Computer Architecture (NTB)

Chapter - 03 (part - 2)

Floating Point addition: (same for subtraction)

35.23142 + 0.00053

step 1: Decimal to Binary and Normalize

a = 35.23142

= 100011.00111011111 (Binary)

= $1.00011\ 00111\ 01111 \times 2^5$ (Normalize

b = 0.00053

= 0.00000 00000 10001011 (Binary)

= 1.0001011×2^{-11} (Normalized)

Step-2: Match the lower exponent with the higher exponent.

$$a = 1.00011\ 00\ 111\ 01111 \times 2^{5}$$

$$b = 1.0001\ 011 \times 2^{-11}$$

$$= 0.00000\ 00000\ 00000\ 10001011 \times 2^{5}$$

Step-3: Add

$$a+b = (1.00011\ 00\ 111\ 01111 +$$

$$0.00000\ 00000\ 00000\ 10001011) \times 2^{5}$$

$$= 1.00011\ 00\ 1110\ 11111\ 0001011 \times 2^{5}$$

$$= 100011.\ 00\ 111\ 0\ 11111\ 0001011$$

$$= 35.2342224121 \text{ (Decimal)}$$

Floating point Multiplication:

$5.234 \times (-0.003)$

Step-01: Decimal to Binary and Normalize

$a = 5.234$

$= 101.00111011111$

1.01001110111×2^2

$b = 0.003 = 0.0000000011000100101$

$1.1000100101 \times 2^{-9}$

Step-02: Do Multiplication

$a \times b = (1.01001110111 \times 2^2) \times (1.1000100101 \times 2^{-9})$

$= (1.01001110111 \times 1.1000100101) 2^{2-9}$

$= 10.000000101 \times 2^{-7}$

$= 0.00000\ 10\ 000000101$

$= 0.\ 01570\ 12939$

Ans: $-0.01570\ 12939$

Overflow/ Underflow Concept:

Overflow/Underflow is detected by the range of the exponent field of floating point representation.

Suppose n bits are allocated for exponent field.

∴ range for bias exponent: 0 to $(2^n - 1)$

as 0 and $(2^n - 1)$ are reserved

∴ actual range: 1 to $(2^n - 1) - 1$

↳ 1 to $(2^n - 2)$

1 → a (lower range)

$(2^n - 2)$ → b (upper range)

Suppose, P → a number to check

if $(a <= P <= b)$ → No overflow/underflow

if $(P < a)$ → Under flow

if $(P > b)$ → overflow