Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 33 No. 1s (2026)

Code Converters Using Verilog

Ponnala Lakshmi Prasanna

Seshadri Rao Gudlavalleru Engineering College, JNTUK

Article History:

Received: *08-01-2026*

Revised: *15-01-2026*

Accepted: *15-01-2026*

Abstract

In digital electronics Code converters are logic circuits that translate data from one binary code format to another (e.g., Binary to BCD, BCD to Gray, BCD to Excess-3) to enable communication between systems.

The paper n-bit Code Converters Using Verilog includes three Decoders 4-bit Gray to Binary and 4-bit Binary to Gray Code Converters. The two Code Converters are designed, simulated and Synthesized Using Verilog. The Verilog Modules of each Code Converter are developed and Synthesized to obtain RTL and Technology schematics. In the next step, The Verilog Test benches are developed for each Code Converter and simulated using Behavioral simulation to obtain the Output waveforms. Next The Output waveforms are verified as per the given Truth Tables.

The design summary of each Code Converter can be obtained after synthesization and simulation. The design summary includes Timing Summary, Device Utilization Summary, Primitive and Black Box Usage and Timing Reports etc.

In future n-bit Code Converters can be further implemented for increased value of n with all possible input combinations. n-bit Code Converters can be designed Using VHDL as well as other HDL languages also and the design can be implemented using Field Programmable Gate Array(FPGA).

Keywords: Verilog, VHDL,HDL, FPGA, RTL,

1. Introduction

In digital electronics Code converters are logic circuits that translate data from one binary code format to another (e.g., Binary to BCD, BCD to Gray, BCD to Excess-3) to enable communication between systems using different coding schemes, making them compatible for processing, display, or error reduction. These essential components use logic gates to perform the conversion, ensuring seamless interoperability in digital systems like calculators, clocks, and encoders.

Key Types of Code Converters:

- **Binary to Gray Code Converter:** Converts standard binary to Gray code (where only one bit changes between successive values), crucial for preventing errors in mechanical systems like encoders.
- **Gray Code to Binary Converter:** Reverses the process, converting Gray code back to standard binary for processing.
- **Binary to BCD Converter (8421):** Converts standard binary numbers (up to 15) to Binary Coded Decimal, used for decimal displays.
- **BCD to Excess-3 Converter:** Adds 3 to a BCD number, often used in arithmetic circuits and error detection.
- **Excess-3 to BCD Converter:** Converts Excess-3 code back to BCD, essential for displaying numbers.

- **BCD to Seven-Segment Display Converter:** Converts a BCD digit into seven output signals to illuminate the correct segments of a display.

2. Objective

The objective of the research paper is to develop 4-bit Binary to Gray and Gray to Binary converters: using Verilog.

To simulate and synthesize the two code converters using Software Xilinx 14.7 and To obtain the corresponding RTL, Technology Schematics, the design summaries, the Output Waveforms for the given Test bench.

3. Methods

4-bit Gray to Binary code converter:-A 4-bit Binary to Gray code converter has 4 inputs and 4 outputs.

Figure1: Block Diagram of 5-bit Binary code converter.

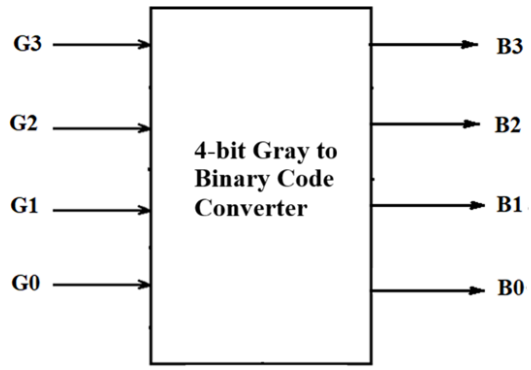


Fig: 4-bit Gray to Binary Code Converter

The Block Diagram of 4-bit Gray to Binary code converter consists of 4 inputs from G3 – G_o and output B_3-B_0.

Working of 4-bit Gray to Binary code converter:-

The working of 4-bit Gray to Binary converter is based on the following equations

$B_3 = G_3$;

$B_2 = G_3$ xor G_2;

$B_1 = G_3$ xor G_2 xor G_1;

$B_0 = G_3$ xor G_2 xor G_1 xor G_0;

The working of 4-bit Gray to Binary code converter is as follows,

When the input $G_3G_2G_1G_0 = 0000$ then the output $B_3B_2B_1B_0 = 0000$.

