Monday, August 27, 2018 2:27 PM

CS 61C Fall 2018

## Number Representation

Discussion 1: August 27, 2018

Notes

## Unsigned Integers

1.1 If we have an n-digit unsigned numeral  $d_{n-1}d_{n-2}\dots d_0$  in radix (or base) r, then the value of that numeral is  $\sum_{i=0}^{n-1} r^i d_i$ , which is just fancy notation to say that instead of a 10's or 100's place we have an r's or  $r^2$ 's place. For the three radices, binary, decimal, and hex, we just let r be 2, 10, and 16, respectively.

We don't have calculators during exams, so let's try this by hand. Recall that our preferred tool for writing large numbers is the IEC prefixing system:

$$\begin{array}{c} \cdot \text{ Ki (Kibi)} = 2^{10} \\ \cdot \cdot \text{ Gi (Gibi)} = 2^{30} \\ \cdot \cdot \text{ Mi (Mebi)} = 2^{20} \\ \cdot \cdot \text{ Ti (Tebi)} = 2^{40} \\ \cdot \cdot \text{ Ei (Exbi)} = 2^{60} \\ \cdot \cdot \text{ Yi (Yobi)} = 2^{80} \\ \end{array}$$

(a) Convert the following numbers from their initial radix into the other two common radices:

1.  $0b_{10010011} = |.2^{\circ} + |.2^{\circ} + |.2^{\circ} + |.2^{\circ} + |.2^{\circ} = |47 = 0 \times 93$ 2.63 = 0 63 =  $0 \times 3$ 3. 0b00100100

3 V2 = 15R1 15/2 = 7R1

$$7/2=3$$
 R 6. 43

(b) Convert the following numbers from hex to binary:

- 1. 0xD3AD=Ob 10 001 (010 (0)
- 2. 0xB33F
- 3. 0x7EC4

(d) Write the following numbers as powers of 2:  $1 \cdot 2^{10}$  11  $2^{9}$  2  $2^{10}$ 

binary solecimal binary so hex decimal so hex X Edigit. 2 Place 11010 Flowest:0 0.20+1.21+0.22 +1.73 +1.24 = 0 + 2 + 0 + 8 + 16= 26 div by 2, find R 37/2 = 18 18/2 = 9 9/2 = 4 4/2 = 2 2/2 = 0 1/2 = 01001011 BC>H group binary into 4 1100 1011 0011->3 1101 - D 0 x D 3

ltil F

## 2 Signed Integers

2.1 Unsigned binary numbers work for natural numbers, but many calculations use negative numbers as well. To deal with this, a number of different schemes have been used to represent signed numbers, but we will focus on two's complement, as it is the standard solution for representing signed integers.

- Most significant bit has a negative value, all others are positive. So the value of an *n*-digit two's complement number can be written as  $\sum_{i=0}^{n-2} 2^i d_i 2^{n-1} d_n$ .
- 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.
- Addition is exactly the same as with an unsigned number.
- Only one 0, and it's located at 0b0.

For questions (a) through (c), assume an 8-bit integer and answer each one for the case of an unsigned number, biased number with a bias of -127, and two's complement number. Indicate if it cannot be answered with a specific representation.

(a) What is the largest integer? The largest integer's representation + 1?

1. Unsigned? 66 | | | | | | | | | = 255 + | -> 06 | 0000 0000

- (b) How would you represent the numbers 0, 1, and -1?
  - 1. Unsigned?
  - 2. Biased?
  - 3. Two's Complement?
- (c) How would you represent 17 and -17?
  - 1. Unsigned?
  - 2. Biased?
  - 3. Two's Complement?
- (d) What is the largest integer that can be represented by *any* encoding scheme that only uses 8 bits?

Sign + may X

Two's comp

Bias

Two's Comp

+: Obo Chang ]

> + value

-: Ob![ ~ ]

Plip bits, add!

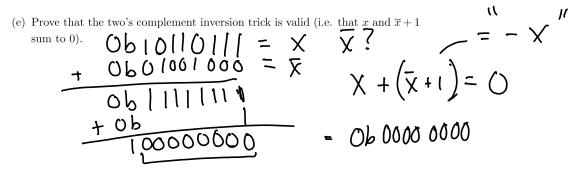
- hew-mag

Bias

Unsigned shifted

Unsigned # - Bias

Ob 000000000



(f) Explain where each of the three radices shines and why it is preferred over other bases in a given context.

## 3 Counting

- 3.1 Bitstrings can be used to represent more than just numbers. In fact, we use bitstrings to represent *everything* inside a computer. And, because we don't want to be wasteful with bits it is important that to remember that n bits can be used to represent  $2^n$  distinct things. For each of the following questions, answer with the minimum number of bits possible.
  - (a) How many bits do we need to represent a variable that can only take on the values  $0,\,\pi$  or e?
  - (b) If we need to address 3 TiB of memory and we want to address every byte of memory, how long does an address need to be?
  - (c) If the only value a variable can take on is e, how many bits are needed to represent it?