

**Study Questions for Patterson & Hennessy**  
Chapter 5 through top of page 315.

As your study guide for the first part of Chapter 5, I have supplied some questions for your consideration. You are looking at Figure 5.24 on page 314, which would be supplied on any test. A single-cycle data path on the main integer processor is assumed; no “delayed branches” is assumed (as promised on p. 297).

---

- Color a data wire.
- Color a control wire.
- Shade in something that is changed at the leading edge of the clock pulse. [Answer: Anything that has state, so Instruction Memory, Data Memory, or Register File.]
- Put an x on a wire whose value is “don't care” during an add instruction.
- Write a slash on a line which represents more than one wire, and near the slash, write the number of wires that the line represents. For example,

----- becomes -----/-----<sup>32</sup>

to indicate that the line represents a 32-bit wide datapath.

- Put a slash on all wires on the diagram which represent n-bit wide datapaths or control paths, n>1.
- Does any instruction write to two places in the same clock cycle? If so, name one. If not, say why not. [This is a variation of the “Check Yourself” question on p. 289.]
- Does any instruction read from two places in the same clock cycle? If so, name one. If not, say why not.
- There are more than 9 wires leaving the Control oval above and to the left of the register file, because some lines like ALUOp are more than 1 bit wide. Explain how it is that only 6 bits, Instruction [31:26], comes in, but more than 9 wires go out. [Page 312 addresses this question.]
- Which instruction takes the longest time to execute? Remember that RAM is slower than the register file, which in turn is slower than simply updating the PC, so the answer is either LW or SW. Which do you think it is and why?
- (Reverse question:) Here is a colored-in version of Figure 5.24. What MIPS instruction does it represent?
- From carefully observing the diagram on p. 314, circle the letter of all true statements. (Answers upside down below.)
  - a. The function field in an R-format instruction is only used to control the ALU.
  - b. 4 is added to PC regardless of whether the next instruction in memory is executed, there is a branch to another instruction, or there is a jump to another instruction.
  - c. RegWrite is asserted in a SW instruction.
  - d. If Branch is asserted, then the instruction must have been BEQ.
- The wire from the ALU control oval to the ALU is three wires if we do not include the NOR instruction. Why will adding just one more instruction like NOR to the ALU demand a fourth wire?

a. T b. T c. F d. T. Why would it be false to require both Branch and Zero to be asserted for an instruction to be a BEQ instruction?