Number System

Decimal.

weight →	10^3	10^2	10^1	10^0	10^{-1}	10^{-2}
Coefficient →	a_3	a_2	a_1	a_0	a_{-1}	a_{-2}

Binary to Decimal Conversion.

$(1011.11)_2 \Rightarrow (?)_{10}$

$= [(1\times2^3)+(0\times2^2)+(1\times2^1)+(1\times2^0)+(1\times2^{-1})+(1\times2^{-2})]_{10}$

$= [8+0+2+1+0.5+0.25]_{10}$

$= (11.75)_{10}$

\# $(721.4)_8 = (?)_{10}$

$= [(7\times8^2)+(2\times8^1)+(1\times8^0)+(4\times8^{-1})]$

$= 448+16+1+0.5$

☆ $= (465.05)_{10}$

\# Find the base value which can satisfy the following 2 eq.

i) $2+3=5$ ii) $2\times4=10$

$2\times x^0+3\times x^0=5\times x^0$ $(2\times x^0)\times(4\times x^0)=(1\times x^1)+(0\times x^0)$

$2+3=5$ $2\times4=8$ ∴ Base 8

∴ $x>5$ $x=8$

Find number of sol. of x & y which can satisfy the eq.

(1) $(43)_8 = (x0)_y$

$4\times8^1 + 3\times8^0 = x\cdot y^1 + 0\times y^0$

$32 + 3 \Rightarrow 35 = xy$

when x & y will result 35.

$35 :=$

x	y
35	1
1	35
5	7
7	5

But $y > x$

$\therefore 35 > 1$

$7 > 5$

$\therefore$ 2 situations.

(2) $(123)_5 = (x8)_y$

$1\times5^2 + 2\times5^1 + 3\times5^0 = x\times y^1 + 8\times y^0$

$25 + 10 + 3 = xy + 8$

$30 = xy$

x	y
30	1
1	30
2	15
15	2
3	10
10	3
6	5
5	6

$y > x$

$\therefore 30 > 1$

$15 > 2$

$5 > 5 \longrightarrow$ Cant possibl. as $8 > 6$

$10 > 3$

~~7 situation~~

$\therefore$ 3 condition.

The result of addition operation 34+43 performed on minimum base is stored in an 8-bit register. The content of register will be.

$(34)_5$
$(43)_5$
———
77

$(34)_5 + (43)_5$

$[(3\times5^1) + (4\times5^0) + (4\times5^1 + 3\times5^0)]_{10}$

$= (19+23)_{10}$

$= (42)_{10}$

$= (101010)_2$

Ans.

$\therefore \underbrace{00101010}_{8bit}$

2	42	
2	21	0
2	10	1
2	5	0
2	2	1
	1	0

Decimal to other base.

$\underbrace{a_3 a_2 a_1 a_0}.\underbrace{a_{-1} a_{-2} a_{-3}}$

$0.a_{-1}a_{-2}a_{-3} \times r = x_0.x_{-1}x_{-2}$

$0.x_{-1}x_{-2} \times r = x_1.x_{-3}x_{-4}$

$0.x_{-3}x_{-4} \times r = x_2.x_{-5}x_{-6}$

$(19.75)_{10} \to (?)_2$

2	19	
2	9	1
2	4	1
2	2	0
	1	0

$0.75 \times 2 = 1.50$

$0.50 \times 2 = 1.00$

$\therefore (10011.11)_2$

$(23)_4 \to (?)_2$

2 → 10, 3 → 11 → $(1011)_2$

Magnitude Representation.

	~~Complete~~ Unsigned	Signed	Complement: r-1's (1's)	r's complement (2's)
+5	101	0101	0101	0101
-5	×	1101	1010	+1 → 1011

If '+' no need to complement all same.

$r = 10$

- r-1: 9's
- r: 10's.

732 find 9's complement.

$$\begin{array}{r} 999 \\ -\ 732 \\ \hline 267 \end{array}$$

Let 0110 is in 1's complement → $(?)_{10}$

$(+6)_{10}$

1 0110 → 1001

$(-9)_{10}$

(0110) in 2's complement

$(+6)_{10}$

(11010) in 2's complement

$-0110 \to (-6)_{10}$

Signed R.

