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

Comp 411 Computer Organization
Prof. Montek Singh
Fall 2012

Problem Set #1
Issued Wed. 9/12/12; Due Wed. 9/19/12 (beginning of class)

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.**

Problem 1: Information Encoding (18 points)

In class we learned that information resolves uncertainty, and that information is measured in units of *bits*. In order to uniquely identify one of $\boldsymbol{N}$ equally likely alternatives, $\lceil \log_2 \boldsymbol{N} \rceil$ bits of information must be communicated. Answer the questions below:

a) How many bits are necessary to encode an integer in the range of 0 to 31 (inclusive)?

Answer:

b) How many bits are necessary to encode an integer in the range of 0 to 512 (inclusive)?

Answer:

c) How many bits are necessary to encode an integer in the range of -32 to 31 (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 its number 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 cards numbered 2 to 6 (inclusive) belonging to only two suits (spades and hearts). How many bits are necessary to encode a card in this deck?

Answer:

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

Most computers choose a particular *word length* (measured in bits) for representing integers. 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? Express your answer as a power of 2.

Answer:

b) If the 64-bit word must allow negative numbers, does that impact the total number of distinct 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 flipping each bit in its representation (i.e., changing 0's to 1's and vice versa) and adding 1 at the end. 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 a ***16-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 biggest positive number that can be represented?

Answer: ____ ____ ____ ____

e) What is the binary representation for the most negative number (i.e., largest magnitude negative number) that can be represented?

Answer: ____ ____ ____ ____

f) What is the decimal value for the biggest positive number in (d) above?

Answer:

g) What is the decimal value for the most negative number in (e) above?

Answer:

h) What is the result in binary of negating the largest-magnitude negative number in (e) above?

Answer: ____ ____ ____ ____

Problem 3: Hexadecimal Representation (15 points)

Since writing a long string of binary digits can be tedious, it is often convenient to use the 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 right). Give the corresponding 8-hexit (a 'hexit' = hex 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) 4100_{ten}

Answer: _ _ _ _ _ _ _ _

b) -512_{ten}

Answer: _ _ _ _ _ _ _ _

c) $1011\ 0101\ 1010\ 1010\ 0011\ 0001\ 0101\ 1010_{two}$

Answer: _ _ _ _ _ _ _ _

d) $1101\ 0101\ 0110\ 1011\ 0101\ 0101\ 0100\ 1010_{two}$

Answer: _ _ _ _ _ _ _ _

e) -1_{ten}

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 is 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 and decimal as indicated.

a) $18_{ten} + 25_{ten}$

Operand 1_{two}: _ _ _ _ _ _ _ _

Operand 2_{two}: _ _ _ _ _ _ _ _

Result_{two}: _ _ _ _ _ _ _ _

Result_{ten}:

b) $74_{ten} - 24_{ten}$

Operand 1_{two}: _ _ _ _ _ _ _ _

Negated Operand 2_{two}: _ _ _ _ _ _ _ _

Result_{two}: _ _ _ _ _ _ _ _

Result_{ten}:

Problem 4: Binary Arithmetic (continued)

c) $24_{ten} - 74_{ten}$

Operand 1_{two}

_ _ _ _   _ _ _ _

Negated Operand 2_{two}

_ _ _ _   _ _ _ _

Result_{two}

_ _ _ _   _ _ _ _

Result_{ten}

d) $100_{ten} + 100_{ten}$

Operand 1_{two}

_ _ _ _   _ _ _ _

Operand 2_{two}

_ _ _ _   _ _ _ _

Result_{two}

_ _ _ _   _ _ _ _

Result_{ten}

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

Problem 5: Fixed-Point Binary (16 points)

Using a ***<u>16-bit</u>*** 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) 88.375_{ten}

Answer:

_ _ _ _ _ _ _ _ . _ _ _ _ _ _ _ _

b) -15.03125_{ten}

Answer:

_ _ _ _ _ _ _ _ . _ _ _ _ _ _ _ _

c) 0011 1000 . 0011 0000_{two}

Answer:

d) 1111 1111 . 0000 1000_{two}

Answer: