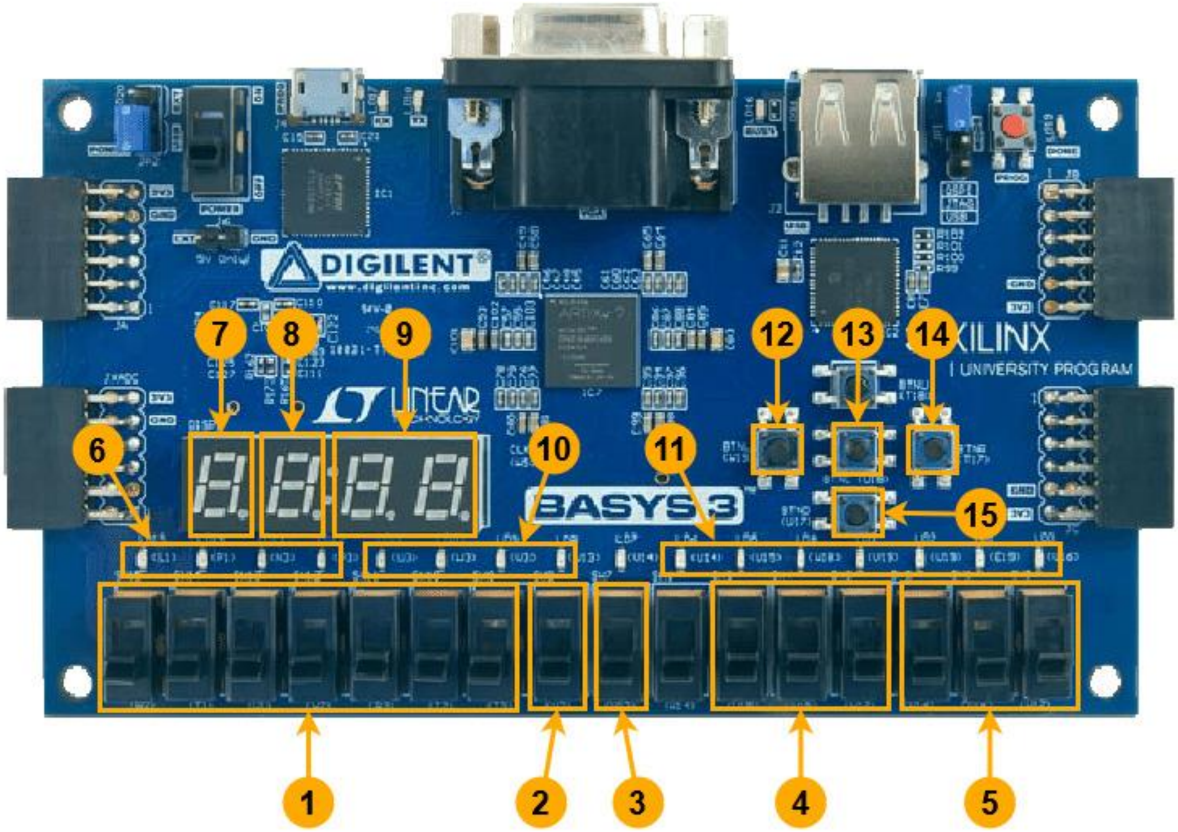
Aedan Jaeson

Date: 21/04/2024

This statement may be considered to assess fair contributions among group members and, if deemed necessary, regulate individual marking.

Contents

User Interface



Label	Component Description
1	Slide switches for **7-digit binary PIN** input.
2	Slide switch to **add people to the vault** (works on both positive and negative edges of the switch).
3	Slide switch to **remove people from the vault** (works on both positive and negative edges of the switch).
4	Slide switches to input the **outside temperature**.
5	Slide switches to input the **desired temperature**.
6	LEDs indicating the **status of the vault door**.
7	SSD indicating the **number of people in the vault**.
8	SSD indicating whether a **morse code dot or dash** has been input.
9	SSDs indicating the **current temperature** in the vault.
10	LEDs indicating the **vault alarm status**.
11	LEDs representing the **entered morse code pin** (if correct) as a **binary coded decimal**.
12	**DOOR_MASTER** button.
13	**SECURITY_RESET** button.
14	**ENTER** button.
15	**Morse code button**.

Functionalities Implemented

Function I - Vault Door Access Control

Vault Entry

When the vault door is closed, indicated by all 4 door status LEDs (**Component 6**) being ON, input the correct 7-digit account PIN (0011011) using the 7 slide switches (**Component 1**) and press ENTER button (**Component 14**).