When the input $G_3G_2G_1G_0 = 0001$ then the output $B_3B_2B_1B_0 = 0001$.

When the input $G_3G_2G_1G_0 = 0010$ then the output $B_3B_2B_1B_0 = 0011$.

When the input $G_3G_2G_1G_0 = 0011$ then the output $B_3B_2B_1B_0 = 0010$.

When the input $G_3G_2G_1G_0 = 0100$ then the output $B_3B_2B_1B_0 = 0111$.

When the input $G_3G_2G_1G_0 = 0101$ then the output $B_3B_2B_1B_0 = 0110$.

When the input $G_3G_2G_1G_0 = 0110$ then the output $B_3B_2B_1B_0 = 0100$.

When the input $G_3G_2G_1G_0 = 0111$ then the output $B_3B_2B_1B_0 = 0101$.

When the input $G_3G_2G_1G_0 = 1000$ then the output $B_3B_2B_1B_0 = 1111$.

When the input $G_3G_2G_1G_0 = 1001$ then the output $B_3B_2B_1B_0 = 1110$.

When the input $G_3G_2G_1G_0 = 1010$ then the output $B_3B_2B_1B_0 = 1100$.

When the input $G_3G_2G_1G_0 = 1011$ then the output $B_3B_2B_1B_0 = 1101$.

When the input $G_3G_2G_1G_0 = 1100$ then the output $B_3B_2B_1B_0 = 1000$.

When the input $G_3G_2G_1G_0 = 1101$ then the output $B_3B_2B_1B_0 = 1001$.

When the input $G_3G_2G_1G_0 = 1110$ then the output $B_3B_2B_1B_0 = 1011$.

When the input $G_3G_2G_1G_0 = 1111$ then the output $B_3B_2B_1B_0 = 1010$.

Table1: Truth Table of 4-bit Gray to Binary code converter

G3	G2	G1	G0	B3	B2	B1	B0
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	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
1	0	0	0	1	1	1	1
1	0	0	1	1	1	1	0
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	0	1	0	1	1
1	1	1	1	1	0	1	0

