Lecture 18

Negative Numbers in Binary

We use modular arithmetic, so for a 4-bit binary number,

$$5 - 1 = 5 + (-1) = 5 + 15 = 4$$

if we ignore the final carry.

- There are different ways to implement this, e.g. 2's complement that we use, where given n bits, $-k = 2^n - k$
  - Example: $n = 4, k = 1 \Rightarrow -k = 15$
  - Example: $n = 4, k = 6, \Rightarrow -k = 10$
  - Example: $n = 5, k = 13, \Rightarrow -k = 19$
- The maximum value that can be represented is then reduced by half, where the first bit denotes the sign
- To represent the same number with more bits, extend the required number of bits on the left by the first digit, e.g.
  - 0b1011 = 0b11111011
  - 0b0011 = 0b00000011
- Short cut calculation: -k = ~k+1
  - e.g. 2 = 0b0010, -2 = 0b1110
  - e.g. 1 = 0b0001, -1 = 0b1111

Addition/Subtraction

- using the same adder as before
- supply 1 input `mux` to decide addition or subtraction
- output = A + B^`mux` + `mux` (second `mux` acts as carry in)

Overflow

- 7+1 in a 4-bit signed binary becomes -8
- If members have opposite signs, overflow cannot occur
- overflow is equivalent to sn-1^cout for an n-bit signed binary