Division of Electronics and Communication Engineering
2024-2025 (ODD SEM)

REPORT

for

FPGA BASED SYSTEM DESIGN-PROJECT BASED COURSE

Title of the Report: FPGA implementation of Digital circuits

A report submitted by

Name of the Student	***Venkatesa Rakshan S***
Register Number	**URK22EC1002**
Subject Name	***FPGA Based System Design***
Subject Code	***22EC2020***
Date of Report submission	***20/09/2024***

Total marks: _____/ 10 Marks

Signature of Faculty with date:

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Introduction to Binary-to-Gray Code Conversion

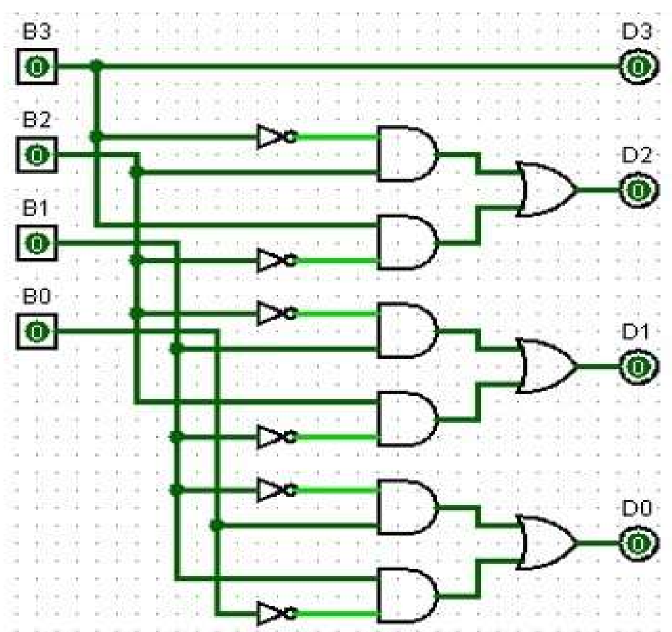
Binary-to-Gray code conversion is a fundamental operation in digital systems that transforms binary numbers into their corresponding Gray code representation. Gray code, where only one bit changes between consecutive numbers, is ideal for applications requiring smooth transitions.

Key benefits:

- Glitch-free operation
- Error detection
- Simplified hardware implementation

This report will cover the algorithm, Verilog implementation, hardware components, and real-world applications of binary-to-Gray code conversion.

Structure of Binary-to-Gray Code Conversion



The above circuit converts a 4-bit Gray code input to a 4-bit BCD output using XOR and AND gates. It's useful in systems where Gray code is used and BCD output is required.

Truth Table of Binary-to-Gray Code Conversion

Natural-binary code				Gray code			
B3	B2	B1	B0	G3	G2	G1	G0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

Applications of Binary to Gray code converstions

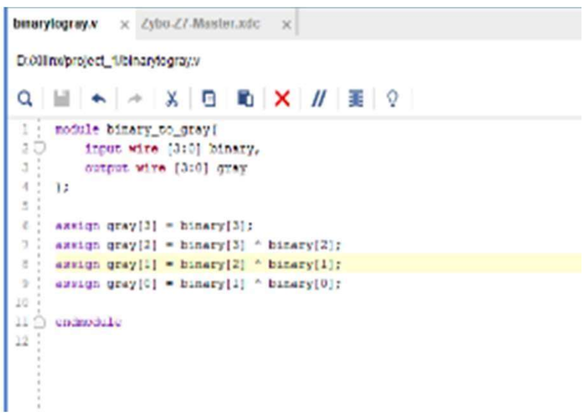
- **Analog-to-digital converters (ADCs):** To reduce glitches and improve accuracy during the conversion process.
- **Rotary encoders:** To provide a smooth and glitch-free output signal.
- **Digital control systems:** To ensure accurate and reliable control signals.
- **Data transmission:** To minimize errors and improve data integrity.