Figure2: Logic Diagram of 4-bit Gray to Binary code converter

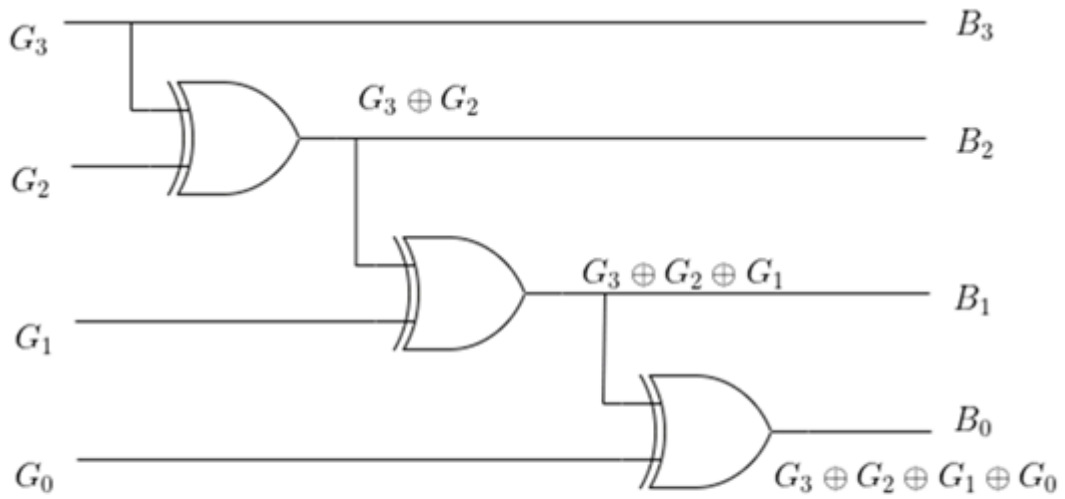


Fig: 4-bit Gray to Binary Code Converter

Results

Figure3: RTL schematic of 4-bit Gray to Binary code converter

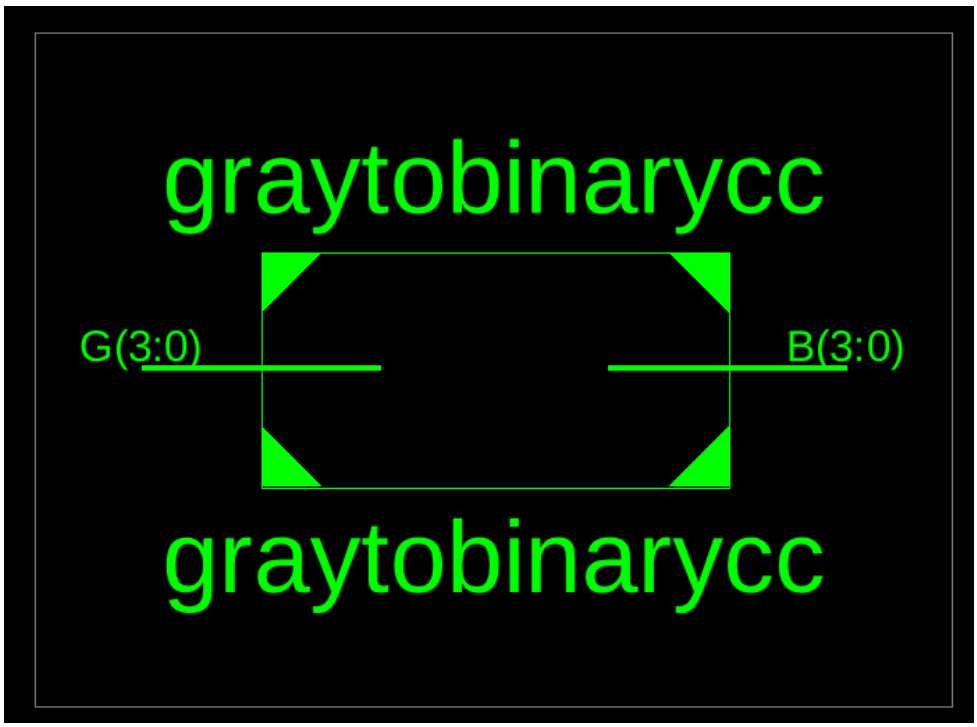


Figure4: Technology schematic of 4-bit Gray to Binary code converter

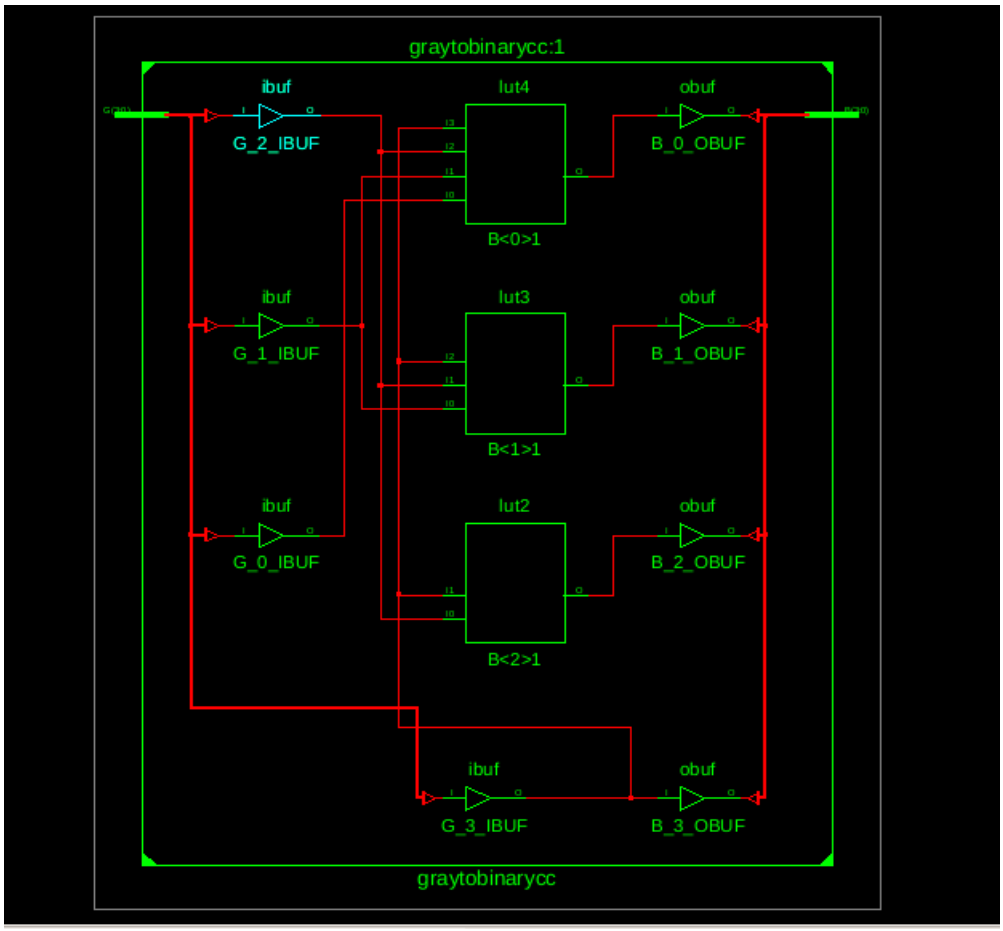
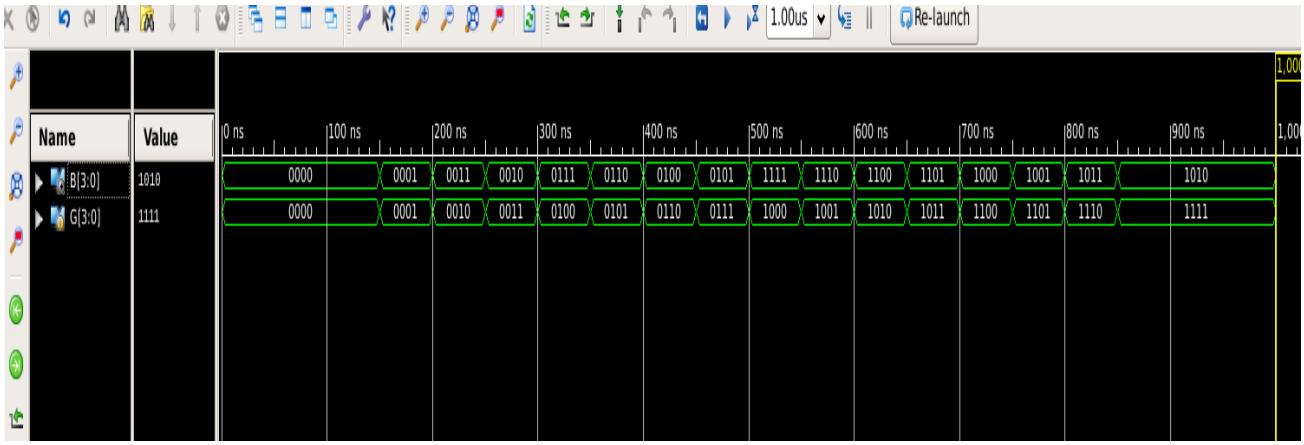


Figure5: Output Waveform of 4-bit Gray to Binary code converter



Discussion of Results:-

The output waveforms are shown in Binary values. From the above output waveform it is clear that, the output is Binary code when the input is in Gray code. G[3:0] represents Gray input and B[3:0] represents Binary code. For example Gray code input 1110 has Binary code equivalent 1011 and so on.

Table2: Timing Summary of 4-bit Gray to Binary code converter

Speed Grade: -3

Minimum period	No path found
Minimum input arrival time before clock	No path found
Maximum output required time after clock	No path found
Maximum combinational path delay	0.934ns

Table3: Device utilization summary of 4-bit Gray to Binary code converter

Selected Device: 7a100tcsg324-3

Slice Logic Utilization:	
Number of Slice LUTs:	3 out of 63400 0%
Number used as Logic:	3 out of 63400 0%
Slice Logic Distribution:	
Number of LUT Flip Flop pairs used	3
Number with an unused Flip Flop	3 out of 3 100%
Number with an unused LUT	0 out of 3 0%
Number of fully used LUT-FF pairs	0 out of 3 0%
Number of unique control sets	0
IO Utilization:	
Number of IOs	8
Number of bonded IOBs	8 out of 210 3%
Specific Feature Utilization:	
Number of BUFG/BUFGCTRLs	-

Table4: Primitive and Black Box Usage of 4-bit Gray to Binary code converter

#BELS	3
#LUT2	1
#LUT3	1
#LUT4	1
# IO Buffers	8
# IBUF	4
# OBUF	4

4-bit Binary to Gray code converter:-

A 4-bit Binary to Gray code converter has 4 inputs $B_3B_2B_1B_0$ and four outputs $G_3G_2G_1G_0$

