5DV008 Computer Architecture Fall 2010
Problem Exercise 1
Due date: December 9, 2010 at 0800 (8am)

This is the first of two problem exercises. It is not obligatory, but it does carry points which may be used to offset a weaker score on the examination, as described in the course syllabus. This exercise is worth at total of 50 points. The due date is strict and late submissions will not be accepted, except under extenuating circumstances.

Time permitting, after the due date, solutions will be discussed in class, and submissions which make a reasonable effort will be corrected to show errors.

1 Problems

1. A single-cycle implementation of the MIPS architecture has the following parameters:

Component	**Delay**
Instruction memory read	500 ps.
Data memory read or write	450 ps.
Register read	200 ps.
Register write	225 ps.
ALU operation	250 ps.
Adder operation	110 ps.
All other operations	0 ps.

Using Figure 4.24 of the textbook as the implementation model, compute the critical-path times for each of the following instructions. In all cases, show the names of the components are included in the path, as well as their values.

(a) The `add` instruction.

(b) The `lw` instruction.

(c) The `sw` instruction.

(d) The `beq` instruction.

(e) The `j` instruction.

2. An implementation of the MIPS architecture following Figure 4.33 of the textbook has the following parameters:

Component	Delay
Instruction fetch (ID)	250 ps.
Instruction decode and register file read (ID)	300 ps.
Execution or address calculation (EX)	400 ps.
Data memory access (MEM)	250 ps.
Write back (WB)	100 ps.

(a) Assuming a non-pipelined implementation, and assuming all delays not explicitly identified above to be negligible, compute the latency for each of the five instructions identified in Problem 1 above.

(b) Repeat part (a), this time assuming a pipelined implementation.

(c) For a pipelined implementation, if one state of the pipeline could be split into two stages, each with a latency one-half of that of the original stage, identify the stage which should be split in order to obtain the least total latency.

(d) Suppose that the instruction mix of a program is as follows: `add`: 25%; `beq`: 30%; `lw`: 30%; `sw`: 15%. For the pipelined implementation, compute the percentage of total cycles which utilize data memory. Assume that there are no stalls or hazards.

3. Given are the following two instruction sequences.

(i):

```
lw   $t1, 0($t2)
addi $t2, $t3, 50
add  $t3, $t1, $t8
addi $t1, $t1, 50
sw   $t1, 0($t2)
```

(ii):

```
addi $t2, $t7, 50
addi $t1, $t2, $zero
sw   $t2, 0($t1)
lw   $t1, 8($t2)
add  $t2, $t2, $t6
```

For each of these instruction sequences, do the following:

(a) Identify all data dependencies, regardless of the distance between the instructions. Classify each dependency as *read after write* (RAW), *write after read* (WAR), or *write after write* (WAW).

(b) For the five-stage pipeline model of the textbook, indicate which of these dependencies results in a hazard in the case that there is no forwarding, and show how `nop` operations may be inserted to avoid these hazards. Find one solution for each program sequence which resolves all hazards, not one solution for each hazard. The solution must not contain any unnecessary `nop`s.

(c) Repeat part (b) in the case that forwarding is employed.

4. A program contains two distinct branch instructions. For the first instruction, the pattern `YYNYY` repeats endlessly, where `Y` indicates that the branch was taken and `N` that it was not. For the second program, the pattern `YYNNN` repeats endlessly. For each of these two patterns, answer the following questions.

(a) Compute the success rate for the always-taken and never-taken branch predictors.

(b) Suppose that a one-bit predictor is employed, with the initial state `0` (previous branch not taken). Compute the accuracy of the prediction for the first k iterations of the loop, with k large enough to identify the repeating pattern.

(c) Now suppose that a two-bit predictor is employed, with the initial state `00` (previous two branches not taken). Compute the accuracy of the prediction for the first k iterations of the loop, with k large enough to identify the repeating pattern.

2 Submission Rules

Solutions may be developed and submitted by groups of up to three individuals.

A printed copy of the solution must be placed in the appropriate course mailbox on the fourth floor of MIT-huset. The user-id of each group member for the submission must be indicated clearly on a cover page of the printed submission.