

CS286-Computer Organization & Architecture

Quiz Question List (Quiz #4) for September 14th, 2026

The following is a list of possible questions for our quiz on September 14th. Some of the questions will not be asked in the quiz. All the questions that will appear in the quiz will appear exactly as shown below (however, numeric parameters may be changed). The quiz is closed textbook, closed notes and closed neighbors. Note that the questions, which did not appear in this quiz, still may appear in the exams. You will find a solution for these questions during lectures.

- #1: What is the essential difference between “lw” instruction and “la” instruction?
- #2: “li \$t0, (1024)” is an illegal instruction (if you try to assemble that instruction using PC-Spim simulator, that instruction will cause a syntax error). What’s wrong?
- #3: What is the difference between “li \$a0, 1024” and “la \$a0, 1024” instructions? Assume that this computer system is a 32-bit system (i.e., all the registers are 32-bit registers and its ALU can deal with up to 32-bit inputs and outputs).
- #4: “li \$t1, \$t0” is an illegal instruction (if you try to assemble that instruction using PC-SPIM simulator, that instruction will cause a syntax error). What’s wrong?
- #5: Translate the “if-then-else” program structure using MIPS instructions (in the following MIPS assembly program structure (by showing all the necessary missing instructions there)).

```
.text
.globl main

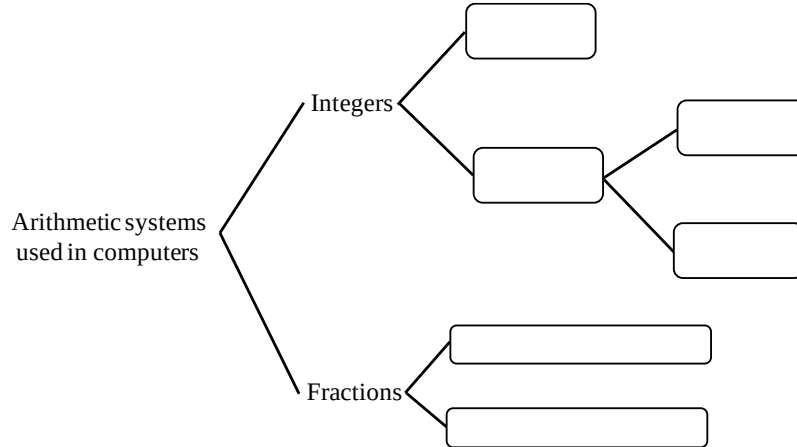
main:      . . . .
           . . . .

           instruction A
           . . . .
           . . . .

if_equal:  instruction B
           . . . .
           . . . .

end_if_else: instruction C
```

#6: Complete the following figure by specifying number systems we discussed in the classroom.



#7: Show “78₍₁₀₎” using unsigned integer format. How many bits are needed?

#8: What are the two major weaknesses in “unsigned integer”?

#9: Show the binary bit pattern for “-5” in 4-bit sign-magnitude integer. Show all your work.

#10: Which numbering system solves the two problems in the “sign magnitude integers”?

#11: What does the following 16-bit two’s complement number mean in the decimal format? Show your calculation.

“1 0 1 1 1 0 0 0 0 0 1 0 0 1 0 1”

#12: What is “Overflow Flag” in a processor (how is it used) for handling operations on two’s complement integers?

#13: Transform the following decimal number to the two’s complement binary number (using the 16-bit format: your processor is a 16-bit architecture processor): -63₍₁₀₎. Show all your work.

#14: What is “Carry Flag” in a processor (how is it used) for handling operations on two’s complement integers?

#15: How will “0₍₁₀₎” be represented using n -bits two’s complement number?

#16: How will “-1₍₁₀₎” be represented using n -bits two’s complement number?

#17: Show the binary bit pattern for “-5” in 4-bit two’s complement integer.

#18: What does the following 16-bit two's complement number mean in the decimal format? Show your calculation.

“1 0 1 1 1 0 0 0 0 0 1 0 0 1 0 1”

#19: What is “Overflow Flag” in a processor (how is it used) for handling operations on two's complement integers?

#20: Assuming 4-bit two's complement integers, will the following calculation cause:

(a) The overflow flag. If yes, show how. If no, show how not.

(b) The carry flag. If yes, show how. If no, show how not.

$$\begin{array}{r} \cdot \cdot \cdot \cdot \quad 0 \ 0 \ 1 \ 1 = +3 \\ + \cdot \cdot \cdot \cdot \quad 0 \ 1 \ 0 \ 0 = +4 \\ \hline \end{array}$$

#21: How will be an eight-bit two's complement integer be converted to a 32-bit two's complement integer? Explain the procedure.

#22: What is the problem in representing fractions using the “fixed-point” method?

#23: What does “floating point” in “floating point fraction numbers” mean?