

Computer Programming

Dr. Deepak B Phatak
Dr. Supratik Chakraborty
Department of Computer Science and Engineering
IIT Bombay

Session: Representing Floating Point Numbers

Quick Recap of Relevant Topics



- Architecture of a simple computer
- Representation of integers

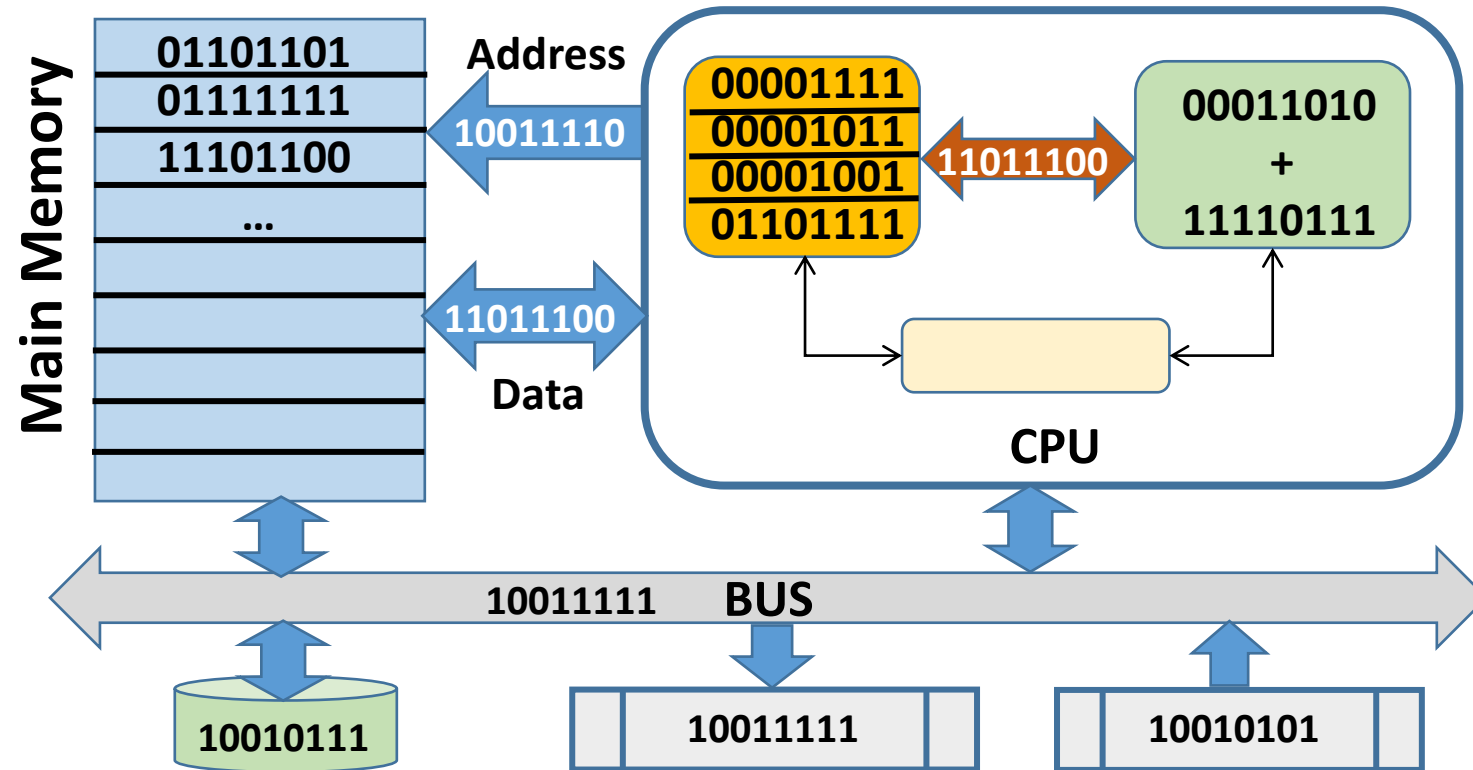
Overview of This Lecture



- A computer's internal representation of numbers
 - Floating point numbers
- C++ declarations of floating point variables

