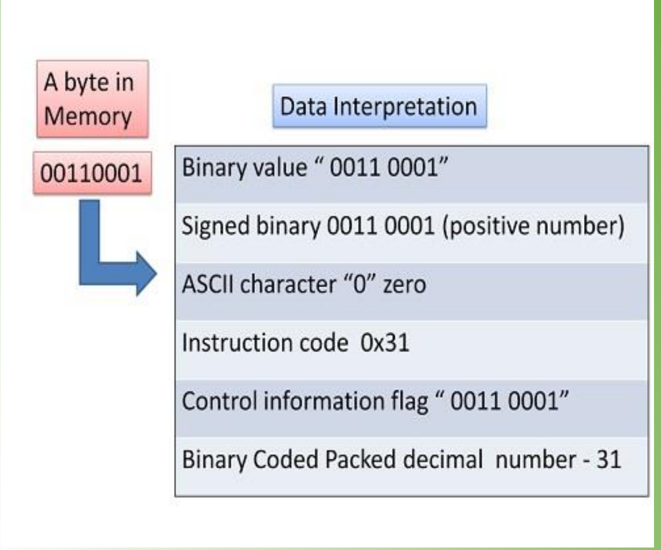
Data Representation and Computer Arithmetic.



Intro:

- Data representation and computer arithmetic are fundamental concepts in computer science that involve how data is stored, processed, and manipulated within a computer system.

- Computer arithmetic deals with the methods and algorithms used to perform mathematical operations like addition, subtraction, multiplication, and division on these binary-encoded data types.

Exact and Approximate numbers

1. Exact numbers are values known with complete precision, like integers or defined fractions. For example, the number of students in a class (e.g., 25) or the fraction 1/2 are exact.
2. Approximate numbers, on the other hand, involve some level of uncertainty or rounding, often seen in measurements or floating-point values. For example, π approximated as 3.14 or a table length of 2.5 meters are approximate numbers.
3. Exact numbers are precise; approximate numbers are close estimates.

The Concept of Significant Digits

- The concept of significant digits (or significant figures) in a number refers to the digits that carry meaning and contribute to the precision of that number.
- **Key Rules for Identifying Significant Digits:**
  1. **Non-zero digits** are always significant.
     Example: In 123.45, all five digits are significant.
  2. **Zeros between non-zero digits** are significant.
     Example: In 105, all three digits are significant.
  3. **Leading zeros** (zeros before the first non-zero digit) are not significant.
     Example: In 0.0025, only the 2 and 5 are significant.
  4. **Trailing zeros in a decimal number** are significant.
     Example: In 50.00, all four digits are significant.
  5. **Trailing zeros in a whole number without a decimal point** may or may not be significant, depending on context or notation.
     Example: In 1500, the number of significant digits can be ambiguous unless clarified.

