31

Monday, August 27, 2018 2:27 PM

CS 61C Fall 2018

Number Representation

Discussion 1: August 27, 2018

D= individual/groups 1 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:

- Ki (Kibi) =
$$2^{10}$$
 - Gi (Gibi) = 2^{30} - Pi (Pebi) = 2^{50} - Zi (Zebi) = 2^{70}

- Mi (Mebi) =
$$2^{20}$$
 - Ti (Tebi) = 2^{40} - Ei (Exbi) = 2^{60} - Yi (Yobi) = 2^{80}

(a) Convert the following numbers from their initial radix into the other two com-

1. $0b_{10010011} = 0 \times 93 = 2^{6} + 2^{1} + 2^{4} + 2^{7} = 147$ $[2]_{63} = 0$ | | | | | | | = 0 | 00 | | | | | | | = 0 | | | 3 | | | $\sqrt{3}$ 0b00100100 = $0\times24 = 2^2+25 = 36$ 4. 0 ObO OxO < home

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

 $\frac{2.~0 \mathrm{xB33F}}{3.~0 \mathrm{x7EC4}}$ $\frac{3}{2}$ home

Conoup

home (d) Write the following numbers as powers of 2: D512 Ki 29.210-219 • 2 Ki 2.2 10=21 • 256 Pi • 128 Ei

home

can represent 2' unique things.

1) Binary - Decimal B-12 place dight 2 + 1.23 + 1.24 =0+2+0+8+16 = 26

1)->B divide by 2+ Remainder

Dec => Hex Hex -> Binary " Group by 4 -> conver ex 11010017 or reverse 0x 123 <mark>0-> 1101</mark> 306 1101 0011 3-3 0011

Bin Hex 0000 0001 1001 10/0 1011 1100 1101 HIL

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? **255** , **0**

 - 2. Biased? 128, -127
 3. Two's Complement? 127, -128
- (b) How would you represent the numbers 0, 1, and -1?
 - 1. Unsigned? O b 0000 0000 000 0001
 - 2. Biased? ObOIIIIII Ob | 000 0000
 - 3. Two's Complement? Obomomo Ob 00000001

0601111110

(c) How would you represent 17 and -173

- 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?

all up to interpretation

& home

Two's Complement . + value 21 in bin

- tip bits, add 0600...01

- new-magnitude

Can do (-) now!

Zeros

Unsigned value - Bias

Bias typically (5-1)

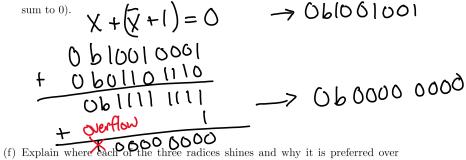
4 bits, bias = 7

for representable range X

Sign + value

Bias

(e) Prove that the two's complement inversion trick is valid (i.e. that x and $\overline{x} + 1$ sum to 0).



other bases in a given context.

home

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?