CS231, DLDCA Lab, Lab 07

Goals

1. Learn to use the WinMIPS64 pipeline simulator for the MIPS64 architecture
2. Understanding various kinds of hazards and stalls
3. Understanding data forwarding

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 "answers.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.
4. Before leaving the lab, ensure the following:
    a. You have marked attendance on SAFE
    b. You have uploaded your submission on BodhiTree, and downloaded and checked if the submissions is right
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
6. You have to use the MIPS conventions, unless mentioned otherwise.

Using the WinMIPS64 simulator

- The WinMIPS64 simulator is a MIPS64 pipeline simulator. The MIPS64 architecture is the 64-bit version of the MIPS architecture. The MIPS64 5-stage pipeline is similar to the MIPS/DLX 5-stage pipeline. The MIPS64 instruction set reference is given to you, please refer to it to write MIPS64 assembly code.
- winmips64 is a windows program, but it can run on Linux using "wine /path/to/winmips64.exe", on the lab machines; "wine" is windows emulator. Refer to the winmips64 tutorial for a brief introduction to winmips64.

- The winmips64 simulator uses a simple "memory-mapped I/O" scheme for input/output, in case you need it. You can look at the testio.s to understand. Such I/O should not be required for the exercises in this lab, but you may need it to understand pre-written code (e.g. if you try to step through factorial.s).
- **Demo to TA [1 mark]:** load and run the "factorial.s" program. Remember, you will find that at some point it is asking for terminal input, so be sure to open the "terminal" sub-window in winmips64.
- **Question [0.5 marks]:** What branch prediction technique does the simulator use? Substantiate your answer. You can use any program other than factorial.s for this, if convenient.
- **Question [0.5 marks]:** In which stage does the branch instruction require its inputs? Substantiate your answer.

Identifying data forwarding, stalls

- You can now use simpler DLX code, like the one in "winmips64-lab.s". In most of these cases, you have to show the corresponding cycle diagram (in the "clock cycle diagram" window in winmips64).
- In the "pipeline" diagram (in one of the WinDLX windows), you can see that apart from the EX, there are three other functional units: multiplier, FP adder, DIV.

1. Write DLX assembly code which causes (a) one or more data stalls, and (b) one or more control stalls.
2. Add to the above code such that there is now a stall in the ID stage of a branch instruction.
3. Add to the above code such that there is data forwarding from the EX stage to the ID stage for some pair of instructions.
4. Add to the above code such that there is data forwarding from the MEM stage to the ID stage for some pair of instructions.
5. Add to the above code to cause the maximum possible stall between a pair of instructions.
6. Add to the above code to cause data forwarding between two instructions that are as far apart as possible.
7. Add to the above code to cause a WAW stall.
8. Add to the above code to cause a structural stall.

- **Demo to TA [1 x 8 = 8 marks]:** Show all the above code and show the "clock cycle diagram" and indicate the above mentioned features. Also show the "statistics" window.