MIT World Peace University

Digital Electronics and Computer Architecture
Second Year B. Tech, Semester 3

4 Bit Code Conversion between Binary and Gray Code using Basic Logic Gates

Practical Report

Prepared By

Krishnaraj Thadesar
Cyber Security and Forensics
Batch A2, PA 34

September 11, 2022

Contents

1 Problem Statement

Design and Implementation of 4 Bit code convertors using Basic Logic Gates.

1. 4 Bit Binary to Gray Code
2. 4 Bit Gray to Binary Code

2 ICs Used

74LS86 (Quad 2-Input Exclusive - OR Gates)

3 Platform Used

Digital Trainer Kit

4 Theory

4.1 Involved Truth Tables

4.1.1 Binary to Gray Code

Binary Code				**Gray Code**			
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

4.1.2 Gray to Binary Code

Gray Code				Binary Code			
G_3	G_2	G_1	G_0	B_3	B_2	B_1	B_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	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	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
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

4.1.3 XOR Gate

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

4.2 Reduced Boolean Expressions from Respective Karnaugh Maps

4.2.1 Karnaugh Maps for Binary to Gray

1. For Variable G_3

B_3B_2 \ B_1B_0	00	01	11	10
00				
01				
11	1	1	1	1
10	1	1	1	1

Simplification of Karnaugh Map Expression from the above K - Map :

$$G_3 = B_3 \tag{1}$$

1. For Variable G_2

B_3B_2 \ B_1B_0	00	01	11	10
00				
01	1	1	1	1
11				
10	1	1	1	1

Simplification of Karnaugh Map Expression from the above K - Map :

$$G_2 = \overline{B_3}B_2 + \overline{B_2}B_3 \tag{2}$$

$$G_2 = B_3 \oplus B_2 \tag{3}$$

1. For Variable G_1

B_3B_2 \ B_1B_0	00	01	11	10
00			1	1
01	1	1		
11	1	1		
10			1	1

Simplification of Karnaugh Map Expression from the above K - Map :

$$G_1 = \overline{B_1}B_2 + \overline{B_2}B_1 \tag{4}$$
$$G_1 = B_2 \oplus B_1 \tag{5}$$

1. For Variable G_0

B_3B_2 \ B_1B_0	00	01	11	10
00		1		1
01		1		1
11		1		1
10		1		1

Simplification of Karnaugh Map Expression from the above K - Map :

$$G_0 = \overline{B_0}B_1 + \overline{B_1}B_0 \tag{6}$$
$$G_0 = B_1 \oplus B_0 \tag{7}$$

Final Expressions for Binary to Gray

$$G_0 = B_3 \tag{8}$$
$$G_2 = B_3 \oplus B_2 \tag{9}$$
$$G_1 = B_2 \oplus B_1 \tag{10}$$
$$G_0 = B_1 \oplus B_0 \tag{11}$$

4.2.2 Karnaugh Maps for Gray to Binary

There is difference when we plot minterms for K maps here, as K Maps are inherently written in Gray code, when we write the min terms, we have to fill them in the respective Gray code > Decimal value, instead of the usual binary > Decimal value.
1. For Variable B_3

		G_1G_0			
		00	01	11	10
	00				
	01				
G_3G_2	11	1	1	1	1
	10	1	1	1	1

Simplification of Karnaugh Map Expression from the above K - Map :

$$B_3 = G_3 \tag{12}$$

1. For Variable B_2

		G_1G_0			
		00	01	11	10
	00				
	01	1	1	1	1
G_3G_2	11				
	10	1	1	1	1

Simplification of Karnaugh Map Expression from the above K - Map :

$$B_2 = \overline{G_2}G_3 + \overline{G_2}G_2B_2 = G_3 \oplus G_2 \tag{13}$$

1. For Variable B_1

G_1G_0

G_3G_2	00	01	11	10
00			1	1
01	1	1		
11			1	1
10	1	1		

Simplification of Karnaugh Map Expression from the above K - Map :

