Floating-Point Arithmetic

Lecture 16
Section 3.5

Robb T. Koether

Hampden-Sydney College

Wed, Oct 2, 2019

1. Floating-Point Addition
2. Floating-Point Multiplication
3. Floating-Point Division
4. Rounding
   - Loss of Precision
   - Guard and Round Bits
5. Assignment

Outline

Floating-Point Addition

- To add floating-point numbers in binary, we must
  - Use the exponents to align the binary points of the mantissas.
  - Add the mantissas.
  - Normalize the result.
  - Round off the result.

Floating-Point Addition

Example (Floating-Point Addition)

$$\begin{aligned} 1.0111 \times 2^2 + 1.1101 \times 2^1 &= 1.0111 \times 2^2 + 0.11101 \times 2^2 \\ &= 10.01011 \times 2^2 \\ &= 1.001011 \times 2^3 \\ &= 1.0011 \times 2^3. \end{aligned}$$

- Add 1.0111×2^2 and 1.1100×2^{-1}.
- Assume 4-bit precision in the mantissas.

Outline

Floating-Point Multiplication

Floating-Point Multiplication

$$\begin{aligned}\left(1.1101 \times 2^4\right) \times \left(1.0011 \times 2^{-2}\right) &= 10.00100111 \times 2^2 \\ &= 1.0001 \times 2^2.\end{aligned}$$

- To multiply floating-point numbers,
  - Multiply the mantissas.
  - Add the exponents.
  - Normalize the result.
  - Round off the mantissa.

Roundoff

Roundoff

$$
\begin{aligned}
&24.0 \times 28.0 - 23.0 \times 29.0 \\
&\qquad = \left(1.1000 \times 2^4\right) \times \left(1.1100 \times 2^4\right) \\
&\qquad\quad - \left(1.0111 \times 2^4\right) \times \left(1.1101 \times 2^4\right) \\
&\qquad = 10.10100000 \times 2^8 - 10.10011011 \times 2^8 \\
&\qquad = 1.0101 \times 2^7 - 1.0101 \times 2^7 \\
&\qquad = 0.0.
\end{aligned}
$$

- Roundoff can severely affect the results.

Outline

Floating-Point Division

Floating-Point Division

$$\begin{aligned}\left(1.0011 \times 2^4\right) \div \left(1.1101 \times 2^{-2}\right) &= 0.101001111011\ldots \times 2^6 \\ &= 1.0101 \times 2^5.\end{aligned}$$

- To divide float-point numbers,
  - Divide the mantissas.
  - Subtract the exponents.
  - Normalize the result.
  - Round off the mantissa.

Outline

Rounding

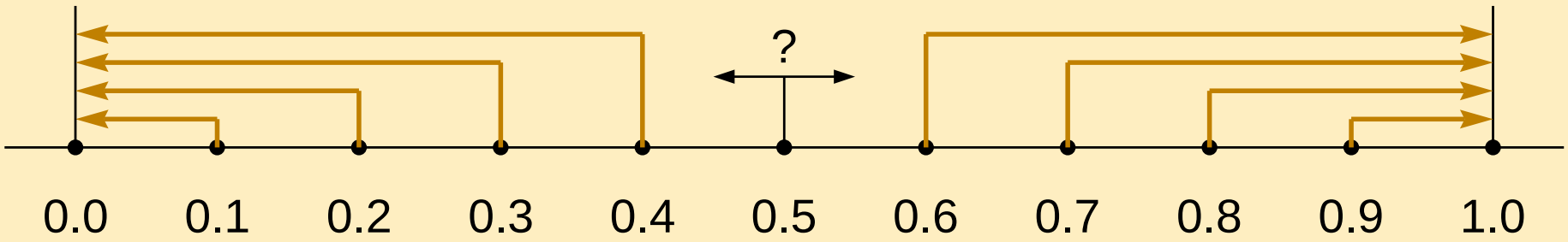
Rounding

1.1	10
10.1	11
11.1	100
+100.1	+101
1100.0	1110

- The conventional rule in decimal is to round down if the next digit is 0 - 4 and to round up if it is 5 - 9.
- The equivalent rule in binary is to round down if the next bit is 0 and to round up if the next bit is 1.
- Both of these rules skew the result when the fractional part is exactly one half.

Rounding

Rounding



Rounding

Round to Nearest Even

1.1	10
10.1	10
11.1	100
+100.1	+100
1100.0	1100

- A more equitable rounding rule when rounding on 1 is to
  - Round down if the previous bit is 0.
  - Round up if the previous bit is 1.
- This will always result in a 0 in the rounded position.

Other Rounding Modes

- IEEE 754 provides four rounding modes.
  - Round to nearest even.
  - Round up (towards $+\infty$).
  - Round down (towards $-\infty$).
  - Round towards 0.
- The processor can be set to use any one of these rounding modes.

Outline

Loss of Precision

Loss of Precision

$$\begin{aligned} 1.1101 \times 2^0 + 1.0100 \times 2^{-6} &= 1.1101 \times 2^0 + 0.0000010100 \times 2^0 \\ &= 1.1101010100 \times 2^0 \\ &= 1.1101 \times 2^0. \end{aligned}$$

- Note that the result is the same as if we had added 0.

Loss of Precision

Loss of Precision

$$(1.1101 \times 2^0 + 1.0100 \times 2^{-6}) - 1.1101 \times 2^0$$

- What would be the result of the above calculation?
- What should it be, mathematically?

Outline

Guard and Round Bits

- Each computed result includes 2 additional bits.
  - The guard bit.
  - The round bit.
- Their purpose is to increase the accuracy of the rounded-off results.

Guard and Round Bits

Floating-Point Division

$$\begin{aligned}\left(1.0011 \times 2^4\right) \times \left(1.1101 \times 2^{-2}\right) &= 0.101001111011\ldots \times 2^6 \\ &= 1.0101 \times 2^5.\end{aligned}$$

- If we computed only the mantissa, the result would be 1.0100×2^5.
- If we computed only the mantissa and the guard bit, the initial result would be 1.01001×2^5 and the round-to-nearest-even rule would round this to 1.0100×2^5.
- If we compute the mantissa and the guard and round bits, the initial result is 1.010011×2^5 and the rounded result is 1.0101×2^5.

Outline

Assignment

Assignment

- Read Section 3.5.