

| Dec | Binary | Gray ^① |
|-----|--------|-------------------|
| 0 | 000 | 000 |
| 1 | 001 | 001 |
| 2 | 010 | 011 |
| 3 | 011 | 010 |
| 4 | 100 | 110 |
| 5 | 101 | 111 |
| 6 | 110 | 101 |
| 7 | 111 | 100 |

Binary \rightarrow Gray

0 1 1
|

MSB (most significant number)

We compare 2 numbers
If same write 0
If different write 1

0 1 1

1) Re-write MSB

0

2. Compare 1st 2 bits.

If same \Rightarrow 0
If diff \Rightarrow 1

0 1

3) Compare 2nd 2 bits
Same \Rightarrow 0
Diff \Rightarrow 1

$\therefore G = 010$ ^②

Example 2: Convert 110 to a Gray code

Solution

1 1 0 \rightarrow 1 0 1
MSB

Example 3: Convert 101011 to a Gray code

Solution

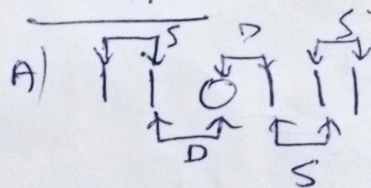
1 0 1 0 1 1 \rightarrow 1 1 1 1 1 0
MSB

Q) Convert the ff

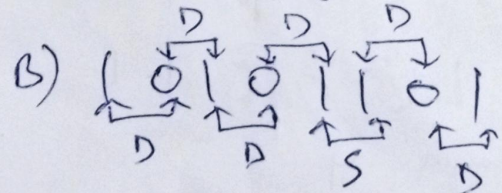
A) 110111

B) 10101101

Solution

A) 

1 0 1 1 0 0

B) 

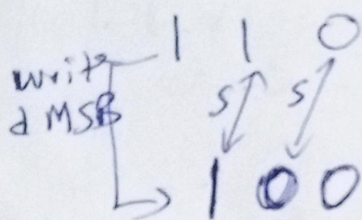
1 1 1 1 1 0 1 1

(3)

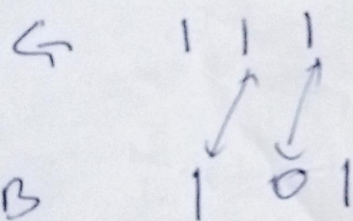
Gray \rightarrow Binary

Example: Let's say we have a Gray code of 110, find the Binary.

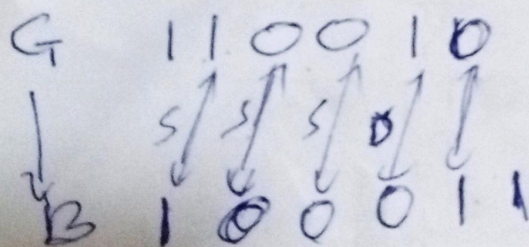
Solution



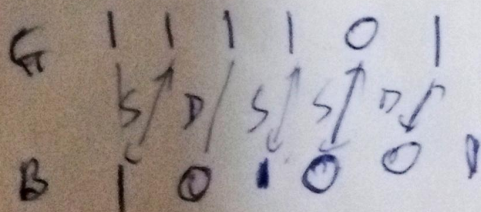
Example 2



Example 3:



Example 4



Example 5 (4)

G 10111011

NOTE

The Gray code as you go from one number to the next number. It only differs by 1 bit. In the Binary code it can differ by more than 1 bit. In successive values.

G. Code reduces switching operations as you go in to designing a circuit using it.

Example:

| Dec | Bin | Gray code |
|-----|-----|-----------|
| 5 | 101 | 111 |
| ↓ | ↓↓ | ↓ |
| 6 | 110 | 101 |

K' MAP (Karnaugh Map) ⑤

The results from Boolean is not always minimum. So it is used to minimize the circuits from a given logic.

→ Given a function ~

$$F = A + BC$$

Make a Truth Table of F.

| A | B | C | F |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

Canonical sum of product
Canonical SOP

$$F = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC$$

Can we reduce it?

1. Use Boolean Algebra.

$$\begin{aligned} F &= \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC \\ &= \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}(C + \bar{C}) + \bar{A}B(\bar{C} + C) + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC \\ &= \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B} + \bar{A}B + A\bar{B} + AB \\ &= \bar{A}\bar{B}\bar{C} + A \end{aligned}$$

Canonical SOP form

| P | q | r | f |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

$$f = P'qr + Pq'r + Pqr' + Pqr$$

Other ways.

$$f = m_3 + m_5 + m_6 + m_7$$

$$f = \sum m(3, 5, 6, 7)$$

Distributive Law:

$$\cancel{A + \bar{A}B} + A + \bar{A}B = (A + \bar{A})(A + B) = A + B$$

$$\begin{aligned} &= A + \bar{A}B\bar{C} \\ &= (A + \bar{A})(A + B\bar{C}) \\ &= A + B\bar{C} \end{aligned}$$

Now we arrange d cells such at the neighboring cell has only one variable changing.

| | | | | |
|-------|---------|---------|---------|---------|
| | BC = 00 | BC = 01 | BC = 11 | BC = 10 |
| A = 0 | m0 | m1 | m3 | m2 |
| A = 1 | m4 | m5 | m7 | m6 |

map rolling.

Rule of adjacency

NB:

The adjacent cells should have only 1 variable changing

K-map for 3 variables

| BC | | 00 | 01 | 11 | 10 |
|----|---|----|----|----|----|
| A | 0 | 0 | 0 | 0 | 1 |
| | 1 | 1 | 1 | 1 | 1 |

Group I: (0,1) (1,1) (0,0) (1,0)
Group II: (0,1) (1,1)

Next is pair if.

Pairing is in powers of 2.

1, 2, 4, 8, 16, ...

$$F = \underline{I} + \underline{II}$$

$$= A \cdot 1 \cdot 1 + B\bar{C}$$

$$F = A + B\bar{C}$$