$$\begin{aligned} B_1 &= \bar{G_3}G_2\bar{G_1} + G_3\bar{G_2}\bar{G_1} + \bar{G_3}\bar{G_2}G_1 + G_3G_2G_1 \\ &= G_3(\bar{G_2}\bar{G_1} + G_2G_1) + \bar{G_3}(G_2\bar{G_1} + \bar{G_2}G_1) \\ &= G_3(\overline{G_2\bar{G_1}} + \bar{G_2}G_1) + \bar{G_3}(G_2\bar{G_1} + \bar{G_2}G_1) \\ &= G_3(\overline{G_2 \oplus G_1}) + \bar{G_3}(\overline{G_2 \oplus G_1}) = G_3 \oplus G_2 \oplus G_1 \end{aligned} \tag{14}$$

$$B_1 = G_3 \oplus G_2 \oplus G_1 \tag{15}$$

1. For Variable B_0

G_1G_0

G_3G_2	00	01	11	10
00		1		1
01	1		1	
11		1		1
10	1		1	

Simplification of Karnaugh Map Expression from the above K - Map :

$$\begin{aligned} B_0 = \bar{G_3}\bar{G_2}\bar{G_0}G_0 + \bar{G_3}\bar{G_2}G_1\bar{G_0} + \bar{G_3}G_2\bar{G_1}\bar{G_0} + \bar{G_3}G_2G_1G_0 + G_3G_2\bar{G_1}G_0 + G_3G_2G_1\bar{G_0} + G_3\bar{G_2}\bar{G_1}\bar{G_0} \\ +G_3\bar{G_2}G_1G_0 = G_3 \oplus G_2 \oplus G_1 \oplus G_0 \end{aligned} \tag{16}$$

$$B_0 = G_3 \oplus G_2 \oplus G_1 \oplus +G_0 \tag{17}$$

Final Expressions for Gray to Binary

$$B_3 = G_3 \tag{18}$$
$$B_2 = G_3 \oplus G_2 \tag{19}$$
$$B_1 = G_3 \oplus G_2 \oplus G_1 \tag{20}$$
$$B_0 = G_3 \oplus G_2 \oplus G_1 \oplus +G_0 \tag{21}$$

5 Design and Implementation

1. We only need XOR gates for this so, we can just start with drawing the Inputs for Binary to Gray B0, B1, B2, B3 and Gray to Binary G0, G1, G2, G3 respectively.
2. Refer to the final Expressions obtained from the K-Maps for each conversion.
3. Join the inputs to the respective Gate Inputs
4. To make the Circuit Diagram, Replace the Gates with their respective ICs, in this case IC7486 for XOR gates, and join inputs of the IC to the Inputs respectively.
5. Write the Pin numbers of the IC after marking $V_c c$ and GND.

