

# **Lecture 14**

## **Code Converter**

# Binary to Gray Code Converter

The Gray Code is unweighted and is not an arithmetic code: that is, there are no specific weights assigned to the bit positions. The important feature of the Gray Code is that it exhibits only a single bit change from one code word to the next in sequence. This property is important in many applications, such as shaft position encoders, where error susceptibility increases with the number of bit changes between adjacent numbers in a sequence.

To convert a binary number to a Gray Code number, the following rules apply.

1. The most significant digit ( Left Most Bit ) in the Gray Code is the same as the corresponding digit in the binary number.
2. Going from left to right, add each adjacent pair of binary digits to get the next Gray code digit, regardless carries.

**For instance** - Let us convert the binary number 1010 to Gray Code.

**Step 1** - The left most Gray digit is the same as the left most binary

1	0	1	0	Binary
↓				
1				Gray

**Step 2** - Add the left most binary digit to the adjacent one.

1	+	0	1	0	Binary
		↓			
1		1			Gray

**Step 3** - Add the next adjacent pair

1	0	+	1	0	Binary
			↓		
1	1		1		Gray

**Step 4** - Add the last adjacent pair

1	0	1	+	0	Binary
			↓		
1	1	1	1		Gray

The conversion is now complete and the Gray Code is 1111.

# Steps to design the converter

1. Design a converter by the following procedures:

- a. Write down the truth table of both input and output bits of the converter.
- b. Apply Karnaugh Map to look for the minimized logic expression for the output bits.
- c. Implement the logic gates by using Circuit Maker.

Example:

For Binary to Gray Code Converter, binary bits are input and gray code bits are output. So first write the truth table for binary bits and gray code. Then k-map for the all bits of gray code, find the simplified expression for each bit of gray code. Then design the logical circuit.

# Truth Table

Decimal	Binary				Gray			
	A	B	C	D	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

# K-Map for each bit of Gray code

For practical consideration code conversions are made for each bit.

(a) For  $Y_3$ , the Karnaugh map can draw as follow.

	$\overline{C}\overline{D}$	$\overline{C}D$	$CD$	$C\overline{D}$
$\overline{A}\overline{B}$	0	0	0	0
$\overline{A}B$	0	0	0	0
$AB$	1	1	1	1
$A\overline{B}$	1	1	1	1

(a)

By minimization,  $Y_3 = A$



(b) For  $Y_2$ , the Karnaugh map can draw as follow.

	$\overline{C}\overline{D}$	$\overline{C}D$	$CD$	$C\overline{D}$
$\overline{A}\overline{B}$	0	0	0	0
$\overline{A}B$	1	1	1	1
$AB$	0	0	0	0
$A\overline{B}$	1	1	1	1

(b)

By minimization,  $Y_2 = A \oplus B$

(c) For  $Y_1$ , the Karnaugh map can draw as follow.

	$\overline{C}\overline{D}$	$\overline{C}D$	$CD$	$C\overline{D}$
$\overline{A}\overline{B}$	0	0	(1   1)	
$\overline{A}B$	(1   1)		0   0	
$AB$	(1   1)		0   0	
$A\overline{B}$	0	0	(1   1)	

(a)

