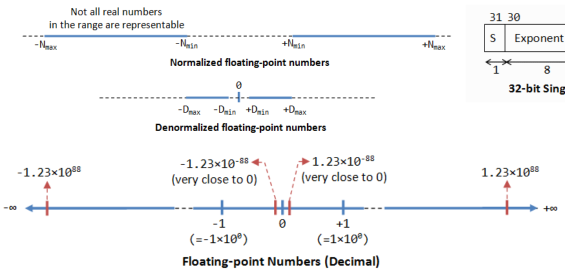
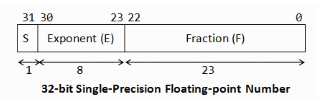
FLOATING POINT VARIABLES

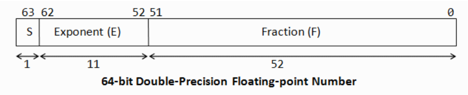
Integers and floats are two different kinds of numerical data. An integer is a number without a decimal point where as a float is a floating point number, i.e. it is a number with a decimal point.Floating point numbers are stored in this format: M x b^e, where m is the mantissa (an integer number), b is base and e is exponent. C supports two floating types: float and double. The float and double are represented using 32-bit single precision and 64-bit double precision. For single precision floating point we have: 1 sign bit, 8 exponent bits, 23 mantissa bits. For double precision floating points we have: 1 sign bit, 11 exponent bits, 52 mantissa bits.Following figure illustrate how floating point number is stored in memory:



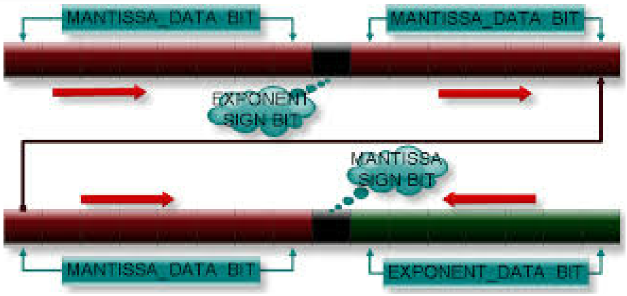
Floating-point Numbers (Decimal)

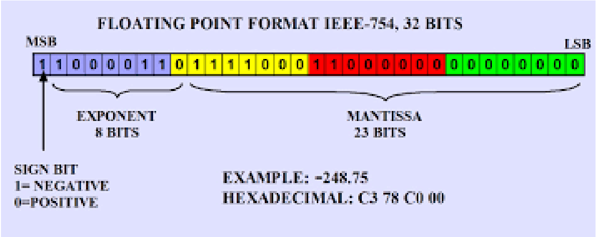


32-bit Single-Precision Floating-point Number



64-bit Double-Precision Floating-point Number

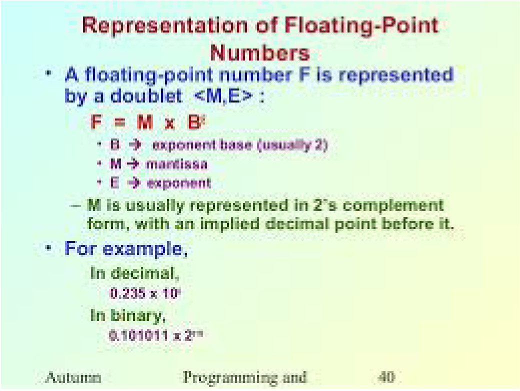




Five important rules to be followed:

Rule 1: To find the mantissa and exponent, we convert data into scientific form.

Rule 2: Before the storing of exponent, 127 is added to the exponent.



Representation of Floating-Point Numbers

- A floating-point number F is represented by a doublet <M,E> :

  $F = M \times B^{E}$

  - B → exponent base (usually 2)
  - M → mantissa
  - E → exponent

  – M is usually represented in 2's complement form, with an implied decimal point before it.
- For example,

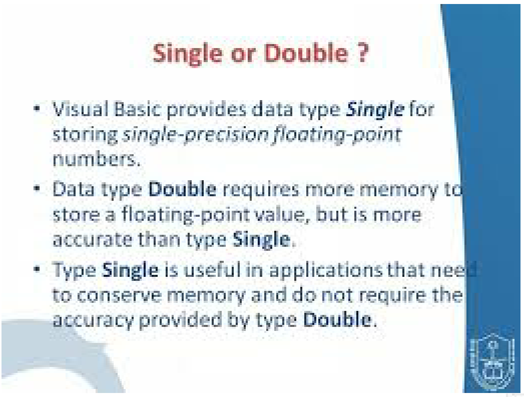
  In decimal,

  0.235×10^{6}

  In binary,

  0.101011×2^{0110}

Autumn Programming and 40



Single or Double ?

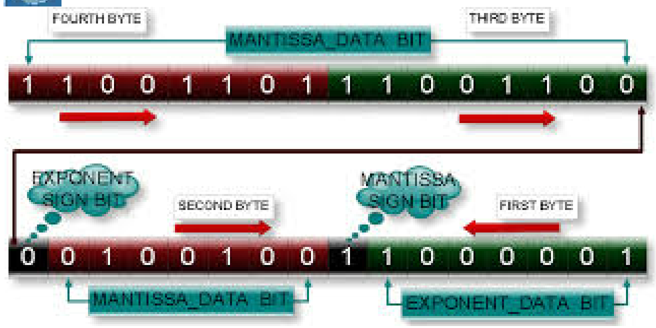
- Visual Basic provides data type ***Single*** for storing *single-precision floating-point* numbers.
- Data type **Double** requires more memory to store a floating-point value, but is more accurate than type **Single**.
- Type **Single** is useful in applications that need to conserve memory and do not require the accuracy provided by type **Double**.

Rule 3: Exponent is stored in memory in first byte from right to left side.

Rule 4: If exponent is negative number it will be stored in 2's complement form.

Rule 5: Mantissa is stored in the memory in second byte from right to left side.

Example: Memory representation of float a= -10.3f


FOURTH BYTE
MANTISSA_DATA_BIT
THIRD BYTE
1 1 0 0 1 1 0 1 1 1 0 0 1 1 0 0
EXPONENT
SIGN BIT
SECOND BYTE
MANTISSA
SIGN BIT
FIRST BYTE
0 0 1 0 0 1 0 0 1 1 0 0 0 0 0 1
MANTISSA_DATA_BIT
EXPONENT_DATA_BIT