CS341, Computer Architecture Lab, Lab 05

Goals

1. Learning to refer/search specification sheets *quickly*
2. Understanding the SPIM exception handler, making modifications

Instructions

1. These exercises are to be done individually.
2. While you are encouraged to discuss with your colleagues, do not cross the fine line between discussion *to understand* versus discussion as a *short-cut* to complete your lab without really understanding.
3. Create a directory called <rollno>-<labno>. Store all relevant files to this lab in that directory.
    a. In the exercises, you will be asked various questions. Note down the answers to these in a file called "<rollno>-<labno>.txt".
    b. In some parts of the exercises, you will have to show a demo to a TA; these are marked as such. The evaluation for each lab will be in the subsequent lab, or during a time-slot agreed upon with the TAs. For this evaluation, you need to upload your code as well.
    c. While submitting (on moodle), you have to create a tar.gz or zip of the entire <rollno>-<labno> directory in which all your relevant files reside.
4. Before leaving the lab, ensure the following:
    a. You have signed the attendance sheet
    b. You have uploaded your submission
5. Things to ensure during TA evaluation of a particular lab submission:
    a. The TA has looked at your text file with the answers to various questions
    b. The TA has given you marks out of 10, and has entered it in the marks sheet, with his/her signature
    c. You have counter-signed the above-mentioned marks
6. You have to use the MIPS conventions, unless mentioned otherwise.
7. **House points:** Some questions carry house points. For these, you are allowed to work along with others, but only if *all* of those working together have completed the lab and shown it to the TA. (You can work on it individually, even before finishing the lab completely, but only if you care about house points more than your own marks!). All questions related to house points have to be shown to the instructor (Bhaskar).

Understanding the SPIM exception handler code

- The SPIM exception handler “exceptions.s” is well commented. Read and understand the code. You may have to refer to the MIPS instruction set reference for instructions you do not understand.

- **Demo to TA [1 mark]:** explain the code briefly to the TA.

- **Demo to TA [1 mark]:** Make a copy of “exception.s” in your directory somewhere and call it “exceptions-<yourname>.s”. Learn to run spim or xspim with your own copy of the exception handler. Show this to the TA. *Hint: you may have to refer to the SPIM command line options.*

Printing the data-address of a mis-aligned store

- Now add to the exception handler code appropriately, to print the mis-aligned data address, corresponding to a store word instruction which causes a mis-aligned store exception. *Hint: you may have to refer to the MIPS data sheets and search in the appropriate place, this is part of the excercise.*

- **Demo to TA [4 marks]:** Write some code in a file called “lab5.s”, such that it causes a store-word mis-aligned exception. Run this code with your modified exception handler, and show that the mis-aligned data address is printed.

Correcting a mis-aligned store

- You have to now change the exception handler further, such that whenever the store is from register $t7, and a mis-aligned exception happens, the handler should store the 4 bytes onto memory despite the mis-alignment. *Hint: you may have to use more than one memory store instruction in the handler for this.*

- **Demo to TA [4 marks]:** show your exception handler code, and how it works.

House points

- **5 house points:** This is to be done within lab hours only. Add to the code such that a load onto $t7 is also handled correctly in spite of an exception.

- **10 house points:** add to the code such that the exception handler can handle store from *any* register. *Hint: you may be better off writing a script to generate the exception handling code.*