Chapter→1
Data Representation
lesser imp chapter as exam point of view.

1.Note:

⊛ Digital computers process data that is in discrete form whereas analog computers process data in continious form. But, Hybrid computers can process data in both discrete as well as continious form.

In digital computers

ON ⟶ represented by 1.

OFF ⟶ represented by 0.

⊛ Amplitude(A) → Maximum displacement that the waveform of an electrical signal can attain.

Frequency(f) → The number of cycles made by signal in 1 second. ~~It~~ is measured in hertz. 1hertz = 1 cycle/second.

Periodic time(T) → Time taken by signal to complete one cycle.

$$\boxed{T = 1/f}$$

⊛ The process of converting a digital signal to an analog signal is known as modulation. Similarly the process of converting back analog signal to digital is known as demodulation.

⊛ Bits ⟶ 0 or 1

Byte → Group of 8 bits used to represent a character.

A nibble → Half byte, Usually a grouping of 4 bytes.

Word ⟶ Two or more bits make a word. Word length is the measure of the number of bits in each word. A word can have length of 16 bits, 32 bits, 64 bits etc.

⊛ Number System → A number system is a set of symbols used to represent values derived from common base or radix.

Note : Number System and their conversion is easy do it yourself.

2. Complement:- Complement is a method or technique used to calculate arithmetic operations like subtraction. Generally complements are of two types:-

i> r's complement

ii> $r-1$'s Complement.

In r's complement we use the formula $r^n - N$.

where, N = Given number

r = base/radix

n = no. of digits in given number.

In $r-1$'s complement we use the formula $r^n - N - 1$.

OR r's $- 1$.

Example:- Find the r's and $r-1$'s complement of $(512)_{10}$

Soln: We have,

$N = 512$
$r = 10$
$n = 3$

r's complement $= r^n - N$
$= 10^3 - 512$
$= 488$

& $r-1$'s complement $= r$'s $- 1$
$= 488 - 1$
$= 487$

Important table related to compliment:

Base/Radix	r's complement	$r-1$'s complement.
$r=2$ for binary numbers	2's complement	1's complement
$r=10$ for decimal numbers	10's complement	9's complement
$r=8$ for octal numbers	8's complement	7's complement
$r=16$ for hexadecimal numbers	16's complement	15's complement.

3. Representation of Signed Binary Numbers.

There are three common ways of representing a signed binary number.

i) Prefixing an extra sign bit to a binary number:- It is the process of representing numbers by adding sign (+, –) infront of a given number.

The value of + sign in binary is zero and – sign is one.

Example

(a) +11 ⇒

+	1011 ← binary of 11
Extra Sign	Actual Binary number
0	1011

⇒ 01011

(b) –11 ⇒

–	1011
Extra Sign	Actual Binary number.
1	1011

⇒ 11011

ii) Using ones complement:- In ones complement ($1^{'s}$) representation we can calculate $1^{'s}$ complement of given binary number by replacing 1 with 0 and 0 with 1.

iii) Using twos complement:- In twoes complement representation we can calculate $2^{'s}$ complement by adding one in $1^{'s}$ complement.

Example:- The $1^{'s}$ complement of $(11)_{10}$ is (0100) and the $2^{'s}$ complement is

$$\begin{array}{r} 0100 \\ +\ 1 \\ \hline 0101 \end{array}$$

4. Alphanumeric Representation:-

In alphanumeric representation we can assign a numeric value to alphabets using ASCII code.

For e.g. 65 is represented by A and binary of 65 is 1000001.

5. Binary Coded Decimal (BCD):

In Binary coded decimal we can use decimal numbers in binary digit upto 9. After 9 we can seperate a decimal number and compute a binary number using 0 to 9.

For example:- 12 $\longrightarrow$ 1100 (In Binary).

In BCD the value of 12 is

1 $\rightarrow$ 0001

2 $\rightarrow$ 0010

On combining 12 = 00010010

$\therefore$ 12 = 10010.

6. Fixed Point Representation:-

In computer architecture fixed point representation is used to represent binary number by using following methods:-

The smallest binary number or decimal number is 0000.0001 & highest is 9999.9999 in decimal and 1111.1111 in binary.

Sign field	Integer field	Fractional field

Example: 1 001 . 1010

sign field, Integer field, fraction field

i.e, −1.10 (Since -ve sign is represented by 1 in sign field / Extra sign)

7. Floating Point Representation:-

The representation of floating point is

Sign field	exponent	Mantisa

Example:- 1354.537

$N = m * r^e$

$= 0.1354537 \times 10^4$.

8. Overflow Detection:-

While adding two n-bit binary numbers, the result maybe a number with n+1 bit this situation is called overflow.

Example:- 9 in binary is 1001 (n-bit)
9 in binary is 1001 (n-bit)
+
10010 (n+1 bit)
overflow

If there is no end carry then, no overflow.

Example
6 in binary is 0110
9 in binary is 1001
+
1111 (No-overflow)

9. Gray Code:

· Gray code is also known as reflected binary code (RBC). We call it the name Frank Gray since, it was named after Frank Gray and was used as solution guide for tower of Hanoi problem.

Conversion from BCD to Gray

Steps
i) Copy the MSB as it is.
ii) Add the most significant bit (MSB) to next bit, write the sum and neglect the carry.
iii) Repeat the process i.e, step no. 2

For Example

Let 1011 be a 4-bit binary number then we convert it into gray code as follows:-

1 0 1 1
+ 1 0 1
1 1 1 0

carry neglected

∴ Gray code of 1011 is 1110.

10. Excess-3 Code:-

Excess 3 code is also known as (XS-3) code. We can calculate excess-3 code by adding binary data with 3 i.e, 0011 in binary.

For example:- 1011
+ 0011
1110 is the required excess-3 code of 1011.

⊛. Extended Binary Coded Decimal Interchange code (EBCDIC):

Extended Binary Coded Decimal Interchange code (EBCDIC) is an 8-bit character-coding scheme used primarily on IBM computers. A total of 2^8 (i.e, 256) characters can be coded using this scheme. For example, the symbolic representation of letter A using Extended Binary Coded Decimal Interchange code is 11000001_2.

⊛. American Standard Code for Information Interchange (ASCII):

It is a 7-bit code, which means that only 2^7 (i.e, 128) characters can be represented. However, manufactures have added an eight bit to this coding scheme, which can now provide for 256 characters. The symbolic representation of letter A using this scheme is 1000001_2. This codes represent text in computers, communication equipment and other devices that use text.

⊛ Error Detection Code:-

An error detection code is a binary code that detects digital errors during data transmission. The detected errors cannot be corrected but their presence is indicated. The most common error detection code used is the parity bit.

Parity → Parity is an extra bit added with original message to detect error during the data transmission. This technique is known as error detection technique.

ⓐ. Even parity → In even parity we count no. of 1's in binary digit and if the count is even then we add 0 otherwise 1.

For example

Binary Digit	Parity
101101	0
110111	1

ⓑ Odd parity → In odd parity we count no. of 1's and if the count is odd then we add 0 otherwise 1.

For example:

Binary Digit	Parity
10110101	0
11110000	1

Parity Generator

Message(xyz)	Parity(odd)	Parity (even)
000	1	0
001	0	1
010	0	1
011	1	0
100	0	1
101	1	0
110	1	0
111	0	1

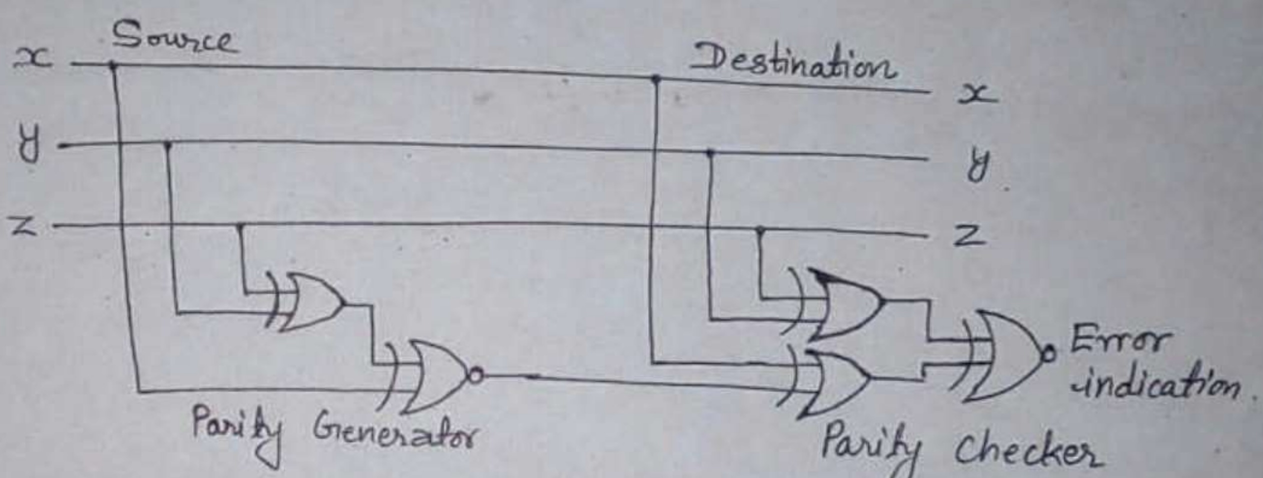

fig. Error detection with odd parity bit.

 # Computer Architecture:- According to Hayes computer architecture is defined as "the study of the structure, behaviour, and design of computers" is called computer architecture. Instruction set, data representation, I/O mechanisms and addressing techniques are its attributes.

 # Computer Organization:- Organization refers to operational units and their interconnections that realize the architecture specifications. Control signals, peripheral interface and memory technology are it's attributes.

⊛ Weighted codes (8421 code and 2421 code):

Binary codes can be classified into two types, weighted and unweighted code. If the code has positional weights, then it is said to be weighted code. Otherwise it is an unweighted code.

8421 code and 2421 code for decimal digits are as in the following table:

Decimal digit	8421 code	2421 code
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	1011
6	0110	1100
7	0111	1101
8	1000	1110
9	1001	1111

8421 code:

→ The weights of this code are 8,4,2 and 1.
→ This code has all positive weights
→ This code is also called as natural BCD code.

2421 code:

→ The weights of this code are 2,4,2 and 1.
→ This code also has all positive weights.
→ It is an unnatural BCD code.
→ It is a self-complementing code.