Computer Design and Organization

Project Part II

In Part II of the project, you will introduce a second pipeline according to the specifications below.

Your processor remains a single issue, in-order (in the sense described below) processor. However, there are now two pipelines, one for load-store instructions and one for all the other instructions. The two pipelines share common IF and ID stages and:

1. The load-store pipe has 3 other stages:
   - An ADDRESS GENERATION stage to compute the address of the load-store
   - A MEM stage to access data memory
   - A WB stage for storing results of load instructions in the register file
2. The second pipe has two other stages
   - An EXE stage to perform arithmetic-logical instructions and branch completion
   - A WB stage for storing results in the register file
3. The register file is multi-ported allowing two reads and two writes in the same cycle as long as the two writes are to two different registers (i.e., the processor completes instructions in "pseudo-order" since a load and a following independent arithmetic instruction can complete at the same time). Of course writing to the same registers makes little sense for consecutive instructions but you should still check for it.

Note that:

1. The ID stage is more complicated now since it needs to schedule the operations on one pipe or the other. (You can schedule the operations to both and flush the one which does not make sense or schedule the operations to only one according to the opcode.)
2. Forwarding is more extensive (you must check for forwarding within one pipe and from one pipe to the other).
3. Instruction fetch should remain essentially the same as in Part I. There should be minor modifications to stalling and branching.

Part II is due Friday March 10