Figure6: Block Diagram of 4-bit Binary to Gray code converter

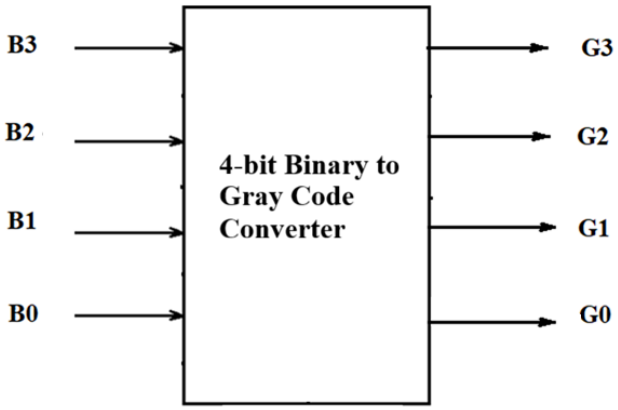


Fig: 4-bit Binary to Gray Code Converter

Working of 4-bit Binary to Gray code converter:-

The working of 4-bit Binary to Gray converter is based on the following equations

$G_3 = B_3$;

$G_2 = B_3$ xor B_2;

$G_1 = B_2$ xor B_1;

$G_0 = B_0$ xor B_1;

The working of 4-bit Binary to Gray code converter is as follows,

When the input $B_3B_2B_1B_0 = 0000$ then the output $G_3G_2G_1G_0 = 0000$.

When the input $B_3B_2B_1B_0 = 0001$ then the output $G_3G_2G_1G_0 = 0001$.

When the input $B_3B_2B_1B_0 = 0010$ then the output $G_3G_2G_1G_0 = 0011$.

When the input $B_3B_2B_1B_0 = 0011$ then the output $G_3G_2G_1G_0 = 0010$.

When the input $B_3B_2B_1B_0 = 0100$ then the output $G_3G_2G_1G_0 = 0110$.

When the input $B_3B_2B_1B_0 = 0101$ then the output $G_3G_2G_1G_0 = 0111$.

When the input $B_3B_2B_1B_0 = 0110$ then the output $G_3G_2G_1G_0 = 0101$.

When the input $B_3B_2B_1B_0 = 0111$ then the output $G_3G_2G_1G_0 = 0100$.

When the input $B_3B_2B_1B_0 = 1000$ then the output $G_3G_2G_1G_0 = 1100$.

When the input $B_3B_2B_1B_0 = 1001$ then the output $G_3G_2G_1G_0 = 1101$.

When the input $B_3B_2B_1B_0 = 1010$ then the output $G_3G_2G_1G_0 = 1111$.

When the input $B_3B_2B_1B_0 = 1011$ then the output $G_3G_2G_1G_0 = 1110$.

When the input $B_3B_2B_1B_0 = 1100$ then the output $G_3G_2G_1G_0 = 1010$.

When the input $B_3B_2B_1B_0 = 1101$ then the output $G_3G_2G_1G_0 = 1011$.

When the input $B_3B_2B_1B_0 = 1110$ then the output $G_3G_2G_1G_0 = 1001$.

When the input $B_3B_2B_1B_0 = 1111$ then the output $G_3G_2G_1G_0 = 1000$.

Table5:-Truth Table 4-bit Binary to Gray code converter

B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0
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

The Truth Table of 4-bit Binary to Gray code converter is given in the following Table. It has 4 inputs B_3- B_0 and four outputs G_3 – G_0. For Example $B_3B_2B_1B_0$=1000 then the corresponding Gray code $G_3G_2G_1G_0$=1100, similarly $B_3B_2B_1B_0$=1010 then the corresponding Gray code $G_3G_2G_1G_0$=1111 and so on.