4 bit $2^{n-1} - 1$ to $2^{n-1} - 1$

0000 +0   1000 −0

0111 +7   1111 −7

1's Complement $2^{n-1} - 1$ to $2^{n-1} - 1$

0000 +0   1000 → 1111 → −7

0111 +7   1111 → 1000 → −0

2's Complement 2^{n-1} to $2^{n-1} - 1$

0000 → +0   1000 → −8

0111 → +7   1111 → 1001 → −1

For representing ~~different~~ 1024 values we need 2^{10}, 10 bits
0 — 1023

But for getting 1024 we need 2^{11}, 11 bits

Self Complementing Code, weights must sum up to 9.

2	4	2	1	= 9
0	0	0	0	
0	0	0	1	

Floating Point Representation.

Provides a large range of numbers as compared to fixed point representation.

Sign	Exponent	Mantissa

Original exponent (e) + bias ⇒ Stored exponent (E)

Biased Exponent → Arithmetic operations on floating point numbers become easy.

Mantissa → Normalized

Bias = 2^{k-1}

101.11

Explicit normalization ⇒ 0.10111×2^3 (should be 1)

∴ e = 3

E = 3 + bias.

M = No. after point = 10111

Implicit Normalization ⇒ 1.0111×2^2 (Should be 1)

∴ e = 2

E = 2 + bias

M = No. of after point = 0111

Value (explicit) $= (-1)^s * 0.M * 2^{E-bias}$

Value (implicit) $= (-1)^s * 1.M * 2^{E-bias}$

Original exponent (e)	Stored exponent (E)
−8	0
−7	1
⋮	⋮
7	15

Excess-8 code → bias

\# A certain well known computer family represents the exponents of its floating-point numbers as 'excess-64' integer i.e a typical exponent $e_6 e_5 \ldots e_0$ represent the no.

Bias

S	$e_6 e_5 \ldots e_0$

i) $e = -64 + \sum_{i=0}^{6} 2^i e_i$ ✓

ii) $e = -64 + \sum_{i=0}^{6} 2e_i$

iii) $e = 64 - \sum_{i=0}^{6} 2^i e_i$

iv) $e = 64 - \sum_{i=0}^{6} 2e_i$

7 bit

$x = (e_6 * 2^6 + e_5 * 2^5 \ldots e_0 * 2^0)$

$e = x - 64$

−Bias.

Consider a 16 bit register used to store floating point no. The mantissa is explicitly normalized signed fraction number. Exponent is represented in excess-32 (bias) form. What is the 16 bit value for $+(11.5)_{10}$ in register.

11.5

$$\begin{array}{r|l} 2 & 11 \\ 2 & 5 \quad 1 \\ 2 & 2 \quad 1 \\ & 1 \quad 0 \end{array}$$

$0.5 \times 2 = \underline{1}.0$

original no. = 1011.1

explicit normalize → 0.10111×2^4

$\therefore$ M = 10111

S = 0

E = 4 + 32 = 36 (bias) → 6 bit

$$\begin{array}{r|l} 2 & 36 \\ 2 & 18 \quad 0 \\ 2 & 9 \quad 0 \\ 2 & 4 \quad 1 \\ 2 & 2 \quad 0 \\ & 1 \quad 0 \end{array}$$

0 100100 101110000

100100

Hexa decimal representation:

$\underbrace{0\,100}\underbrace{100\,1}\underbrace{011\,1}\underbrace{000\,0}$

$(4 \quad 9 \quad 7 \quad 0)_{16}$ = 0x4970 = 4970H

What is 4 digit hexa-decimal rep. of $+(0.000101)_2$ in prev q. with explicit normalization.

$0.000101 \rightarrow 0.101 \times 2^{-3}$ ⇒ e = −3

$\therefore$ E = −3 + 32 = 29.

M = 1010....