By minimization,  $Y_1 = B \oplus C$

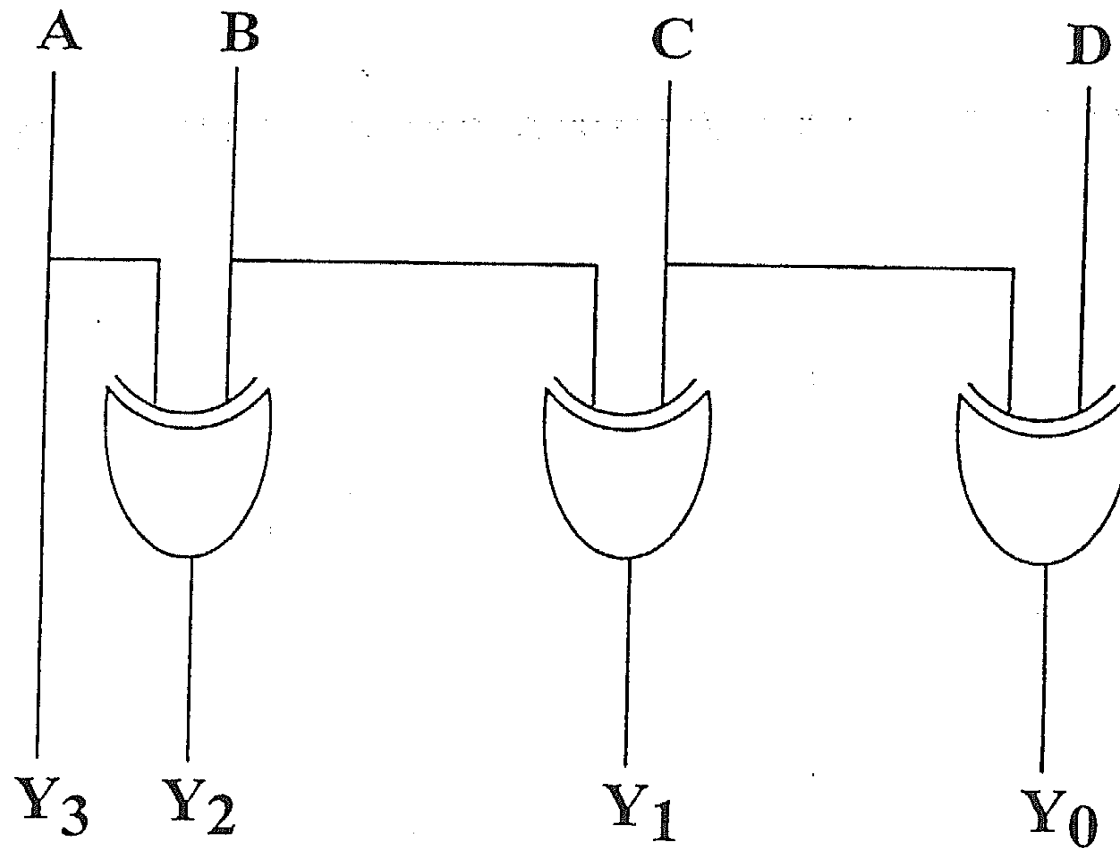
(d) For  $Y_0$ , the Karnaugh map can draw as follow.

	$\overline{C}\overline{D}$	$\overline{C}D$	$CD$	$C\overline{D}$
$\overline{A}\overline{B}$	0	1	0	1
$\overline{A}B$	0	1	0	1
$AB$	0	1	0	1
$A\overline{B}$	0	1	0	1

(b)