Figure7: Logic Diagram of 4-bit Binary to Gray code converter

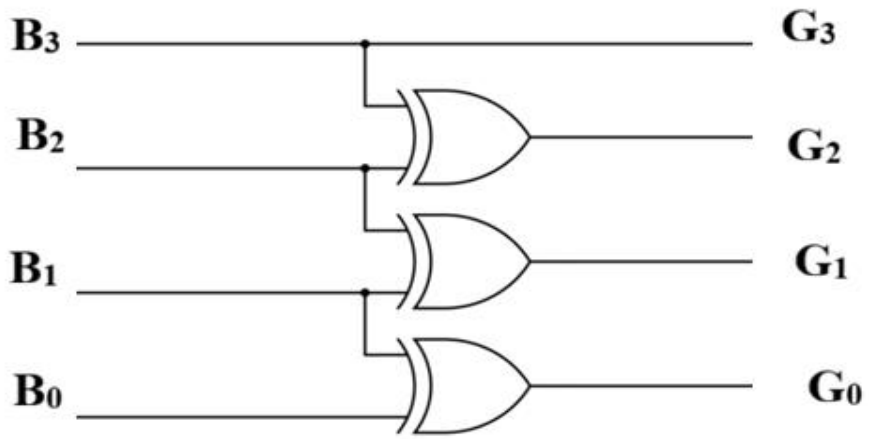


Fig: 4-bit Binary to Gray Code Converter

Results

Figure8: RTL schematic of 4-bit Binary to Gray code converter