If an incorrect PIN is entered, the vault will enter the ALARM state and the alarm LEDs (**Component 10**) will toggle every 2 seconds for 20 seconds. During this time, you may try to input the correct PIN again. If the correct PIN is entered, the alarm LEDs (**Component 10**) will turn off and the door will open, otherwise the vault will enter the TRAP state and the alarm LEDs will toggle every 0.5 seconds. To disable the alarm during the ALARM or TRAP state, press the security reset button (**Component 13**) or in the TRAP state, input the 7-digit security PIN (1010101) using the 7 slide switches (**Component 1**) and press ENTER (**Component 14**).

To open the door from the outside at any time, press DOOR_MASTER (**12**).

While the door is opening/closing, the door status LEDs (**Component 6**) will turn ON/OFF one-by-one until after 4 seconds the door is either completely open (all LEDs ON) or closed (all LEDs OFF).

Vault Open

While the vault is open, the door will start closing after 30 seconds. People may enter or exit the vault while the door is completely open using the respective slide switches (**Component 2** for entry, **Component 3** for exit). The number of people in the vault at any one time is displayed on the left SSD (**Component 7**).

Vault Exit

To open the vault door when there are people inside, a correct 10-digit morse sequence must be input using the morse code button (**Component 15**). After 10 digits have been input, if the sequence is correct the vault door will open and the binary coded equivalent of the 10-digit code will be displayed on the LEDs (**Component 11**). If the 10-digit sequence is incorrect, the vault door will remain closed and the morse code sequence may be re-attempted any number of times.

To register a dot, hold the button down for 0.5-1.5 seconds, to register a dash, hold the button down for 1.5+ seconds. The second SSD (**Component 8**) will indicate whether a dot or dash has been input.

Function II – Automated Climate Control

While the vault is occupied, the two right SSDs (**Component 9**) will turn on, indicating that the climate controller is active. These two SSDs will display the current temperature inside the vault. The climate controller will adjust the vault's temperature to match the desired temperature, which is specified by the 3 right-most slide switches (**Component 5**). The rate at which the temperature changes is determined by the number of people in the vault. 2° per second if 5 or less people are in the vault, and 1° per second if there are more than 5 people.

Once the vault has been vacated (the door is open, and no one is left in the vault), the temperature will start to equalise towards the outside temperature at a rate of 1° per 2 seconds which is specified by 3 slide switches (**Component 4**). After 10 seconds since the last person exited the vault, the climate controller will stop changing the temperature, then 20 seconds later the current temperature SSDs (**Component 9**) will turn off.

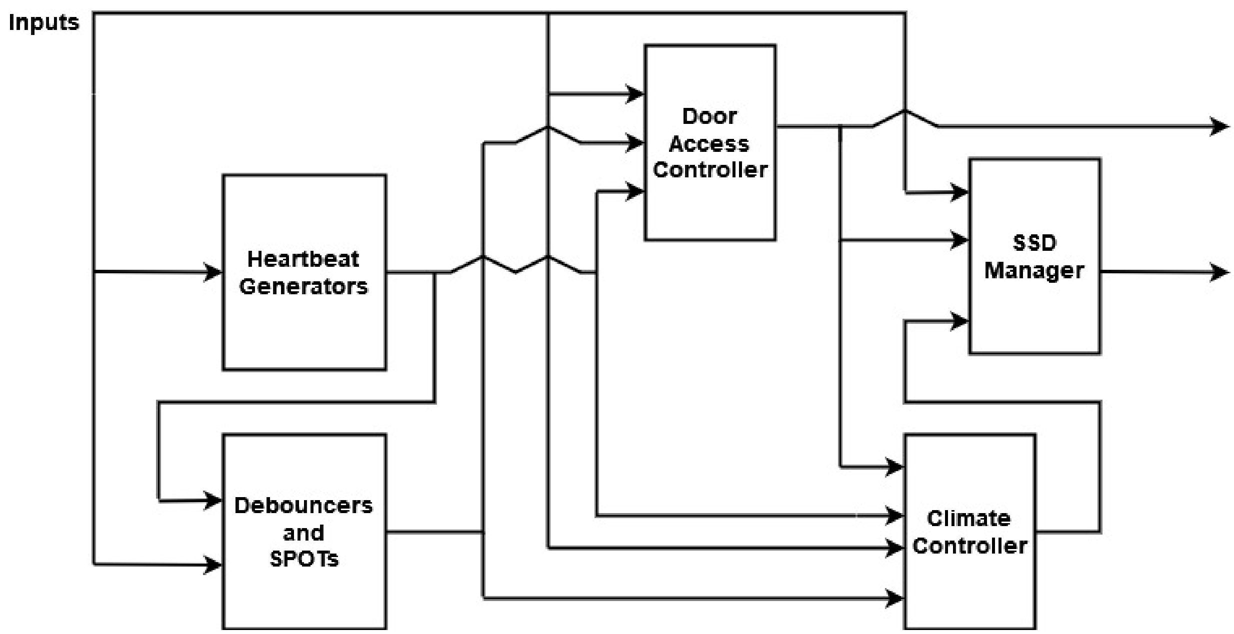
Project Structure

Top Module

Overview

This module integrates both the Vault Door Access Control and Automated Climate Control features. It is also responsible for processing all input/output data to the configured IO components on the Basys3 board.

Block Circuit Diagram



Block	Description
Heartbeat Generators	Generates synchronized timing signals used across various modules.
Debouncers and SPOTs	Filters out noise from user inputs to produce stable signals for processing
Door Access Controller	Manages access control and security features of the vault door
Climate Controller	Automates the climate control within the vault based on occupancy and temperature settings
SSD Manager	Manages the display of information based on the seven segment displays

The Door Access Controller and Climate Controller modules operate independently, except for the number of people being passed from the Door Access module to the Climate Controller. Both modules interface with the SSD manager, which displays relevant status information without influencing their logic. Heartbeat generators supply timing signals across the system, while debouncers and SPOTs ensure clean input signals.

Design Choices

The smart vault controller's top module is the main orchestrator, tasked with integrating various submodules in a coherent system.