Significant Figures

0.00003400

Zeros are not significant after decimal before non-zero numbers

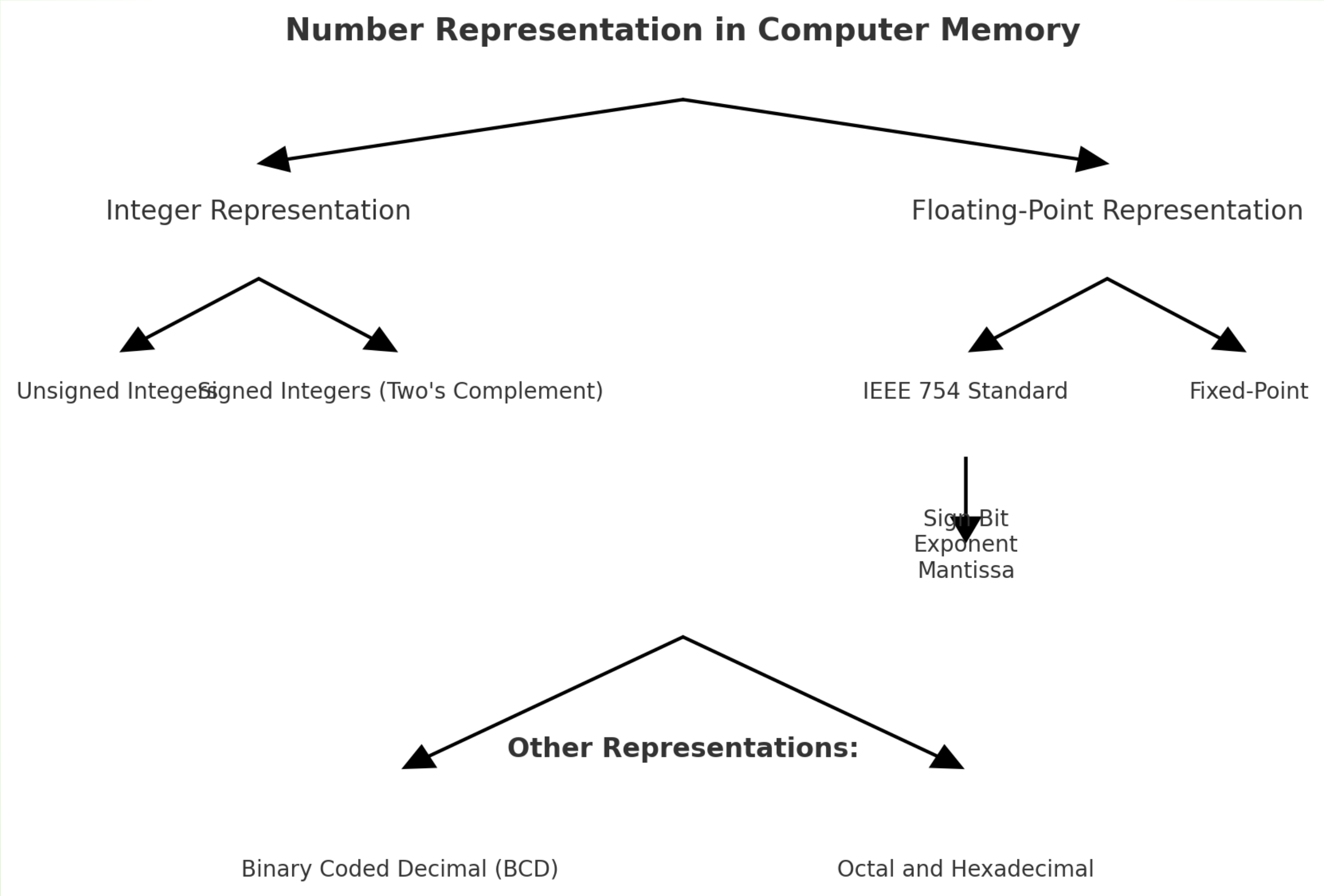
All nonzero numbers are significant

Zeros after nonzero numbers in a decimal are significant

Significant Figures Practice Problems

1) 3.0800	5	12) 2.84 km	3
2) 0.00418	3	13) 0.029 m	2
3) 7.09×10^{-5}		14) 0.003068 m	4
4) 91,600	3	15) 4.6×10^{-5} m	2
5) 0.003005	4	16) 4.06×10^{-9} m	3
6) 3.200 x 109	3	17) 750 m	2
7) 250	2	18) 75m	2
8) 780,000,000	2	19) 75,000 m	2
9) 0.0101	3	20) 75,000. m	5
10) 0.00800	3	21) 75,000.0 m	6
11) 2804	4	22) 10 cm	1

Representation of Numbers In Computer Memory.



Storage of Integer Numbers

1. **Signed Representation:** In signed representation, the most significant bit (MSB) of the binary number is used to indicate the sign of the number: **0** for positive numbers **1** for negative numbers.
   Example: ‘5’ in an 8-byte system: ‘00000101’
   ‘-5’ in an 8-byte system: ‘11111010’

2. **1's Complement Representation:** In 1's complement representation, positive numbers are represented in the same way as in unsigned binary. To represent a negative number, you invert (complement) all the bits of its positive counterpart. Example: ‘5’ in an 8-byte system: ‘00000101’
   ‘-5’ in an 8-byte system: ‘11111010’

Storage of Integer Numbers

3. 2's Complement Representation: 2's complement is the most widely used method for representing signed integers in computer systems. In this method, positive numbers are represented in the same way as in unsigned binary. To obtain the 2's complement of a negative number, you take the 1's complement of the number and then add 1 to the least significant bit (LSB).

