Floating Point

Tom Kelliher, CS 220

1 Administrivia

Today's Objectives

1. Explain the structure of IEEE single-precision binary numbers.
2. Explain the meanings of overflow and underflow for floating point numbers.
3. Convert decimal numbers in scientific notation to and from IEEE single precision binary numbers.
4. Add, subtract, and multiply IEEE single-precision binary numbers, producing correctly normalized results.

Next Up

Read 3.1 and 3.2.

1. Demonstrate an understanding of the structure and components of the hypothetical stored program machine by explaining the purpose of each component and tracing the fetch and execution stages of simple instructions through the machine.
2. Assemble very simple sequences of C code involving sequential and conditional execution.
3. Determine the effect of these ISA features on instruction size: registers, addressing modes, and instruction format.

2 Warm-Up

1. By looking at a binary number, how do you know if it's an unsigned integer, a two's complement integer, or a floating point number?

   (a) By looking at the sign bit.

   (b) By checking the exponent field for the presence of the bias factor.

   (c) You can't know.

   (d) By checking for overflow.

2. An IEEE single-precision number is stored in memory in the order sign bit (msb), exponent, significand because

   (a) It doesn't really matter; it's just an agreed-upon convention

   (b) The sign bit is always the msb and the biasing of the exponent makes room for the significand to be to the right (towards the lsb) of the exponent

   (c) The sign bit is always the msb and the sign bit and the exponent have to fit into the same byte in memory

   (d) The sign bit matters most in determining the comparative value of a number, followed by the biased exponent, and finally the significand

   (e) None of the above

3. IEEE single-precision exponents are biased by 127 because

   (a) it prevents underflow.

   (b) biasing allows floating point numbers to be compared using integer comparators.

   (c) to make the range of the floating point numbers symmetric.

   (d) to allow the exponent to be stored in eight bits.

4. IEEE floating point numbers are normalized because

   (a) it prevents underflow.

   (b) normalization allows floating point numbers to be compared using integer comparators.

   (c) to make the range of the floating point numbers symmetric.

   (d) to allow the significand to be stored in 23 bits.

5. Which statement is true of floating point arithmetic?

   (a) Significands are aligned when adding and multiplying.

   (b) Significands are aligned when adding but not when multiplying.

   (c) Significands are aligned multiplying but not when adding.

   (d) Significands are not aligned.

3 Problems

1. What is the difference between overflow in the context of two's complement numbers and overflow in the context of floating-point numbers?

2. Convert the decimal numbers -57.5 and 31.125 to IEEE single-precision. Express your answers in hexadecimal notation.

3. Convert the IEEE single-precision numbers 0x24A60000 and 0xC43C0000 to decimal numbers in scientific notation.

4. Add each of the pairs of binary numbers in scientific notation, following the four steps shown on pg. 77 of the textbook. Each of these numbers has five bits of precision. These pairs were carefully chosen to illustrate some of the pitfalls of fixed precision floating point arithmetic. Identify the pitfall illustrated by each problem. How does the actual floating point addition algorithm address these pitfalls?

   (a) $1.1101 \times 2^4 + -1.1110 \times 2^4$

   (b) $1.1111 \times 2^1 + 1.0000 \times 2^6$

   (c) $1.1111 \times 2^5 + 1.0000 \times 2^1$