The choice to implement debouncers and single pulse triggers (SPOTs) arises from the inherent physical characteristics of the Basys3 FPGA's input mechanisms. Mechanical switches and push buttons are prone to 'bouncing' – a common issue where the electrical contact "bounces" upon actuation, causing multiple transitions. This can lead to erroneous readings, hence the need for debouncing to filter these extraneous signals, ensuring that only clean, stable transitions reach the system's logic.

SPOTs were used alongside debouncers for buttons that need to trigger an action on a single press. This is for critical operations such as the ENTER function, where a single clean input is needed to process the user's input accurately. The morse code button was debounced, but not passed through a SPOT module, as its operation required output signals longer than a single clock pulse.

Resource Saving

By instantiating heartbeat generators used by multiple modules only once and distributing their signals, the design conserves resources that would otherwise be used up by multiple redundant timing mechanisms. This shared approach to heartbeat signals aligns with the synchronous nature of the system and supports the concurrent functioning of the door control logic, Morse code decoding, and climate control systems.

Seven Segment Display Manager

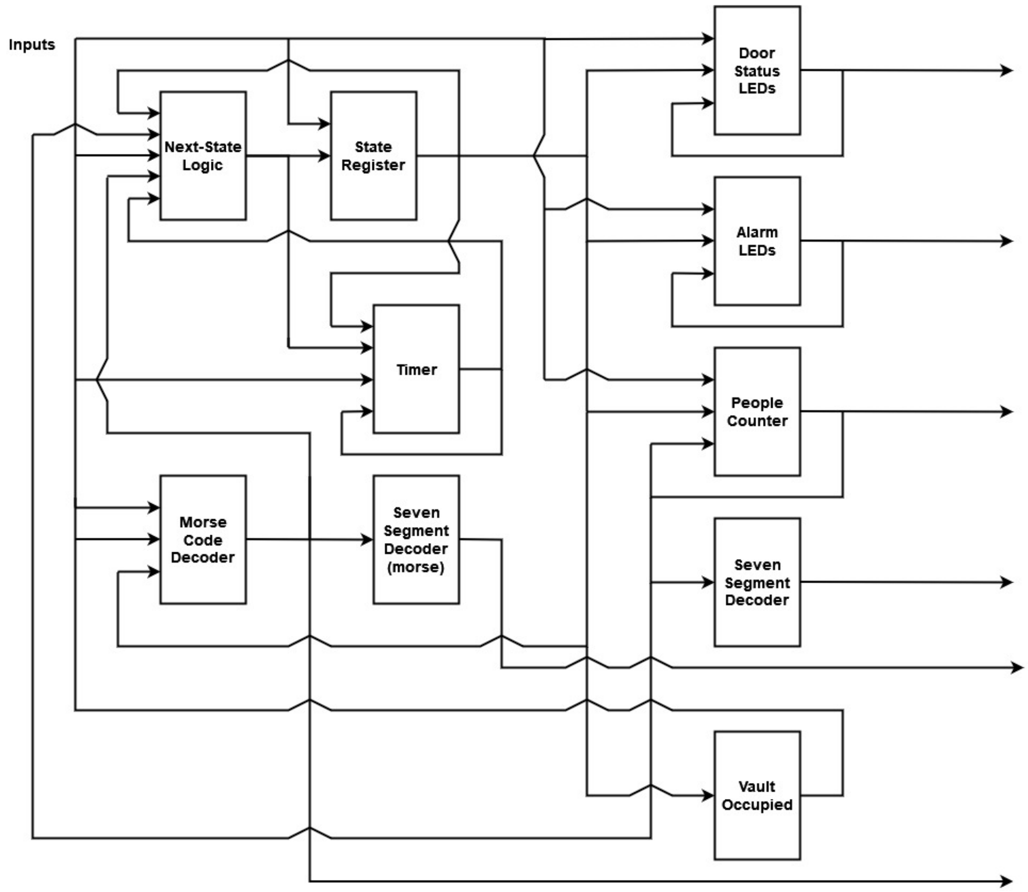
The SSD manager interprets binary-coded inputs and governs the display across multiple SSDs. It multiplexes display data to the appropriate SSD based on a rapid cycling mechanism, creating the illusion of persistent visualisation.

