Course Name: EMBEDDED SYSTEMS I / III
Course Number and Section: **14:332:493:03 / 16:332:579:05**
Year: Spring 2024

Embedded Systems Final Project

Home Security System

Student Name and RUID: Deshna Doshi, 206009273

Date Submitted: 5/6/2024

GitHub Link:
https://github.com/embedded-systems-1-spring-2024-labs/final-project-deshnadoshi

Table of Contents

I. Introduction

For this final project, I have chosen to develop a Home Security System that relies on four distinct PMODs with unique functionality and the Zybo Z7 board for its implementation. This report will discuss the technical and non-technical aspects of the Home Security System and details of its implementation.

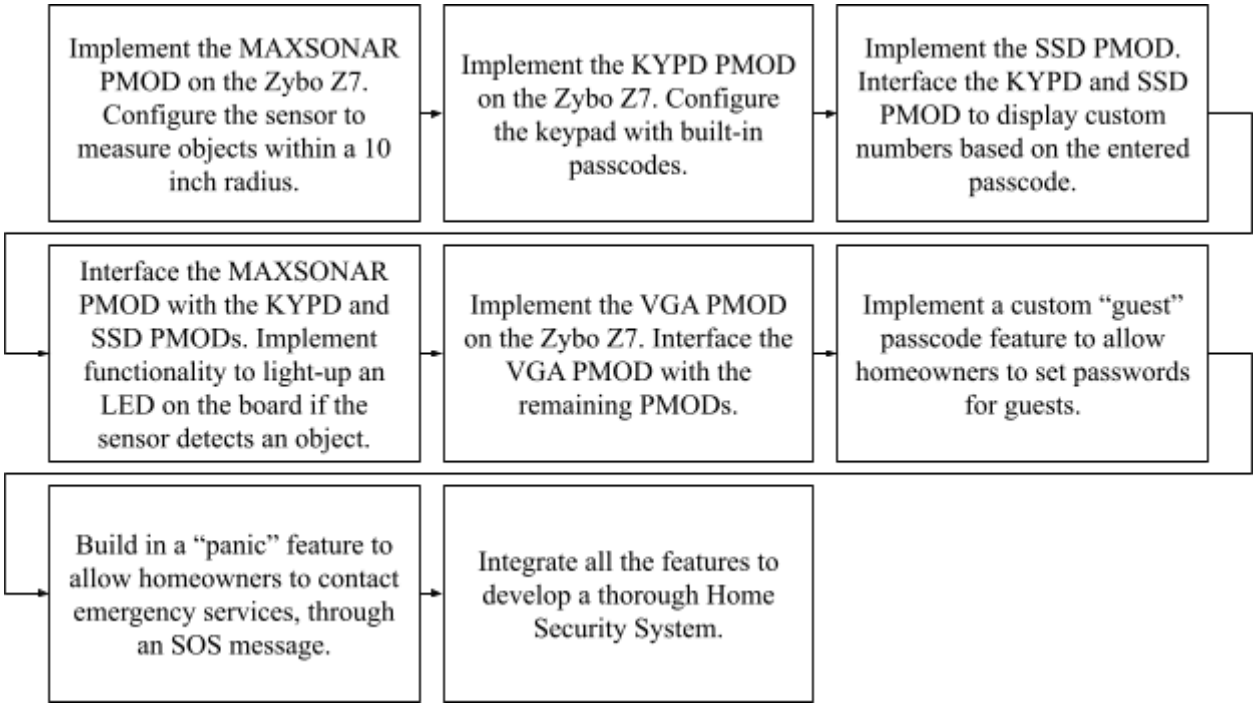
The inspiration for this project stemmed from the ubiquitous need for security measures in all fields of work, ranging from home security to commercial industries or government industries. Through this project I attempted to model existing home security systems that implement various sensors, cameras, and built-in programs to contact emergency responders. I chose to use similar PMODs that could achieve these goals. The Ultrasonic Range Sensor was used to mimic motion detection or trespassing, the 16-Button Keypad was used for an alarm locking/unlocking mechanism, the Seven Segment Display was used for displaying who triggered the alarm, and the Video Graphics Array was used to mimic emergency communication.

II. Purpose/Objective

The purpose of this project was to design an embedded system running on the Zybo Z7 board with the use of two or more PMODs. I chose to use the Ultrasonic Range Finder, 16-Button Keypad, Seven Segment Display, and Video Graphics Array to implement a Home Security System. The embedded system combines the functionality of all four PMODs and the functionality of the Zybo Z7 board to design a well-rounded security system.

While the target user of this project was home owners, this project is aimed at providing solutions for security concerns in industrial and commercial spaces as well. The ultimate objective in designing this embedded system was to safeguard a person's property and belongings.

III. Theory of Operation



IV. Component Descriptions & Simulation Diagrams/Schematics

A. PMODs

For this project, I have used four PMODs: Ultrasonic Range Finder (MAXSONAR), 16-Button Keypad (KYPD), Seven Segment Display, and Video Graphics Array (VGA). I combined the functionality of all four PMODs to design my Home Security System.

The purpose of the Ultrasonic Range Finder was to determine if an object or person was within a certain distance from the sensor. In other words, it was used to determine if they had entered a restricted area. Due to the limitations of the Ultrasonic Range Finder, I chose for the sensor to detect any object that appears within ten inches.

The purpose of the 16-Button Keypad was to function as an unlocking system. There were three built-in codes for owners of the house to bypass the sensor, 'A', 'B', and 'F'. There is also a fourth code that can be customized for a guest.

The purpose of the Seven Segment Display was to determine the ID of the person that unlocked the sensor and to detect if there was an intruder. 'A', 'B', and 'F', were used for the built-in codes. '6' represented when the system was unlocked by a guest user. '1' represented an intruder.

The purpose of the Video Graphics Array was to integrate the panic button functionality. If there was a need for a user to purposefully trigger the alarm, there would be an external display that showed an ‘SOS’ symbol.

These four PMODs were integrated with the buttons, switches, and LEDs on the Zybo Z7 board to create a Home Security System.

B. Component Descriptions

1. pmod_maxsonar: Controller for the Ultrasonic Range Sensor PMOD. Detects if an object is within 10 inches of the sensor.
2. distance_led_controller: Controller for the LED that lights up when an object is within 10 inches of the Ultrasonic Range Sensor PMOD.
3. kypd_decoder: Decodes the key presses on the 16-Button Keypad PMOD.
4. kypd_led_controller: Controls the “locking” and “unlocking” mechanism for the entire system, based on the entries on the keypad.
5. kypd_maxsonar_interface: Connects the functionality of the Ultrasonic Range Sensor PMOD and Keypad PMOD.
6. load_passkey: Controller for loading a custom guest passkey into the system. .
7. panic_button_led_controller: FSM and controller for enabling the panic LEDs and display.
8. ssd_unlocker_controller: Controller for the seven segment display based on the values from the Ultrasonic Range Sensor and Keypad PMOD.
9. pixel_pusher: Generates the pixels for the SOS image file.
10. vga_ctrl: Controller for the VGA PMOD.
11. panic_picture: IP Block Memory segment for the SOS image corresponding to the panic button.
12. maxsonar_kypd_ssd_vga_top_level: Intermediate top level design for interfacing the 16-button Keypad, Ultrasonic Range Finder, Seven Segment Display, and VGA PMOD.
13. maxsonar_kypd_ssd_top_level: Intermediate top level design for interfacing the 16-button Keypad, Ultrasonic Range Finder, and Seven Segment Display PMOD.
14. maxsonar_kypd_top_level: Intermediate top level design for interfacing the 16-button Keypad and Ultrasonic Range Finder PMOD.
15. clock_div: Clock divider to drive the VGA output.
16. debounce: 20 ms debouncer for the buttons on the board.

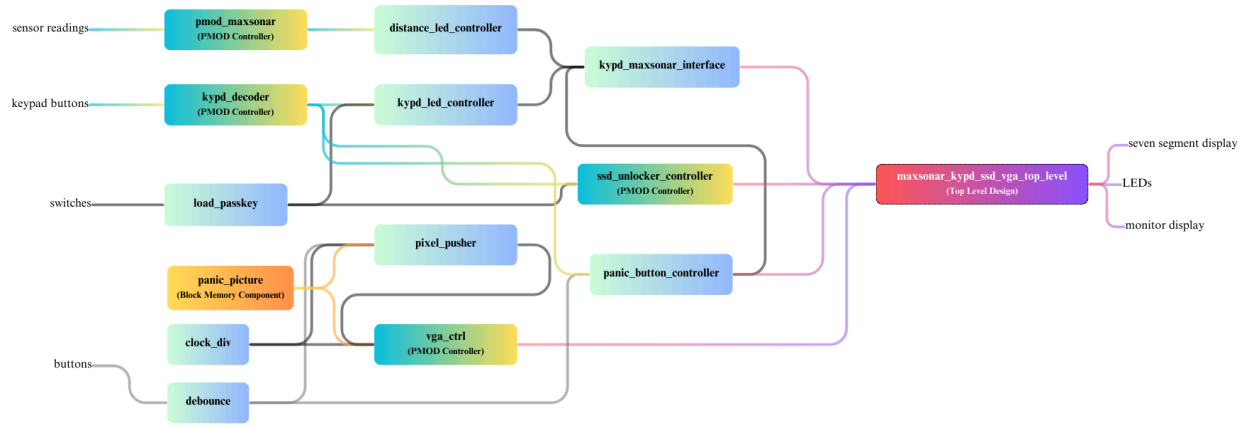
Note: The components below have code written for them and are on GitHub, however I faced issues when trying to make this functionality work on the board. I would have implemented this if I had more time to work on the project.

These components were meant to be incorporated with the built-in login keys, to function as identifiers for the users that "logged" into the system to unlock or disable the motion sensor.

17. charles_picture: IP Block Memory segment for the User 'F' image corresponding to the panic button.
18. pixel_pusher_ssd_charles: Generates the pixels for the User 'F' image file.
19. vga_ctrl_ssd_charles: Controller for the VGA PMOD.
20. pierre_picture: IP Block Memory segment for the User 'A' image corresponding to the panic button.
21. pixel_pusher_ssd_pierre: Generates the pixels for the User 'A' image file.
22. vga_ctrl_ssd_pierre: Controller for the VGA PMOD.
23. nick_picture: IP Block Memory segment for the User 'B' image corresponding to the panic button.
24. pixel_pusher_ssd_nick: Generates the pixels for the User 'B' image file.
25. vga_ctrl_ssd_nick: Controller for the VGA PMOD.

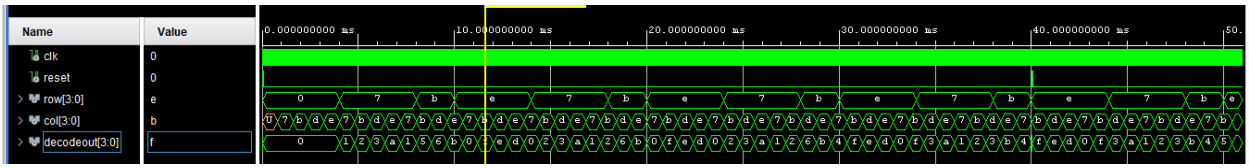
C. System Block Diagram

Note: The System Block Diagram below only contains the components that were implemented on the board correctly. Components 17 to 25 are not included.