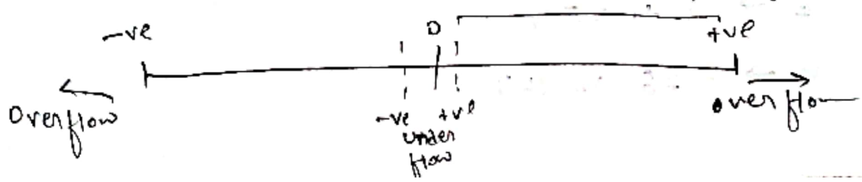
$$\begin{array}{r|l} 2 & 29 \\ 2 & 14 \quad 1 \\ 2 & 7 \quad 0 \\ 2 & 3 \quad 1 \\ & 1 \quad 1 \end{array}$$

$\underbrace{0\,011}\underbrace{101\,1}\underbrace{010\,0}\underbrace{000\,0}$

$(3 \quad B \quad 4 \quad 0)_{16}$

More bits in Mantissa → More precision or accuracy.

" " in exponent → Range of no. will be large



Disadvantage of Conventional Representation.

- 0 can't be stored.
- Suffer from underflow.

IEEE-754

Single Precision

S	E	M
1	8	23

bias = 127 $(2^{k-1}-1)$

Double Precision

S	E	M
1	11	52

bias = 1023 $(2^{k-1}-1)$

If E = 000---0
or
E = 111---1 } Special Case.

Always implicit Normalization.
(E ≠ 0000---00
E ≠ 1111---11

S	E	M	Number
0	00---0	0000---0	+0
1	00---0	0000---0	−0
0	111---1	0000---0	$+\infty$
1	111---1	000---0	$-\infty$
0 or 1	111---1	M ≠ 0---0	NAN (Not a Number)
0 or 1	000---0	M ≠ 0---0	Denormalized no. ↳ The no. which can not be normalized
0 or 1	E ≠ 000---0 or E ≠ 111---1	xxxxx	Implicitly normalized value.

bias = 127

for a normalized no. E => Min $(00000001)_2 \rightarrow (1)_{10}$ max $(11111110)_2 \rightarrow (254)_{10}$

range of E → 1 to 254

range of e → −126 to 127

Denormalized No. → for Underflow

→ very very small no. which can't be implicitly normalized.

Ex → 1.01×10^{-127} ← After implicitly normalization.
out of range. $-127+127 \to 0$ ↳ Not allowed

↓

0.101×10^{-126} ← Denormalized

0	00000000	101 . . . . . .

0.00011×10^{-126}

0	00000000	00011 . . .

$$\text{Value (Denormalized)} = (-1)^s \times 0.M \times (2)^{-126}$$
(−1022 ← double precision)

$$\text{Value (implicit)} = (-1)^s \times 1.M \times (2)^{E-bias}$$

$\underbrace{0}\,\underbrace{10000011}\,1110 - - - 0$

S = '+'   E = 10000011 → 131
(128 64 32 16 8 4 2 1)

$\therefore e = 131 - 127 = 4.$

$\therefore + (1.1100) \times 2^4$

$+ (11100) = +28$
(16 8 4 2 1)

0 $\underbrace{00000000}_{\text{denormalized}}$ 1100- - -

Direct −126 in Denormalized case.

$\therefore + (0.1100---) \times 2^{-126}$

$+ (11.0) \times 2^{-2} \times 2^{-126}$

$+ 3 \times 2^{-128}$

minimum normalized $= 2^{-126}$ → '+'
$= -2^{-126}$ → '−'

Max. '+' in Single precision

E = e+b
e = E−b.

$\underbrace{0}_{1}\ \underbrace{11111110}_{8}\ \underbrace{111- - - 1}_{23}$

↓

254

$+ (1.111- - - 1) \times 2^{254-127}$

$+ (\overbrace{111- - -1}^{24}.0) \times 2^{-23} \times 2^{127}$ ← in '−'

$+ (2^{24}-1) \times 2^{104} \rightarrow -(2^{24}-1) \times 2^{104}$