Door Access Controller

Overview

This module manages all features related to the access control of the vault and operating the vault door.

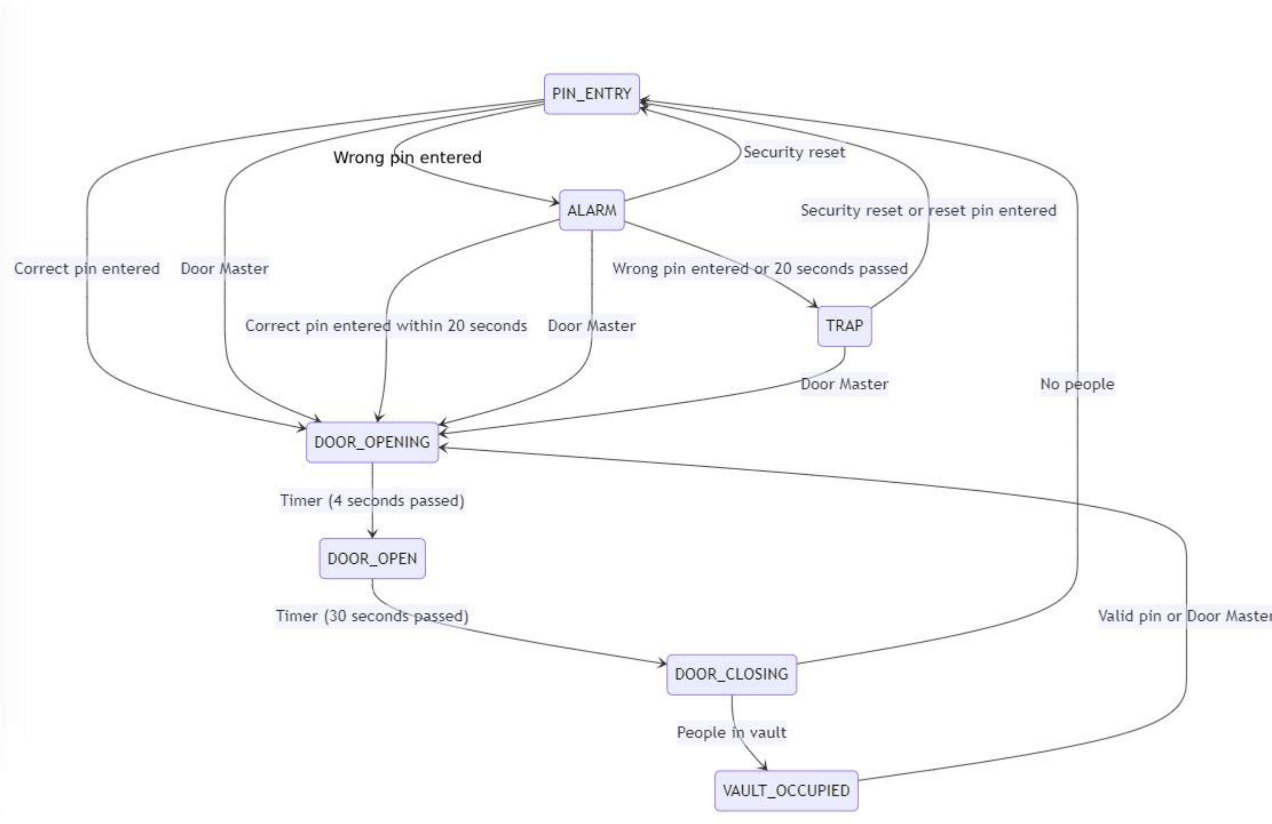
Block Circuit Diagram



Block Name	Description
Next-State Logic	Combinational logic block using outputs from various other blocks to determine the logic for state transitions.
State Register	A clocked process that changes the state if next-state is different to the current state.
Timer	A clocked process that increments a counter every second to implement timing features in this submodule. The timer is reset whenever there is a state change.
Morse Code Decoder	A sub-module to handle the morse code feature for opening the vault door when the vault is occupied.
Seven Segment Decoder (morse)	Converts the binary coded decimal output of the Morse Code Decoder to an 8-bit register for the seven-segment display (the 8th bit is the dot).
Door Status LEDs	A clocked process that handles the animation of the opening/closing of the vault door, and sets the LEDs to either ON or OFF depending on if the vault door is open or closed.
Alarm LEDs	A clocked process that handles the flashing alarm LEDs if in the ALARM or TRAP state.
People Counter	A clocked process that adds/removes people from the vault based on the corresponding input switches and the current state.

Seven Segment Decoder	Converts the binary coded decimal from People Counter to an 8-bit register for the seven-segment display (8th bit is the dot).
Vault Occupied	Combinational logic block that sets a 1-bit register to indicate whether the vault is occupied or not (as other modules such as Morse Code Decoder, and Climate Controller use this).

State Diagram



FSM design

State	Description
PIN_ENTRY	Represents the vault door being closed and empty, waiting for an official to enter the correct entry PIN.
ALARM	Represents the alarm state of the vault, where an incorrect PIN was entered in the PIN_ENTRY state.
TRAP	Represents the trap structure being enclosed due to multiple failed entry attempts, or if the correct PIN was not entered within 20 seconds of entering the ALARM state.
DOOR_OPENING	Represents the vault door opening and only lasts for 4 seconds as the door status LEDs are turning off one-by-one.
DOOR_OPEN	Represents the vault door being completely open, enabling people to enter and exit the vault.

DOOR_CLOSING	Represents the vault door closing and only lasts for 4 seconds as the door status LEDs are turning off one-by-one.
VAULT_OCCUPIED	Represents the vault door being closed and the vault being occupied by at least 1 person.

We chose to use an FSM to implement the logic of this module because it compartmentalises the features into separate and clear logical states. We chose to use a 3-bit register to store the 7 states (1 state is left unused). We chose to use binary encoding over grey encoding and one-hot encoding, as the state transition frequency and number of states is low. This means that the likelihood of encountering metastability is lower as there were very few bit flips required for each state transition.

