CPSC 418	Advanced Computer Architecture	Spring 1995

Homework 6: Due 28 Feb

General Information for This Homework

MOPS (Canada) Ltd is a low-cost competitor of a more well known microprocessor company. The MOPS line of microprocessors (q1999, q2999, q3999, q7999) has been relatively successful and MOPS is working on the q8999. In this homework we will examine the different techniques for dynamic instruction scheduling that MOPS is considering for the q8999 and it's successor the q9999.

Problem 1 (8 Points)

The current design for the q8999 uses *non-pipelined* floating-point execution units. Other relevant data:

- Dual-issue superscalar
- 8 architected floating-point registers
- One floating-point add/subtract unit: (latency 3 cycles)
- One floating-point multiply unit:
  (latency 5 cycles)
- One floating-point divide unit:
  (latency 20 cycles)
- The floating-point register file has 6 read ports and 3 write ports devoted to the arithmetic units (there are other ports for the load/store unit).

The latency data for the units includes write-back.

MOPS is considering two instruction scheduling techniques for the q8999: a scoreboard and Tomosulo's algorithm with reservation-stations. There are two reservation station for each execution unit.

1(a) (2 Points) Write two fragments of code that will run faster with one technique than with the other. Each fragment must illustrate a different reason why the code will run faster. The two fragments don't have to run fastest with the same technique (*i.e.* one may run faster with a scoreboard and one with reservation stations).

1(b) (6 Points) For each dynamic scheduling technique (scoreboard and reservation-stations) label the hazards in the code and explain the affect of the hazard on the execution time for the code.

Problem 2 (9 Points)

The q8999 was so successful that MOPS has already begun work on the q9999, which will use pipelined floating-point execution units, register-renaming, and contain 16 physical registers for the 8 architected registers. As with the r8999 proposal that used Tomosulo's algorithm, the reservation station for each execution pipeline (functional unit) can contain up to two instructions. The data for the execution-pipelines are:

- One floating-point add/subtract pipeline:
  latency 3 cycles, throughput 1-instruction/cycle
- One floating-point multiply pipeline:
  latency 5 cycles, throughput 1-instruction/2-cycles
- One floating-point divide pipeline:
  latency 20 cycles, throughput 1-instruction/20-cycles
- The floating-point register file has 4 read ports and 3 write ports devoted to the arithmetic pipelines (there are other ports for the load/store pipeline).

Given:

- the code fragment in Figure 1
- at clock-cycle zero all of the reservation stations are empty and all of the registers are valid
- at clock-cycle zero the register map and free list are as shown in Figures 2 and 3.

2(a) (5 Points) For each of the instructions, list the time at which the instruction begins execution and when it writes its result to the register file.

2(b) (4 Points) Show the processor state (reservation stations, register map (including validity flag), and freelist) for the clock cycle when the add instruction begins execution.

```
100     MULF   F3, F2, F1
104     DIVF   F2, F1, F2
104     MULF   F7, F6, F1
10C     SUBF   F4, F2, F1
110     ADDF   F2, F1, F3
114     SUBF   F1, F3, F2
```

Figure 1: Code fragment for register renaming

Arch Reg	F0	F1	F2	F3	F4	F5	F6	F7
Phys Reg	F0'	F2'	F14'	F1'	F12'	F5'	F15'	F9'

Figure 2: Register map at clock cycle zero

F3'	F4'	F6'	F7'	F8'	F10'	F11'	F13'
hd							tl

Figure 3: Freelist at clock cycle zero

Problem 3 (1 Points) How much time did you spend **total** on this homework? Which problems needed more preparation in lecture or better explanations?