Computer Organisation COMP2008

Lab Sheet 2 (starts session week 3, due in week 4)

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 goals of today's lab: analyse simple program performing arithmetic operations, learn how to debug programs.

1. Study the lecture notes. Also use **HP_AppA.pdf** (on the website) as a reference. Study **at least** the following sections from the Patterson and Hennessy "Computer Organization and Design" textbook:
   - If you have Edition 2: Chapters 3.1, 3.2, first part of 3.3, or:
   - If you have Edition 3: Chapters 2.1, 2.2, first part of 2.3, or
   - If you have Edition 4: Chapters 2.1 to 2.5
   - If you have Edition 5: Chapters 2.1 to 2.5
   - If you have Edition 6: Chapters 2.1 to 2.5

   /* Text readings are listed in Teaching Schedule and Learning Guide */

General Data	UnitOutline | LearningGuide | Teaching Schedule | Aligning Assessments |
Extra Materials	ascii_chart.pdf | bias_representation.pdf | HP_AppA.pdf | instruction decoding.pdf | masking help.pdf | PCSpim.pdf | PCSpim Portable Version | Library materials

2. Explain stored-program concept [**0.5**].

3. Consider three types of information: ① *calculated values*, ② *ASCII characters*, and ③ *instructions* to be executed. Explain what out of the three types of information (e.g. some of them; all of them; or none of them) can be found in the memory of MIPS computers? Justify your answer [**0.5**].

   Note: if the question is unclear, you can alternatively answer the following multiple choice question: Consider three types of information given above, what can be found in the memory? (circle the answer and justify your answer).

   a) ③ **only**
   b) ① **and** ② **only**
   c) **Mixture of** ① ② ③
   d) **None of them**

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

Please answer the lab Questions listed below in writing, print or neatly write your answers:

1. Create a subfolder for lab 2, and download the assembly language file: *simplecalc.s*

2. Open *simplecalc.s* in a text editor and analyse the program. Identify its text segment, data segment, and the registers used in it. Run the program in PCSpim, and observe its results in the console window.

   Question (0.5 marks):
   a. Write on a piece of paper the formula for the calculation performed by the program *simplecalc.s*, and perform this calculation manually, on paper, using number values defined in *simplecalc.s* Note: The formula is a mathematical expression with variables used for the calculation performed by the program. For writing the formula, you should first introduce/declare some necessary variables based on the calculation performed by the program. Variable and expression are standard and common terms in programming context. It's assumed you have understood them from learning Programming Fundamentals.

   b. write what the program prints in the console window (before the errors corrected in d).

   c. are the results from (a) and (b) identical? If not, explain why.

   d. two lines of the code are marked "ERROR". Correct the errors in both lines.

3. Continue to work on *simplecalc.s*

   Question (1 mark): add **ALL** missing comments (**everywhere you see "__??"**). You need to understand what the program does (the formula for the calculation performed by the program), and what is the purpose of each line of the code. A good comment should explain purpose of an instruction, which is clear and concise for coding implementation. Do NOT just copy the description of instructions from the textbook. Below examples of a good and a bad comment:

   ```
   addu    $ra, $0, $s7      # restore the return address (this is good, descriptive comment)
   addu    $ra, $0, $s7      # register $ra gets contents of register $s7 (this is bad comment which does
                             # not explain why this operation is performed, what is its purpose)
   ```

   Expressions are straightforward way to comment on most calculation steps (for clear semantic meaning and easy implementation).

4. Still with program *simplecalc.s*

   Question (0.5 marks): Write a list of all registers used in the program, and write next to each register what was its role (for example: $ra – used to store return address). Also, compare the registers used in the program with "Policy of Registers Use Convention" table (see the lecture notes and the textbook). Is the program using the registers in accordance with the convention?

Practical advice: keep all lab work for future reference. It helps to analyse code if you print it from TextPad or other text editor using smaller (8-9pts), monospaced ("fixed pitch") fonts, such as **`Courier`**. It is best to fit one line of code with comments in one line of print, and include the line numbers.

3. Assessment notice

When you are 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 (note that amount of questions vary per lab).

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

Your tutor may decide to keep the source code printout, but you should always keep the cover sheet marked and signed by the tutor.