E11

Lecture 1 - Supplemental Notes on Two's Complements

1. Unsigned Binary Numbers

We have discussed how a collection of bits, i.e. binary variables that can take on the value of 0 or 1, can represent a number larger than 1. The left-most bit is the Most significant bit, and the right-most bit is the least significant bit. For example, a 4 bit binary number has maximum value 1111 that represents $2^3 + 2^2 + 2^1 + 2^0 = 15$. The smallest binary number in this case would be 0000 = 0.

2. Signed Binary Numbers

Since we want to represent negative numbers as well, we need a method of incorporating the sign of the number into the bits.

2.1 Method 1: First bit represents the sign

A naïve approach might be to simply use the first bit to represent the sign of the number, and the remaining bits to count up as before. For example, to represent +3 with 4 bits we use **0**011, while to represent -3 we use **1**011.

2.2 Method 2: Two's Complements

A better way is to use two's complements. For any given positive number represented with n bits, one obtains the negative of that number by flipping all the bits and adding 1.

For example, to obtain -3 using 4 bits, we start with 0011, flip the bits to obtain 1100, then add 1 to obtain 1101. Note how the left most bit still signifies the sign as being negative.

2.3 Why Two's Complements?

We want to add and subtract our numbers. Lets consider adding 4 and -3 with both methods. In method 1, we get:

```
 0100
+1011
_____
 1111   This is -7 in method 1 (Is this correct for 4 – 3 ?)
```

Now let's try two's complement:

```
 0100
+1101
_____
 0001   This is +1 in method 2 (Is this correct for 4 – 3 ?)
```