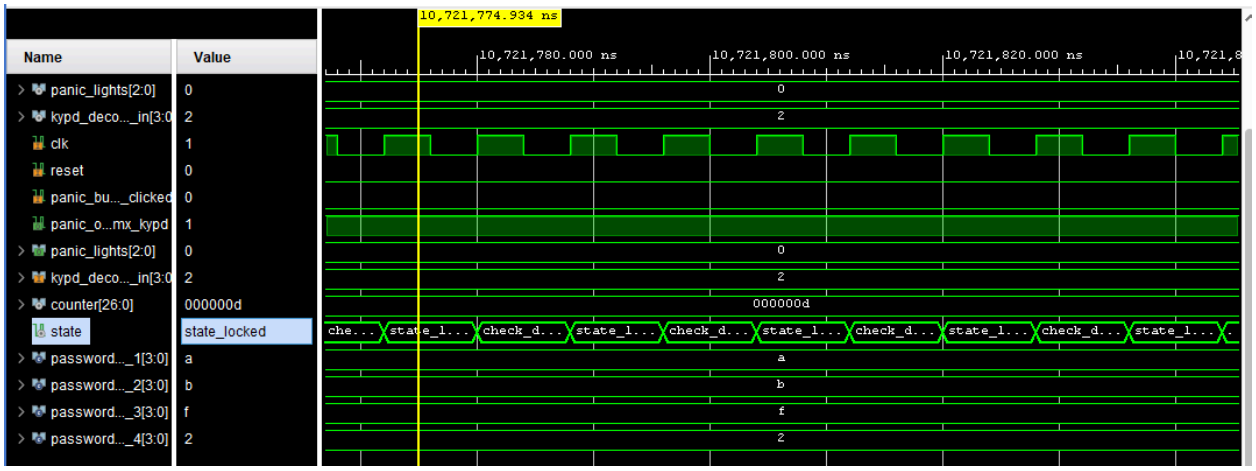


D. Simulation Diagrams

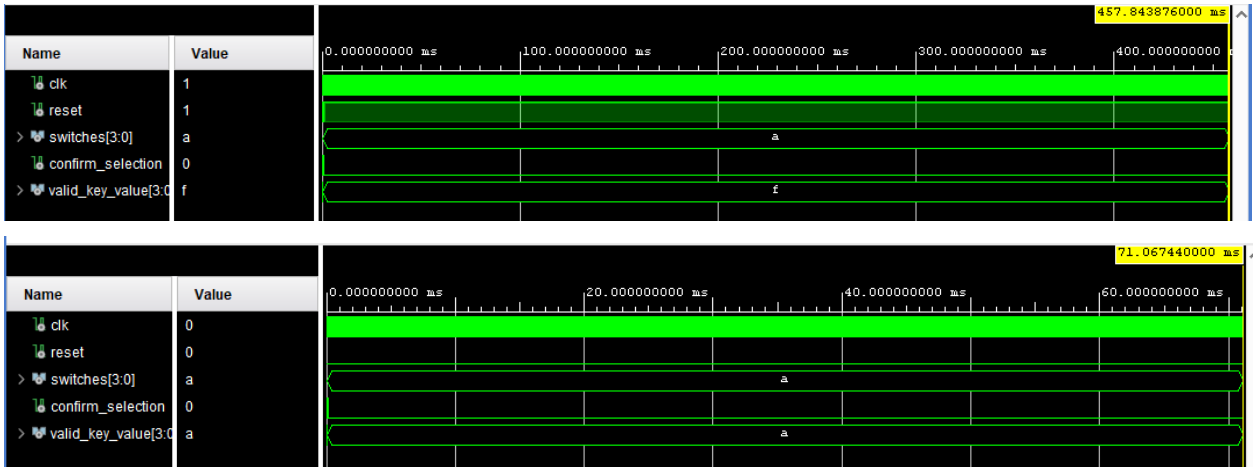
Case 1: Depicts pressing various different keys.



Case 2: Depicts panic button functionality.



Case 3: Depicts loading passkey.



E. Schematics

<u>pmod_maxsonar:</u>

Diagram 1a: Elaboration Schematic

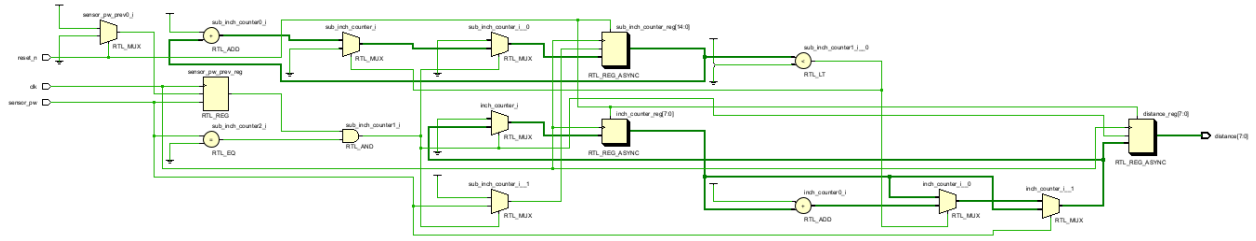


Diagram 1b: Synthesis Schematic

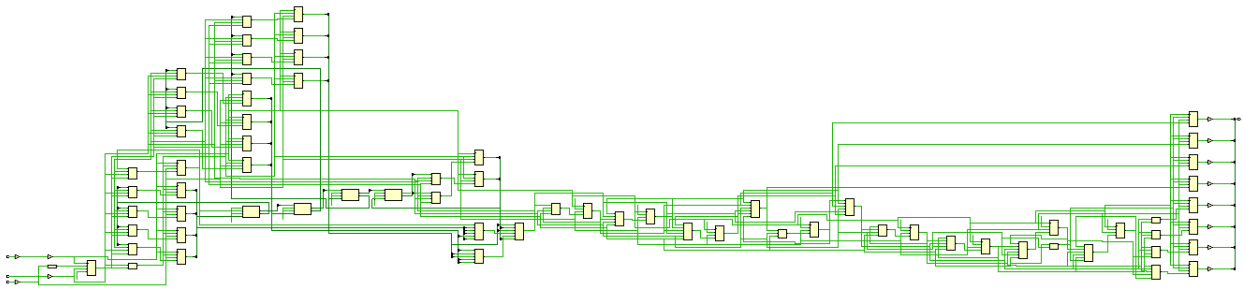


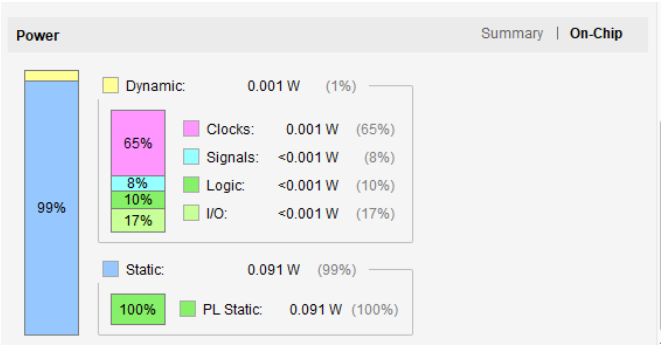
Diagram 1c: Post-Synthesis Utilization Table

Utilization — **Post-Synthesis** | Post-Implementation

Graph | **Table**

Resource	Estimation	Available	Utilization %
LUT	24	17600	0.14
FF	32	35200	0.09
IO	11	100	11.00
BUFG	1	32	3.13

Diagram 1d: On-Chip Power



distance_led_controller:

Diagram 2a: Elaboration Schematic

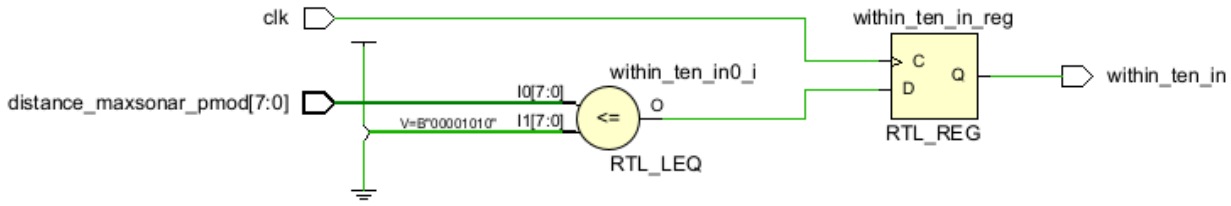


Diagram 2b: Synthesis Schematic

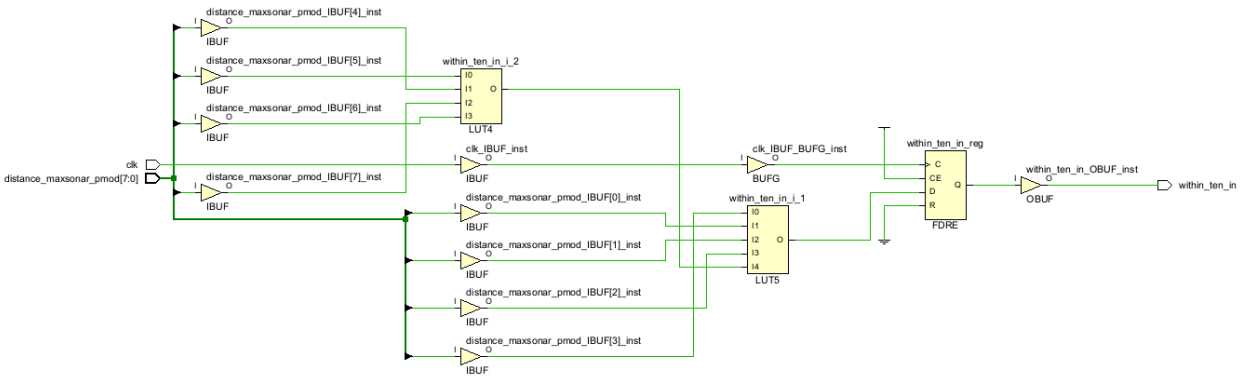


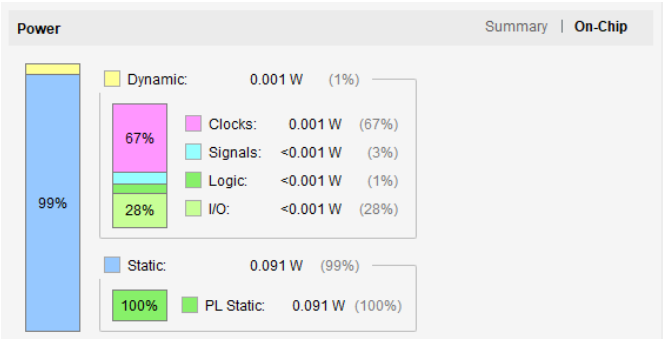
Diagram 2c: Post-Synthesis Utilization Table

Utilization — **Post-Synthesis** | Post-Implementation

Graph | **Table**

Resource	Estimation	Available	Utilization %
LUT	2	17600	0.01
FF	1	35200	0.01
IO	10	100	10.00
BUFG	1	32	3.13

Diagram 2d: On-Chip Power



kypd_decoder:

