Faculty of Science, School of Computer Science

COMP-2650
Computer Architecture I: Digital Design
Fall 2024

Assignment#	Date	Title	Due Date	Grade Release Date
Lec03	Week 03	**Signed Numbers**	Oct. 01, Tuesday Midnight	Oct. 07

The objectives of the lecture (weekly) assignments are to practice on topics covered in the lectures as well as improve the student's critical thinking and problem-solving skills in ad hoc topics that are closely related but not covered in the lectures. Lecture assignments also help students with research skills, including the ability to access, retrieve, and evaluate information (information literacy).

Deliverables

You should answer 2 of the below questions based on your preference using an editor like MS Word, Notepad, and the likes or pen in papers. In the latter case, you have to write and scan the papers clearly and merge them into a single file. In the end, you have to submit all your answers in one single pdf file lec03_UWinID.pdf containing the question ids for the answer. Please note that if your answers cannot be read, you will lose marks. Please follow the naming convention as you lose marks otherwise. Instead of UWinID, use your own UWindsor account name, e.g., mine is hfani@uwindsor.ca, so my submission would be: lec03_hfani.pdf

Questions

(select only 2 questions based on your preference)

1. To represent $(1.5)_{10}$ and $(1.05)_{10}$ in base-2 with no error, how many positions we need? Show the steps.

2. Floating Point. The term floating point refers to the fact that a number's fraction point (e.g., the decimal point in base-10, or binary point in base-2) can *float*; that is, it can be placed anywhere relative to the significant digits of the number. For instance, given n=3 positions in base-10, the decimal point can be placed in:

3 integers and 0 fraction: xxx. → Smallest Unit (precision): 1
2 integers and 1 fraction: xx.x → Smallest Unit (precision): 0.1
1 integer and 2 fractions: x.xx → Smallest Unit (precision): 0.01
0 integer and 3 fractions: .xxx → Smallest Unit (precision): 0.001

Given n=4 positions, put the fraction point in a position for the following numbers in base-2 to minimize the conversion error? Show the steps and all the possibilities if more positions are possible with the same error.

a. $(1.5)_{10}$
b. $(5.5)_{10}$
c. $(10.5)_{10}$

3. Make the following numbers negative if they are positive or make them positive if they are negative. Show the steps for when the numbers are in signed-magnitude and signed-radix-complement.

a. $(E1A01)_{16}$
b. $(380101)_{9}$
c. $(1001011)_{2}$

4. Padding. The padding refers to the fact that one or more digits are added to a number without having an effect, such as changing the value or the sign of a signed number. For instance, adding 0 to the right side of a fraction number does not affect it. Adding 0 to the left side of the integer part of a number does not affect if the number is not in a signed-magnitude or signed-radix-complement. For example, $(1.05)_{10} = (001.0500)_{10}$. Padding is done to make a pair of numbers with an equal size for arithmetic or make a number equal to the size of given positions.

Given the decimal numbers +5 and -5, pad the numbers to 5 positions in the required base when they are in signed-magnitude and signed-radix-complement *without fraction part*.

a. $(?)_{16}$
b. $(?)_{2}$
c. $(?)_{5}$
d. $(?)_{10}$

5. Perform the following arithmetic in signed-magnitude and signed-5's-complement in base-5.

a. 4031 + 0024
b. 4031 – 0024

6. Show that in 2's complement binary system, the highest significant position acts like a sign (not the same) as in the signed-magnitude binary system. Is this true for any radix-r number system? Justify your answer.

7. Give examples that overflow happens for each addition and subtraction in signed-magnitude and signed radix complement in base-2 and base-5. (in total 2×2×2 = 8 examples)

8. Given n is the number of positions in radix-r, how many possibilities are wasted due to the sign position in the signed-magnitude numbering system.