Indian Institute of Technology Kharagpur

Computer Science and Engineering

CS 31007 COMPUTER ORGANIZATION AND ARCHITECTURE Autumn 2020

Date: 17.10.2020 **Maximum points = 70** Credit: 20%

Online Test-3 **Time:** 3:30 PM - 4:45 PM

Instructions:

This is an OPEN-BOOK, OPEN-NOTES, MCQ/Short-Answer type test. For an **MCQ-type** question, please choose one answer from the given choices. Each correct answer will fetch 4 points, incorrect answer will contribute 0 point, and no answer leads to 1 point. For **Short-Answer** type questions, **show your work** illustrating how you have derived your answer. You may use calculators if required. This question paper has three pages.

Submission of answers: Please create a file including **your name, roll-number**, and your **answers**, and submit it to the CSE Moodle Page by **5:00 PM, 17 October 2020.**

---

1. (4 points): MCQ

The carry-bit leaving the most significant position while performing a 2's complement computer arithmetic (addition/subtraction) is observed to be "1". It plays the following role (choose one):

A. It is to be added back to the most significant bit of the result to obtain the correct sum;
B. It plays a role in determining the sign of the result;
C. Overflow occurs when this bit is "1";
D. It is discarded and the result is read without it, provided there is no overflow;
E. None of the above.

2. (4 points): MCQ

You are given an *n*-bit ripple-carry adder that performs addition on two *n*-bit 2's-complement binary numbers. Assume that the two input numbers comprise random strings of 0's and 1's, i.e., they appear on input lines with equal probabilities of 0's and 1's. The probability that an overflow will occur in the result when $n \to \infty$, is closest to (choose one):

A. 1/8, B. 1/4, C. 1/3, D. 1/2, E. None of these

3. (4 points): MCQ

The exponent in IEEE 754 single-precision floating-point number is excess-biased by 127 to (choose one):

A. enhance the range of representation;
B. avoid signed comparison of exponents while performing arithmetic;
C. help in normalization;
D. achieve both A and C;
E. none of these.

4 (4 points): MCQ
In MIPS architecture, assume that register \$t1 and \$t2 hold *two* 2's complement integers *X* and *Y* (32-bit each), respectively. The following MIPS code is now run:

addu \$t3, \$t1, \$t2
sltu \$t3, \$t3, \$t1

The carry-bit generated **out of** the most significant bit position after execution of the first instruction is observed to be "1". Then, the content of register \$t3 at the end of the code in decimal is (choose one):

A. 0, B. 1, C. depends on values of *X* and *Y*,
D. *sltu* instruction may not be executed because of overflow; E. none of these;

5. (9 points): Short-Answer type
Write the largest normalized number that is smaller than 3.0 in the IEEE 754 single-precision format:

6. (9 points): Short-Answer type
Represent the decimal number (- 1/8) in IEEE 754 single-precision format:

7. (9 points): Short-Answer type
Consider the following floating-point number shown in IEEE 754 single-precision format below:

1	0000 0000	1100 0000 0000 0000 0000 000

Write the corresponding number in binary or in decimal.

8. (10 points): Short-Answer type
Consider the following two floating point numbers A and B in IEEE 754 single-precision format:

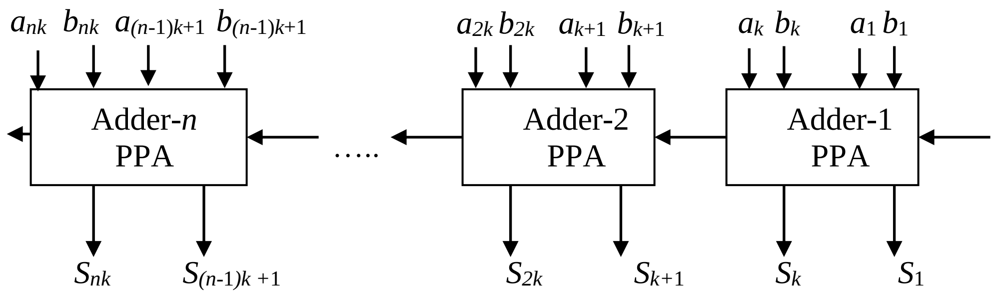
A: 0 0111 1110 1000 0000 0000 0000 0000 000

B: 1 1000 0001 0100 0000 0000 0000 0000 000

The result of floating-point multiplication (A * B) in IEEE 754 single precision format is:

9. (10 points): Short-Answer type

Two (nk)-bit numbers $A = a_{nk}\ a_{nk-1} \ldots . a_1$ and $B = b_{nk}\ b_{nk-1} \ldots . b_1$ are being added using the following scheme. The bits are partitioned into n groups, each group consisting of k bits. For each group of k bits, a PPA (Brent-Kung Parallel Prefix Adder) is employed to compute the sum. These PPA's are then serially cascaded as in ripple-carry adders. Estimate the cost and delay of the proposed adder in terms of n and k.



10. (7 points): Short-Answer type

We would like to perform multiplication of two signed n-bit integers A, B expressed in 2's complement form. The multiplicand and the multiplier can be of any sign. Recall that in Booth's algorithm several addition and subtraction steps are needed depending on the bits present in the multiplier, and at the end, correct result is obtained with the desired sign. Assume that both A and B lie within the range: $-2^{n-1} < A, B \leq (2^{n-1} - 1)$. Can any overflow occur while performing these additions or subtractions during the execution of Booth's algorithm? If not, why not? If yes, how are they handled? Justify your views accordingly.