Diagram 3a: Elaboration Schematic

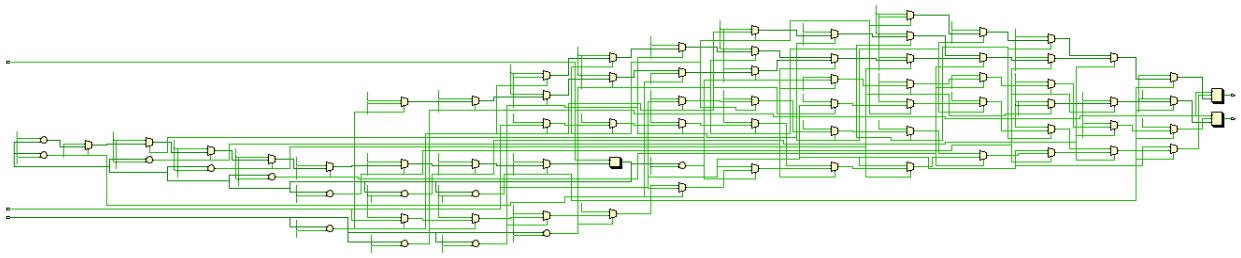


Diagram 3b: Synthesis Schematic

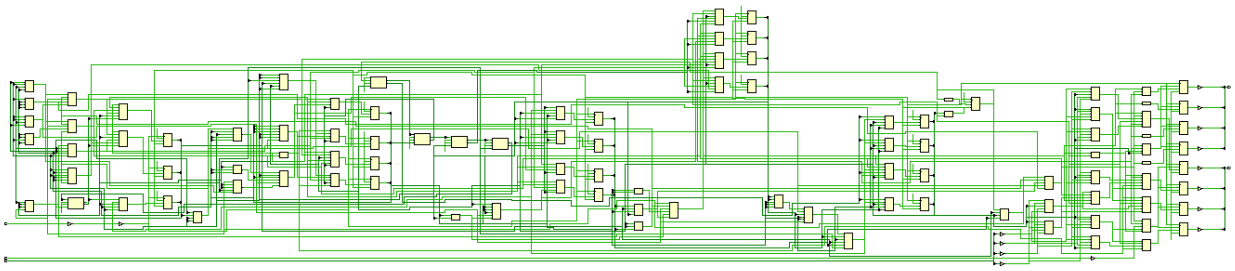


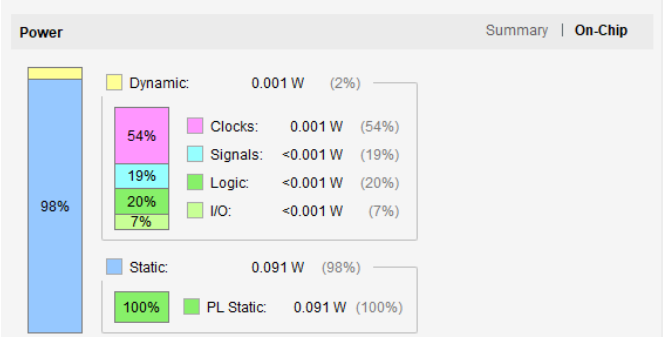
Diagram 3c: Post-Synthesis Utilization Table

Utilization — **Post-Synthesis** | Post-Implementation

Graph | **Table**

Resource	Estimation	Available	Utilization %
LUT	55	17600	0.31
FF	28	35200	0.08
IO	14	100	14.00
BUFG	1	32	3.13

Diagram 3d: On-Chip Power



kypd_led_controller:

Diagram 4a: Elaboration Schematic

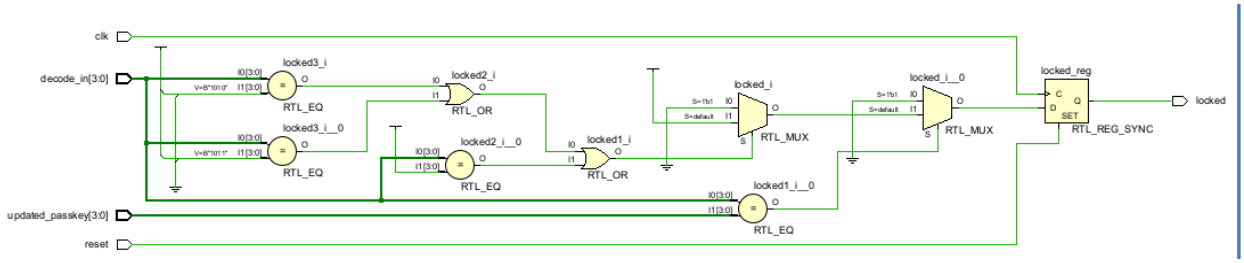


Diagram 4b: Synthesis Schematic

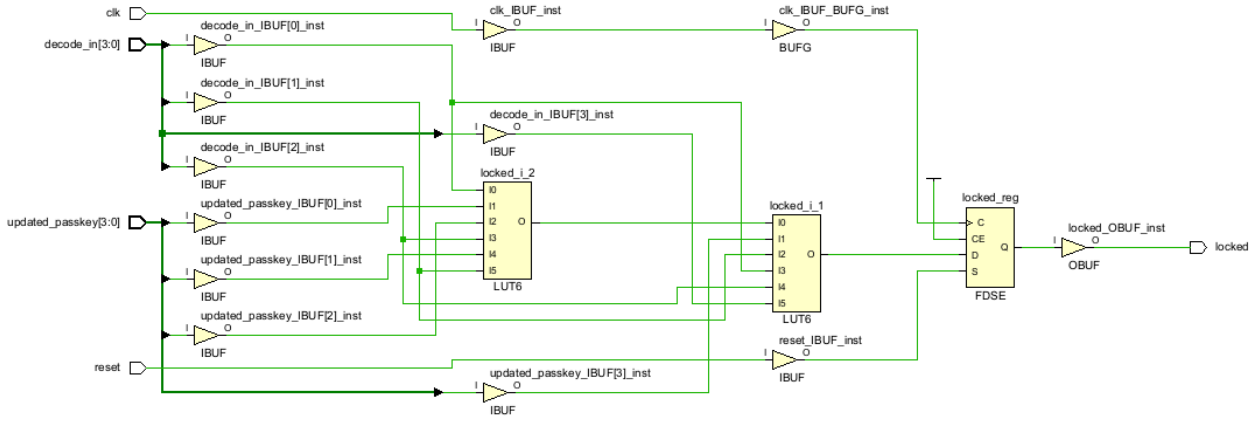
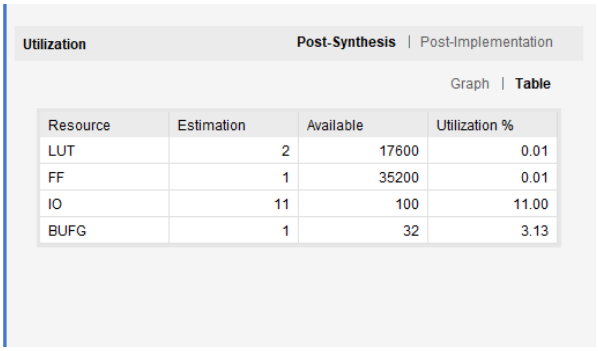


Diagram 4c: Post-Synthesis Utilization Table

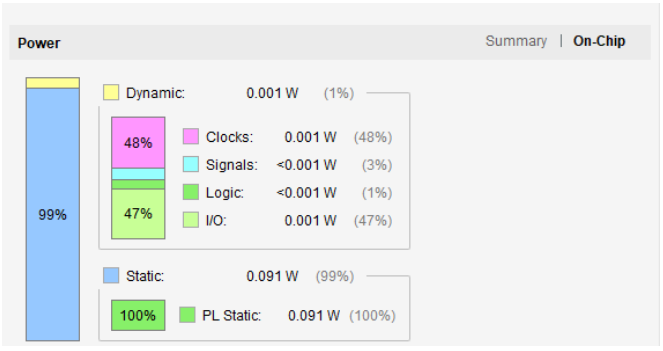


Utilization — **Post-Synthesis** | Post-Implementation

Graph | **Table**

Resource	Estimation	Available	Utilization %
LUT	2	17600	0.01
FF	1	35200	0.01
IO	11	100	11.00
BUFG	1	32	3.13

Diagram 4d: On-Chip Power



kypd_maxsonar_interface:

Diagram 5a: Elaboration Schematic

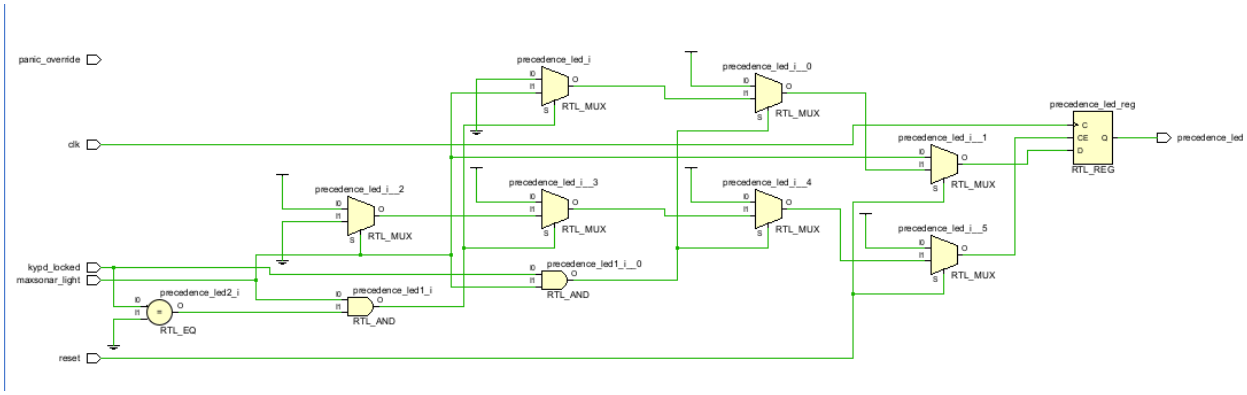


Diagram 5b: Synthesis Schematic

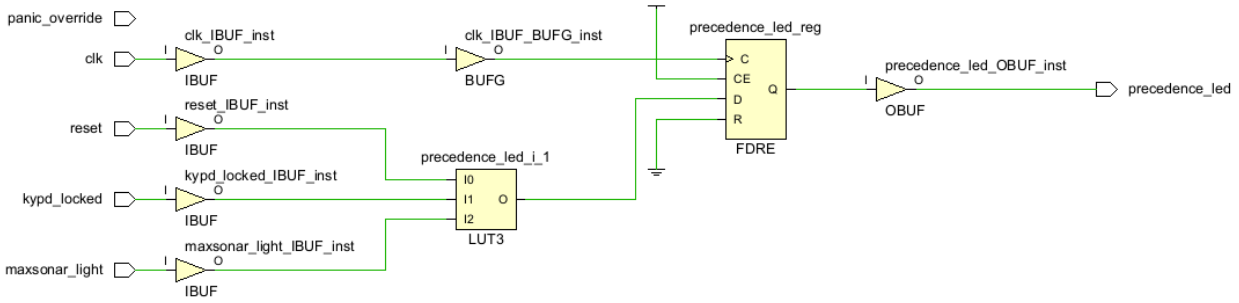


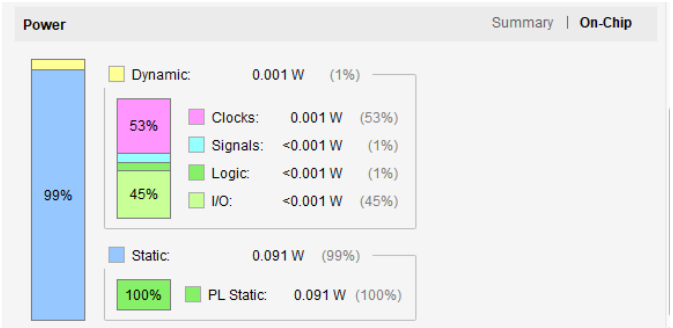
Diagram 5c: Post-Synthesis Utilization Table

Utilization — **Post-Synthesis** | Post-Implementation

Graph | **Table**

Resource	Estimation	Available	Utilization %
LUT	1	17600	0.01
FF	1	35200	0.01
IO	5	100	5.00
BUFG	1	32	3.13

Diagram 5d: On-Chip Power



load_passkey:

