The UNIVERSITY *of* NORTH CAROLINA *at* CHAPEL HILL

Comp 411 Computer Organization
Spring 2011

Problem Set #1
Issued Wed. 1/12/11; Due Wed. 1/26/11 (beginning of class)

Homework Information: Some of the problems are probably too long to be done the night before the due date, so plan accordingly. Late homework will not be accepted. Feel free to get help from others, but the work you hand in should be your own. You may **not** use solutions to a previous year's homework to aid you in completing this assignment. Please **enter your answers in the space provided,** but feel free to attach any additional sheets of scratch work.

Problem 1: Information Encoding (18 points)

In lecture we learned that information resolves uncertainty, and that information is measured in units of *bits*. In order to uniquely identify one of **N** equally likely alternatives, $\log_2\mathbf{N}$ bits of information must be communicated. Use this information to solve the problems below:

a) How many bits are necessary to encode an integer in the range of 0 to 63 (inclusive)?

Answer:

b) How many bits are necessary to encode an integer in the range of 0 to 256 (inclusive)?

Answer:

c) How many bits are necessary to encode an integer in the range of -256 to 255 (inclusive)?

Answer:

d) Consider a standard card deck (4 distinct suits; values 2-10, J, Q, K, A). How many bits are necessary to encode a card's suit (ignoring value)?

Answer:

e) How many bits are necessary to encode a card's value (ignoring suit)?

Answer:

f) Suppose we create a deck of cards that consists only of the number cards (2-9) and only has two suits. What is the smallest number of bits necessary to uniquely encode the new deck?

Answer:

Problem 2: Two's Complement Representation (24 points)

Most computers choose a particular *word length* (measured in bits) for representing integers and provide hardware that performs operations on word-size operands. Many current generation processors have word lengths of 64 bits. Restricting the size of the operands and the result to a single word means that the arithmetic operations are actually performing arithmetic modulo 2^{64}.

a) How many different values can be encoded in a 64-bit word? Give your answer in terms of a power of 2.

Answer:

b) If the 64-bit word must allow negative numbers, does that impact the total number of values that can be encoded? If so, how many distinct values can now be encoded?

Almost all modern computers use a 2's complement representation for integers since the 2's complement addition operation is the same for both positive and negative numbers. In 2's complement notation, one negates a number by complementing each bit in its representation (i.e., changing 0's to 1's and vice versa) and adding 1. By convention, we write 2's complement integers with the most-significant bit (MSB) on the left and the least-significant bit (LSB) on the right. Also by convention, if the MSB is 1, the number is negative; otherwise it's non-negative. Use an ***8-bit*** 2's-complement representation to answer the following questions:

c) What is the binary representation for 0?

Answer:

d) What is the binary representation for the most positive number that can be represented?

Answer:

e) What is the binary representation for the most negative number that can be represented?

Answer:

f) What is the decimal value for the most positive number?

Answer:

g) What is the decimal value for the most negative number?

Answer:

h) What is the result of negating the largest-magnitude negative integer?

Answer:

Problem 3: Hexadecimal Representation (15 points)

Since writing a long string of binary digits gets tedious, it's often convenient to use hexadecimal notation where a single digit in the range 0—9 or A—F is used to represent adjacent groups of 4 bits (starting from the left). Give the corresponding 8-digit hexadecimal encoding for each of the numbers below. (*Hint:* For decimal numbers, you can either convert them first to binary using the successive division method discussed in class and then convert binary to hex, or you could directly convert decimal to hex using successive division by 16.)

a) 2053 (decimal)

Answer:

b) -128 (decimal)

Answer:

c) 1111 0001 0000 0011 1111 0111 1010 1001 (binary)

Answer:

d) 1010 0101 1100 1000 0000 0101 0011 1111 (binary)

Answer:

e) -7 (decimal)

Answer:

Problem 4: Binary Arithmetic (27 points)

Calculate the following using ***8-bit*** 2's-complement arithmetic (which is just a fancy way of saying to do ordinary addition in base 2 ***keeping only 8 bits of your answer***). Remember that subtraction can be performed by negating the second operand to form its 2's complement, and then adding it to the first operand. Give your answers in binary.

a) 15 + 37

Operand 1:

Operand 2:

Result:

b) 59 - 20

Operand 1:

Negated Operand 2:

Result:

Problem 4: Binary Arithmetic (continued)

c) 20-59

Operand 1:

Negated Operand 2:

Result:

d) 120 + 110

Operand 1:

Operand 2:

Result:

e) Explain in a sentence or two what happened in part d) above.

Problem 5: Fixed-Point Binary (16 points)

Using a ***16-bit*** fixed-point binary representation, where the leftmost 8 bits are the integer part (i.e., before the binary point), and the rightmost 8 bits are the fractional part, convert the first two decimal numbers below into binary, and the remaining two from binary into decimal.

a) 100.25 — Answer:

b) -17.0625 — Answer:

c) 0111 1100 . 1010 0000 — Answer:

d) 1111 1111 . 1001 0000 — Answer: