Computer Science CSCI 261

Computer Architecture and Assembly Language

Dr. Peter Walsh
Department of Computer Science
Vancouver Island University
peter.walsh@viu.ca

Negative Numbers

- Sign Magnitude
- 2s Complement
- Excess n (Biased)
  - e.g. excess 3 (4 bit numbers)

Number		Excess 3	
0	0000	3	0011
1	0001	4	0100
2	0010	5	0101

etc

Excess 8 (4 bit numbers)

Number	Excess 8	Value
0	8	-8
1	9	-7
2	10	-6
3	11	-5
4	12	-4
5	13	-3
6	14	-2
7	15	-1
8	0	0
9	1	1
10	2	2
11	3	3
12	4	4
13	5	5
14	6	6
15	7	7

Excess 8 (4 bit numbers) cont.

- Modulo 16 Arithmetic

Addition		Subtracton	
(5)	5	(5)	5
(2)	2	(-2)	14
(7)	7	(3)	19 mod 16

Excess 127 (8 bit numbers)

- IEEE 754
  - use modulo 256 arithmetic
  - see Excess n Table Generator (home page)

Number	Value
0	reserved
1	-126
2	-125
. . . .	
127	0
. . . .	
254	127
255	reserved

Integers and Characters (8 bit numbers)

- e.g. 1110 1010
  - decimal: 234
  - hex: EA
  - Ascii: å see Ascii Tables (home page)
  - Sign Magnitude: -106
  - 2s Complement: -22
  - IEEE 754 Excess 127: 107

Integers and Strings

- e.g. "Hi"
  - sequence of ASCII characters
  - often null terminated
  - hex: 48 69 0
  - binary: 1001000 01101001 00000000
- e.g. "234"
  - hex: 32 33 34 0
  - binary: 0110010 0110011 0110100 00000000

Binary Coded Decimal (BCD)

- e.g. 234
  - BCD: 0010 0011 0100

Digit	BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Converting Fractional Parts

- Decimal To Binary

e.g. 0.24

2 * 0.24	
2 * 0.48	0 (msb)
2 * 0.96	0
2 * 0.92	1
2 * 0.84	1

↓

Converting Fractional Parts cont.

- Binary To Decimal

e.g. 0.011

$$= 0 * 2^{-1} + 1 * 2^{-2} + 1 * 2^{-3}$$

$$= \frac{1}{4} + \frac{1}{8} = \frac{3}{8}$$

$$= 0.375$$

Converting Fractional Parts cont.

- Precision
  - to maintain the same degree of precision, a decimal number with m fractional digits must be represented by a binary number with k fractional digits such that $2^k * 10^{-m} \geq 1$
- e.g. 0.247

$$2^k * 10^{-3} \geq 1$$

$$\Rightarrow 2^k \geq 1000$$

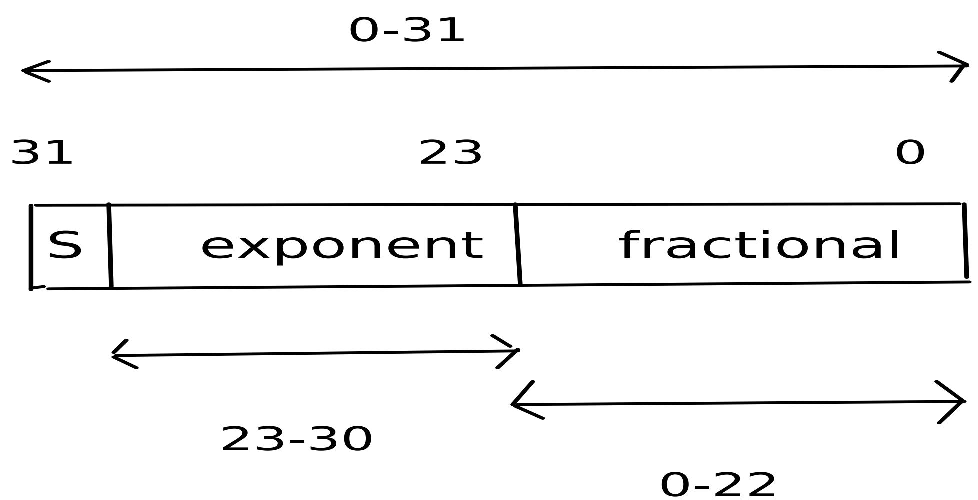
$$\Rightarrow k \geq 10$$

Floating Point Numbers

- Decimal (scientific notation)
  - e.g. $62345.0 = 6.2345 * 10^4$
    the mantissa is 6.2345 and the exponent is 4
- Binary (derived from scientific notation)
  - e.g. $10101.0 = 1.0101 * 2^4$
    the fractional part is 0101 and the exponent is 4
  - notice that the leading digit in the mantissa is always 1, consequently, the mantissa can always be re-constructed from the fractional part

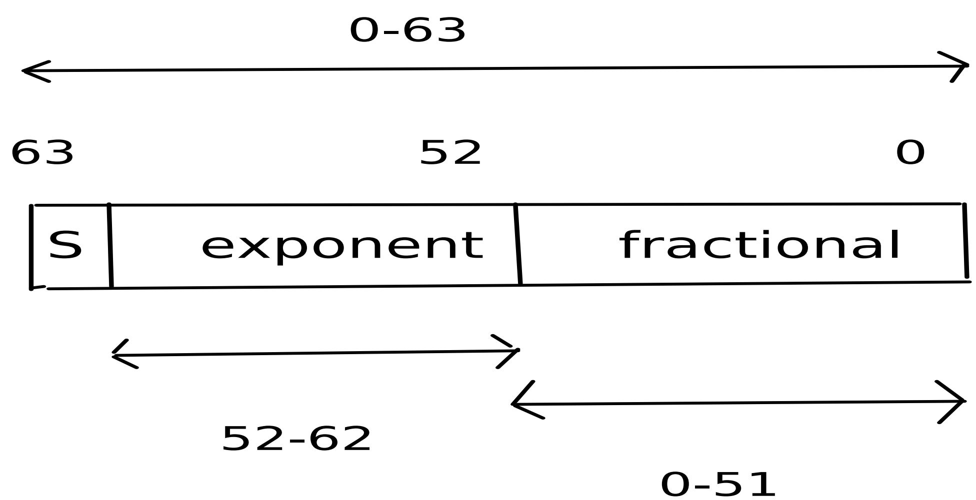
IEEE 754

- Single Precision (32 bits)
  - 1 sign bit
  - 8 bit exponent (biased 127)
  - 23 bit fractional part (unsigned)



IEEE 754

- Double Precision (64 bits)
  - 1 sign bit
  - 11 bit exponent (biased 1023)
  - 52 bit fractional part (unsigned)



IEEE 754 Single Precision

- e.g. $19.5 = 10011.1 = 1.00111 * 2^4$
  - see IEEE 765 Calculator (home page)

S (sign) = 0

Exponent (biased 127) = 127 + 4 =131

= (8 bits) 10000011

Fractional Part = 00111

= (23 bits) 00111000000000000000000

0	10000011	00111000000000000000000