5.1 Binary to Gray Conversion Logic Gate Design

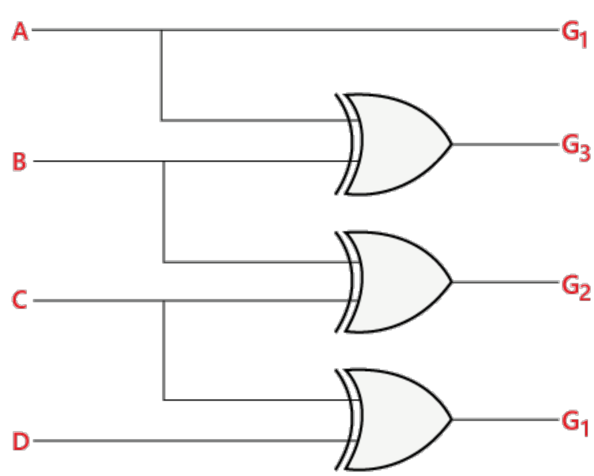


Figure 1: Binary to Gray Code conversion using Logic Gates

5.2 Gray to Binary Conversion Logic Gate Design

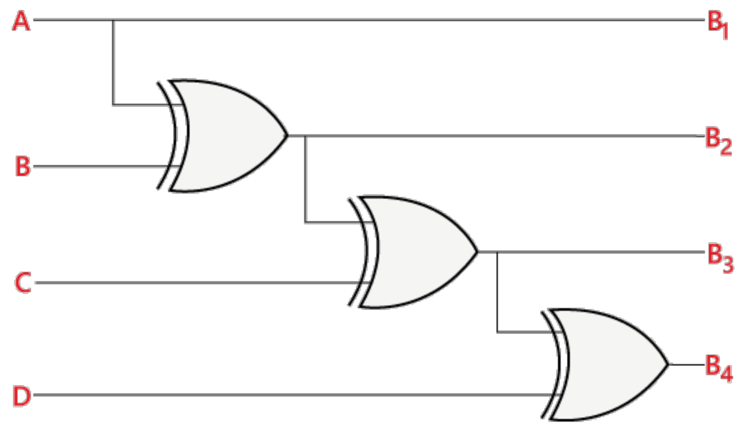
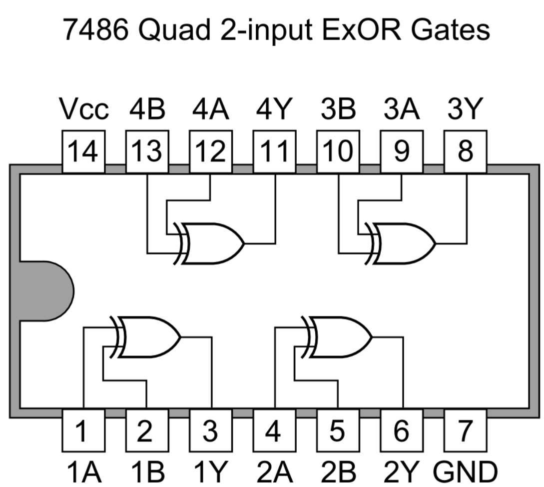


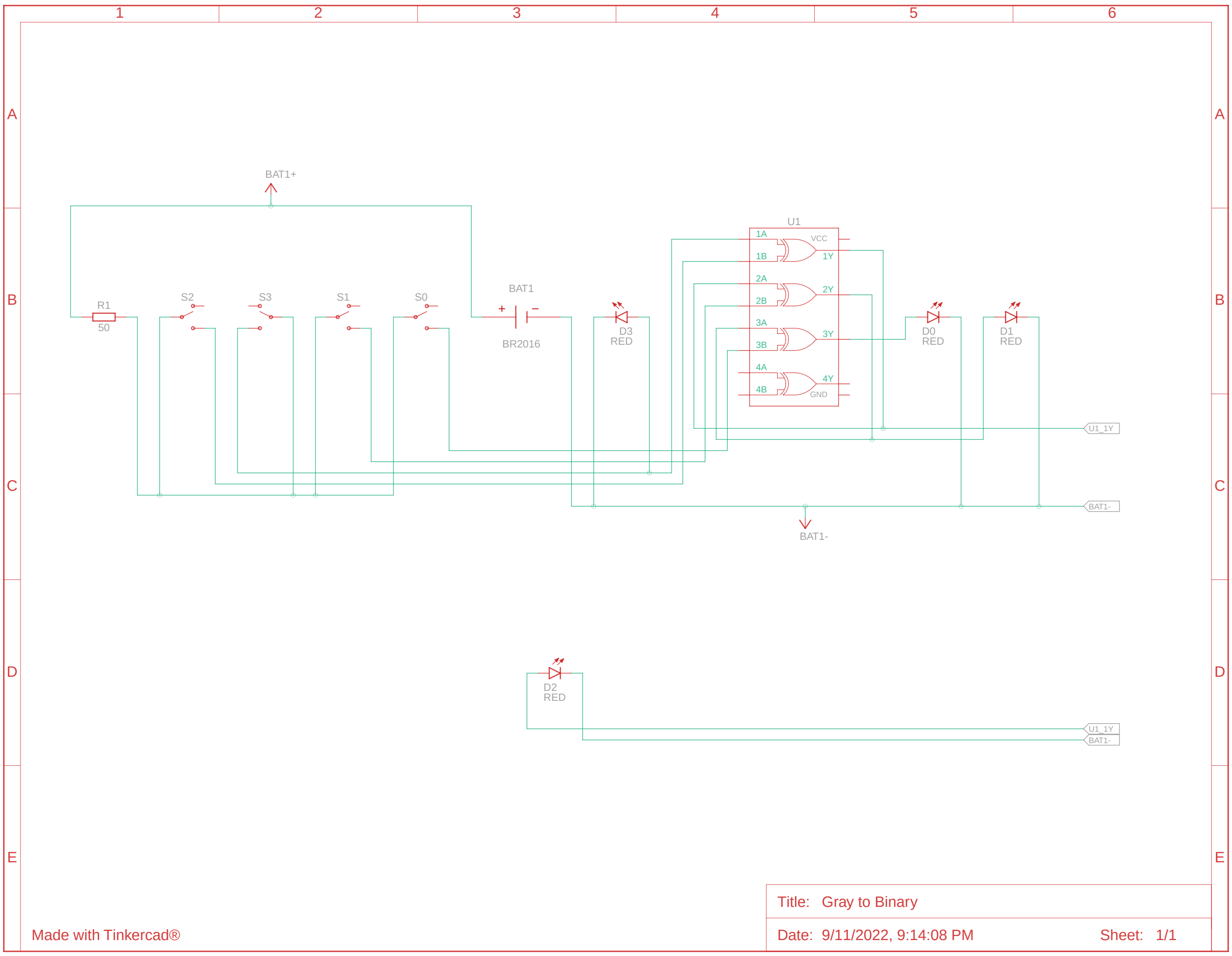
Figure 2: Gray to Binary conversion using Logic Gates

