

Study Questions for Patterson & Hennesey, Section 2.4

Section 2.4

Because the text is clear, these study questions will focus on testing your understanding of the section and lecture six. Answers appear upside down at the bottom of this sheet.

- a. How much memory can be addressed at all by a 32-bit wide MIPS machine?
- b. How much memory can be addressed by **lw** or **sw** if the base address is kept fixed?
- c. In what way is **addi** more like **lw** and **sw** than it is like **add** and **sub**? In what way is **addi** more like **add** and **sub** than it is like **lw** and **sw**?
- d. **Add** and **sub** both have a 0 op field. How does MIPS tell the difference between the two instructions?
- e. Using the table on p. 67 of your text, answer the following two questions. (Or using the green insert in the front of your text, because that's what you'll be allowed to use on the exam.)

(1) Assume that this bit pattern is an instruction. Write it in MIPS assembly language.

000000 01000 10000 00010 100000

(2) Here is a MIPS assembly language instruction. Write it in machine language. I was vague in lecture about which of the two registers goes first. See p. 67 for example. The source register **rs** and target **rt** are different for **sw** and **lw**, as their semantics suggests.

sw \$t0, 4 (\$s0)

Answer key:

a. 2^{32} KB. That's about 4 GB = 4,000,000,000 B.
 b. 2^{16} KB, which is about 64,000 KB although because signed offsets are allowed, that's only 2^{15} (=32,768) in either direction, addresses as high as $(2^{15}-1)$ or as low as $-(2^{15}-1)$, which is to say offsets ranging from $+0$.. $32,767$ and from $-32,767$.. -0 .
 [Idea to discuss later in the course: In terms of bytes, does this give us two byte patterns, one which represents $+0$ and one which represents -0 ?]
 e.(1) add \$v0, \$t0, \$s0
 because op=0, funct=32 means **add**; so it's add \$2, \$8, \$16 [Remember where rt is.]
 e.(2) 101011 10000 01000 000000000100
 op rs rt offset
 For lw and sw, rt is first named argument whether coming or going, rs is second named argument. Since \$t0=0x8=8 and \$s0=0x10=16, we put 8 in rt and 15 in rs.
 sw is op=43 or 0x2B, and the offset=0x4; so those are placed in the word as shown.