

- byte : 8 bits
- nibble : 4 bits (half-byte)
- word : multiple bytes (1, 2, 4) (for MIPS it's 4)
- int : 4 bytes (1 bit for sign, 31 for magnitude)
- float : 4 bytes
- double : 8 bytes
- char : 1 byte

- Convert decimal whole numbers to base R : divide by R , first remainder is LSB, last is MSB
- Convert decimal fractions to base R : multiply by R , first carry is MSB, last is LSB
- base R to base R^N : partition in groups of N e.g groups of 4 for base 2 to base 16
- Convert to R-1s complement : Flip the digits; **digit** = $R - \text{digit}$
- Convert to Rs complement : Flip the digits, then add 1 to the number
- 1s complement has +ive and -ive 0
- 2s complement has only 1 representation of 0
- 2s complement can represent an additional negative number e.g for binary, 1000 represents -8 (+8 cannot be represented in a signed 4 bit number)
- Convert to excess X: Take number minus X (0 refers to -x)
- IEEE 754 Floating-Point Representation:
sign|exponent|mantissa
- Single-precision float has 1 bit sign, 8 bit excess-127 exponent, 23 bit mantissa (normalized with a leading bit 1 i.e the mantissa is the X in 1.X)
- Double has 1 bit sign, 11 bit excess-1023 exponent, 52 bit mantissa

- 2s complement addition: Simply add & ignore carry out of MSB
- 2s complement subtraction: take 2s complement of number to be subtracted, then do 2s addition.
- 1s complement addition: Add; If there is a carry out, add 1 to the result
- 1s complement subtraction: take 1s complement of number to be subtracted, then do 1s addition.
- check for **overflow: If result is opposite sign of both operands (that have the same sign)**

R, I, J format

- ## Instruction Set Architecture Architectures & Endianness

- Sum-Of-Products (SOP): Product term or a logical sum of product terms
- *minterm*: Product term that contains n literals from all the variables
- Product-Of-Sum (POS): Sum term or a logical product of sum terms
- *Maxterm*: Sum term that contains n literals from all the variables
- $Mx = mx'$ because of De Morgan's
- Sum of 2 distinct Maxterms is 1 e.g $M_{1234} + M_{1120} = 1$

- For m flip-flops, up to 2^m states exist.
- SR has invalid code while JK uses that for the toggle code

Input	0X DE AD BE EF
Big-Endian	0: DE, 1: AD ...
Little-Endian	0: EF, 1: BE ...