Chapter #10
Computer Arithmetic

Computer Arithmetic

- Operations performed by Arithmetic Logic Unit (ALU)
- Arithmetic Operands:
  - —Integers
  - —Fractional numbers
- Issues
  - —Representation
  - —Range of values

Fractional Numbers

- Integer + fraction
  - Ex: 1001.1010 = $2^3 + 2^0 + 2^{-1} + 2^{-3} = 9.625$
- Problem:
  - Locating radix point (radix point not stored)
- Solution:
  - Fixed point
    - Adv: simple to represent (radix point position is fixed)
    - Disadv: limited range of representable numbers
  - Floating point
    - Adv: wider range of representable numbers
    - Disadv: more complex to represent (radix point position varies)

Floating Point Arithmetic

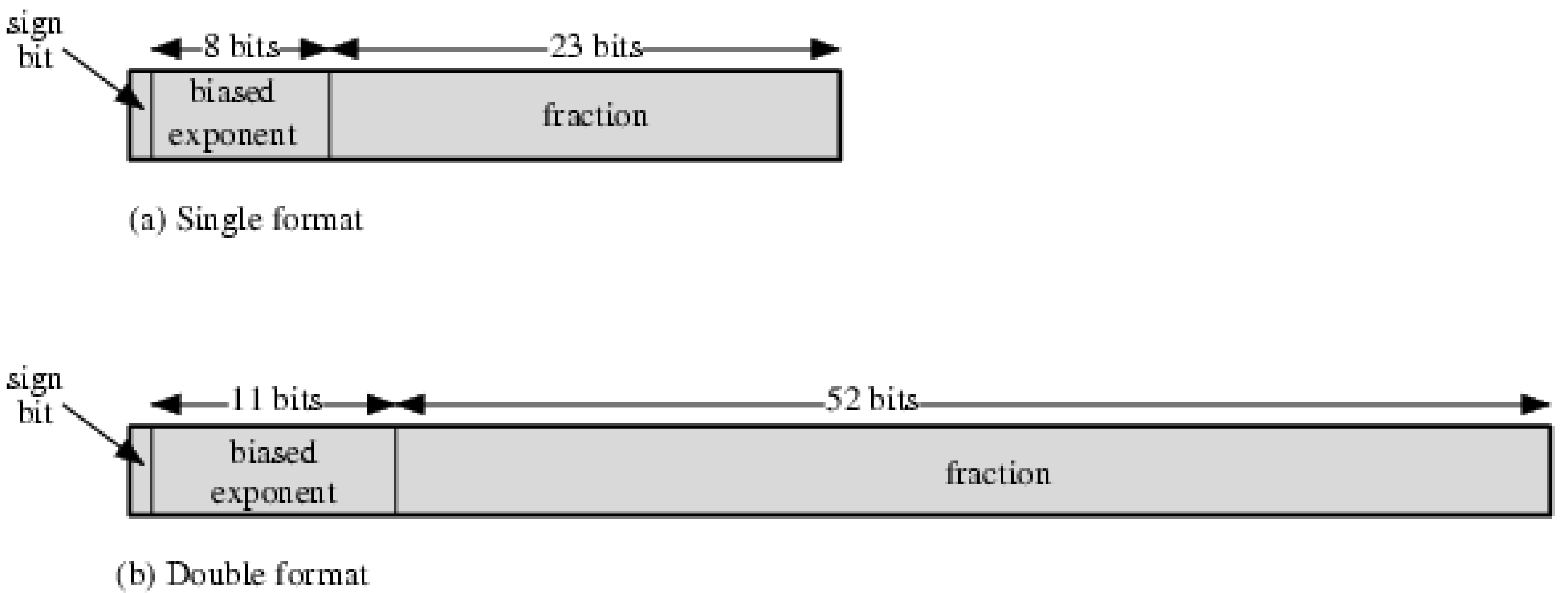
- Number representation:
  $\pm\ m \times b^e$ {m=mantissa, b=base, e=exponent}

- Standard binary format:
  - Sign *s* (0 or 1)
  - Exponent *e* (often biased to represent negative #'s)
  - Mantissa *m* (normalized so leading bit is 1)

IEEE-754 Floating Point Standard

Characteristic	Single-precision	Double-precision
Length in bits	32	64
Fraction part in bits	23	52
Hidden bits	1	1
Exponent length in bits	8	11
Bias	127	1023
Approximate range	$2^{128} \approx 3.8 \times 10^{38}$	$2^{1024} \approx 9.0 \times 10^{307}$
Smallest normalized number	$2^{-126} \approx 10^{-38}$	$2^{-1022} \approx 10^{308}$

IEEE-754 Floating Point Standard



IEEE-754 Floating Point Features

- Sign:
  - 0 for positive, 1 for negative
- Exponent:
  - Biased +127 (single-precision) or +1023 (double-precision)
  - Actual exponent = Represented exponent – bias
- Significand:
  - Normalized number in range [1…2)
  - First digit is assumed to be 1, so it is hidden

IEEE-754 Number Types (Single precision)

Type	Sign (1 bit)	Exp (8 bits)	Mantissa (23 bits)
Normalized	0 or 1	$0 < Exp < 255$ (biased +127)	Any pattern of 0's and 1's
Zero	0 or 1	Exp=0	All 0's
Denormalized	0 or 1	Exp=0 (biased +126)	Any pattern except all 0's
Infinity	0 or 1	Exp=255	All 0's
NaN	0 or 1	Exp=255	Any pattern except all 0's

Decimal → IEEE format

1. Convert decimal number to binary
2. Normalize number w/in range: [1,…,2), adjusting exponent & hide leading 1
3. Bias exponent by +127
4. Append sign, exponent, mantissa
5. Convert from binary to hex

Decimal → IEEE Example

Example:

$(-23.5625)_{10} \rightarrow (?)_{IEEE}$

1. Decimal→binary: $(-23.5625)_{10} \rightarrow (-10111.1001)_2$
2. Normalize: $-10111.1001 \rightarrow -1.01111001 \times 2^4$
3. Biased exponent: $4+127=131$
4. Append:

1	10000011	011110010…0

5. Binary→HEX: $(C1BC8000)_{IEEE}$

IEEE format → Decimal

1. Convert from hex to binary
2. Extract sign, exponent, mantissa
3. Unbias exponent by -127
4. Unnormalize number, removing exponent & adding leading 1
5. Convert binary number to decimal

IEEE → Decimal Example

Example:

(C1BC8000)$_{IEEE}$

1. HEX→Binary: 1100 0001 1011 1100 1000 0…0
2. Extract: Sign=neg, exp=10000011, mant=011110010…0
3. Unbiased exponent: 131-127=4
4. Unnormalize: -1.011110010×2^4→-10111.1001
5. Binary→Decimal: -23.5625