

Indian Institute of Technology, Kharagpur
Department of Computer Science and Engineering
Mid-Semester Examination

High Performance Computer Architecture (CS60003)

Time=2 Hours

Max Marks=65

Important Instructions:

- Answer all questions.
 - No clarifications to any question shall be provided. In case you have any queries, you can make suitable assumptions, but please write down your assumptions clearly.
 - All answers should be brief and concise. Lengthy and irrelevant answers will be penalized.
-

1. What is the asymptotic prediction accuracy of a two-bit branch predictor on the following repeating pattern for a certain branch? ... **NNTNNTNNTNNTNNTT**... . (length of repeating segment =17) [3]
2. Assume that the cost of a processor using a simple 5 stage MIPS pipeline is 25% of the total cost of a computer system. The disks, main memory, power supply and enclosure make up the other 75% of the total cost. It is now proposed to increase the speed of the processor by a factor of 10 using a superscalar design approach. But this will increase the cost of the processor by a factor of 10. Further, simulation studies show that the superscalar processor would have wait on the average 30% of time for I/O. On the other hand, the original pipelined processor stalled only 10% time for I/O. From a cost/performance viewpoint, is increasing the speed by a factor of 10 desirable? Assume that in both the processors there are no stalls on account of data or control hazards. Justify your answer with a quantitative analysis of the two computers. [5]
3. An engineer is trying to design a branch predictor for the MIPS 5 stage processor. Branches constitute 20% of all instructions. The engineer has essentially two options for the branch predictor: PicoPruner and ChartCooser. For both these predictors, the branch mispredict penalty is 3 cycles. Branches correctly predicted undergo no penalty cycles. Simulation of PicoPruner shows 10% misprediction rate, but implementing PicoPruner will increase the cycle time by 20%. Simulation of ChartCooser shows 20% misprediction rate, but its implementation will increase the cycle time by only 10% . Which predictor should the designer choose? Clearly show all details of your computations. [8]
4. A program has the following instruction mix: 40% integer operations, 40% loads and stores, and 20% floating point operations. The processor uses a diversified pipeline in which integer operations take 1 cycle, loads and stores take 2 cycles, and floating point operations take 3 cycles. A compiler writer suggests a transformation in which every floating point operation is replaced with 4 integer operations. The hardware-designers indicate that by not implementing floating point arithmetic, the clock cycle time can be decreased by 15%. Will the combination of these two changes improve or degrade the overall performance of the processor, and by how much? [5]

5. A certain 6 stage pipelined processor has two branch delay slots. An optimizing compiler can fill the first slot 80% of the time and can also fill the second slot 20% of the time. Of the filled slots, 10% eventually get discarded on account of being taken from the fall through path. What is the percentage improvement in performance achieved by this optimizing compiler relative to a compiler that does not fill any of the branch delay slots? Assume a branch occurs once every 7 instructions on the average. [7]
6. Suppose a processor has a 10-stage pipeline with the following stages:
- F1 – start the fetch and predict - if the instr. is a branch, whether it is taken or not, and if taken, the target address.
 - F2 – complete the fetch
 - D1 – decode the instruction – know it's a branch at the end of D1,
 - D2 – complete decode
 - RR – read the registers– compute branch target address
 - A1 – start ALU operation; resolve branch condition and update CPU
 - A2 – complete ALU operation
 - M1 – start memory operation
 - M2 – complete memory operation
 - WB – write the result to registers
- a) What is the penalty for a mispredicted branch? [2]
 - b) What is the penalty when a branch is correctly predicted as taken, but branch target address is incorrectly predicted? [2]
 - c) Assuming the following characteristics of the processor and the program, what is the CPI for the processor? [4]
 - There are no stalls in this pipeline except for branch instructions.
 - Branch instructions account for 25% of all instructions
 - 60% of branch instructions are taken
 - Branch direction is correctly predicted 90% of the time
 - The target address for a taken branch is correctly predicted 80% of the time
7. Consider a 5 stage MIPS pipeline in which a dynamic predictor with target prediction is used at the IF stage. A tagged branch target prediction (BTB) scheme for taken branches is used. The programs run on this processor have 20% branches. Of these, on the average 60% are taken and 40% are not taken. For taken branches, there is a 10% chance of miss in the branch target buffer. Also, 10% of the branches matching the BTB turn out to be not taken. Further, the branch target address is predicted with 90% accuracy. In case of misprediction, the PC is updated with correct target address in MEM stage. On account of data hazards, the pipeline stalls 30% of time. What would be the CPI for this processor? Clearly show all steps of your computation. [8]

8. For the simple 5-stage pipelined MIPS processor, we are required to design a new more complicated instruction. Two design options (given below) are being considered. Determine which design option would be superior for executing large programs and how much faster (ratio of CPIs) is it expected to run as compared to the other option? You may ignore the effect of hazards. Clearly show your work out. [5]

- a) Adding extra logic circuits to the execute stage of the pipeline. This would increase the latency of the EXE stage by 20%.
- b) Adding a new stage after the EXE stage altogether, making it a 6-stage pipeline. This arrangement would leave the cycle time unaffected.

9. Consider the following pseudo code:

```
int i=1, j, c;
loop1:    j = 1;
loop2:    c = c + i + j;
          j++;
          if (j <= 3) goto loop2;
          i++;
          if (i <= 1000) goto loop1;
```

The compiler will produce conditional branches for the two if statements. Using a branch history table with 2-bit saturating counters, **exactly how many mispredictions will there be?** Assume there are enough table entries to avoid conflicts. Assume all table entries are initialized to 0. Clearly show your reasoning and computations. [8]

10. In the following double-nested loop S1; S2; ... Sk; are the statements that form the body of the inner loop and are simple arithmetic statements.

```
for(i=0 ; i < m ; i++)
    for(j=0; j<n ; j++){
        S1; S2; ... Sk;}
```

- a) Suppose we have the option of using either a 1-bit local predictor or a 2-bit saturating counter local predictor for the inner loop. Which predictor would be more accurate for the inner loop and by how much? [4]
- b) Suppose a (1,2) correlating predictor is used. Compare the performance of the (1,2) correlating predictor with the 2-bit saturating counter. [4]

--- The End ---