Recap from Earlier Lecture

- Snapshot:

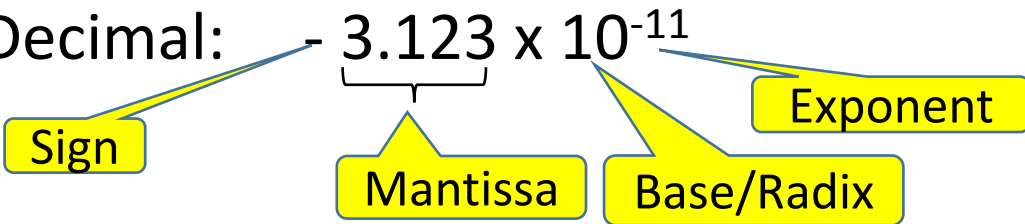


- How do we represent numbers like 3.14×10^{-23} in a computer?

Representing Floating Point Numbers

- Numbers with fractional values, very small or very large numbers cannot be represented as integers
- Floating point number

• Decimal: -3.123×10^{-11}



The diagram shows the components of the decimal number -3.123×10^{-11} . A yellow box labeled 'Sign' points to the minus sign. A yellow box labeled 'Mantissa' points to the decimal part '3.123'. A yellow box labeled 'Base/Radix' points to the '10' in the power of ten. A yellow box labeled 'Exponent' points to the power '-11'.

- Mantissa = $-(3 \times 10^0 + 1 \times 10^{-1} + 2 \times 10^{-2} + 3 \times 10^{-3})$
- Binary: $-1.1101 \times 2^{00000110}$
 - Mantissa = $-(1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4}) = -1.8125$
 - Exponent = $(1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0) = 6$

Representing Floating Point Numbers



- **Normalized mantissa**: single non-0 digit to left of radix point
 - $0.02345 \times 10^{12} = 2.345 \times 10^{10}$
 - $110.101 \times 2^{00000110} = 1.10101 \times 2^{00001000}$
 - Binary: Implicit 1 always on left of radix point; need not be stored
- Floating point numbers represented by allocating fixed number of bits for mantissa and exponent
 - Cannot represent all real numbers
 - Finite precision artifacts
 - What is $0.101 \times 2^{111} + 1$ if we have only 3 bits to represent mantissa?

Floating Point Numbers in C++

- **float** and **double** data types
- **float**
 - 32 bits (4 bytes): 1 sign, 8 exponent (2's complement), 23 mantissa (other than implicit leading "1.")
 - Approximate range of magnitude: $10^{-44.85}$ to $10^{34.83}$
- **double**
 - 64 bits (8 bytes): 1 sign, 11 exponent (2's complement), 52 mantissa (other than implicit leading "1.")
 - Approximate range of magnitude: $10^{-323.3}$ to $10^{308.3}$
- Special bit patterns reserved for 0, infinity, NaN (not-a-number: result of 0/0), ...
- C++ declarations: **float** temperature; **double** verticalSpeed;

Floating Point Numbers in C++



- Floating point constants can be specified in C++ programs as
 - 23.572 (can have non-normalized mantissa in programs)
 - 2357.2e-2 or 2357.2E-2 (scientific notation)
 - 2357.2×10^{-2} (base 10)
- C++ constant floating point declaration
 - `const float pi = 3.1415`
 - `const double e = 2.7183`
 - Values of `pi` and `e` cannot change during program execution

Summary



- Binary representation of floating point numbers
 - Sign, mantissa and exponent
 - C++ declarations

Practice Quiz

- Consider the C++ declaration

```
const float scaleFactor = -1.875 x 2-32;
```

Show how `scaleFactor` is represented in the computer's memory.

Recall float has 32 bits: 1 sign, 8 exponent, 23 mantissa and assume the number is represented as



Practice Quiz



- Now consider

```
const float accGravity = 9.875;
```

Show how `accGravity` is represented in the computer's memory.