6 Circuit Diagrams

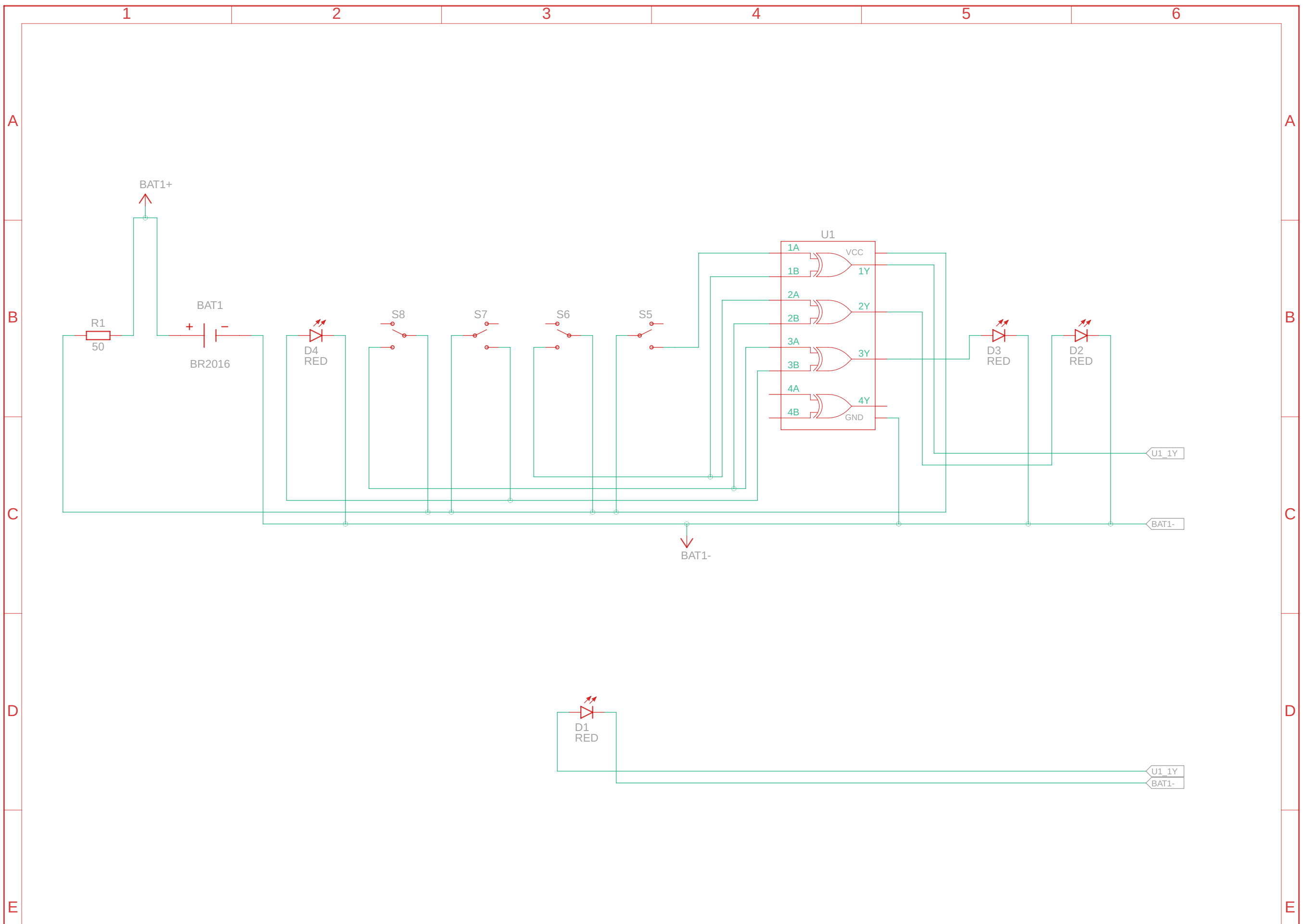
6.1 Pin Diagram of IC 7486



6.2 Circuit Diagrams for Conversions



BAT1+
U1
1A
1B
2A
2B
3A
3B
4A
4B
VCC
1Y
2Y
3Y
4Y
GND
R1
50
S2
S3
S1
S0
BAT1
BR2016
D3
RED
D0
RED
D1
RED
D2
RED
U1_1Y
BAT1-
BAT1-
U1_1Y
BAT1-
Made with Tinkercad®
Title: Gray to Binary
Date: 9/11/2022, 9:14:08 PM
Sheet: 1/1



Made with Tinkercad®

Title: Binary to gray

Date: 9/11/2022, 8:51:15 PM

Sheet: 1/1

7 Procedure

1. Design Combinational logic circuits as per given problem statement.
2. connect the IC 7486 and other basic logic gate ICs as per diagram.
3. Give V_{cc} supply and ground connection to each IC.
4. Give variaous combinations to select lines.
5. Observe the output and verify the truth table.
6. Switch off the power supply off the trainer kit.

8 Conclusion

We have learned the Implementation of Binary to Gray and Gray to Binary code converter using logic gates.

9 FAQs

1. **Why is code conversion Necessary?**
   (a) Code conversions are most commonly used in computers, digital electronics, and microprocessors, etc. There are numerous codes like binary, octal, hexadecimal, Binary Coded Decimal (BCD), Excess-3, Gray code, Error Correcting Codes (ECCs) and ASCII code ...
   (b) Each Code just represents the basic numbers 1, 2, 3, 4, etc in a different Way. Each method is unique, and has some specific use.
   (c) In performing calculations and execution of certain algrithms, Certain types of codes may provide input that can be easily applied to the algorithm, as opposed to another type of code.
   (d) To Reduce the number of components used in a circuit, different types of such conversions from one code to another may be used, by applying different logic.
   (e) An Example is Gray code, which is used in shaft encoders because The code of successive numbers differs exactly by one bit from its preceder.
   (f) Excess- 3 is extensively used for subtraction because every code in XS-3 has its complement. 1s complement of the code yields 9s complement of a number itself.
   (g) Alphanumeric codes by ASCII standards are widely used as a representation systems to the character set in computers.
   (h) BCD is used in 7 Segment Displays
2. **Write any 2 Applications of Gray Code**
   (a) The gray code is used in a few specific applications. The main applications include being used in analog to digital converters, as well as being used for error correction in digital communication. Gray code is used to minimize errors in converting analog signals to digital signals.
   (b) Boolean circuit minimization

(c) Communication between clock domains

(d) Error correction

(e) Genetic algorithms

(f) Mathematical puzzles

(g) Position encoders

3. **Explain Weighted and Non Weighted Codes with Examples**

(a) Weighted Code: The weighted codes are those that obey the position weighting principle,which states that the position of each number represent a specific weight.

(b) Example of a Weight Code would be Binary, BCD, hex, Octal, etc where each position has a certain weight.

(c) Non Weighted Codes: It is a type of code that does not have any positional weights associated with it.

(d) Example would be Decimal, Binary, ASCII, Gray Code, etc.