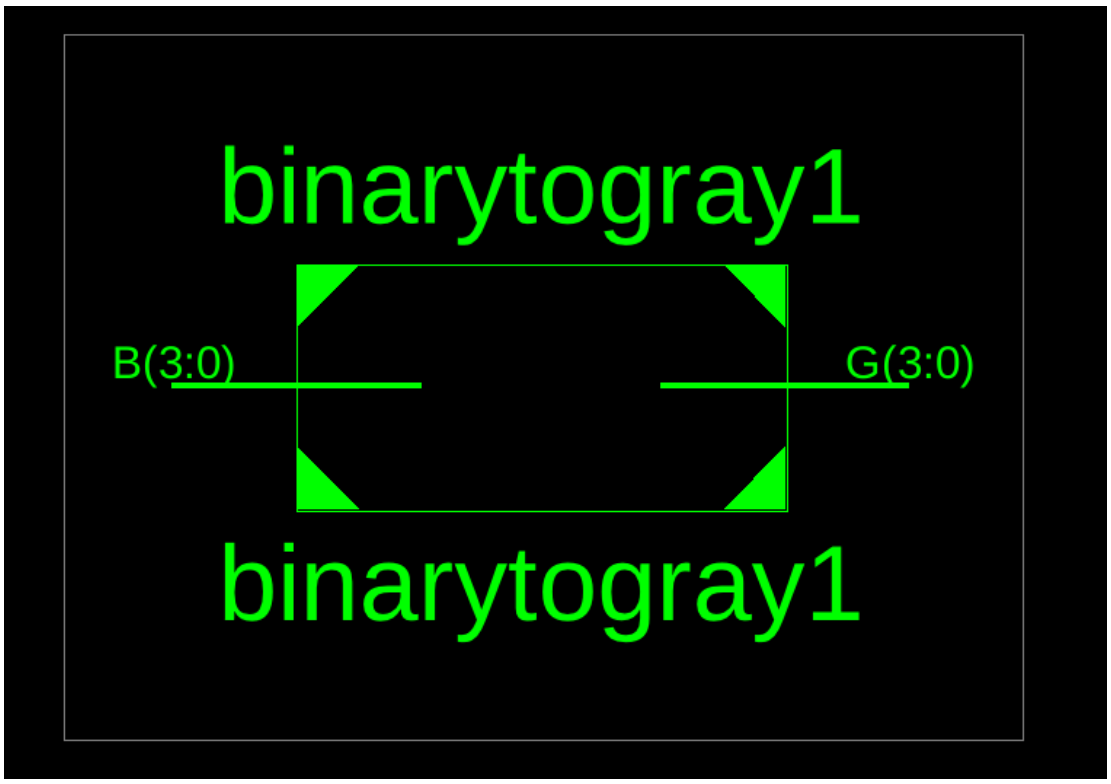


Figure9: Technology schematic of 4-bit Binary to Gray code converter

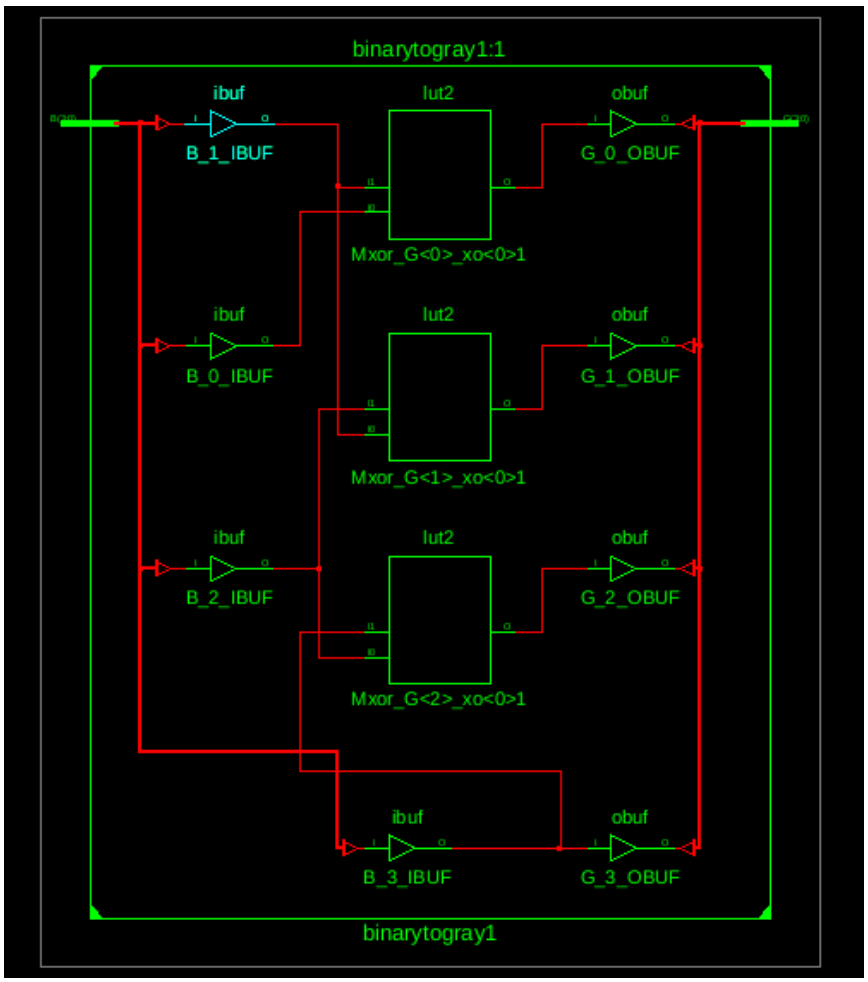
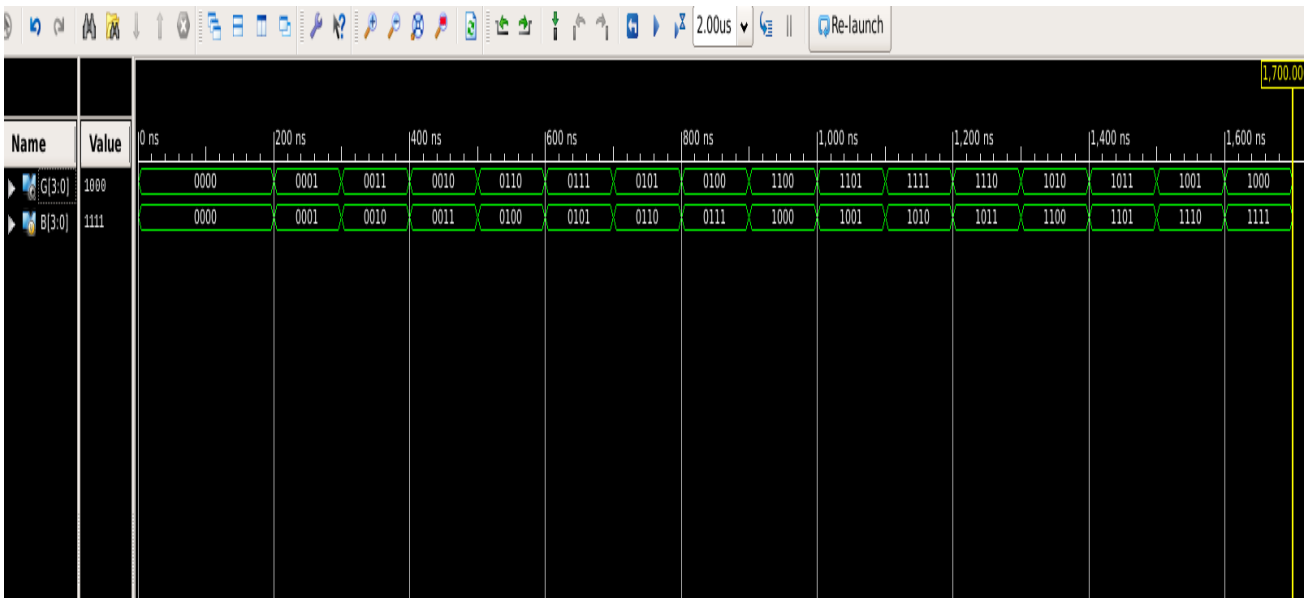


