

CS286-Computer Organization & Architecture

Quiz Question List (Quiz #3)

for September 7th, 2026

The following is a list of possible questions for our quiz on September 7th. 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.

***** the left-over topics from Week #1 *****

- #1: What does “efficient” in “efficient computer programs” mean? Mention at least two.
- #2: What is “Moore’s Law” (a specific definition of “Moore’s Law” is needed)?
- #3: What does “MIPS” stand for?
- #4: What are “super computers” (how fast, what purpose and how large)?
- #5: What is “Von-Neumann Architecture”? (Your HW – our textbook or some search engines in the Internet)
- #6: What does “performance” in computer system mean (at least four examples)?
- #7: What are “registers” in processors?
- #8: How are registers in processors used when a binary executable program is executed?

***** the topics from Week #2 *****

- #9: What is “PC-SPIM”?
- #10: Why do we need “jr \$31” at the end of an assembly program?
- #11: What do “lw” instruction does (performs)?
- #12: What does “()” for the second argument for “lw” instruction means?
- #13: What do “sw” instruction does (performs)?
- #14: What do “labels” do?

- #15: What is the primary advantage in using “labels”?
- #16: What is the essential difference between “lw” instruction and “la” instruction?
- #17: What are “system calls” in “MIPS Simulator”?
- #18: “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?
- #19: 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).
- #20: “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?
- #21: 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]

```