CHAPTER 2

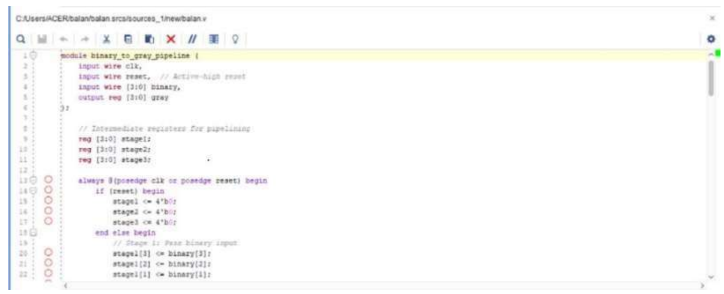
IMPLEMENTATION

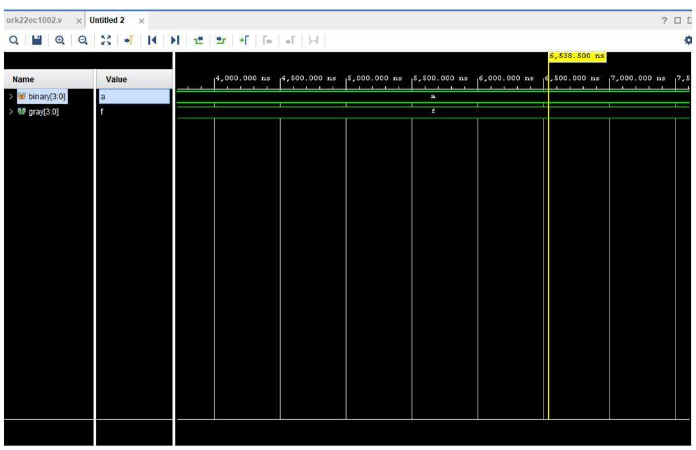
CODE:

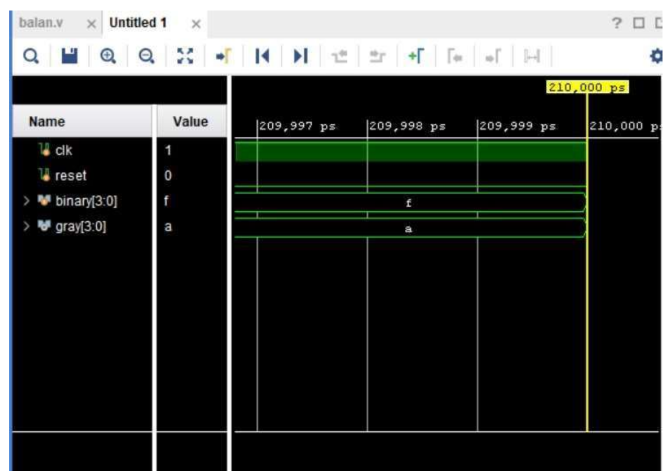
WITH OPTIMIZATION



WITHOUT OPTIMIZATION







SIMULATION:

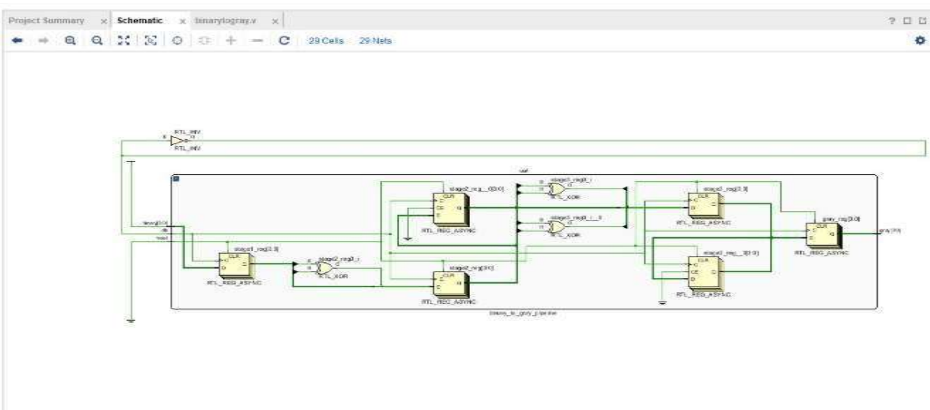
SYNTHESIZE THE DESIGN:

- After writing the code, click **Run Synthesis** in the **Flow Navigator**.
- If there are any errors in your code, resolve them and rerun the synthesis.

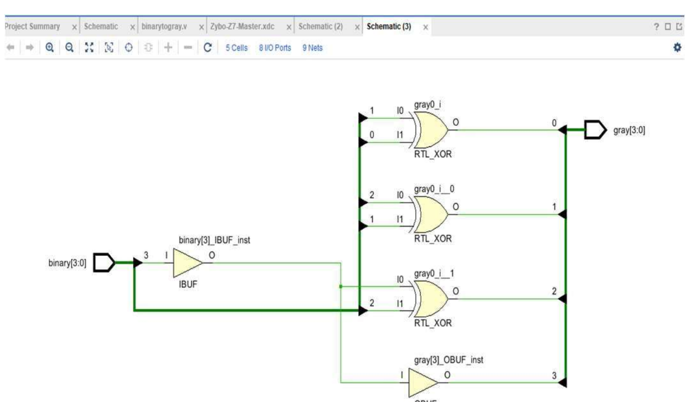
IMPLEMENT THE DESIGN:

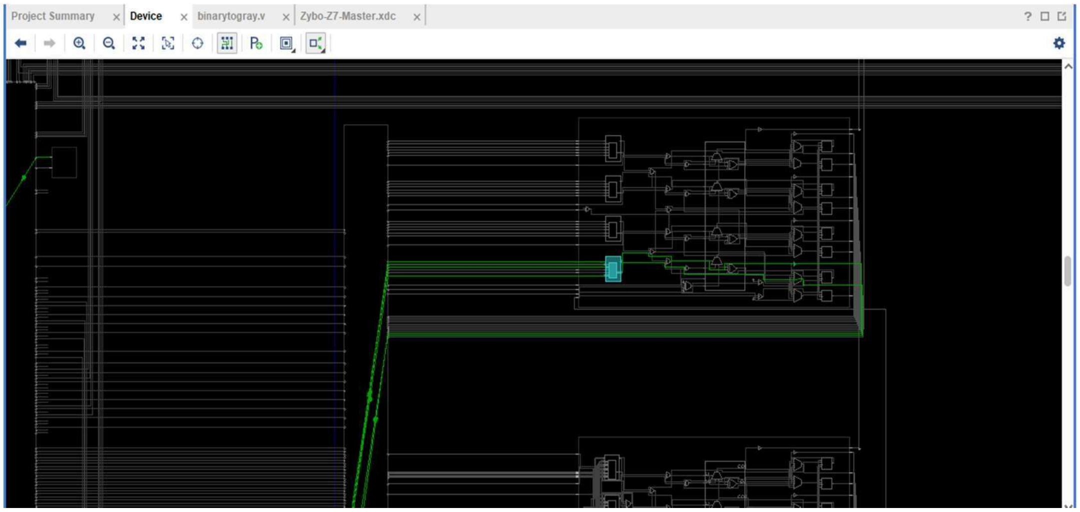
Once synthesis is complete, click **Run Implementation** in the **Flow Navigator**.

- This step maps your design to the actual FPGA resources.



After





GENERATE BITSTREAM:

- After the implementation is completed, click **Generate Bitstream**.

This generates the .bit file, which is required to program your FPGA.

- If prompted to save the design, click **Yes**.

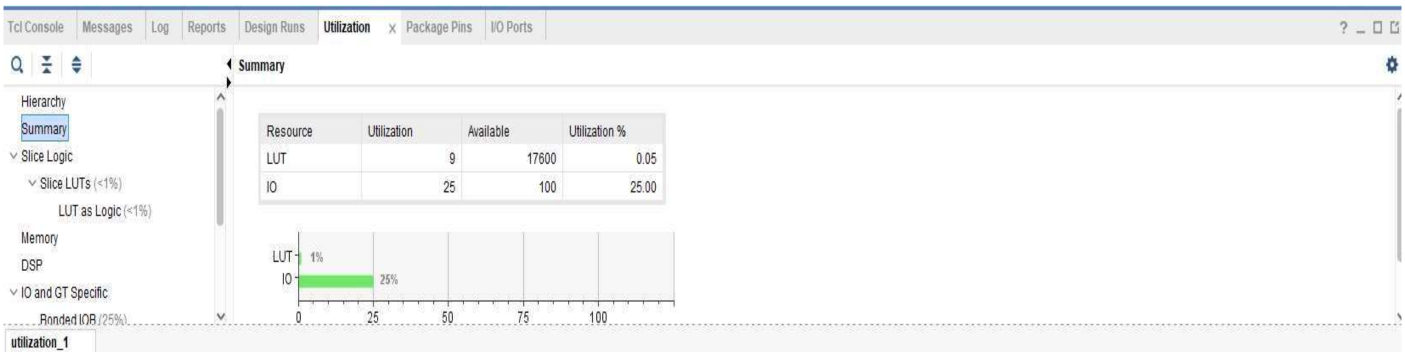
CONFIGURE THE FPGA (DOWNLOAD BITSTREAM):

- Once the bitstream file is generated, open the **Hardware Manager** by clicking **Open Hardware Manager** in the Flow Navigator.
- Click **Open Target** > **Auto Connect** to connect to your FPGA board.
- After the board is connected, click **Program Device** and select the generated bitstream file (.bit).

CHATPER 4

CONCLUSION

BEFORE OPTIMIZATION



AFTER OPTIMIZATION

CHATPER 4

CONCLUSION

TABULATION

CODE	IUT USED	IOB USED
ORIGINAL CODE	9	25
MODIFIED	1	4

CHATPER 4

CONCLUSION

This report has explored the concept of binary-to-Gray code conversion, a fundamental operation in digital systems. We have discussed the algorithm, Verilog implementation, hardware components, and real-world applications of this conversion.

Gray code offers several advantages, including glitch-free operation, error detection, and simplified hardware implementation. By understanding the principles and techniques involved in binary-to-Gray code conversion, you can design and implement digital systems that require precise and reliable operation