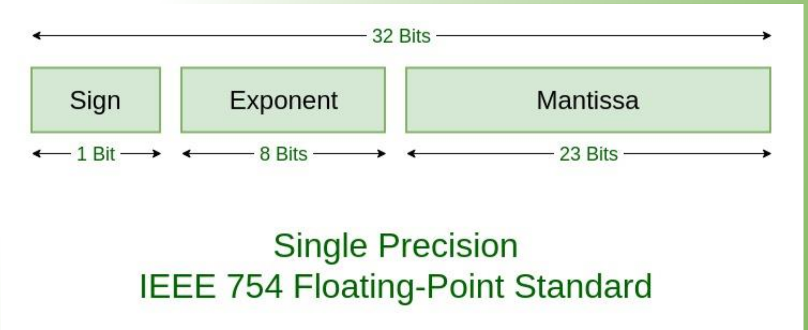
Example:
‘+5’ in an 8-byte system: ‘00000101’
‘-5’ in an 8-byte system: ‘11111011’

Storage of Floating Point Numbers

- The floating-point form is used to represent real numbers of greatly varying magnitude although there is a maximum length to the digits that can be stored.

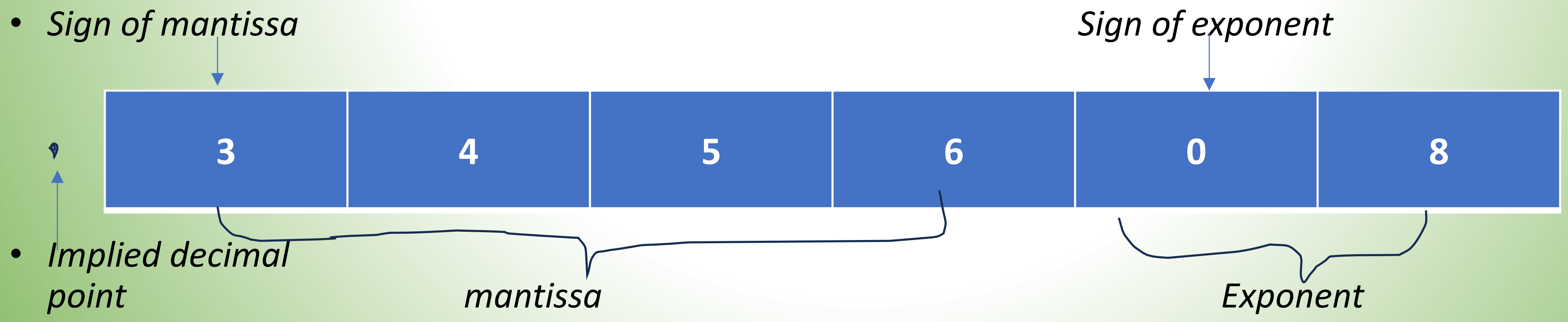
- In the Floating-point form of representing real numbers, a real number is expressed as a combination of a mantissa and exponent.

- For example: 2.998×10^8 is written as 2.998E8.



Concept of Normalization

- It is a standard practice to make mantissa less than 1 and greater than or equal to .1.
- The shifting of the decimal point to the left of the most significant digit is called normalization and the representation of a number in this form is called *normalized floating point numbers.*
- *Example:* the normalized form of the number 32.58 X 10^6 is .3258E8.



Floating Point Arithmetic

1. ***<u>Addition Operation</u>*** (using the example **of 0.4142×10^6** and **0.5156×10^6**)
   1. Represent the Numbers in Standard Floating Point Format
      0.4142×10^6 and 0.5156×10^6
   2. Align the Exponents: In this case, both numbers already have the same exponent of 6, so no further alignment is needed.
   3. Add the Mantissas: $0.4142+0.5156=0.9298$
   4. Normalize the Result: The result of the addition is 0.9298×10^6. This is already in normalized form, where the mantissa is between 0.5 and 1.0 (or in some cases between 1.0 and 2.0, depending on the floating-point representation).
   5. Final Result: 0.9298×10^6

Floating Point Arithmetic

2. *<u>Subtraction Operation</u>* (using the example **of 0.5462E-99** and **0.5483E-99**)

1. Represent the Numbers in Standard Floating Point Format
   0.5462×10^{-99} and 0.5483×10^{-99}
2. Align the Exponents: Since both numbers have the same exponent, -99, there is no need for alignment. We can directly proceed with the subtraction.
3. Subtract the Mantissas: $0.5483 - 0.5462 = 0.0021$
4. Normalize the Result: In this case, the result 0.0021×10^{-99} is already in a form that doesn't require further normalization. The mantissa is within a valid range for floating point representation.
5. Final Result: .0021E-99.

