Two's complement

- Unsigned binary numbers work to store natural numbers, but many calculations use negative numbers as well.
  - Sign and magnitude is an obvious solution, but it has a rather serious flaw (what?).
  - Bias notation works well, but has another less obvious problem (what?).
- Eventually, two's complement became the standard solution for representing signed integers.
  - Most significant bit has a negative value, all others have positive.
  - Otherwise exactly the same as unsigned integers.
- A neat trick for flipping the sign of a two's complement number : flip all the bits and add 1.

Two's Complement Questions

For the following questions assume a 32 bit integer.

- What is the largest integer? The largest integer + 1?
- How do you represent with an integer the numbers 0, 1, and -1?
- How do you represent 17, -17 with a signed int?

Float Introduction

- Floats were made to increase the range of values to include very small and large reals (technically, extended reals).
- This added range comes at the cost of precision - remember that N bits can at most represent 2^N things.
- Done with normalized binary numbers (similar to scientific notation, but in base 2)
- The IEEE single precision floating point: $(-1)^{\text{sign}} \times 1.\text{mantissa} \times 2^{(\text{exponent}-127)}$
  - 1 bit for sign
  - 8 bits for exponent
  - 23 bits for mantissa
- Notice: Implicit 1 to the left of binary point, mantissa is only to the right
- Exponent uses biased notation instead of two's complement. Why?
- Note also that floats use Sign and Magnitude and therefore have 2 representations for 0.

Special Floats

- zero : e = 0, m = 0
- denorm: e = 0, m $\neq$ 0
- $\pm\infty$: e = 255 (MAX), m = 0
- NaN: e = 255 (MAX), m $\neq$ 0

Denormalized Numbers

- Serve to fill in the between 2^{-126} and 0
- "Denormalized" comes from number not being normalized (there is no longer a 1 before the '.')
- Has an implicit exponent of -126, but no implicit leading 1 in its mantissa
- Thus they take the form $(-1)^{\text{sign}} \times 0.\text{mantissa} \times 2^{-126}$

Float Questions

For the following assume a single precision IEEE float.

- What is the largest float?
- What is the second largest float?
- What is the smallest positive float?
- Suppose you wanted to multiply a float held in $t0 by 8, but could only use the addu, addiu, and sll instructions. How would you (You may assume that the float $< 2^{124}$)?