Delft University of Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
Computer Engineering Lab
2022-2023

Computer Architecture and Organization (CAO) EE3D11

Assignment 3

Deadline Mar 14, 2023, before 08:00!

Note:
Upload a single file having the name "YourFirstName-LastName-HW-3" to the corresponding assignment in Brightspace.
When needed, the answer should be justified. Show clearly which theory is used to find your final answer.

Exercise 1 [3pts]

Based on Neumann architecture, any piece of information (e.g., 32-bits in MIPS architecture) is saved as a string of '0's and '1's. This information can be an instruction, 4 characters, or a number. If it is a number, it can be an integer or a floating-point number. If it is an integer, it can be signed or unsigned. Now with this introduction let's have a look at a word of 32-bits in memory; it is 1100 0101 0111 0000 0000 0000 0000 0000.

- Assume it is an instruction, what is the assembly format of this instruction? [0.5pt]
- Assume it is a single precision number based on IEEE 754 standard, what is it? Answer this question with and without considering the hidden bit in fraction. [1pt]
- Assume it is a signed integer, what is it in hex sign magnitude? [0.5pt]
- Assume it is an unsigned integer value, what is it in hex? [0.5pt]
- Assume it is a set of four characters coded using ASCII, what are these 4 characters? [0.5pt]

Exercise 2 [1.5pts]

I have already given the formulations to figure out what is the sign of remainder and quotient in division for floating point numbers, does it work for integer numbers division also? Can you do similarly for multiplications, both integers and floating-point numbers?

Exercise 3 [1.5pts]

Can you formulate overflow detection for signed integers after addition?

Exercise 4 [2pts]

The revised IEEE 754-2008 standard added a 16-bit floating point format with 5 exponent bits. What do you think is the likely range of the numbers that it could represent? Justify your answer.

1) 1.0000 00 x 2^0 to 1.1111 1111 11 x 2^{31}, 0
2) $\pm$1.0000 0000 0 x 2^{-14} to $\pm$1.1111 1111 1 x 2^{15} , ± 0, $\pm\infty$, NaN
3) $\pm$1.0000 0000 00 x 2^{-14} to $\pm$1.1111 1111 11 x 2^{15} , ± 0, $\pm\infty$, NaN
4) $\pm$1.0000 0000 00 x 2^{-15} to $\pm$1.1111 1111 11 x 2^{14} , ± 0, $\pm\infty$, NaN

Exercise 5 [1.5pts]

If we have three different multiplier hardware (the ones given in slides 15, 16 and 17) for integers of size 64 bits, what will be the cost of each of them in terms of number of adders?

Exercise 6 [5.5 pts]
This exercise deals with 4-bit binary unsigned division. We will use the following octal numbers: A=Dividend= 11 and B= Divisor =4. Assume Quotient register to be 4 bits, Divisor and Remainder registers are 8 bits each.

- Use the algorithm of Figure 3.9 in the book (slide 21 of the lecture) and a table similar to the one used in Figure 3.10 (see below), calculate A divided by B using the hardware described in Figure of slide 21 for 64-bit architecture (which is slightly different from Figure 3.8 of the book); the figure in the slide has a bidirectional signal between ALU and "control test".

  ALU informs "Control test" after subtraction if the result is positive or not and based on that the "Control Test" decides to allow the Remainder Register to update (or not) its contents via "Write" enable.

  Fill in the table for the Quotient, Divisor, and Remainder for each step. You need to provide the DESCRIPTION of the step being performed (shift left, shift right, sub, update Remainder Register, etc.). [3pts]

iteration	Description of the Step	Quotient	Divisor	Remainder
0	Initial values	0000	0100 0000	0000 1001
1				

- The figure of slide 22 (which is that of Figure 11 with an additional control signal) is an improved version of Figure 3.8 (slide 21). Explain the purpose/function of each input and output to/from the control unit for both figures. Indicates clearly if there is a difference in the functions. [2.5 pts]

Signal	Figure 3.8 (slide 21)	Figure 3.11(improved version in slide 22)