Diagram 6a: Elaboration Schematic

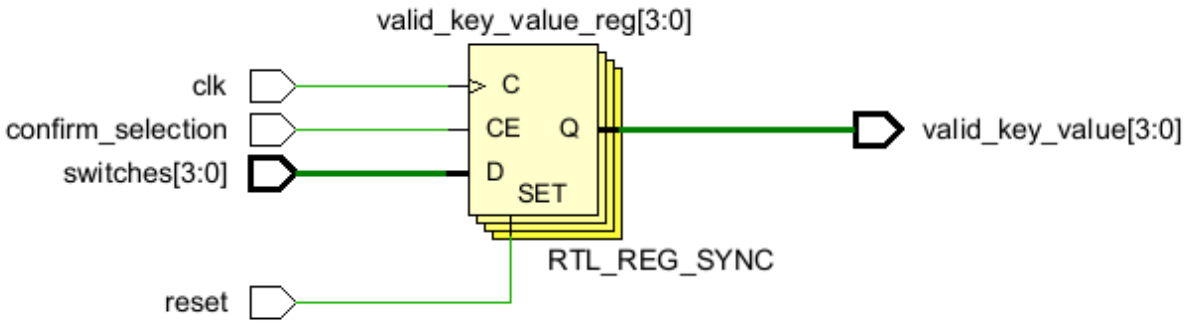


Diagram 6b: Synthesis Schematic

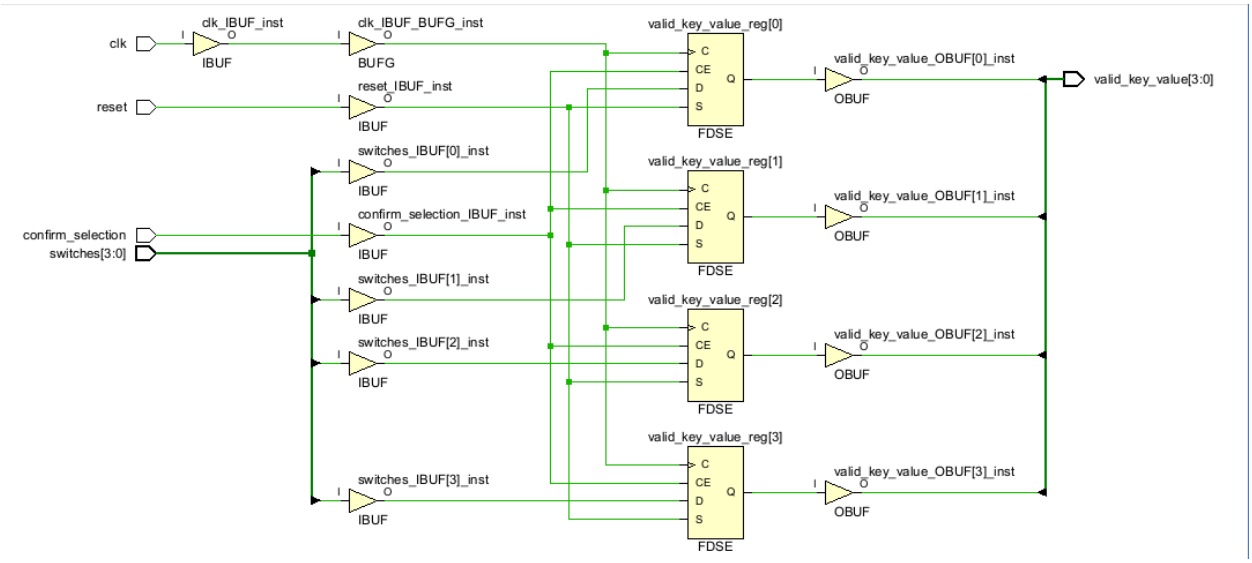
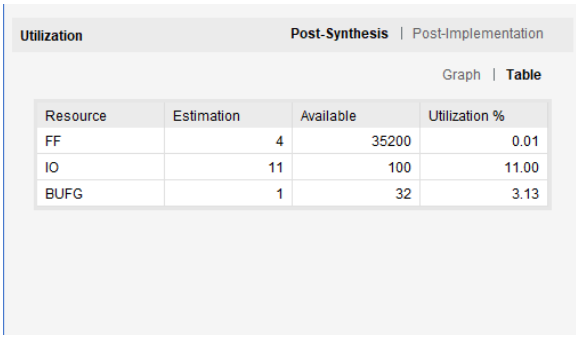


Diagram 6c: Post-Synthesis Utilization Table

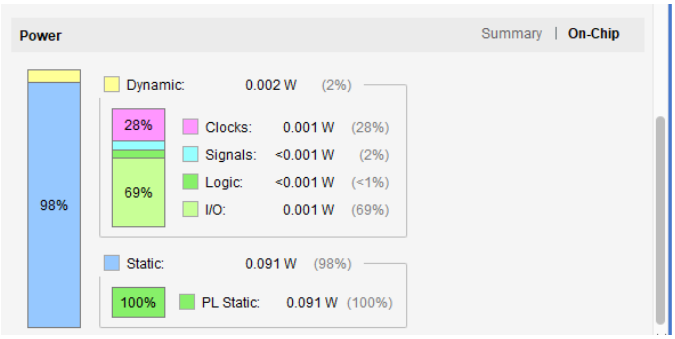


Utilization — Post-Synthesis | Post-Implementation

Graph | Table

Resource	Estimation	Available	Utilization %
FF	4	35200	0.01
IO	11	100	11.00
BUFG	1	32	3.13

Diagram 6d: On-Chip Power



<u>panic_button_led_controller:</u>

Diagram 7a: Elaboration Schematic

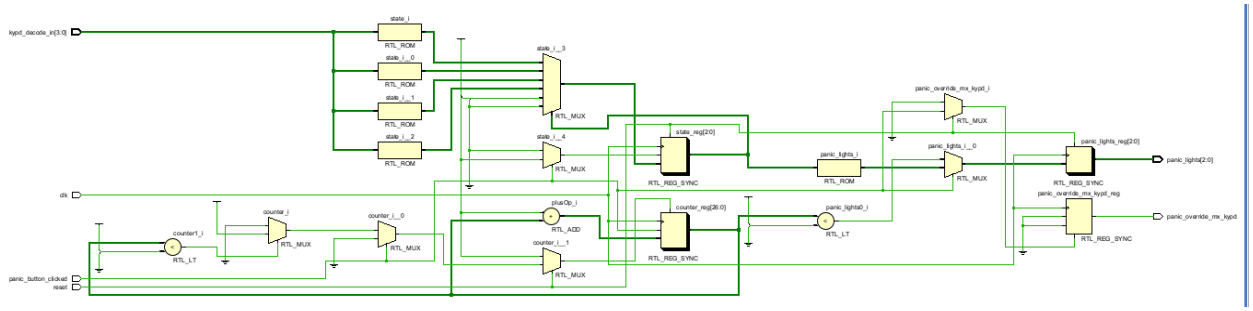


Diagram 7b: Synthesis Schematic

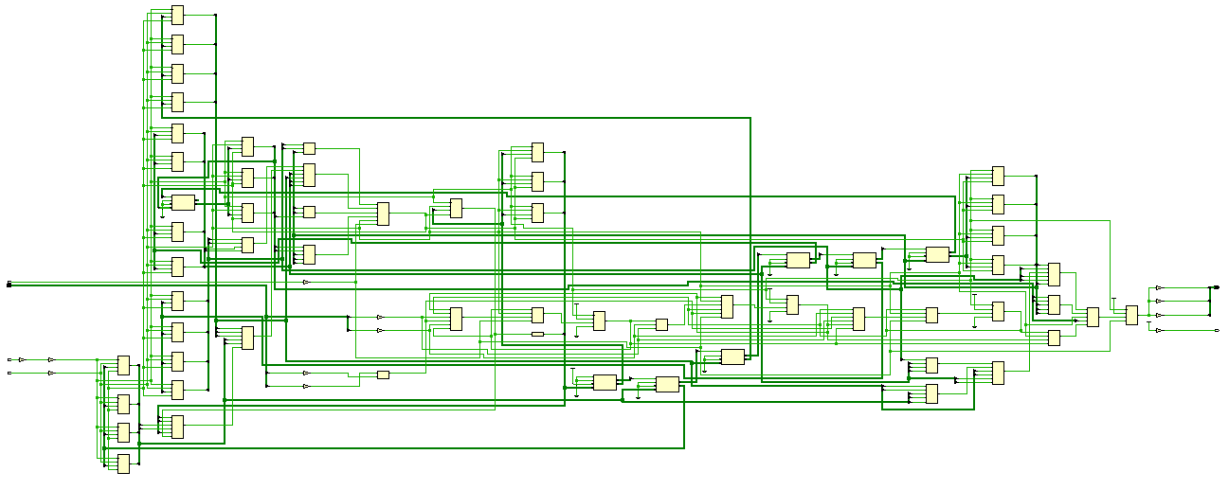


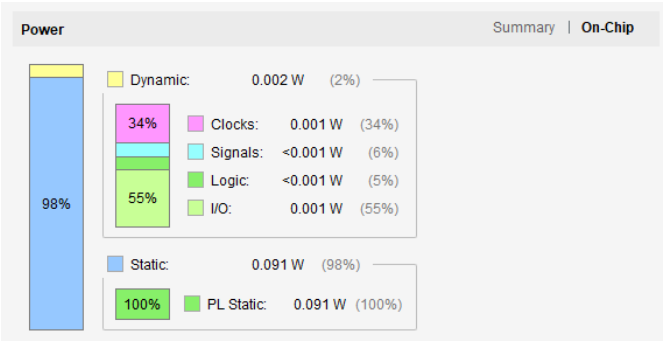
Diagram 7c: Post-Synthesis Utilization Table

Utilization | Post-Synthesis | Post-Implementation

Graph | Table

Resource	Estimation	Available	Utilization %
LUT	13	17600	0.07
FF	28	35200	0.08
IO	7	100	7.00
BUFG	1	32	3.13

Diagram 7d: On-Chip Power



ssd_unlocker_controller:

