Computer Organisation COMP2008

Lab Sheet 4 (starts session week 5, due in week 6)

Student Name and Number	
Date, Grade and Tutor signature, **max mark 3**	

Keep this cover sheet marked and signed by the tutor.

1. Preparation [Total max. mark: 1]

The main goal of today's lab is to become familiar with programming arithmetic operations with operands available in registers, and in memory locations. Study the lecture notes, and relevant sections of the textbook, at least: Ed2: 3.1, 3.2, 3.3, 3.11, or Ed3: 2.1, 2.2, 2.3, Ed4: 2.1, 2.2, 2.3, 2.14, Ed5: 2.1, 2.2, 2.3, 2.14, Ed6: 2.1, 2.2, 2.3, 2.14

1. Describe below in writing two basic structures used in assembly programs (as interface) to organise and access data in memory and describe the difference [**0.5 mark**].

2. Illustrate both methods with few lines of sample code [**0.5 mark**].

   Hint: For answering this question, let's focus on the static data structure in memory. Refer to slides 9-13 in lecture 2 notes for concepts regarding data allocation in memory and methods for addressing memory locations of data. Also work on this question together with *Workshop Tasks* -> *Item 2* (below in this lab sheet) to examine the methods for data in memory. Also recall the corresponding essential methods in high-level languages for addressing data in memory.

2. Workshop Tasks [Total max. mark: 2]

1. Create a subdirectory for lab 4, and get assembly language files *simplearray.s* and *simplemem.s*

2. Invoke the PCSpim simulator. Run the program *simplearray.s*, and then *simplemem.s*. Experiment with these programs to fully understand how they work. Specifically analyse the assembly language instructions (not pseudoinstructions) in the programs which access memory for loading and storing. **Recall the Preparation questions:** How do both programs gain access to data memory locations? Notice the difference.

3. **Task I:** [**1 mark**] write your own program which calculates and displays in the console window the values of the following expression:

   $$X = Z[3] – Z[5]$$

   where: the appropriate initial value of the first 10 elements of array Z are defined in memory (you need to define them). The result (variable X which you are calculating) also has to be stored in memory. Thus you need to define two arrays: Z (at least 10 words) and X (one word).

   Make sure that your program is properly documented, and it uses the appropriate registers, according to MIPS conventions. You may find it useful to write the programs in C, or any other high level language first (you do not have to write anything in C, do it only if it helps you).

   Demonstration: Demonstrate to the tutor your running program, and be prepared to explain the code.

4. **Task II:** [**1 mark**] Write a program which calculates and displays in the console the values of the following expression:

   $$Z[12] = Z[k] + Z[k\text{+m}]$$

   Make appropriate definition of array Z. The program should read variables ***k*** *and* ***m*** from the keyboard (refer to the sample program *simplemem.s,* which demonstrate how *p* and *q* are read from the keyboard).

   Demonstration: Demonstrate to your tutor your running program, and be prepared to explain the code.

3. Assessment notice

When you ready, present to the tutor a printed copy of your program source code, with your name and student number included in the comments (#…), and typed or neatly written answers to all questions listed in the lab sheet. Your tutor may decide to keep the source code printout, but you should keep marked and signed cover sheet.

Warning: Any source code duplicated amongst students will result in a zero mark, and possible further action according to the WSU policy on plagiarism.