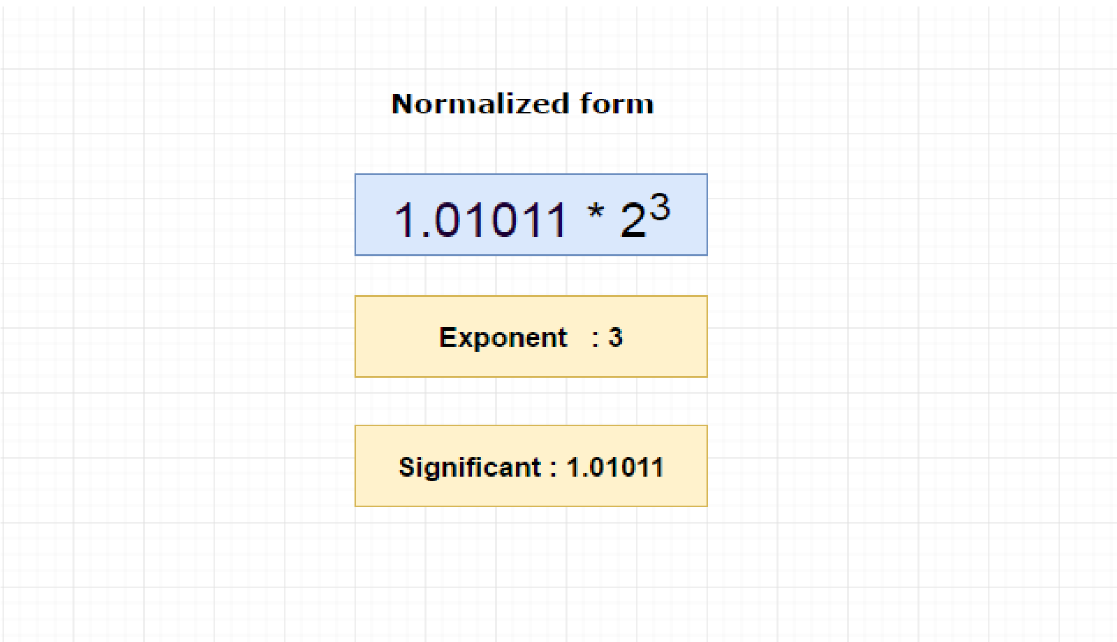
Storing of Decimal Number

➢ float:

- To store a floating-point number, 4-byte (32 bit) memory will be allocated in computer.
  - 1 bit for sign
  - 8 bits for exponent part
  - 23 bits for significant part
- Floating number will be converted to binary number.
  - 10.75 to (1010.11).
- Converted binary number to normalize form.

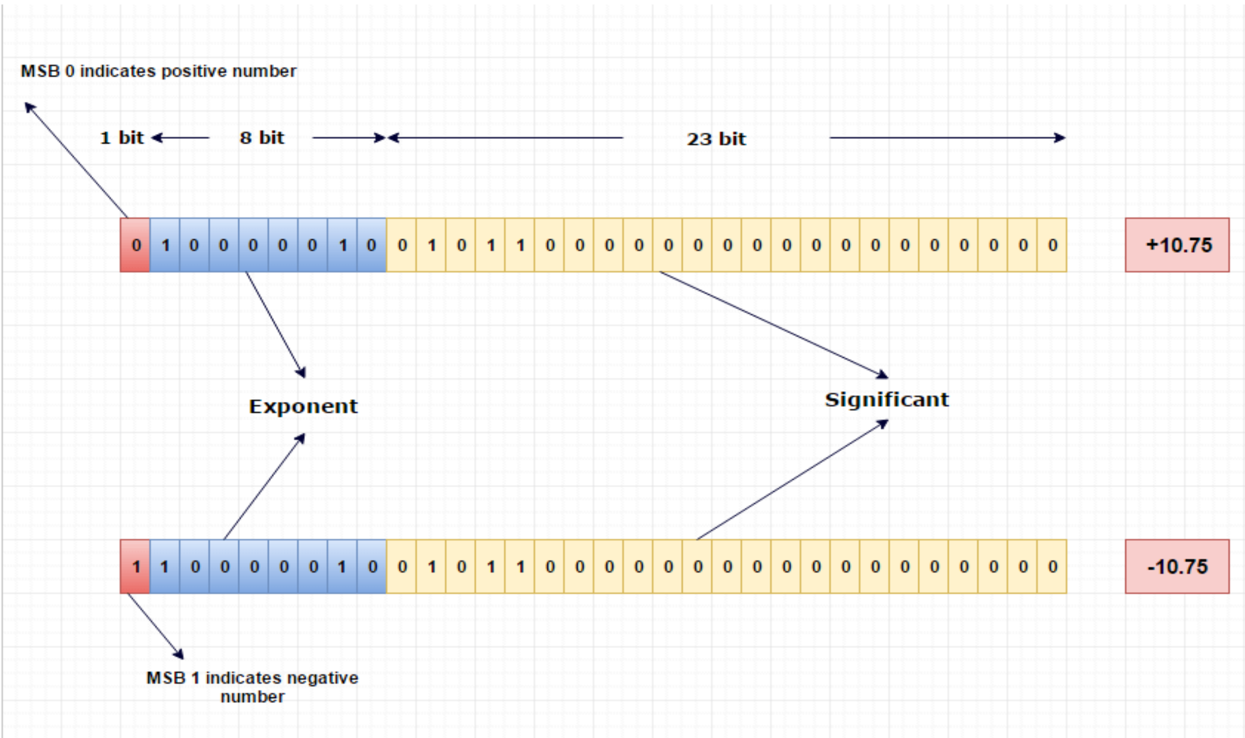


- Add bias to exponent.
  - Formula to calculate bias value is-
    - bias = **(2^ (n - 1)) – 1**.
  - Here, we have allocated 8 bits for exponent. So, n will be **8**.
    - (2^(8-1))-1 = 127
  - Hence the normalized exponent value will be,
    - Actual exponent + bias value, i.e. 3 + 127 = 130.
    - Binary form of 130 is 10000010.

- Representation
    - Sign bit 0 because 10.75 is positive number.
    - Exponent value is 130 which is 10000010.
    - Significant value is 01011.

✓ **Syntax:** Sign Bit (1 bit) + Exponent Value (8 bits) + Significant Value (23 bits)

❖ **Output:** 0 10000010 01011000000000000000000



➢ double:

- To store a double-point number, 8-byte (64 bit) memory will be allocated in computer.
    - **1** bit for sign
    - **11** bits for exponent part
    - **52** bits for significant part
- Formula to calculate bias value is-
    - bias = (2^(n-1))-1.
    - (2^(11-1))-1 = (2^10)-1 = 1024-1 = **1023**.

✓ **Syntax:** Sign Bit (1 bit) + Exponent Value (11 bits) + Significant Value (52 bits)