Diagram 8a: Elaboration Schematic

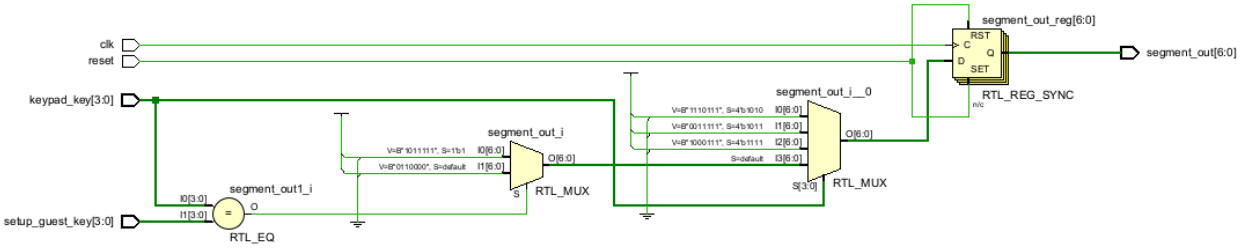


Diagram 8b: Synthesis Schematic

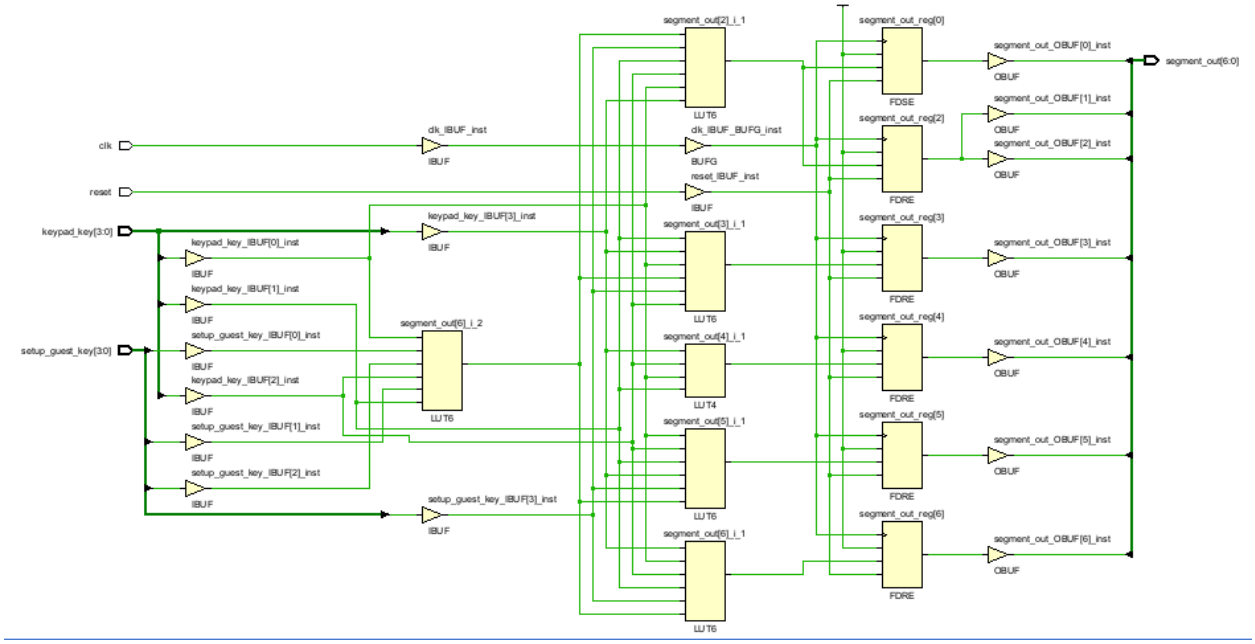


Diagram 8c: Post-Synthesis Utilization Table

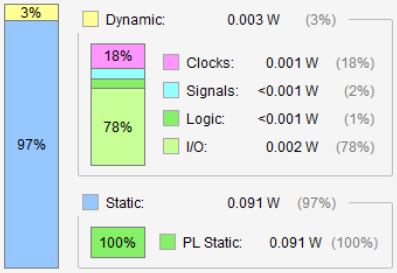
Utilization **Post-Synthesis** | Post-Implementation

Graph | **Table**

Resource	Estimation	Available	Utilization %
LUT	6	17600	0.03
FF	6	35200	0.02
IO	17	100	17.00
BUFG	1	32	3.13

Diagram 8d: On-Chip Power

Power Summary | **On-Chip**



pixel_pusher:

Diagram 9a: Elaboration Schematic

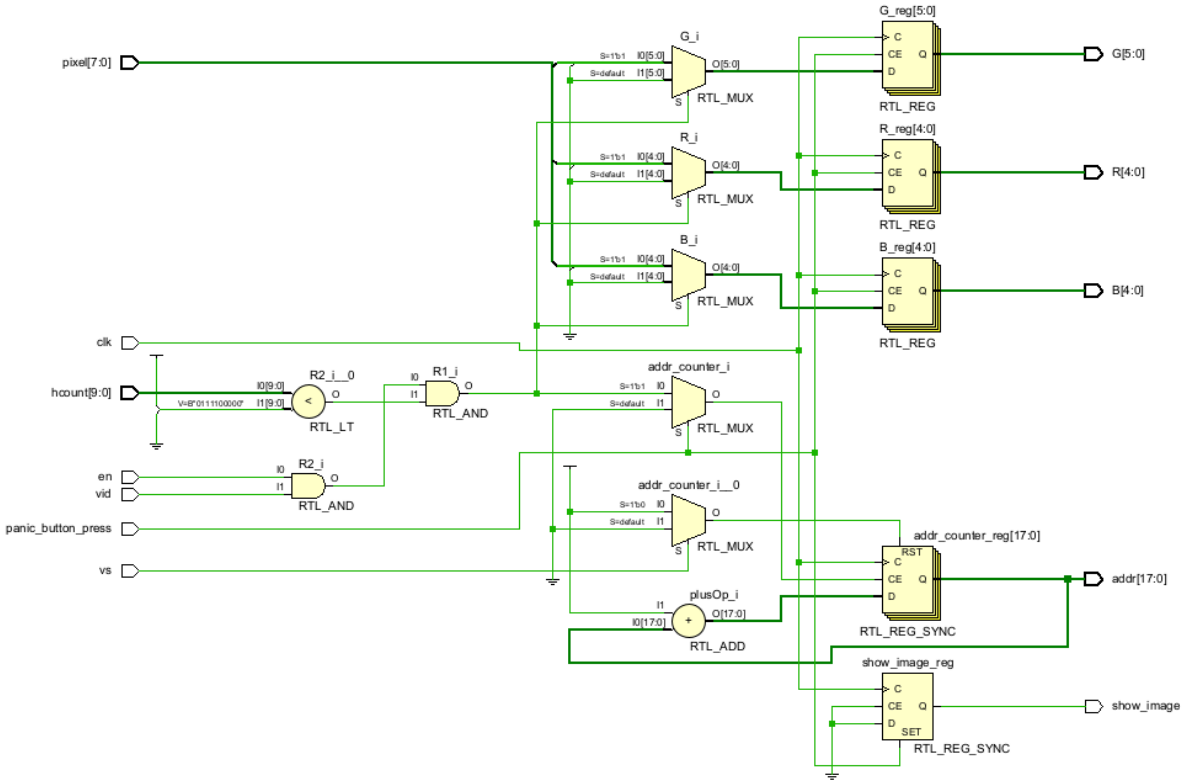


Diagram 9b: Synthesis Schematic

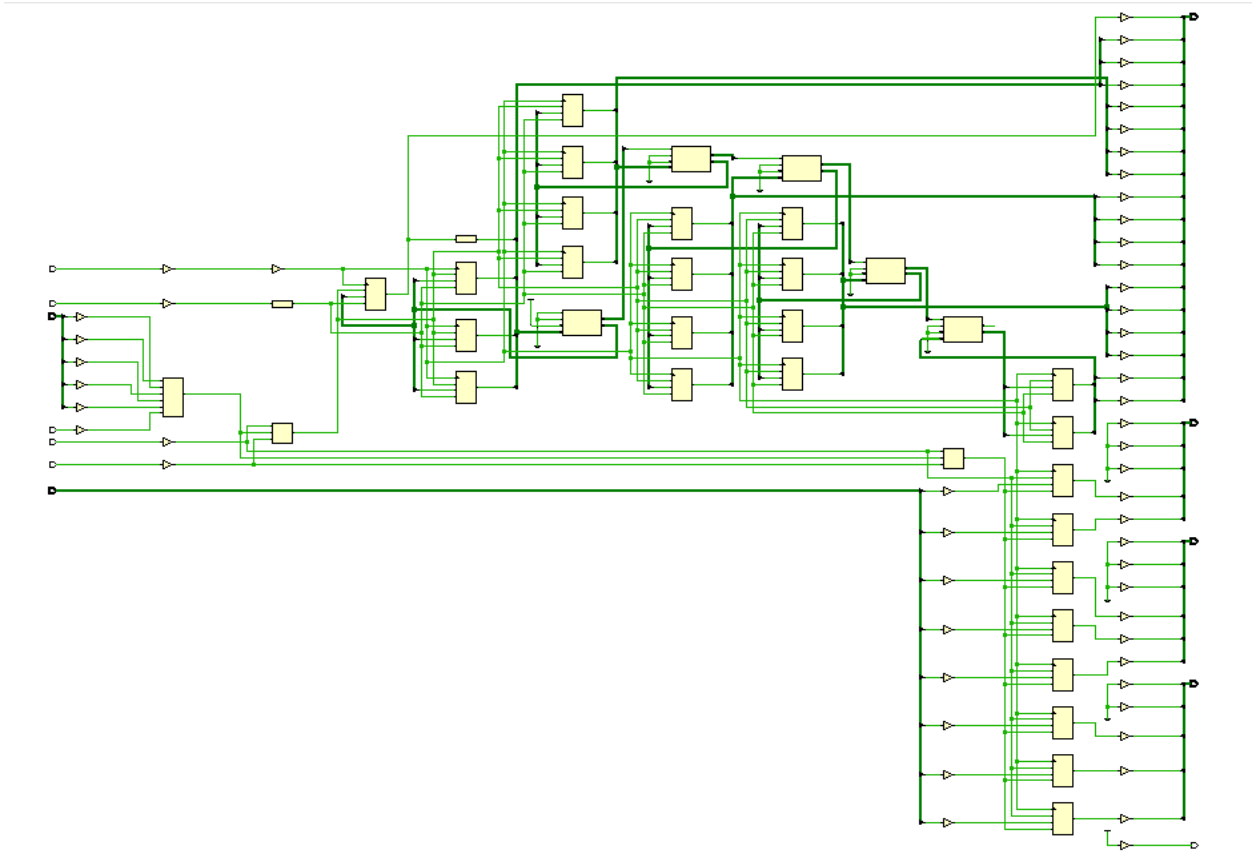


Diagram 9c: Post-Synthesis Utilization Table

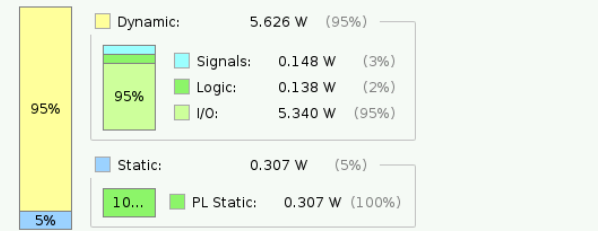
Utilization — **Post-Synthesis** | Post-Implementation

Graph | **Table**

Resource	Estimation	Available	Utilization %
LUT	5	17600	0.03
FF	26	35200	0.07
IO	53	100	53.00
BUFG	1	32	3.13

Diagram 9d: On-Chip Power

Power — Summary | **On-Chip**



vga_ctrl:

Diagram 10a: Elaboration Schematic

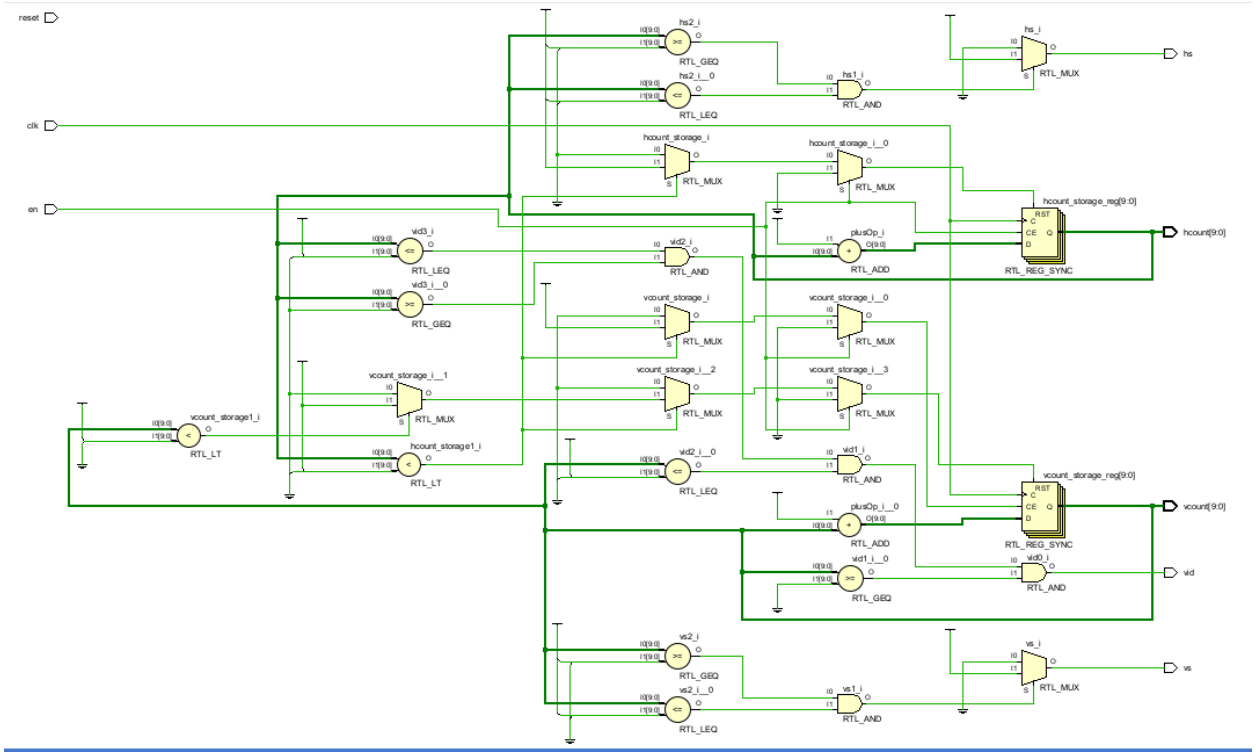


Diagram 10b: Synthesis Schematic

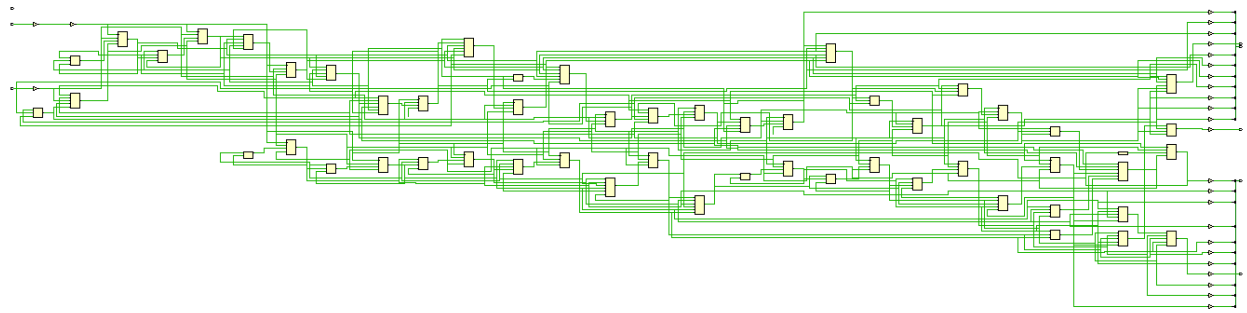


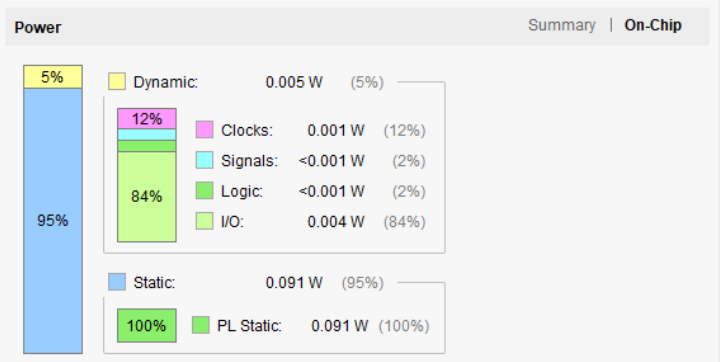
Diagram 10c: Post-Synthesis Utilization Table

Utilization **Post-Synthesis** | Post-Implementation

Graph | **Table**

Resource	Estimation	Available	Utilization...
LUT	26	17600	0.15
FF	20	35200	0.06
IO	25	100	25.00
BUFG	1	32	3.13

Diagram 10d: On-Chip Power



maxsonar_kypd_ssd_vga_top_level:

Diagram 12a: Elaboration Schematic

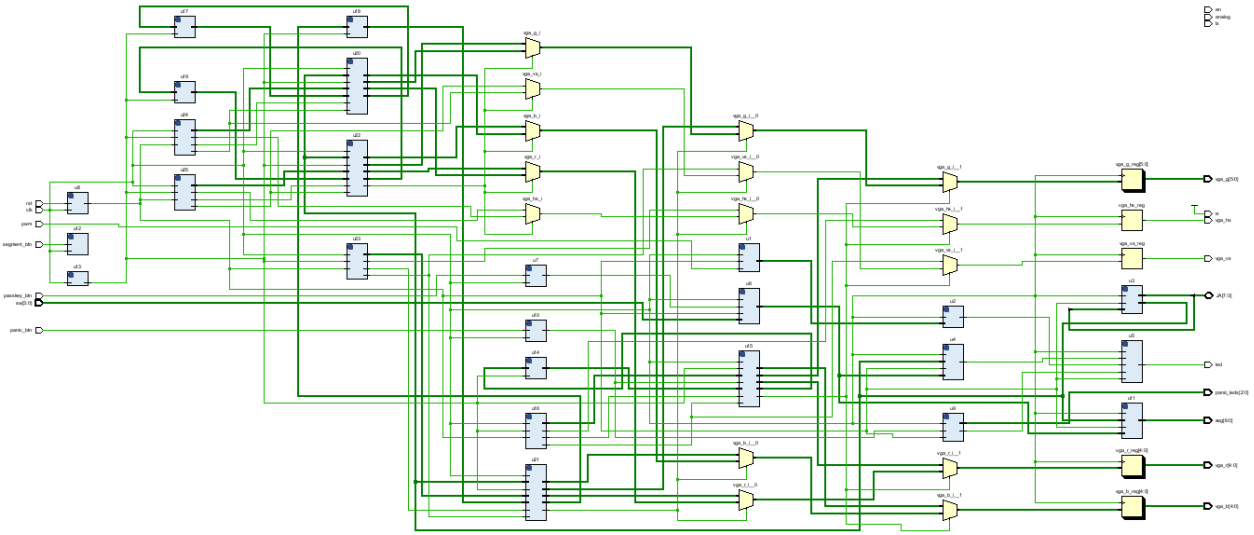


Diagram 12b: Synthesis Schematic

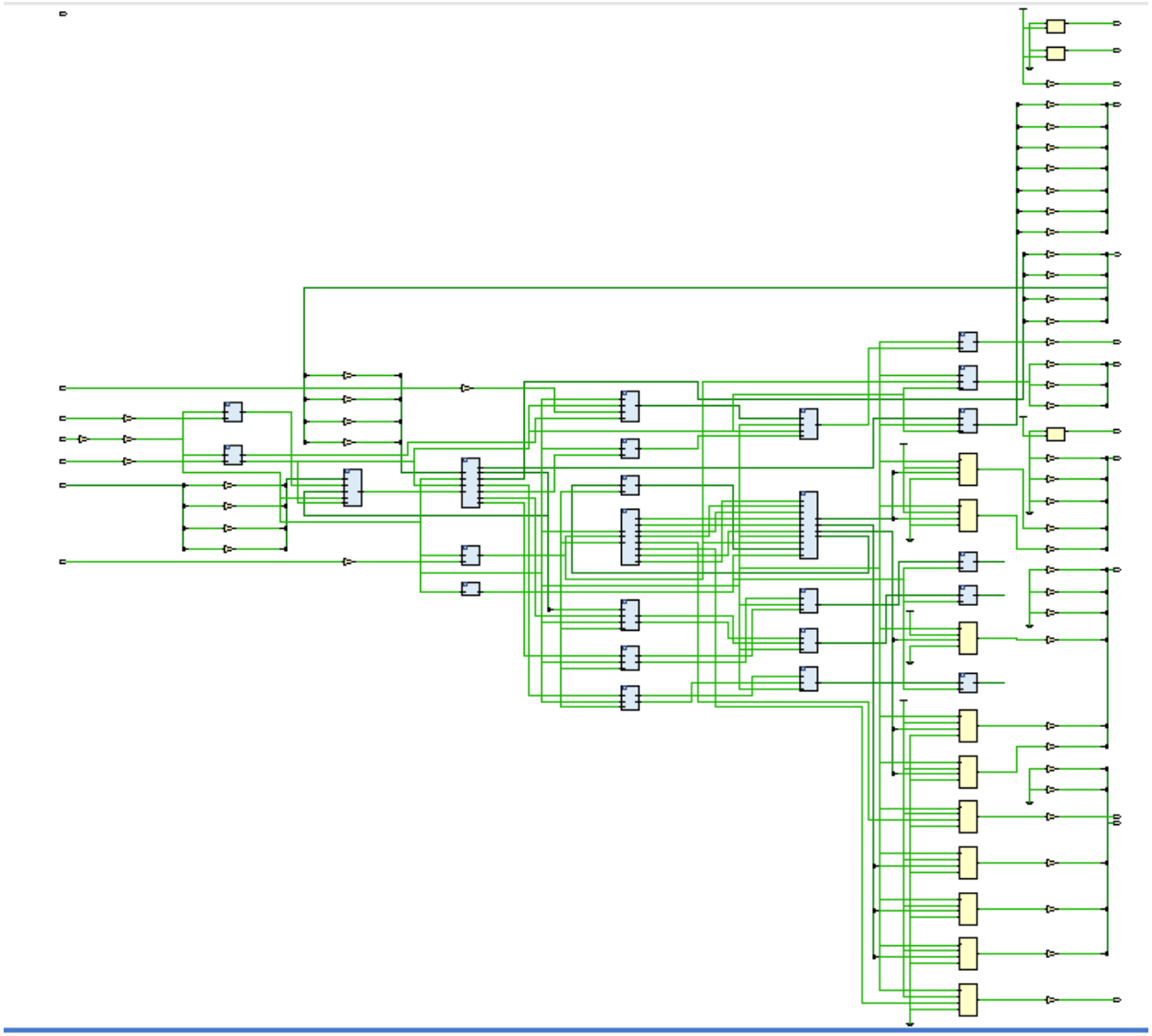


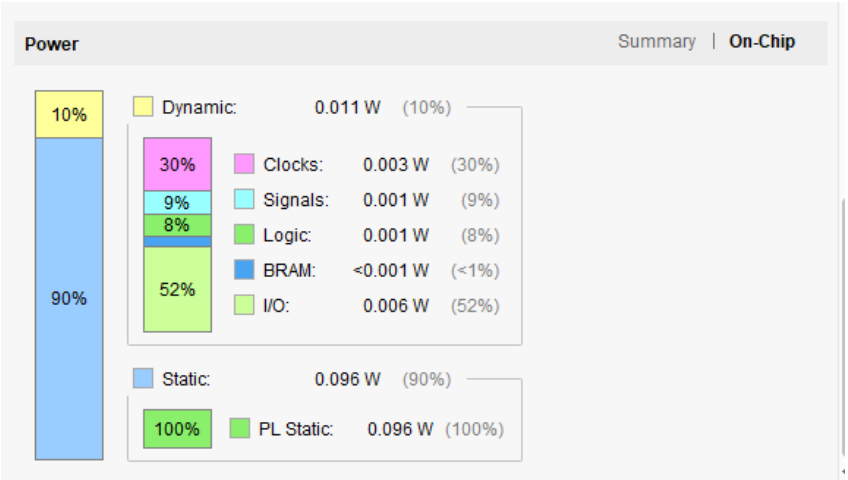
Diagram 12c: Post-Synthesis Utilization Table

Utilization **Post-Synthesis** | Post-Implementation

Graph | **Table**

Resource	Estimation	Available	Utilization %
LUT	251	17600	1.43
FF	383	35200	1.09
IO	50	100	50.00
BUFG	1	32	3.13

