CS 213, Fall 2000
Homework Assignment H2
Handed out: Oct. 5, Due: No handin (for exam practice only)

Dave O'Hallaron (`droh@cs.cmu.edu`) is the lead person for this assignment.

In this assignment, we will work with two small floating point formats based on the IEEE standard:

- LITTLE format:
  - 8 bits
  - There is a sign bit in the most significant bit.
  - The next four bits are the exponent. The exponent bias is 7.
  - The last three bits are the significand.
- TINY format:
  - 6 bits
  - There is a sign bit in the most significant bit.
  - The next three bits are the exponent. The exponent bias is 3.
  - The last two bits are the significand.

Otherwise, the rules are like those in the IEEE standard (normalized, denormalized, representation of 0, infinity, and NAN).

Problem 1 – Special Values

For *both* formats, give the following values (in decimal):

1. Largest positive finite number
2. Positive normalized number closest to zero
3. Largest positive denormalized number
4. Positive denormalized number closest to zero

Problem 2 – Encoding LITTLE numbers

Encode the following values as 8-bit LITTLE floating point numbers: $3/4$, $-13/16$, 44, and -104

Problem 3 – Decoding LITTLE numbers

Determine the values (real number, $\pm\infty$, or `NaN`) corresponding to the following LITTLE format bit patterns (most significant bit on the left):

1. 10110011
2. 01111010
3. 10010001
4. 01001111
5. 11000001

Problem 4 – Converting LITTLE numbers to TINY numbers

Convert the following 8-bit LITTLES to 6-bit TINIES. Overflow should yield $\pm\infty$, underflow should yield ± 0.0, and rounding should follow the "round-to-nearest-even" tie-breaking rule covered in class.

1. 00010000
2. 11101001
3. 00110011
4. 11001110
5. 11000101