Digital Electronics

Number representation

1. Binary
2. hexa octal number
3. Decimal number
4. hexa decimal

to Convert Binary to octal make a grouping of 3 bits and for hexa make it with 4 bits.

Radix Complement: →

(i) 9's Complement = Sub. the given no from 9's (in number 9 should be in same order as that no.)

Ex in (100)→ ie 3 bit it should be 999 for 4 bit = 9999

9's complement of 0987 = 9999
−0987
1012 → 9's comp.

(ii) 10's Complement

Sub. the given no. as in 9's Complement the only change is ones digit (i.e last digit ~~जगह बदल~~) must be sub. from 10.

or simply take 9's compliment and then add 1 to the no. you got. Ex 98 = 02

③ 1's Compliment

to find 1's Compliment just Convert 1's into zeroes and 0's into ones.

Ex.

110101 → 001010

(4) 2's compliment :→

to find 2's Compliment. write all zeroes same upto 1st 1 (including 1 also) from right side and then Convert 1 ⇄ 0. or add 1 to 1's compliment

Ex 1101100 ⇒ 0010100

Substraction with Compliment

1). using 10's Compliment :→

Step.1. take Compliment which is going to substract

Step.2 add it with other no.

Step.3. discard carry if any.

Step.4. if you do not get any carry then

put a -ve sign before the 10's Compliment of result you got.

Same for 9's Compliment also.

using 1's compliment :→

In 1's Compliment all steps are same the only change is, add the carry to the result if you got if no carry the then same process you did last time.

Signed binary number :→

0 → for +ve no. } → used at leftmost place
1 → for -ve no }

Binary Code →

we use these Codes because decoding becomes easy than binary system.

1. Weighted Code → Ex 439 (10^0 10^1 10^2) weight of → in binary (2^0, 2^1, 2^2 ...)
2. Non weighted Code Some wight code 8421, 84$\bar{2}\bar{1}$
3. Reflective Code 5421, 2421
4. Alphanumeric Code 54$\bar{1}\bar{1}$, 7421
5. Error detecting and correcting codes.

Note after 5 we give (more) high weightage से from left side to write any no. in Binary System.

BCD (Binary Code decimal)

396 → 0011 1001 0110

Note after 9 we can not rep. a digit in BCD.

BCD addition

if Sum > 9 then add 0110

Ex

6	= 0110		1101
+7	0111		0110
	1101 → (13) X (not valid) then.		0011
		0001	(3)
		(1)	

Self complementing code

Excess-3

↓ unweighted code

add 0011 to each Corresponding BCD

Ex
  0001 0011 → in BCD
+ 0011 0011
  0100 0100 → in Excess 3 code

ay Code / Cyclic Code / unit distance Code

- Unweight Code
- not an arithmetic ode

0 → 0000	6 → 0101
1 → 0001	7 → 0100
2 → 0011	8 → 1100
3 → 0010	9 → 1101
4 → 0110	10 → 1111
5 → 0111	11 → 1110

	00	01	11	10
00	0	1	2	3
01	7	6	5	4
11	8	9	10	11
10	15	14	13	12

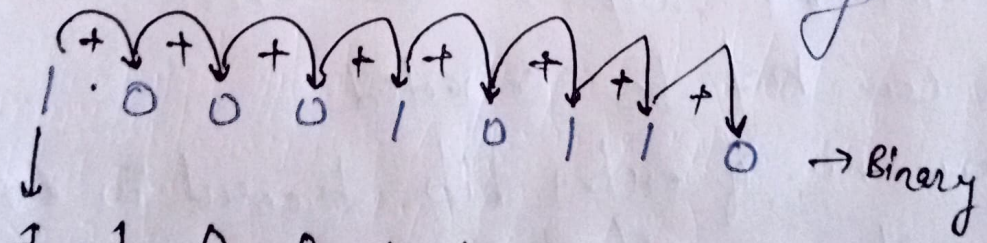
12 → 1010
13 → 1011
14 → 1001
15 → 1000

Conversion of Binary to Gray Code

Step 1. write as it is MSD

Step 2 → then add MSD to next digit (bit) and next bit to next bit and so on record only sum discard the carry.

