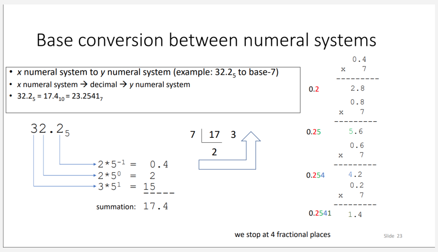
Cheatography

do not use Cheat Sheet
by wayneswu via cheatography.com/193768/cs/40338/

Positional Number System

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

Number System Base conversion



2x vs 10y

Decimal term	Abbreviation	Value	Binary term	Abbreviation	Value	% Larger
kilobyte	KB	10^3	kibibyte	KiB	2^{10}	2%
megabyte	MB	10^6	mebibyte	MiB	2^{20}	5%
gigabyte	GB	10^9	gibibyte	GiB	2^{30}	7%
terabyte	TB	10^{12}	tebibyte	TiB	2^{40}	10%
petabyte	PB	10^{15}	pebibyte	PiB	2^{50}	13%
exabyte	EB	10^{18}	exbibyte	EiB	2^{60}	15%
zettabyte	ZB	10^{21}	zebibyte	ZiB	2^{70}	18%
yottabyte	YB	10^{24}	yobibyte	YiB	2^{80}	21%

Value	Value
2^{10}	1 024
2^{20}	1 048 576
2^{30}	1 073 741 824
2^{40}	1 099 511 627 776
2^{50}	1 125 899 906 842 624
2^{60}	1 152 921 504 606 846 976

- 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, BCD 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*

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 x 21
3. $e' = 128_{10} == 1000\_0000_2$
Answer: 1_1000000_110 0000...00000

Special cases floating single precision

Sign Bit	E'	Significand	Value
0	0000 0000	000 0000 0000 0000 0000 0000	+0 (Positive Zero)
1	0000 0000	000 0000 0000 0000 0000 0000	-0 (Negative Zero)
0/1	0000 0000	≠ 0	Denomalized
0	1111 1111	000 0000 0000 0000 0000 0000	+ Infinity
1	1111 1111	000 0000 0000 0000 0000 0000	- Infinity
x	1111 1111	01x xxxx xxxx xxxx xxxx xxxx	sNaN
x	1111 1111	1xx xxxx xxxx xxxx xxxx xxxx	qNaN