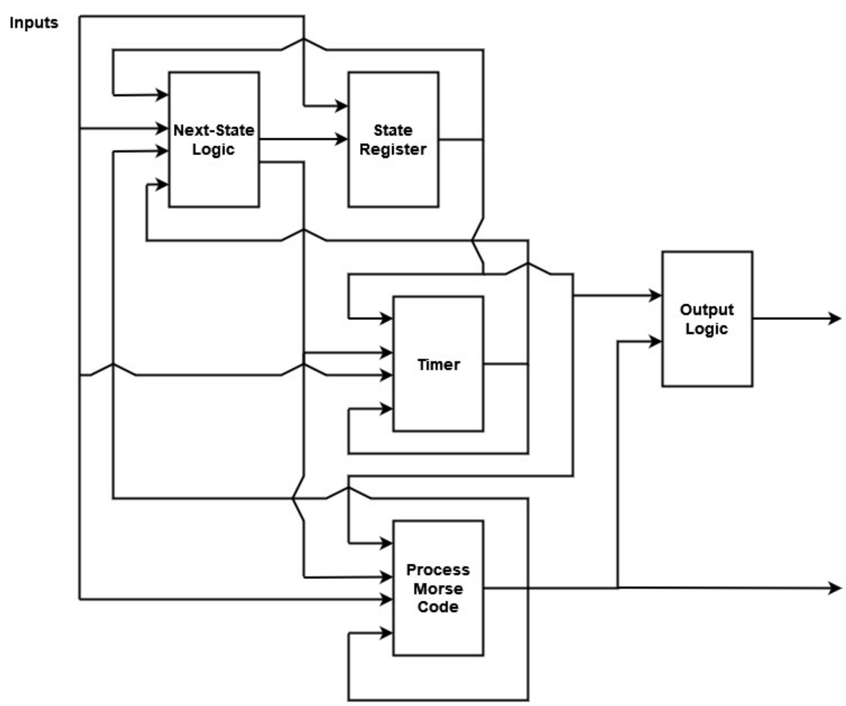
Output Logic

The implementation of an FSM for this module is slightly modified to better suit the task. Instead of using a combinational logic block for the output logic, we used multiple clocked processes to modify all output registers. This choice was suitable because all output registers from the door access controller operate independently of one another, so there was no need to conditionally map certain registers to different output wires/registers.

Morse Code Decoder

This is a submodule within the Door Access Controller. It handles all logic and processing required to implement the morse code vault exit features.

Block Circuit Diagram

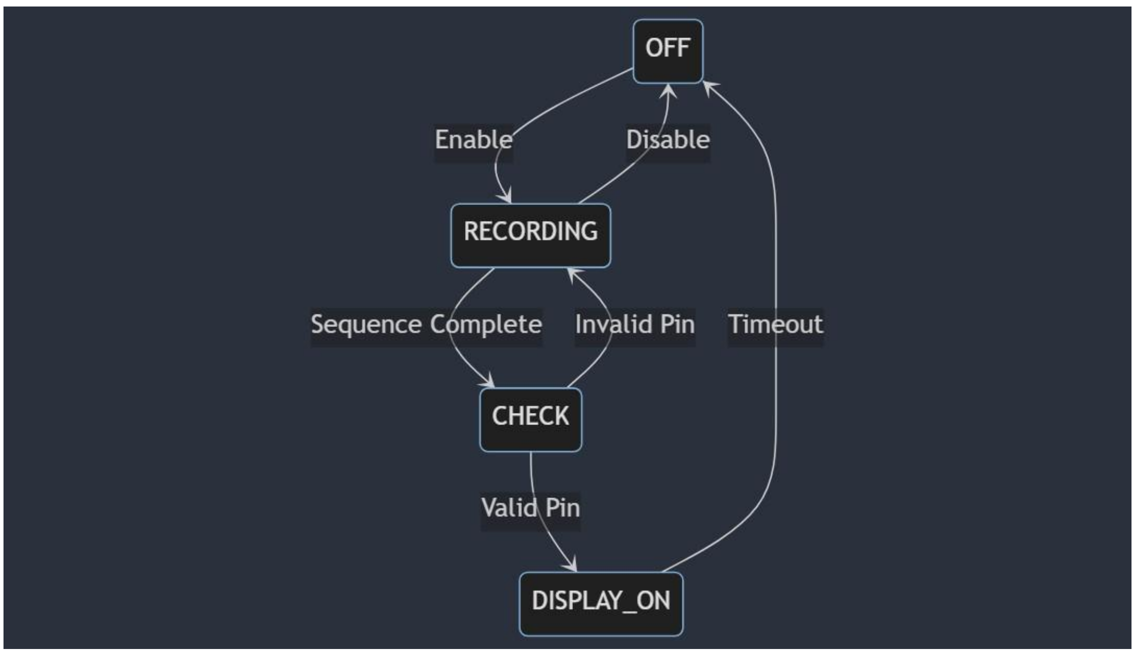


FSM Design

Block	Description
Next-State Logic	Determines the next state of the FSM based on the current state and the input conditions. It evaluates whether the button press indicates a dot, dash, or no input and whether the sequence is complete.

State Register	Holds the current state of the FSM. It is updated with the value from the Next-State Logic block at every clock cycle, effectively changing the state of the FSM as required.
Timer	Tracks time intervals for different states. It is used to measure the duration of the Morse code inputs (dot or dash) and to manage timing for state transitions, such as moving from RECORDING to CHECK.
Process Morse Code	Interprets the Morse code inputs by measuring the duration of the button press and updating the pin sequence accordingly. It also resets the sequence after processing and prepares for new input.
Output Logic	Manages the outputs based on the current state of the FSM. In the DISPLAY_ON state, it controls the LED display to provide feedback of a valid sequence. It also manages the resetting of the display after the sequence has been shown.