Diagram 12d: On-Chip Power



maxsonar_kypd_ssd_top_level:

Diagram 13a: Elaboration Schematic

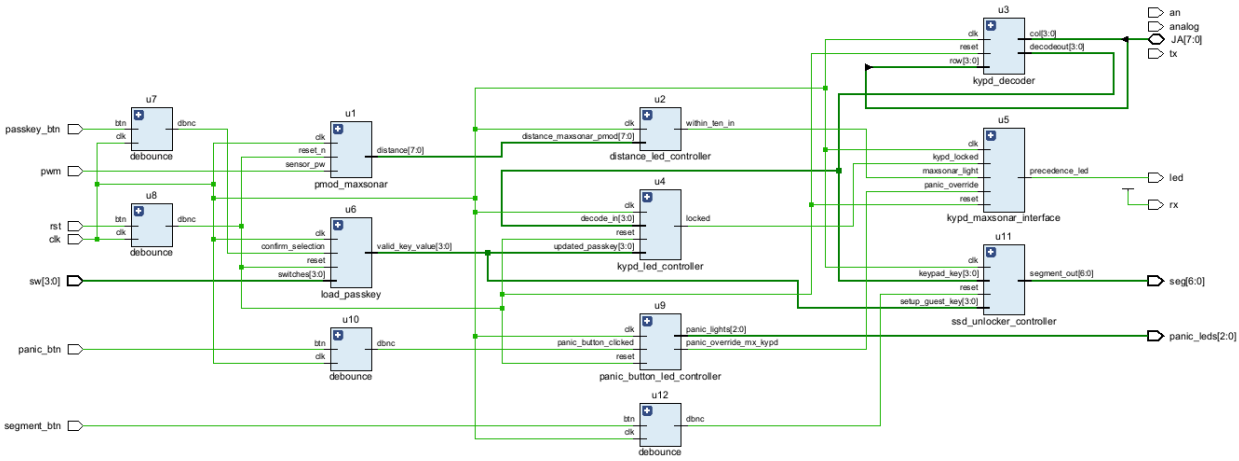


Diagram 13b: Synthesis Schematic

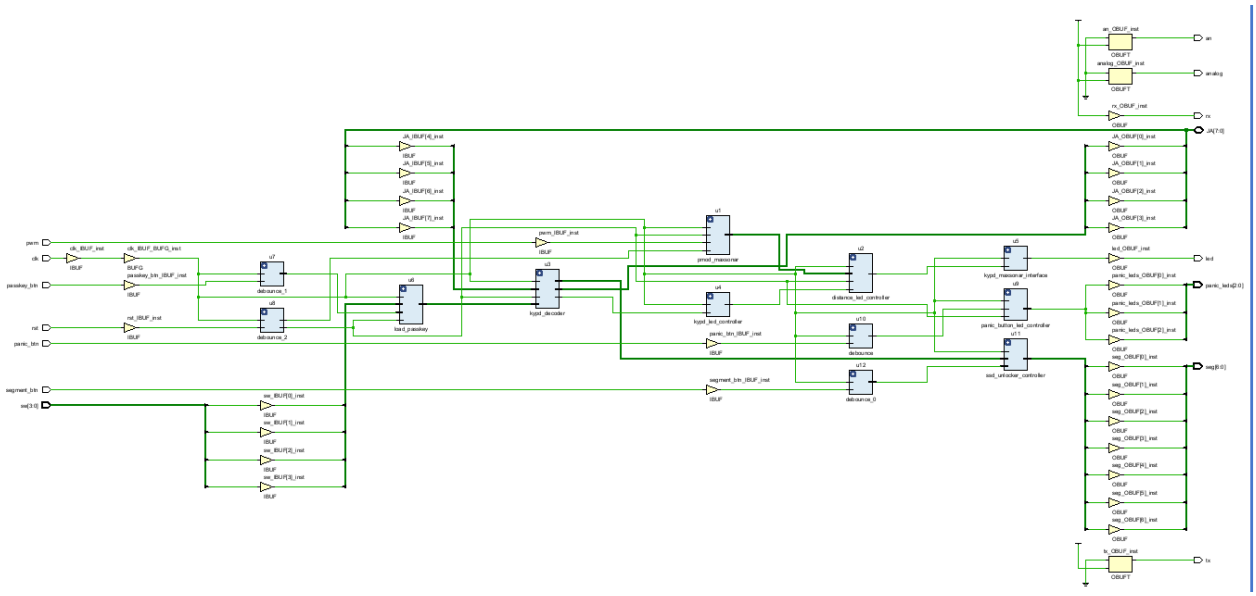


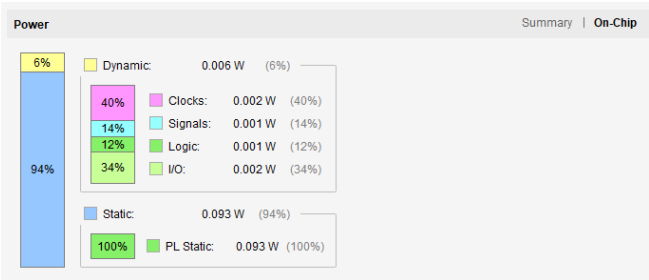
Diagram 13c: Post-Synthesis Utilization Table

Utilization — **Post-Synthesis** | Post-Implementation

Graph | **Table**

Resource	Estimation	Available	Utilization %
LUT	153	17600	0.87
FF	213	35200	0.61
IO	33	100	33.00
BUFG	1	32	3.13

Diagram 13d: On-Chip Power



<u>maxsonar_kypd_top_level:</u>

Diagram 14a: Elaboration Schematic

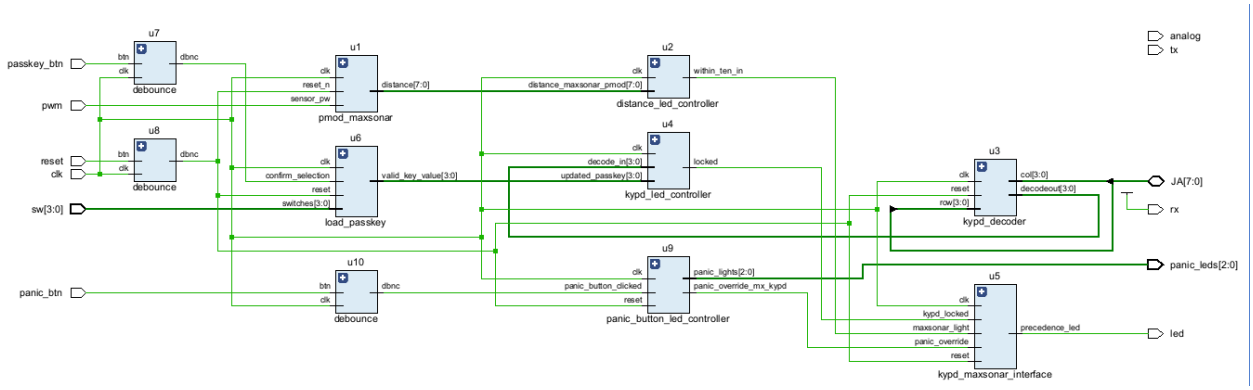


Diagram 14b: Synthesis Schematic

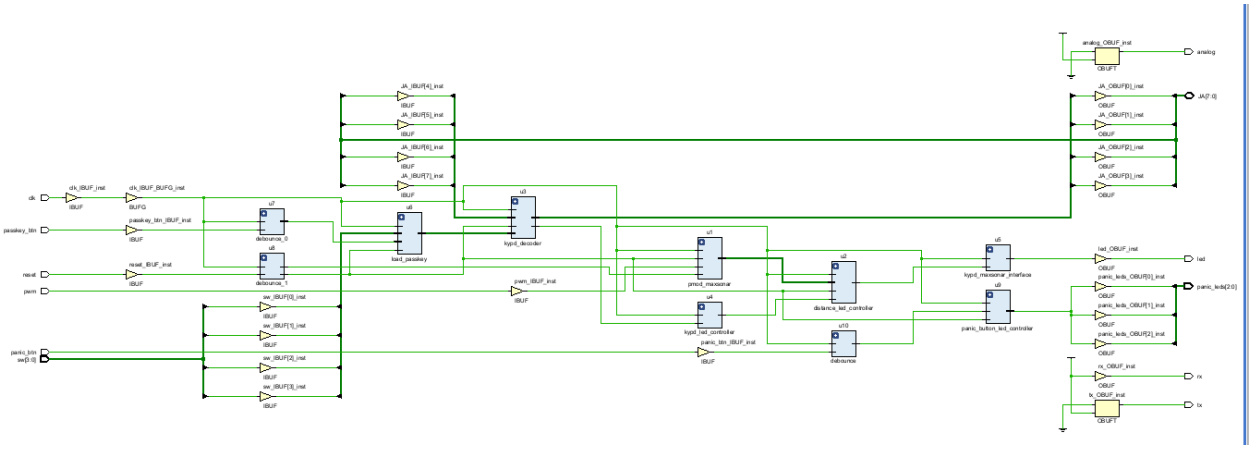


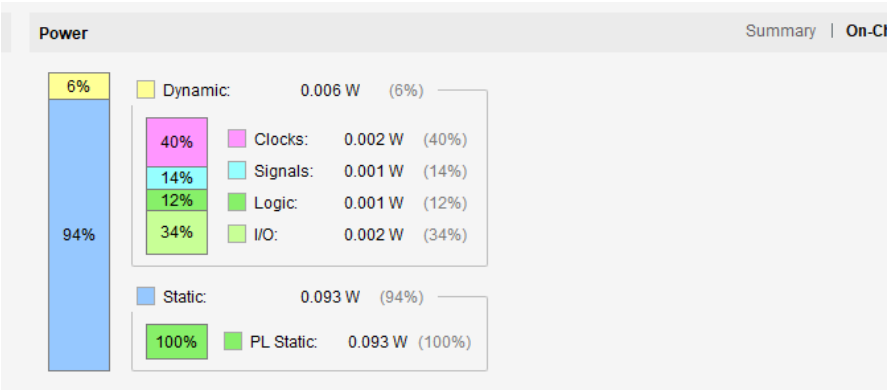
Diagram 14c: Post-Synthesis Utilization Table

Utilization **Post-Synthesis** | Post-Implementation

Graph | **Table**

Resource	Estimation	Available	Utilization...
LUT	139	17600	0.79
FF	179	35200	0.51
IO	24	100	24.00
BUFG	1	32	3.13

Diagram 14d: On-Chip Power



clock_div:

