CSE 240A Winter 2012 Assignment 2

Assigned 1/27 Due Thursday 2/2 (In class at start of lecture)
Homework should be typed. Do not forget to put your name and staple all sheets together.

1. Show forwarding and stalls for the following code, assuming the 5 stage MIPS pipelined machine discussed in class including all reasonable forwarding logic. Show it the way we did in class, with IF ID EX M WB, and arrows and bubbles. How many cycles does this code take to complete?

   lw R3, 100(R5)
   add R6, R3, R2
   sub R9, R3, R8
   lw R1, 2000(R9)
   add R5, R4, R3
   addi R7, R1, #8

2. What is the steady-state CPI (ie, assume the branch is taken very many times) of the following code, if we assume the 5-stage MIPS pipeline, but with no support for forwarding (except reg-file forwarding). Also assume no branch delay slot, but rather the "assume branch not taken" approach.

   loop: lw R1, 100(R2)
        addi R1, R1, #1
        sw R1, 0(R2)
        addi R2, R2, #4
        sub R4, R3, R2
        bnez R4, loop

3. What is the steady-state CPI for the following code (note that they are the same code as above but a small trick have been played to fill the branch delay slot). Assuming normal forwarding logic. Also, assume a branch delay slot. Show exactly how you arrive at your answer.

   loop: lw R1, 100(R2)
        addi R1, R1, #1
        addi R2, R2, #4
        sub R4, R3, R2
        bnez R4, loop
        sw R1, -4(R2)

4. Another way to fill the branch delay slot is to try to re-order instructions between the start and end of the loop. For example, in some cases you can move instructions from the start of the loop to the end of the loop. Does this apply to the code in problem 2? If so give the new code. Does the CPI improve over the code in problem 3?

5. Assume code where 20% of instruction are conditional branches and 5% are jumps and procedure calls. Assume an alternate scalar pipeline where the branch target is resolved at the end of the second pipeline stage, and the branch condition is resolved at the end of the third cycle. Also, assume branches are taken 60% of the time, and that we have a useful instruction in the branch delay slot 75% of the time. What is the CPI of this code if we only experience stalls for branch hazards? Assume the ISA uses branch delay slots for jumps/procedure calls as well as conditional branches. Also, assume the architecture takes an "assume branch taken" strategy.

6. Repeat problem 1, but this time assume a 6-stage pipeline, with memory access requiring two stages (IF ID EX M1 M2 WB). The result of the memory access is not available until the end of M2. How many cycles does the code now take?

7. Circle all of the instructions in the following code that could safely be moved into the branch delay slot (replacing the nop). Assume other instructions cannot be reordered or changed. Assume you know nothing about the missing code (the ...). Assume no direct path from the fall through code to the taken path.

   add R1, R2, R3
   sub R5, R2, R3
   and R7, R5, R2
   lw R8, 1000(R5)
   beq R7, R15, label:
nop
add R6, R7, R5
sub R5, R7, R5
lw R9, 2000(R2)
...
label: add R9, R7, R5
addi R5, R0, #0
...

8. Using the web, find the number of pipeline stages for the following processors: Intel Nehalem, Intel Core 2 Duo, AMD Opteron, AMD Phenom. Make sure to mention your source.