

332:333 Computer Architecture and Assembly Language Lab

www.ece.rutgers.edu/~yyzhang/fall03

- **Related Reference**

The MIPS programs assigned in this lab course are all meant for execution on a SPIM simulator, described in detail in Appendix A of the text (*Computer Organization & Design – The Hardware/Software Interface Second Edition*) for the course (14:332:331) that accompanies this laboratory.

MIPS is introduced starting in section 3.2 of the text. Section 3.3 through 3.8 gradually detail many MIPS properties. In particular, 3.3 explains the alternate notations for MIPS registers and the format used for memory operands; 3.4 shows how MIPS instructions look in binary; and 3.5 describes the use of some of the branch and jump instructions. Summaries like that on page 131 are particularly useful. 3.6 adds more register notation and discusses procedure calling and the use of the MIPS stack and stack pointer \$sp. The summaries on pages 140 and 141 are especially helpful. 3.7 reminds you about ASCII codes; and 3.8 describes immediate operands, PC-relative branch and jump offsets, and the special register \$at (page 147). Note the illustration on page 152 and the summary on page 155. 3.9 introduces the important concept of pseudoinstructions. A pseudoinstruction can be treated by the programmer as if it were part of the MIPS repertoire. In fact, it is not. It is converted, by the assembler, into a sequence of one or more non-pseudoinstructions (i.e., those that are part of the MIPS hardware's instruction set). 3.9 also reminds you about hex notation. Finally, section 3.10 and 3.11 show several examples of real MIPS assembly code. Note the two tables on page 189. Many of the exercises at the end of Chapter 3 test your understanding of MIPS programming practices.

Sections 4.2 through 4.4 review signed numbers, negation via the two's complement, the important concept of sign extension (page 214), binary addition, and bitwise logical operations. Note again the tables on pages 219, 224 and 228. MIPS integer multiply and divide instructions are summarized on pages 264 and 273. Another good summary table appears on page 274. MIPS floating-point instructions are briefly introduced on pages 288 and 290. Additional useful summary tables appear on pages 291, 299, 309 and 310.

Finally, in Appendix A, sections A.1 through A.4 review assemblers and assembly language, introducing some significant new notions – like assembler directives (page A-14). Sections A.5 and A.6 describe how MIPS memory is subdivided and review MIPS procedure calling conventions. Notice the register-naming summary on page A-22 and its accompanying table on page A-23. (They add the last two register names \$k0 and \$k1.) Skipping sections A.7 and A.8, you reach section A.9, where the SPIM simulator user manual begins. After you have studied it, through the first part of section A.10 (ending at page A.53), you will be ready to attack the programming assignments given in this course. Notice that the rest of Appendix A (starting on page A-53) contains a detailed description of each of the MIPS instructions and pseudoinstructions that are available to you.

For the additional SPIM information, please refer to 331 course web page www.ece.rutgers.edu/~yyzhang/fall03, and then click “useful links”.

- **SPIM simulator availability**

Many PCs containing already-loaded simulators PCSpim (the graphical/windows version) are available in the lab in EE building. You may also download and install PCSpim (it is free!!!) from SPIM homepage (www.cs.wisc.edu/~larus/spim.html).

To active PCSpim on a Windows machine, double-click on the PCSpim icon. To get a display that looks like that on text page A-41: First, using the Window menu, omit the Toolbar and Status Bar, and then maximize the PCSpim window; Second, if necessary, close the Messages and Console windows. (They can be reopened at any time. During a run, the latter automatically opens whenever input from the keyboard is called for.)

SPIM is also available on workstations for both EDEN (ARC Bldg) and ECE (EE Bldg) systems. Please type `/user/local/spim/bin/xspim` for the graphical spim.

As you can see from the sample display on text page A-41, SPIM shows the hex value of 32-bit words with the most significant byte (bits 31-24) on the left and the least significant byte (bits 7-0) on the right. Notice that every program text segment begins, at memory location 0x00400000, with eight words of default start-up code that ends with a call (jal) to the global label main followed by an exit syscall. Your program will normally begin (with the main routine) at address 0x00400020. Further, your data area will usually begin at address 0x10010000.

- **Syllabus**

Week 1	Dates	Events
1	09/01 – 09/05	Lab cancelled
2	09/08 – 09/12	Syllabus handed out; Get familiar with the SPIM simulator;
3	09/15 – 09/19	Project 1 handed out;
4	09/22 – 09/26	
5	09/29 – 10/03	Project 1 Due; Project 2 handed out;
6	10/06 – 10/10	
7	10/13 – 10/17	
8	10/20 – 10/24	Project 2 Due; Project 3 handed out;
9	10/27 – 10/31	
10	11/03 – 11/07	
11	11/10 – 11/15	Project 3 Due; Project 4 handed out;
12	11/17 – 11/21	
13	11/24 – 11/28	
14	12/01 – 12/05	
15	12/08 – 12/12	Project 4 Due;

- **About the projects**

- Except the week when the projects are due, you are not required to attend the lab. You can download the project from course webpage.
- During the week when the lab is due, you need to attend YOUR session and the TA will grade your project.
- No more than 2 people can work in a group. You may also choose to work alone.
- Proj 1: 20%, Proj 2: 25%, Proj 3: 25%, Proj 4: 30%