Ex

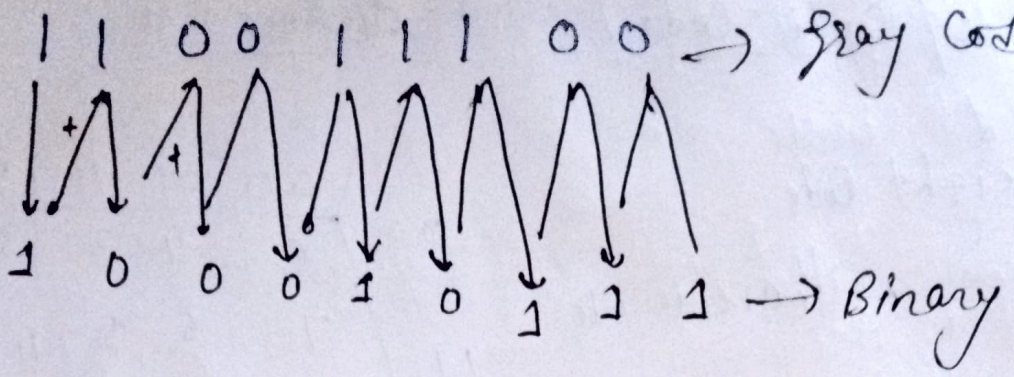


1 1 0 0 1 1 1 0 1 → gray Code

Gray Code to Binary →

write MSD same then add each result with next digit (bit) of gray code (as result is MSD).

Ex



Alphanumeric Codes: →

ASCII total alphanumeric group > 36
it include, ten decimal digit, 26 letter, and certain special character such as $.
- it must be coded with a minimum 6 bits.

Ex A = 1000001, B = 1000010

Error detecting Code →

- code should be with even or odd 1's.
- one extra bit helps to detect error.
- error can not be corrected.
- if even no. of extra bits present in code error then it may not give error.

Application of Code

1). BCD used in Digit clocks digit thermometers, digit meters. and other devices with seven segment display

Gray codes only one bit changes in next number are used in shaft position encoders.

- Ex-3 used in old computer and self complementary.

logic gates: →

1. Basic logic gate: → AND, OR, NOT
2. universal gate → NAND, NOR
3) other gates → Buffer, XOR (Exclusive OR), Exclusive NOR

$x\bar{y} + \bar{x}y$ → $xy + \bar{x}\bar{y}$

$x \oplus y$ → $x \odot y$

literal: → it is variable or the compliment of variable.

← : Boolean Algebra: →

1. Commutative law $A+B = B+A$, $AB = BA$
2. Associative law $A+(B+C) = (A+B)+C$, $A(BC) = (AB)C$
3) Distributive law $A(B+C) = AB+AC$

$A+0 = A$ $\quad A+\bar{A} = 1$ $\quad A+AB = A$

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

$A \cdot 0 = 0$ $\quad A \cdot \bar{A} = 0$ $\quad (A+B)(A+C) = A+BC$

$A+A = A$ $\quad \bar{\bar{A}} = A$

Demorgan's Law :→

$$\overline{xy} = \bar{x} + \bar{y} \quad , \quad \overline{x+y} = \bar{x} \cdot \bar{y}$$

Min term :→

write ~~A for 1~~ $1 = A$ Symbol of sop = Σ

$0 = \bar{A}$

we find here sop (sum of product)

Ex $x\bar{y}z + xyz + \bar{x}.\bar{y}z$

NOTE we write sum of all those term in which we are getting 1 as output.

Max term :→

write $1 = \bar{A}$ Symbol of pos $\Rightarrow \Pi$

$0 = A$

Here we calculate pos (product of sum).

Ex $(x+y)(\bar{x}+y)$

Note→ we write product of sum of all those there in which are getting 0 as output

NOTE Sop and pos are compliment to each other.

K Map

AB \ CD	00	01	11	10
00				
01				
11				
10				

- for minterm make grouping of 1's.
- In grouping we can include only 2^n terms.
- try to include more terms.
- then we write SOP

* for Max terms grouping should include zeroes.
and other thing are same.
* Here we will write POS.

- **Don't Care Condition:→**

There are some condition at which our output do not depend, there we can use 1 or 0 acc. our requirement to solve logic and it will not affect the output.