

# CS61C Course Notes

Anmol Parande

Fall 2019 - Professors Dan Garcia and Miki Lustig

## Contents

<b>1</b>	<b>Binary Representation</b>	<b>2</b>
1.1	Base conversions . . . . .	2
1.2	Numeric Representations . . . . .	2
1.2.1	Sign and Magnitude . . . . .	2
1.2.2	One's Complement . . . . .	2
1.2.3	Two's Complement . . . . .	3
1.2.4	Bias Encoding . . . . .	3

**Disclaimer:** These notes reflect 61C when I took the course (Spring 2019). They may not accurately reflect current course content, so use at your own risk. If you find any typos, errors, etc, please report them on the GitHub repository.

## 1 Binary Representation

Each bit of information can either be 1 or 0. As a result,  $N$  bits can represent at most  $2^N$  values.

### 1.1 Base conversions

#### Binary to Hex conversion

- Left pad the number with 0's to make 4 bit groups
- Convert each group its appropriate hex representation

#### Hex to Binary Conversion

- Expand each digit to its binary representation
- Drop any leading zeros

### 1.2 Numeric Representations

When binary bit patterns are used to represent numbers, there are nuances to how the resulting representations are used.

**Definition 1** *Unsigned integers are decimal integers directly converted into binary. They cannot be negative.*

**Definition 2** *Signed integers are decimal integers whose representation in binary contains information denoting negative numbers.*

Binary math follows the same algorithms as decimal math. However, because computers only have a finite number of bits allocated to each number, if an operation's result exceeds the number of allocated bits, the leftmost bits are lost. This is called **overflow**

#### 1.2.1 Sign and Magnitude

In the sign and magnitude representation, the leftmost bit is the sign bit. A 1 means the number is negative where as 0 means it is positive. The downside to this is when overflow occurs, adding one does not wrap the bits properly.

#### 1.2.2 One's Complement

To fix the bit wrapping, to form the negative number, we can flip each bit. This solves the wrapping issue so even with overflow, when adding 1, the result will continue increasing.

### 1.2.3 Two's Complement

To convert from decimal to two's complement:

- Invert each bit of the positive version of the number
- Add one to the result

The same process can be utilized to go from the negative version of a number to the positive version.

### 1.2.4 Bias Encoding

With bias encoding, the number is equal to its unsigned representation plus a bias turn. With a negative bias, we can center a positive range of  $0 \rightarrow 2^N - 1$  on 0.