Binary to decimal → 8421 approach

$$\begin{array}{cccccc} 32 & 16 & 8 & 4 & 2 & 1 \\ \end{array}$$
$$(1\ 1\ 0\ 0\ 0\ 1)_2$$

$32 + 16 + 1 = (59)_{10}$

Binary Number

$$1\ 0\ 1\ 1\ 0\ 1$$

MSB, LSB

LSB → 1 (odd)

LSB → 0 (even)

Ex

$$\begin{array}{cccccc} 32 & 16 & 8 & 4 & 2 & 1 \\ \end{array}$$
$$(1\ 0\ 1\ 0\ 1\ 1)_2$$

$32 + 8 + 2 + 1$

$= (43)_2$

$$\begin{array}{cccccc} 32 & 16 & 8 & 4 & 2 & 1 \\ \end{array}$$
$$(1\ 0\ 1\ 0\ 1\ \underline{\underline{0}})_2 \rightarrow \text{even}$$

$32 + 8 + 2$

$= (42)_{10}$

Error detecting & Correcting Code

↳ Malfunction, which disturb the actual flow of Execution

↳ Change in data

↳ Change / manipulation in H/W

☆ Whenever a message is transmitted, it may get scrambled by noise or data gets Corrupted.

☆ To avoid these kind of things we use error detection & Correction code

Ex 'Parity Check', 'Hamming Code', 'CRC (cyclic Redundancy Check)'

Parity Check:

It ensures accurate data transmission between two nodes during Communication.

→ It can detect 1 bit error.

↳ even Parity : Total no. of 1s in a bitstream are even

odd Parity : Total no. of 1s in a bitstream are odd

$\underbrace{0\ 1\ 0\ 1}$

no. of 1s = 2

↓

even number (even Parity)

$\underbrace{0\ 1\ 1\ 1}$

no. of 1s = 3

↓

odd number (odd Parity)

Parity Check

Suppose we've 8 bit number

MSB →

	d_6	d_5	d_4	d_3	d_2	d_1	d_0

Reserve → Parity Bit

$d_6 \ldots d_0$: 7 bits ← data

Ex

$\underline{0}$ $\underbrace{1011010}_{\text{data}}$

0 → even parity

no. of 1's = 4 → even

$\underline{1}$ $\underbrace{1\ 0\ 1\ 1\ 0\ 1\ 0}_{\text{data}}$

data → no. of 1^s = 4 (even)

no. of 1^s = 5 (odd)

↳ odd parity

Sender

0 1 0 1 1 0 1 0

no. of 1s = 4 (even)
parity

Receiver

0 1 0 1 1 0 1 1

no. of 1s = 5 (odd)

↓

Error

0 1 0 1 1 0 0 1 } 2 bits Change

no. of 1s = 4 (even)

⇓

[Limitation]

odd bits

Even parity → 1, 3, 5, 7, 9....
(detectable)

→ [2n+1] bits ← detectable
↖ odd

even Parity → no. of Bit switch ⇒ 2n (even)
↳ undetectable

Data Representation :

↳ Inside memory & decimal Representation

- Unsigned Magnitude
- Signed Magnitude } important
- 1's Complement
- 2's Complement } Concept
- 9's Complement
- fixed & floating point representation (Important)

(A) Unsigned Magnitude :

↳ Unsigned number does not has any sign

↳ +ve number

→ Range 0 to ∞

→ $0 \leq n \leq \infty$

1 bit $= 0, 1 \rightarrow 2^0$

2 bits $= 4 \rightarrow 2^2$

3 bits $= 8 \rightarrow 2^3$

4 bits $= 16 \rightarrow 2^4$

5 bits $= 32 \rightarrow 2^5$

n bits $= 2^n$

If we have 'n' bits, then we can represent 2^n numbers in unsigned Mag. Representation

Range $[0$ to $2^n - 1]$

* No negative numbers.

If we have 4 bits,

Binary : Range 0000 to 1111

Decimal : Range 0 to 15

0 to $2^n - 1$

(B) Signed Magnitude :

→ We can represent both −ve & +ve number.

→ MSB is reserved for sign bit .

- ↳ sign bit = 0 then positive number
- ↳ sign bit = 1 then negative number

Suppose we have 4 bits (Signed magnitude form)

MSB

⇓ reserve sign bit

- ↳ 0
- ↳ 1

data = 3bits

बचा = 3 bit

Range
- → Sign bit = 0 : $2^3 = 8$ numbers } +ve number
- → Sign bit = 1 : $2^3 = 8$ numbers } −ve number

Range = −7 to +7

$$\underbrace{-7\ -6\ -5\ -4\ -3\ -2\ -1}_{-(2^{n-1}-1)}\ \boxed{-0\ \ +0}\ \underbrace{+1\ +2\ +3\ +4\ +5\ +6\ +7}_{(2^{n-1}-1)}$$

$\boxed{-0\ \ +0}$ ↓ 2^s Comp

5 bits

$n=5$

sign | data

sign $\downarrow$ $+ve \rightarrow 0$

data: 4 bits $n=4$

$\downarrow$

$2^n = 2^4 = 16$ Comb

Range $= [0 \text{ to } 2^n - 1]$

sign $\downarrow$ 1 $\downarrow$ $-ve$ number

data: 4 bits $\rightarrow$

$\Downarrow$

$2^n = 2^4$ Comb. $= 16$

$\downarrow$

Range $[0 \text{ to } 2^n - 1]$

$\Rightarrow -[0 \text{ to } 2^n - 1]$

$= [-0 \text{ to } -2^n - 1]$

total Range $-2^n - 1$ to $2^n - 1$

$$\left[\underbrace{-2^{n-1} - 1}_{-ve \text{ number}} \text{ to } \underbrace{2^{n-1} - 1}_{+ve}\right]$$

$\rightarrow$ n bits $\rightarrow$ 1 bit sign $\underbrace{(n-1) \text{ bits}}$ data

$\rightarrow$ Range $\rightarrow$ $[2^{(n-1)} - 1]$ $\rightarrow$ $-[2^{(n-1)} - 1]$ to $[2^{(n-1)} - 1]$

Representation	Range
Unsigned Magnitude	0 to $2^n - 1$
Signed Magnitude	$-(2^{n-1}-1)$ to $(2^{n-1}-1)$
Signed 1's Complement	$-(2^{n-1}-1)$ to $(2^{n-1}-1)$
Signed 2's Complement	$-(2^{n-1})$ to $(2^{n-1}-1)$

1 bit reserve