Notes: Negative Binary Integers

Goals

At the end of today, you should know

- The three ways to represent signed binary integers.
- The pros and cons of each system.
- How to negate a binary number in each system.
- How to do subtraction in two's complement.
- What overflow is, what it looks like, and when it occurs.

Unsigned Integers

- Unsigned integers are ≥ 0; n-bit number can represent $0 - 2^n$-1. Leftmost bit indicates 2^{n-1}.
- Binary addition is like decimal addition: Add right to left, add carry to left if necessary. Binary subtraction is like decimal subtraction (borrow from left).

Signed Integers

- Signed integers: Leftmost bit used as sign bit (0 for nonnegative numbers, 1 for negative numbers).
- To get symmetry, we'd like to represent (for some k), the 2k+1 numbers -k, -k+1, -k +2, …, -1, 0, 1, 2, …, k. But n bits gives us 2^n bit patterns.
- Sign magnitude: To negate number, flip its sign bit
  - Problem: "Negative" zero
- One's complement: To negate number, flip all its bits: -k is unsigned $(2^{n-1}-1)$-k.
  - Problem: "Negative" zero
  - How much is 00000 - 1?
- Two's complement: Take one's complement and add 1: -k is unsigned 2^{n-1}-k.
  - Shortcut
  - Problem: One more negative than positive number.
  - Note $-2^{n-1}+k$ is unsigned $2^{n-1}+k$.
  - $-2^{n-1}+k$ is $-(2^{n-1}-k)$ is unsigned $2^{n-1}-(2^{n-1}-k)$ is k
  - In two's complement, x-y is x+(-y)

Overflow

- Overflow: Result of operation on n bits doesn't fit into n bits.
- When can this occur?