Floating Point Arithmetic

3. *Multiplication Operation*
   1. The mantissa of two numbers are multiplied
   2. Exponent of two numbers are added.
   3. The final result is obtained by rounding off the mantissa to four decimal places after normalizing the mantissa and adjusting the exponent accordingly.

   Examples:

   i) .6534E5 X .2525E7 = .16498..E(5+7)=.1650E12

   ii) .1122E15 X .1222E-20 = .1371E-6

Floating Point Arithmetic

4. *<u>Division Operation</u>*

   1. The mantissa of numerator is divided by the mantissa of the denominator.
   2. Exponent of the denominator is subtracted from the exponent of the numerator.
   3. The final result is obtained by rounding off the mantissa to four decimal places after normalizing the mantissa and adjusting the exponent accordingly.

   Examples:

   i) .5431E0 / .4552E1 = .119310..E(0-1)=.1193E0

   ii) .2753E1 / .9873E-2 = .2788E3

Errors

- Errors in computer numerical calculations arise due to the limitations of digital representations of numbers and the algorithms used to perform arithmetic operations. These errors can accumulate and significantly affect the accuracy of results

- <u>In other words, the numerical results obtained are often approximate values i.e. have an error associated with it. The error associated with an approximate value is defined as the difference between the true value and the approximate value.</u>

Sources of Errors

1. **Mathematical Modeling Errors**:

   •Occur when simplifying real-world problems into mathematical models.
   •Result from assumptions, approximations, and inadequate representation of complex phenomena.

2. **Inherent Errors**:

   •Arise from limitations in measurement instruments and natural variability.
   •Include uncertainties in physical constants and data.

3. **Rounding Errors**:

   •Caused by the finite precision of computers, leading to small discrepancies when numbers are rounded.
   •Can accumulate in successive calculations, affecting accuracy.

Sources of Errors

- **Truncation Errors**:

  •Result from approximating infinite processes (e.g., series, integrals) with finite steps.
  •Common in numerical methods like integration, differentiation, and iterative algorithms.

- **Blunders**:

  •Human errors such as incorrect data entry, programming mistakes, or misinterpretation of results.
  •Often preventable with careful attention and verification.

MEASURES OF ACCURACY

a) *Absolute error:* The absolute error is defined as the absolute difference between the true value of the quantity and its approximate value as given or obtained by measurement or calculation.

   Example: If x is the exact value and x* is the approximate value of a quantity then absolute error is given by: $e_{abs} = |x - x^*|$

MEASURES OF ACCURACY

b) <u>Relative Error:</u> The relative error is defined as the ratio of absolute error to the absolute true value of the quantity.

That is, $e_{rel} = \frac{e_{abs}}{|x|}$, where x is the true value of the quantity.

c) <u>Percentage Error:</u> The percentage error is defined as the product of the relative error and 100.

That is, $e_{per} = 100 \times e_{rel}$

ERROR PROPAGATION

Propagated error in an arithmetic operation occurs due to approximate values of numbers taken by computer with only finite digits.

<u>(I) Propagation of error in addition operation</u>

Let X_A and Y_A be the approximate values of two numbers whose exact values are X and Y. Let e_X and e_Y be the errors in these numbers. Let Z denote the sum of X and Y That is,

$Z = X + Y$

Then ZA the approximate value of Z is given by $Z_A = X_A + Y_A$

ERROR PROPAGATION

The error in Z is $e_Z = Z - Z_A$

Now $\quad Z = X + Y$

$$Z_A + e_Z = (X_A + e_X) + (Y_A + e_Y)$$

$$Z_A + e_Z = (X_A + Y_A) + (e_X + e_Y)$$

$$e_Z = e_X + e_Y$$

Therefor, the error in the sum of two numbers is equal to the sum of their errors. If eX and eY are of opposite signs; then the resultant error eZ is reduced.

