1) Give 2 reasons why the speedup from pipelining is less than the number of stages in the pipeline. Refer to the Hrishikesh paper if necessary.

2) If we can dynamically translate the ISA (either in software, like Transmeta, or in hardware, like the Pentium 4), does ISA design matter any more? Explain why or why not.
3) In the following code, show the dependences and label what kind each is. Then, to the right of the code, show how a dynamically scheduled processor (e.g., MIPS R10K) might use renaming to eliminate the false dependences. Assume that all registers are initially mapped to the physical register of the same name (e.g., r4 is mapped to l4) and that the processor has 20 logical registers at its disposal.

```assembly
add r4, r1, r2
sub r4, r4, r3
mul r3, r4, r1
```

4) Why do SPARC processors (from Sun Microsystems) still use register windows to enable caller/callee procedures to communicate with each other, even though almost everyone now agrees that they’re a bad idea?
5) Assume that a processor can dispatch, on average, 3 instructions per cycle. Assume that all instructions take 3 cycles in the pipeline before dispatch (for fetching) and 5 cycles from beginning dispatch to beginning to execute (inclusive) and 14 cycles from beginning to execute to the commit stage (inclusive).
(a) On average, how many instructions are in the ROB (assuming the ROB is infinite)?

(b) How big would you make the ROB? Explain your answer.

6) Compare and contrast the Intel P6 (which combines the reservation stations with the ROB, much like Sohi’s RUU) and the MIPS R10K, in terms of:
(a) How they rename registers to avoid WAR and WAR hazards. Hint: how are in-flight instructions tagged?

(b) How they achieve precise state and how they recover to a precise state upon an exception or mis-speculation.
7) Compare having one ALU that is twice as fast as normal to having 2 ALUs at the normal speed, in terms of performance, amount of hardware, and power (refer to Mudge’s power paper if necessary). If any part of your answer depend on aspects of the software being run (e.g., available ILP), explain how.

8) Name 3 of the toughest problems currently facing microarchitects. Make sure that the problems you specify are non-overlapping.