Figure10: Output Waveform of 4-bit Binary to Gray code converter



Discussion of Results:-

The output waveforms are shown in Binary values. From the above output waveform it is clear that, the output is Gray code when the input is in Binary code. B[3:0] represents Binary input and G[3:0] represents Gray code. For example Binary code input 1111 has Gray code equivalent 1000 and so on.

Table6: Timing Summary of 4-bit Binary to Gray code converter

Speed Grade: -3

Minimum period	No path found
Minimum input arrival time before clock	No path found
Maximum output required time after clock	No path found
Maximum combinational path delay	0.761ns

Table7: Device utilization summary of 4-bit Binary to Gray code converter

Selected Device: 7a100tcsg324-3

Slice Logic Utilization:	
Number of Slice LUTs: Number used as Logic:	3 out of 63400 0% 3 out of 63400 0%
Slice Logic Distribution:	
Number of LUT Flip Flop pairs used	3
Number with an unused Flip Flop	3 out of 3 100%
Number with an unused LUT	0 out of 3 0%
Number of fully used LUT-FF pairs	0 out of 3 0%
Number of unique control sets	0
IO Utilization:	
Number of IOs	8
Number of bonded IOBs	8 out of 210 3%
Specific Feature Utilization:	
Number of BUFG/BUFGCTRLs	-

Table8: Primitive and Black Box Usage of 4-bit Binary to Gray code converter

#BELS	3
#LUT2	3
# IO Buffers	8
# IBUF	4
# OBUF	4

Conclusions

Thus 4-bit Gray to Binary and Binary to Gray Code Converters are synthesized and simulated Using Verilog and the Results are obtained.

Future scope

In future Code Converters can be further implemented for increased value of n. n-bit Code Converters can be designed Using VHDL as well as other HDL languages also and the design can be implemented using Field Programmable Gate Array (FPGA).

References

[1] Cihun-Siyong Alex Gong, Muh-Tian Shiue, Ci-Tong Hong, and KaliWen Yao, "Analysis and Design of an Efficient Irreversible Energy Recovery Logic in 0.18m CMOS," *IEEE Trans. Circuits Syst. I*, vol. 55, No. 9, pp. 2595–2607, Oct. 2008.

[2] Hirokatsu Shirahama and Takahiro Hanyu, "Design of High-Performance Quaternary Adders Based on Output-Generator Sharing ", *Proceedings of the 38th International Symposium on Multiple Valued Logic*, pp. 8–13. 2008.

[3] S.M. Basir, G.A. Mhdiraji, A. Malekmohammadi, N.H. Ibrahim, A.F. Abas and M.K. Abdullah, "Demonstration of Duty Cycle Division Multiplexing with bit error rate estimation," in *2009 IEEE 9th Malaysis International Conference on Communications (MICC)*, 2009.

[4] S. Vijayakumar and B. Karthikeyan, "Mixed style of Low Power Multiplexer Design for Arithmetic Architectures using 90nm Technology ", *Proceedings of the 12th international conference on Networking, VLSI and signal processing (ICNVS)*, pp. 83–87, February 2010.

[5] Ji-Seop Song, Sang-Hyuk Yang, Eung-ju Kim, Shin-II Lim and Suki Kim, "A High Speed and Low Power 4:1 Multiplexer Architecture for Graphic Memory Interface " accepted to 2010 International Conference on Electronics, Information, and Communication.

[6] Vasundara Patel K.S., K.S. Gurumurthy, "Arithmetic operations in multi-valued logic ", *International journal of VLSI design and communication system (VLSICS)*, vol. 1, no. 1, pp. 21–32, March 2010.

[7] Anindya Dal, Ifat Jahangir and Masud Hasan, "Design of Quaternary Serial and Parallel Adders ", *ICECE 2010, 6th International Conference on Electrical and Computer Engineering*, 18-20 December 2010.