Assignment 5: MIPS Pipeline

This assignment is based on MIPS pipeline simulation using WinMIPS64 simulator. The simulator and its documentation/tutorial is available at http://indigo.ie/~mscott/. Since the simulator is based on windows, you will have to use something like Wine (https://www.winehq.org) to run it on ubuntu or macOS.

Submission Instructions: Make a directory named with your roll number. For each problem *i*, write your code (if required by the problem) in a file named *i*.s and include them in your submission directory. Also write answer to each question just below it and include this file in your submission. Finally submit a tar.gz i.e., [roll-no.].tar.gz (as follows) on moodle:

```
[roll-no.]
   |----1.s
   |----2.s
   :
   |----n.s
   |----a5.docx (or any other format as you like)
```

1. Write a snippet of assembly code and use the simulator timing diagram to deduce whether each of the following is pipelined in the execution step and, if so, the number of stages involved in execution of each operation.
(i) Integer multiply
(ii) integer divide
(iii) FP add
(iv) FP multiply
(v) FP divide

2. Configure the pipeline to (a) suppress forwarding and (b) enable forwarding for different snippets of code. Write your observations i.e., differences between the case when forwarding is enabled and when it is disabled.

3. Write an independent 2-instruction sequence to maximize the time between the start of the first instruction and the end of the next. Repeat for a 3-instruction sequence i.e., maximize the time between the start of the 1st instruction and end of the 3rd.

4. Write a snippet of code with the potential to create a structural hazard in the multi-cycle execution case. Is this resolved in the pipeline and, if so, how?

5. What is the time to execute a pair of multiply instructions with the second having a RAW dependency on the first, with and without forwarding?

6. Write a snippet of code to determine the branch penalty in each case – (a) branch not taken and (b) branch taken? What are the penalties in each case and what do you conclude about the stage in which the branch condition is evaluated and the stage in which the target address is computed?

7. Experiment by configuring the pipeline to support a branch delay slot. Show a snippet of code in which this feature is useful and one where this feature is not. In case the feature is useful, show how.

8. Write a snippet to illustrate a potential WAW hazard. Is this situation handled in the pipeline and, if so, how?

9. Can a WAR hazard exist in this pipeline? If so, how? If no, why not? Explain using a code snippet.

10. It is required to sum 12 FP numbers residing in registers f1 through f12. Write the snippet of code to perform the summation with the minimum number of clock cycles. What is the minimum time required? Assume that the final sum should be in f13 and that the contents of registers f1 through f12 should not be destroyed.