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

Comp 411 Computer Organization
Spring 2009

Problem Set #1
Issued Tuesday, 27 Jan 2009; Due Tuesday, 3 Feb 2009

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.

Problem 1. Information Encoding (20 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.

(A) How many bits are required to uniquely encode all playing cards in a standard deck of 52? (That is, if each of the 52 cards is to be assigned a unique binary pattern of the same length, how many bits long would it be?)

(B) If all cards with values 2 to 10 inclusive are removed from the deck (all suits), and the remaining cards are each assigned a unique binary pattern of the same length, how long would each pattern now be?

Problem 2. Modular Arithmetic and 2's Complement Representation (80 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?

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.

(B) Please use a 32-bit 2's complement representation to answer the following questions. What's the representation for 0? For the most positive integer that can be represented? For the most negative integer that can be represented? What are the decimal values for the most positive and most negative integers? What do you get if you negate the largest negative integer (give both the binary and decimal values)?

(C) Since writing a string of 32 bits gets tedious, it's often convenient to use hexadecimal notation where a single "hexit" 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-hexit hexadecimal encoding for each of the following numbers:

(C.1) 527_{10}
(C.2) -131456_{10}
(C.3) $00011001111100100000100101011100_2$
(C.4) $11111111111111111111111111110010_2$
(C.5) -8_{10}

(D) 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 and then adding it to the first operand.

(D.1) 41 + 27
(D.2) 92 - 29
(D.3) 29 - 92
(D.4) 100 - 57
(D.5) 97 + (-97)
(D.6) 89 + 89

Explain what happened in the last addition and in what sense your answer is "right".

(E) Fixed-Point Binary: Assuming a 32-bit fixed-point binary representation, where the leftmost 16 bits are the integer part (i.e., before the binary point), and the rightmost 16 bits are the fractional part. Assume the numbers are to be represented in 2's complement. Convert the first two decimal numbers below into binary, and the remaining two from binary into decimal.

(E.1) 123.0125_{10}
(E.2) -16.2_{10}
(E.3) $0000\ 0000\ 1100\ 0101\ .\ 0001\ 0001\ 0000\ 0000_2$
(E.4) $1111\ 1111\ 1100\ 0101\ .\ 0001\ 0001\ 0000\ 0000_2$