

CS 286-001 – Computer Organization & Architecture

Midterm Exam

Fall, 2026 (**PARTIAL SOLUTIONS**)

1:30-2:45 P.M.
October 14, 2026

This exam is a closed textbook and closed note exam. There are **5** questions in this exam. You have 75 minutes to finish the questions. Please write your answers on separated pieces of papers. To avoid grading problems, please staple your solutions in the ascending order in the question number. All the questions in this exam are mandatory (no optional question).

Notice 1: Please carefully read each question and follow the requirement(s) in each question. Failing to follow the requirement(s) may cause minor or major penalties. If you answered anything not asked by a question, you will not get much credit.

Notice 2: It is your responsibility to present your solutions in such a way that Dr. Fujinoki can read your writing. It is assumed that whatever important idea(s) in your solutions is clearly presented or even emphasized.

Student ID (the last four digits): _____

QUESTION #1 (15 minutes) – 4 points for each of the five questions below (**DONE**)

(1) If a sufficient condition is not satisfied, what conclusion can we draw?

No conclusion (we can not draw any conclusion), if a sufficient condition is not satisfied.

(2) What is the essential difference between “li \$a0, 1024” and “la \$a0, 1024” instructions?

“li” loads the given constant (immediate) to a register as a two’s complement integer, while “la” loads the given constant (immediate) to a register as an unsigned integer

(3) What are the primary purposes of “assembly source code files”?

Note: This question does NOT ask: “What are the two primary advantages of learning computer organization and architecture by CS majors?”

Writing programs in an assembly language allows human programmers to:

(a) Apply the best possible code optimizations to programs (programs that fit into small memory space or programs that run as fast as they can).

(b) Fix hard-to-fix software bugs (stack over flow, infinite loops, memory leaks, segmentation faults, and etc.), which are hard or even impossible to be fixed by programmers who do not have assembly programming skills.

(4) Which of the following *lw* instructions is (are) not valid (the one that will cause error in PC-SPIM assembler (select all that apply)?

- (a) `lw $v0, 89`
- (b) `lw $v0, ($a2)`
- (c) `lw $s1, $s2`
- (d) `lw $a1, MY_LABEL` – assume that “MY_LABEL” is properly defined somewhere in your assembly source code file.
- (e) `lw $v0, -812($a2)`

Solution: (a) and (c)

(5) What are the two problems in sign-magnitude integers?

- There are two different binary bit patterns for “0”
- “+N” + “-N” will produce an incorrect output

QUESTION #2 (15 minutes) – 20 points

Suppose that you wrote the following statements in C/C++;

```
int my_int_01;
unsigned int my_int_02;

void main (void)
{
    my_int_02 = 35200;

    my_int_01 = my_int_02;

    printf("my integer: %d\n", my_int_01);
}
```

Assuming that the computer that runs this program is a 16-bit architecture computer (i.e., the registers in its processor(s) are all 16 bits), what number will you see when you print the value of “my_int_01” in the program?

Note that “%d” in the **printf** statement means “print it as a decimal number”.

Show all your work to find the number (i.e., please explain how you got the number).

QUESTION #3 (10 minutes) – 20 points

We have a processor, called Processor A. The processor clock cycle rate and CPI of processor A are:

Processor A: **Clock cycle rate = 850 MHz**
 CPI = 1.6

Now a another new is being developed, called Processor B. The CPI of the processor B is:

Processor B: **CPI = 0.92**

Suppose we have an executable program that both A and B can execute. How can Processor A execute the program as fast as Processor B? Assume that 1MHz = 10^6 Hz (processor cycles).

Show all your work for full credit.

QUESTION #4 (20 minutes) – 30 points, 3 points for each of the 10 questions

Is each of the following statements true or false? Please choose either TRUE or FALSE (no “it depends”). It is your responsibility to clearly indicate which option you chose. For this question, you should answer either TRUE or FALSE below (no need to provide answers in separate paper).

- (1) Given N bits for sign-magnitude integers, the maximum (positive) and the minimum (negative) numbers are $+ (2^{(N/2)} - 1)$ and $- 2^{(N/2)}$.

[TRUE FALSE]

- (2) Fixed-point fraction numbers and sign-magnitude integers have a fixed interval between two numbers that can be accurately represented.

[TRUE FALSE]

- (3) Processors with a lower clock rate (referred as “Processor A”) execute the same binary programs faster than the processors with a higher clock rate (referred as “Processor B”), if A’s CPI is higher than that of B.

[TRUE FALSE]

- (4) When a value (a value meant to human users) assigned to an “*unsigned integer*” variable is copied to an “*int*” variable, the value will never be correctly copied (again, as a value mean to human users) to the “*int*” variable.

[TRUE FALSE]

- (5) Source code files written in a high-level programming language are first translated to assembly source code files and then, the assembly source code files are translated to binary executable files by a compiler.

[TRUE FALSE]

- (6) Binary executable machine codes and assembly instructions are in one-to-one relationship, although assembly instructions and statements in a high-level programming language are in many-to-one relationship respectively.

[TRUE FALSE]

(7) For scalar, super-scalar, pipeline, and super-pipeline datapaths processors, super-scalar, all of them can possibly achieve CPI 1.0 or lower.

[TRUE FALSE]

(8) In two's complement integers, it is possible for a result of a mathematical operation to be in error even if the carry flag is not raised.

[TRUE FALSE]

(9) Each instruction takes less time to execute (the actual time a processor spends for executing one instruction) by a pipeline processor than by a scalar processor (assuming that all other aspects of the processors being the same).

[TRUE FALSE]

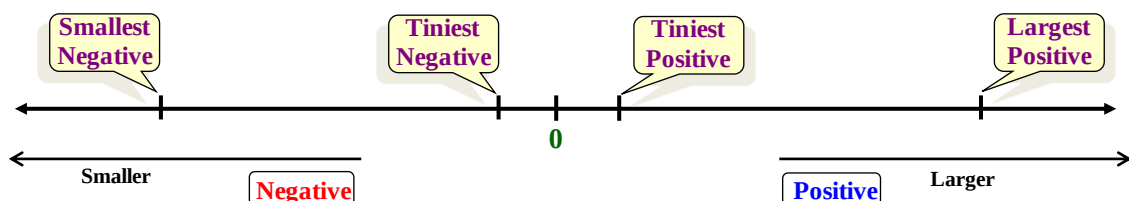
(10) The IEEE-754 floating-point standard uses “the normalized format” for saving one bit in the exponent bits (which will improve the accuracy).

[TRUE FALSE]

QUESTION #5 (15 minutes): – 20 points

IEEE (Institute of Electrical and Electronics Engineers) has another standard for floating-point numbers, called “double precision”. Its only differences are the number of bits for the exponent and the significand bits, and the use of “bias of 1023”. The double precision format uses 11 bits for the exponent and 52 bits for the significand. For the exponent, it applies “bias of 1023”. Note “1111111111” and “0000000000” for the exponent bits are reserved for “exceptions”.

Decimals	Bias of 0	Bias of 127	Bias of 1023	• • •
1024			1111111111	
255	11111111	N. A.		
• • •	• • •	N. A.		
128	10000000	11111111		
• • •	• • •	• • •		
0	00000000	01111111	0111111111	
• • •	N. A.	• • •		
-127	N. A.	00000000		
	N. A.	N. A.		
-1023			0000000000	



Question: In IEEE 754 double-precision floating point numbers, what is the absolute difference between its “second smallest negative number” and “third smallest negative number”? Show all your work for full credit.