State Diagram



The Morse Code Decoder operates through a finite state machine (FSM) that includes the following states:

State	Description
OFF	The initial state where the system is inactive awaits activation. Activation occurs when the vault is detected as occupied.
RECORDING	In this state, the system records inputs from the Morse code button. It distinguishes between dots and dashes based on input duration and stores them in a buffer to construct the pin sequence.
CHECK	Once a complete sequence of 10 signals is recorded, this state initiates. Here, the sequence is compared against predefined valid sequences to determine its validity.
DISPLAY_ON	If the sequence is valid, this state is engaged. The corresponding pin, in binary coded decimal (BCD) format is displayed on the LEDs to provide user feedback. This state is temporary, allowing the user to see before resetting.

The selection of states was implemented to manage each phase of the Morse code input process distinctly: Idle/waiting, input capture, verification, and feedback display. This structured approach simplifies the logic from a potentially complex series of conditional checks to a more manageable and efficient state-driven process.

Design Choices

A design decision was made to clear the SSD display both when an invalid pin was entered and when the input was less than 0.5 seconds. This was put in place as opposed to leaving the last previous input on the display. For the former case, this gives the user clear indication to re-enter the Morse sequence, and for the latter, this helps the user understand that the input entered was not registered. Both enhance the user interface and reduce potential confusion.

The 10-bit ‘pinSequence’ register handles sequence input efficiently. By using a shift operation, the design can append new inputs to the existing sequence in a resource-effective manner.

Inclusion of the advanced feature showing the BCD equivalent on the LEDs provided an immediate readable indication of the Morse code input’s validity, improving user feedback. We also employed a case statement approach to manage valid sequences, creating a concise and easily manageable list that aids in validating the entered sequence.

Assumptions

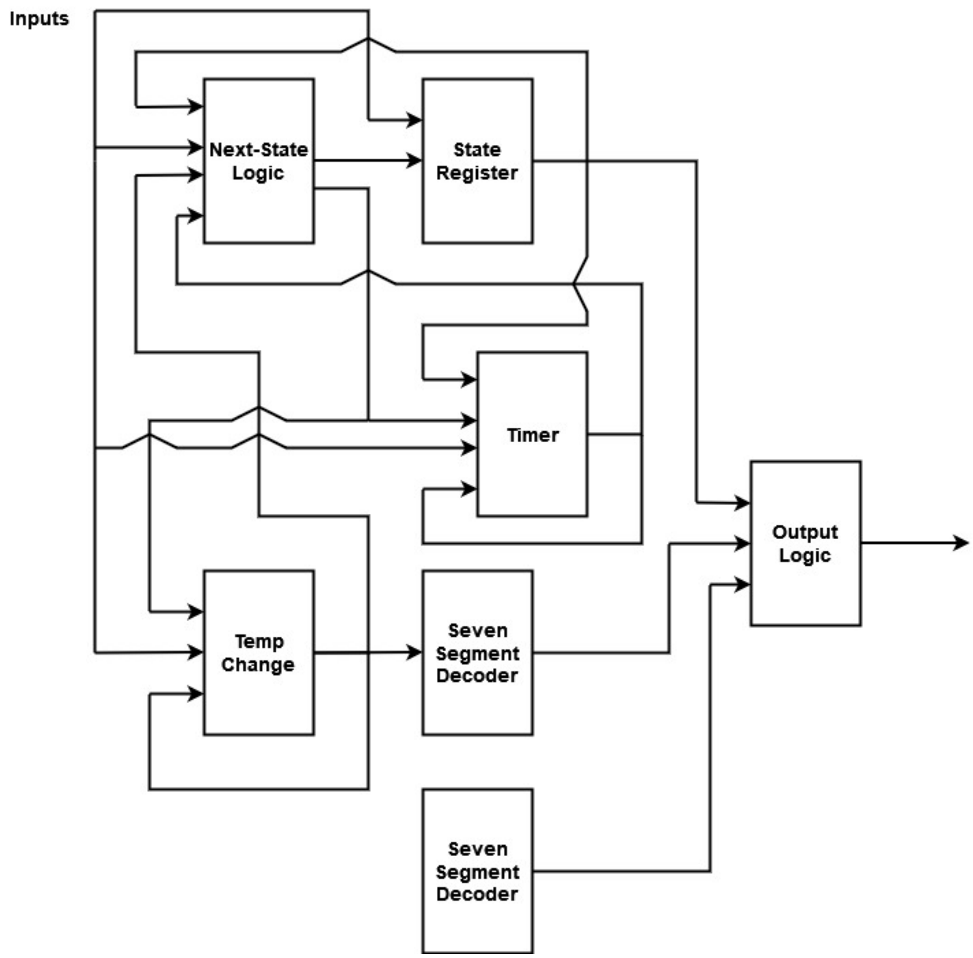
- Button presses shorter than 0.5 seconds are accidental and do not form part of the Morse code sequence.
- Upon entering a Morse code sequence, the system assumes the need to clear the display after checking for validity. If an invalid sequence is entered, the SSD display resets to prompt the user to re-enter the sequence.

Climate Controller

Overview

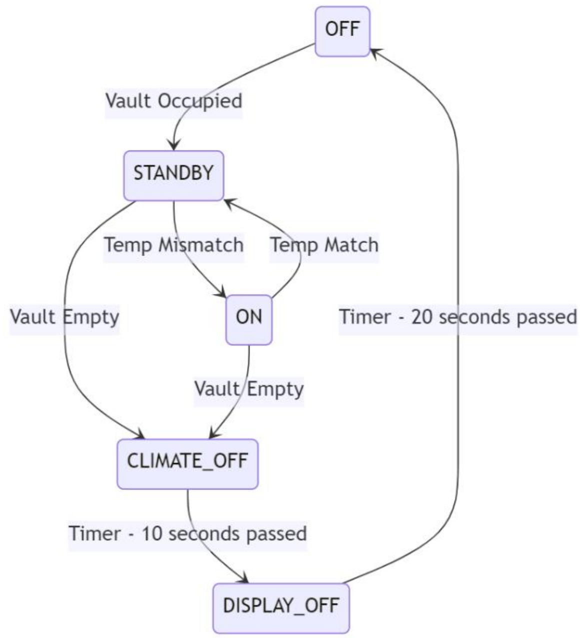
This module manages all features related to the vault’s Automated Climate Control.

Block Circuit Diagram



Block Name	Description
Next-State Logic	Combinational logic block using outputs from various other blocks to determine the logic for state transitions.
State Register	A clocked process that changes the state if next-state is different to the current state.
Timer	A clocked process that increments a counter every second to implement timing features in this submodule. The timer is reset whenever there is a state change.
Temp Change	A clocked process that increments or decrements the current temperature towards the desired temperature of the climate controller is ON.
Seven Segment Decoders	Convert binary coded decimals to decimal digits on a seven-segment-display. The bottom SSD block is only ever “2” or blank.
Output Logic	Maps the seven-segment-display registers to the output SSD registers when the display should be on (based on the current state).

State Diagram



FSM design

State	Description
OFF	This state represents the Climate Controller being completely off, including the display SSDs.
STANDBY	This state represents the Climate Controller being on, but not actively increasing/decreasing the temperature because the current and desired temperature are the same.
ON	This state represents the Climate Controller being on and actively changing the current temperature toward the desired temperature.
CLIMATE_OFF	This state represents the Climate Controller preparing to turn off due to the last person exiting the vault. In this state the temperature will begin to equalise to the outside temperature. This state only lasts for 10 seconds.
DISPLAY_OFF	This state represents the Climate Controller being completely off and preparing to turn the display SSDs off after 20 seconds.

We chose to use an FSM to separate the climate controller functionalities into separate logical states. We chose to use a 3-bit register to store the 5 states (3 states are left unused). We chose to use binary encoding over grey encoding and one-hot encoding, as the state transition frequency and number of states is low. This means that the likelihood of encountering metastability is lower as there were very few bit flips required for each state transition.

Assumptions

The Climate Control function requirement:

> *"If the occupied vault is being vacated, the climate control will turn OFF in 10 seconds after the last person exits the vault.*
>
> - *In this condition, room temperature decrease/increase towards the outside temperature at the rate of 1° per 2 seconds."*

It’s assumed that the temperature stops being equalised to the outside temperature after the given 10 seconds. For example, if the current temperature does not equal the outside temperature after 10 seconds, the current temperature will not change until the climate controller turns back on again when people start re-entering the vault.

Additionally, we assume that:

> *“By default, ... the current room temperature will be equal to the outside temperature.”*

Is applied when the program is deployed onto the FPGA, not when the climate controller first turns on (as this conflicts with the previous assumption made above).