

Positional Number System

- **Radix** - number of unique symbols in a number system
- usually 0-9, then A-Z

Number System Base conversion

Base conversion between numeral systems

- x numeral system to y numeral system (example: 32.2_5 to base-7)
- x numeral system $\rightarrow$ decimal $\rightarrow$ y numeral system
- $32.2_5 = 17.4_{10} = 23.2541_7$

32.2_5

$+ 2 \times 5^{-1} = 0.4$

$+ 2 \times 5^0 = 2$

$+ 3 \times 5^1 = 15$

summation: 17.4

$7 \overline{) 17.4}$

2

3

↑

0.4

$\times 7$

2.8

$\times 7$

0.8

$\times 7$

0.6

$\times 7$

0.4

$\times 7$

0.2

$\times 7$

0.2541

$\times 7$

1.4

we stop at 4 fractional places

Slide 13

2x vs 10y

Decimal Name	Abbreviation	Value	Binary Name	Abbreviation	Value	% Large	Value	Value
kilo	K	10^3	kilo	K	2^{10}	1%	2^{10}	1 024
megabit	Mb	10^6	megabyte	MB	2^{20}	1%	2^{20}	1 048 576
gigabit	Gb	10^9	gigabyte	GB	2^{30}	1%	2^{30}	1 073 741 824
terabit	Tb	10^{12}	terabyte	TB	2^{40}	1%	2^{40}	1 099 511 627 776
petabit	Pb	10^{15}	petabyte	PB	2^{50}	1%	2^{50}	1 125 899 906 842 624
exabit	Eb	10^{18}	exabyte	EB	2^{60}	1%	2^{60}	1 152 921 504 606 846 976
zetta	Z	10^{21}	zettabyte	ZB	2^{70}	1%	2^{70}	

- Binary prefix are mainly use in memory capacity
- SI prefix are usually use in data transfer rate or storage space

- abbreviation * value = number of bits

Binary Data Organization

Organization	Number of bits	Usage
Bit (binary digit)	2 cells - 0 or 1	Basic unit
Crumb	2 bits	*largely defunct term - rarely used
Nibble	4 bits	Hex digit, ICD digit
Byte	8 bits	Smallest addressable data unit
Half word	16 bits	Definition of word is architecture-dependent
Word	32 bits	A 32-bit architecture considers 1 word as 32-bit
Double word	64 bits	
Quad word	128 bits	

- a bit has 2 cells
- most significant (left) ----- least significant (right)
- bit(b), byte(B)
- little endian - **top** address to **bottom**
- big endian - **bottom** address to **top**

Integer representation

UNSIGNED

0 to $(2^n)-1$

normal

fill the rest with 0 (MSb)

SIGNED

$-(2^{n-1})$ to $+(2^{n-1})-1$

sign and magnitude

sign bit | positive int

1's complement (n-1's)

flip for negative int

2's complement (n's)

flip then + 1, for negative int

- **unsigned** integers use **zero extension**

- **signed** integers use **sign extension**

in short, extend the MSb until you have reached the sufficient num of bits

integer operation overflow

SHOULD ____; otherwise, overflow

ADDITION

UNSIGNED SHOULD NOT have carry

SIGNED [same sign] SHOULD remain the same sign

SIGNED [different sign] add using 2's complement representation (never overflow)

SUBTRACTION

UNSIGNED SHOULD HAVE carry

SIGNED $A-B = A+B'$ (2's complement B)

addition of signed integers [same sign]

1. *first bit should never change*
2. *ignore carry if there is*



By **wayneswu**

cheatography.com/wayneswu/

Not published yet.

Last updated 18th September, 2023.

Page 1 of 3.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>

IEEE 754 Floating point for single precision

1 - sign bit	8 - exponent	23 - mantissa
0 for positive	$e' = e + 127$	f in 1.f notation

Example:

Given: 3.5_{10}

1. $3.5_{10} = 11.1_2$

2. 1.11×2^1

3. $e' = 128_{10} == 1000_0000_2$

Answer: 1_1000000_110 0000...00000

IEEE 754 Floating point for single precision

1 - sign bit	8 - exponent	23 - mantissa
0 for positive		

test

1 - sign bit	8 - exponent	23 - mantissa
0 for positive	$e' = e + 127$	f in 1.f notation

Example:

Given: 3.5_{10}

1. $3.5_{10} = 11.1_2$

2. 1.11×2^1

3. $e' = 128_{10} == 1000_0000_2$

Answer: 1_1000000_110 0000...00000

Special cases floating single precision

Sign Bit	E'	Significant	Value
0	0000 0000	000 0000 0000 0000 0000	+0 (Positive Zero)
1	0000 0000	000 0000 0000 0000 0000	-0 (Negative Zero)
0/1	0000 0000	$\neq 0$	Denormalized
0	1111 1111	000 0000 0000 0000 0000	+ Infinity
1	1111 1111	000 0000 0000 0000 0000	- Infinity
x	1111 1111	01x xxxx xxxx xxxx xxxx	sNaN
x	1111 1111	1xx xxxx xxxx xxxx xxxx	qNaN