By minimization,  $Y_0 = C \oplus D$

**Binary Input**



**Gray Code Output**

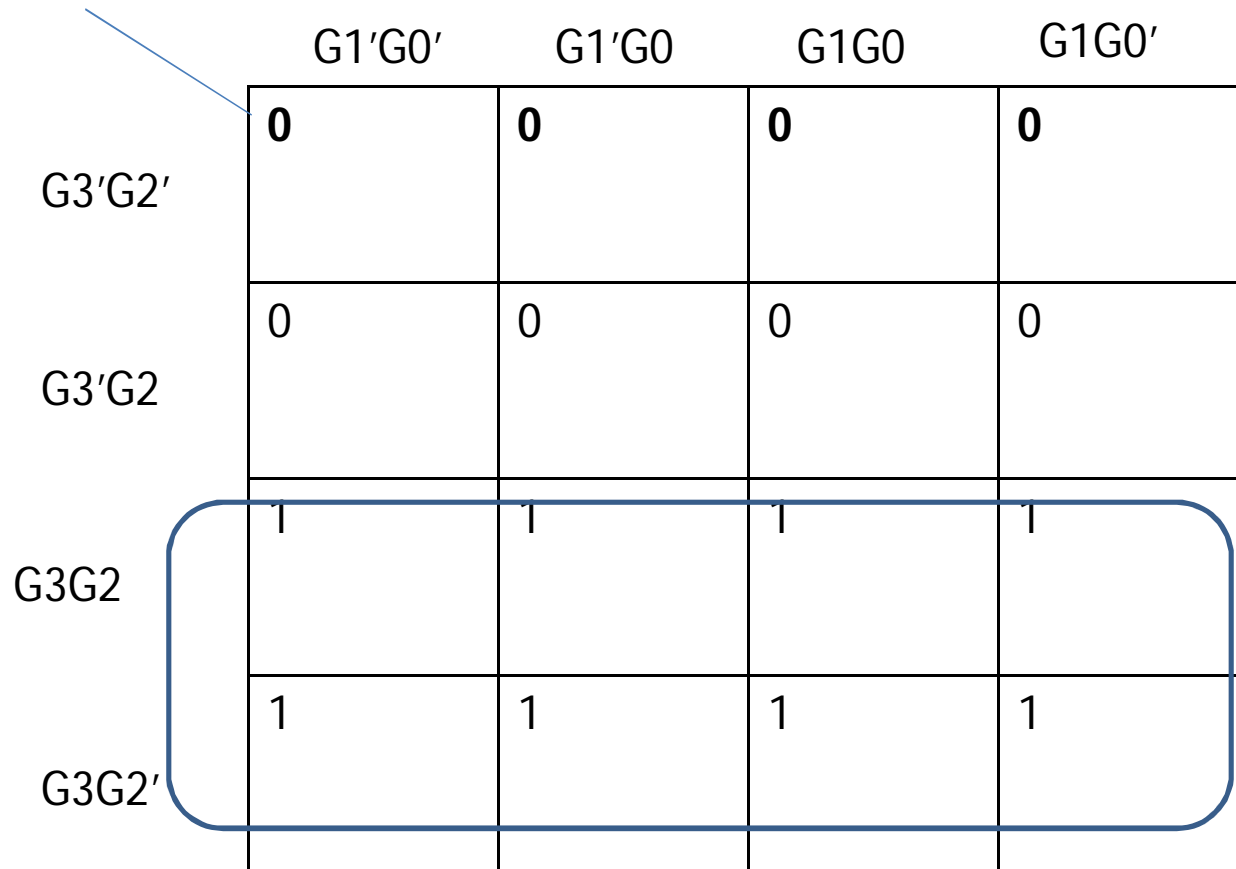
Figure 3.1 Binary-to-Gray Code

# Gray to Binary Converter

# Truth Table

Dec	G3	G2	G1	G0	B3	B2	B1	B0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
3	0	0	1	1	0	0	1	0
2	0	0	1	0	0	0	1	1
6	0	1	1	0	0	1	0	0
7	0	1	1	1	0	1	0	1
5	0	1	0	1	0	1	1	0
4	0	1	0	0	0	1	1	1
12	1	1	0	0	1	0	0	0
13	1	1	0	1	1	0	0	1
15	1	1	1	1	1	0	1	0
14	1	1	1	0	1	0	1	1
10	1	0	1	0	1	1	0	0
11	1	0	1	1	1	1	0	1
9	1	0	0	1	1	1	1	0
8	1	0	0	0	1	1	1	1

# K MAP For B3



A Karnaugh map for the function B3. The map is a 4x4 grid. The columns are labeled G1'G0', G1'G0, G1G0, and G1G0'. The rows are labeled G3'G2', G3'G2, G3G2, and G3G2'. The cells in the third row (G3G2) and the fourth row (G3G2') all contain the value 1. A blue loop is drawn around these four cells, indicating a group of four 1s. A blue line points from the title 'K MAP For B3' to the top-left cell of the map.

	$G1'G0'$	$G1'G0$	$G1G0$	$G1G0'$
$G3'G2'$	0	0	0	0
$G3'G2$	0	0	0	0
$G3G2$	1	1	1	1
$G3G2'$	1	1	1	1

$$B3 = G3$$

# K MAP For B2

	$G1'G0'$	$G1'G0$	$G1G0$	$G1G0'$
$G3'G2'$	0	0	0	0
$G3'G2$	1	1	1	1
$G3G2$	0	0	0	0
$G3G2'$	1	1	1	1

$$B1 = G3 \oplus G2 \oplus G1$$

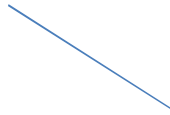


# K MAP For B1

	$G1'G0'$	$G1'G0$	$G1G0$	$G1G0'$
$G3'G2'$	0	0	1	1
$G3'G2$	1	1	0	0
$G3G2$	0	0	1	1
$G3G2'$	1	1	0	0

$$B1 = G3 \oplus G2 \oplus G1$$

# K MAP For B0



	$G1'G0'$	$G1'G0$	$G1G0$	$G1G0'$
$G3'G2'$	0	1	0	1
$G3'G2$	1	0	1	0
$G3G2$	0	1	0	1
$G3G2'$	1	0	1	0

$$B0 = G3 \oplus G2 \oplus G1 \oplus G0$$