Diagram 15a: Elaboration Schematic

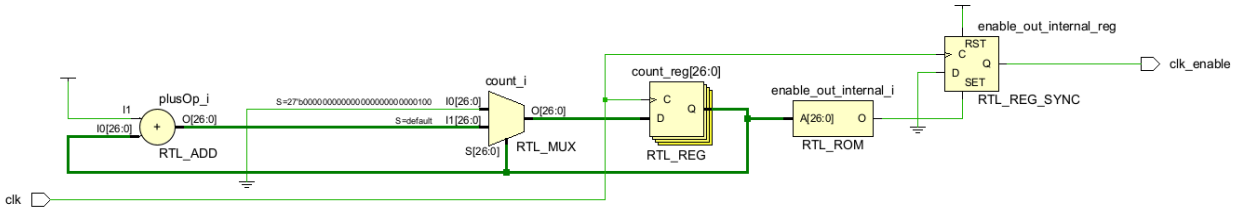


Diagram 15b: Synthesis Schematic

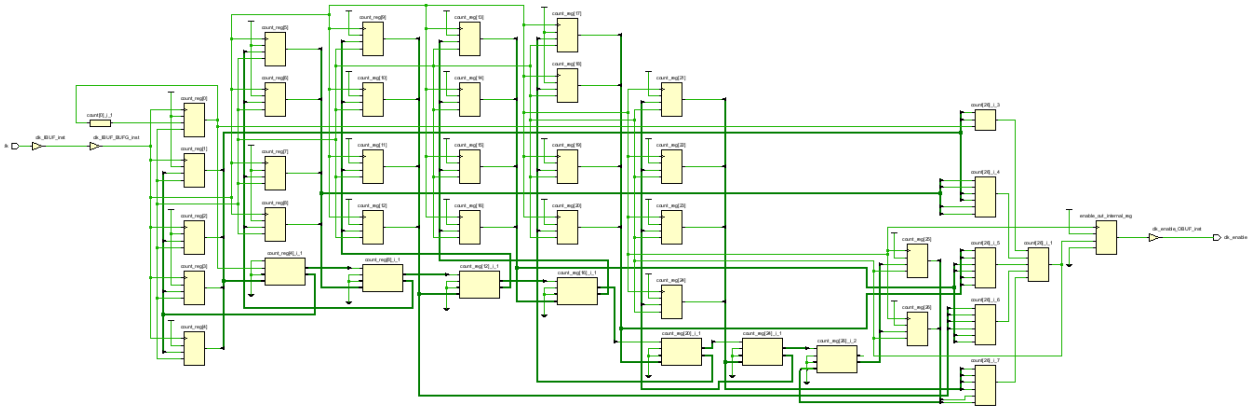


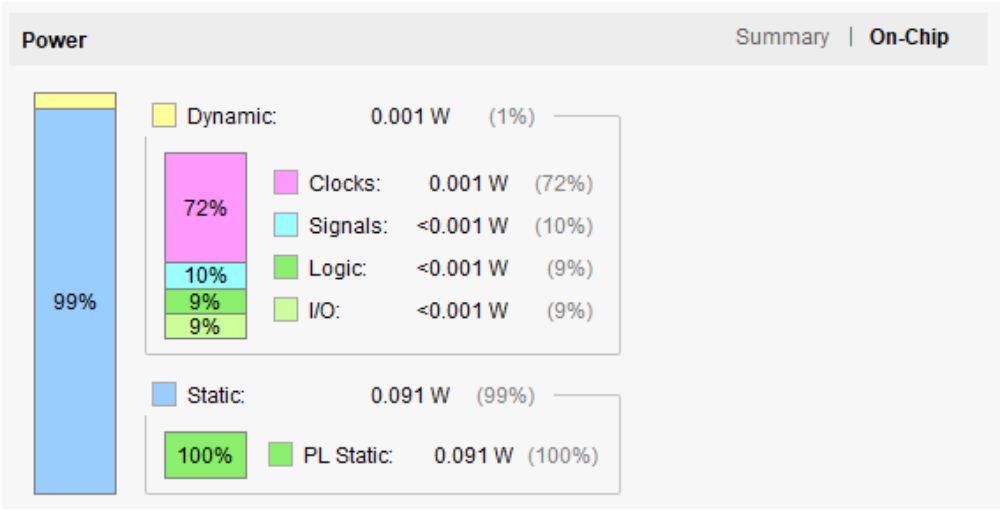
Diagram 15c: Post-Synthesis Utilization Table

Utilization | **Post-Synthesis** | Post-Implementation

Graph | **Table**

Resource	Estimation	Available	Utilization %
LUT	6	17600	0.03
FF	28	35200	0.08
IO	2	100	2.00
BUFG	1	32	3.13

Diagram 15d: On-Chip Power



<u>debounce:</u>

Diagram 16a: Elaboration Schematic

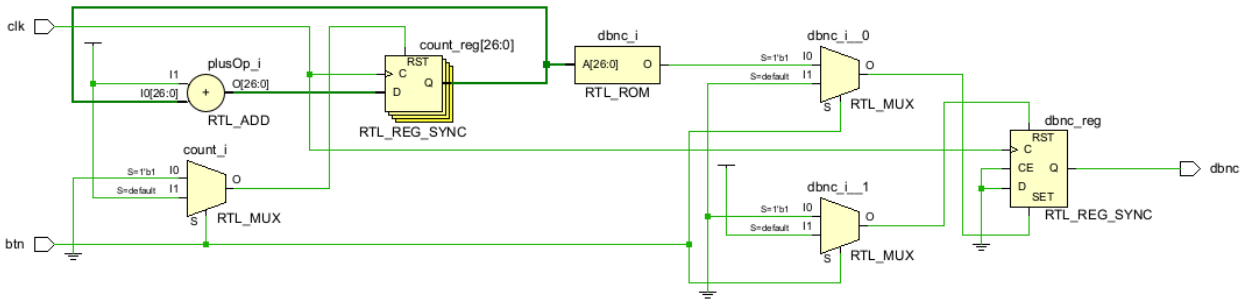


Diagram 16b: Synthesis Schematic

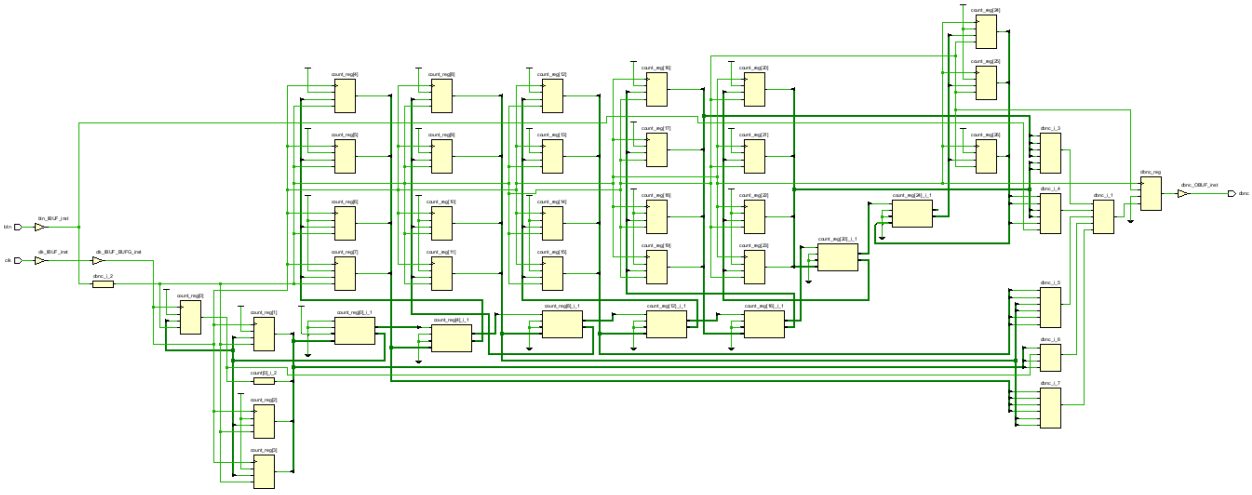


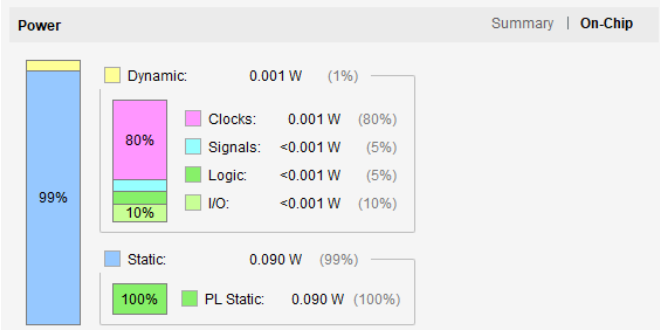
Diagram 16c: Post-Synthesis Utilization Table

Utilization | **Post-Synthesis** | Post-Implementation

Graph | **Table**

Resource	Estimation	Available	Utilization %
LUT	8	17600	0.05
FF	28	35200	0.08
IO	3	100	3.00
BUFG	1	32	3.13

Diagram 16d: On-Chip Power



V. Non-functional Requirements & Project Limitations

A. Non-functional Requirements

My system is designed to function in real-time based on the current inputs or outputs of the PMODs and the Zybo Z7 board. The design of the system is such that it should immediately report an intruder if one is detected by the MAXSONAR PMOD. Additionally, the SSD should be responding to the stimulus from the KYPD and MAXSONAR PMODs, as they occur. As a result, I have used the built in 125 MHz clock and one clock divider to drive my system.

If we take a look at the resource utilization of the top level design we'll notice that the greatest resource utilization occurs for the IO ports (50% usage). This is expected due to the number of PMODs used in the design. The KYPD, VGA, and SSD all use 12 pins through which they communicate input and output information with the Zybo board. The MAXSONAR PMOD only uses 6 pins to communicate input and output with the Zybo board. The 50% usage of the I/O resources indicates that there is not much more leeway for making any additions in terms of new connections to the Zybo board. As for the lookup tables and flip-flops, we notice that their utilization is down to approximately 1%. Therefore, if I choose to increase the complexity of this project, by further developing the VHDL code, I would not face many issues in terms of resources available.

If we take a look at my top level design, the total power consumption is only 0.011 W (as shown in Diagram 12d). Despite there being four PMODs, the power consumption for the entire system is very low. According to the design specifications, most of the PMODs require a power supply. For instance, all of the PMODs require a 3.3V power supply to function. Overall, the power consumption of this entire system is fairly low, which indicates that it can be replicated easily as an off-the-shelf device. Additionally, the low power consumption indicates low energy consumption, which enhances the systems

B. Project Limitations

The limitations of this project arose mostly from the functionality of the PMODs.
The Ultrasonic Range Finder PMOD is able to detect objects between the range of 6 to 255 (up to 20 feet), however, it is only able to detect objects that are directly in front of the sensor. Unlike other sensors that might be able to detect objects within a certain radius of their locations, the MAXSONAR PMOD is only able to detect objects or instructions that appear within a direct line. Therefore, there is a large blind spot in the view of the PMOD. The project is limited in the directions it can detect an object in. Since the MAXSONAR PMOD forms the crux of this project, it prevents intrusion detection from being more advanced.

Furthermore, with the KYPD PMOD, I was only able to use 16 buttons. This restricted my ability to create diverse and secure built-in codes for users. As a result, I was limited in the types of codes I could create. If I were to have more time, I would have attempted to use the PS2 PMOD to integrate a complete keyboard into the system.

VI. Conclusion/Follow-Up

Overall, I faced many challenges when creating this design. The first and foremost being, the interaction between all four PMODs. Since the output or input of any given PMOD was directly connected to all other PMODs, I had to ensure that the reading of data in and out of the system to

toggle the various features was done effectively. To achieve this, I generated testbenches at intermediate steps to ensure that the system would function as I had intended.

Given more time, I would have liked to add an additional feature. I hoped to use the Wi-Fi PMOD to integrate the alarm system with a device like Google Home, Alexa, etc. to build in functionality that could directly call 911 in case of an emergency. If I wanted to build upon the existing functionality, I could also use the RTCC PMOD to record the time and date of each intrusion or entry into the house. This data could then be provided to the user for their personal use. If I were to extend the application of this embedded system to recognize other types of emergencies besides an intruder, I would integrate the HYGRO PMOD. Using its temperature and humidity sensing functionality, I could detect if and when there was a fire or gas leak in the home. With more time and the ability to integrate more PMODs, I would have been able to further develop this project and improve its functionality.

To conclude, in this project, I implemented a Home Security System with the help of four PMODs: an Ultrasonic Range Finder, Keypad, Seven Segment Display, and Video Graphics Array. Through various controller files, a finite state machine, and testbenches, I was able to develop an embedded system that detects any motion within a specified range and reports the results in real-time through the Zybo Z7 and monitor.