Further $| e_Z | = | e_X + e_Y |$

$\Rightarrow | e_Z | \leq | e_x | + | e_y |$

=> The absolute error of a sum of two numbers is less than or equal to the sum of their absolute errors.

ERROR PROPAGATION

(II) Propagation of error in subtraction operation

Let XA and YA be the approximate values of two numbers whose exact values are X and Y. Let eX and eY be the errors in these numbers.

Let Z denote the difference of X and Y

That is,

$$Z = X - Y$$

Then ZA the approximate value of Z is given by

$$Z_A = X_A - Y_A$$

ERROR PROPAGATION

<u>(II) Propagation of error in subtraction operation</u>

The error in Z is $e_Z = Z – Z_A$

Now $Z = X - Y$

$$Z_A + e_Z = (X_A + e_X) - (Y_A + e_Y)$$

$$Z_A + e_Z = (X_A + Y_A) - (e_X + e_Y)$$

$$e_Z = e_X - e_Y$$

Therefor, the error in the difference of two numbers is equal to the difference of their errors. If eX and eY are of opposite signs; then the resultant error eZ is reduced. Further

$$| e_Z | = | e_X - e_Y |$$

$$| e_Z | \leq | e_x | + | e_y |$$

The absolute error of a difference of two numbers is less than or equal to the sum of their absolute errors.

ERROR PROPAGATION

(III) Propagation of error in multiplication operation

Let X_A and Y_A be the approximate values of two numbers whose exact values are X and Y. Let e_X and e_Y be the errors in these numbers.

Let Z denote the product of X and Y
That is,

$$Z = X_Y$$

Then ZA the approximate value of Z is given by

$$Z_A = X_A Y_A$$

The error in Z is

$$e_Z = Z - Z_A$$

Now

$$Z_A = X_A Y_A$$

ERROR PROPAGATION

$Z - e_Z = (X - e_X)(Y - e_Y)$

$Z - e_Z = X_Y - Xe_Y - Ye_X + e_Xe_Y$

Now neglecting the product eXeY which is very small being the products of errors.

$e_Z \approx Xe_Y + Ye_X$

Dividing both side by $Z(= X_Y)$

$e_Z/Z \approx e_Y/Y + e_X/X$

Further $|e_Z/Z| \approx |e_Y/Y + e_X/X|$

$|e_Z/Z| \leq |e_Y/Y| + |e_X/X|$

The relative error of a product of two numbers is less than or equal to the sum of the relative errors of the factors.

ERROR PROPAGATION

(IV) Propagation of error in division operation

Let X_A and Y_A be the approximate values of two numbers whose exact values are X and Y. Let e_X and e_Y be the errors in these numbers.

Let Z denote the product of X and Y

That is, $Z = X/Y$

Then Z_A the approximate value of Z is given by

$$Z_A = X_A / Y_A$$

The error in Z is: $e_Z = Z - Z_A$

Now $Z_A = X_A / Y_A$

ERROR PROPAGATION

$Z - e_Z = (X - e_X) / (Y - e_Y)$

$e_Z = Z + (X - e_X) / (Y - e_Y)$

$e_Z = (X / Y) + (X(1 - e_X / X) / Y(1 - e_Y / Y))$

By binomial expansion

$(1 - e_Y / Y)^{-1} = (1 + e_Y / Y)$

$e_Z \approx e_X / Y - Xe_Y / Y^2 + e_X e_Y / Y^2$

Now neglecting the product $e_X e_Y / Y^2$ which is very small being the products of errors

$e_Z \approx e_X / Y - Xe_Y / Y^2$

ERROR PROPAGATION

Dividing both side by Z(= X / Y)

$e_Z\,/Z \approx e_X\,/X - e_Y$ / Y

Further

$|e_Z\,/Z| \approx |e_X\,/X - e_Y\,/\mathrm{Y}|$

$|e_Z\,/Z| \leq |e_Y\,/Y| + |e_X\,/\mathrm{X}|$

The relative error of a product of two numbers is less than or equal to the sum of their relative errors.

Submitted By: Kartik Kumar Sahu 4007/23 BCA - II

Submitted To: Ms. Nisha Gupta