COMP 740 Homework 4

Prof. Montek Singh
Fall 2016
Due Monday, December 5th (in class)

1. *Compiler Techniques and VLIW* (35 points). Do Textbook Problem 3.14 parts (a), (b) and (c).

2. *Multiple-Issue* (15 points). Do Textbook Problem 3.15 part (b). You already did part (a) as part of Homework 3; please refer to its solution.

3. *Thread-Level Parallelism* (25 points). Consider the following MIPS code fragment. Assume that it processes an array X1 of 64-bit integers, which is stored at address 3000. Assume the following initialization: R1 = 1000, R2 = 0, R6 = 0, R3 = 0.

```
loop :   DSLL R4, R3, #3
         SUBI R1, R1, #1
         LD R5, 3000(R4)
         ADDI R3, R3, #1
b1 :     BEQZ R5, t0
         ADDI R2, R2, #1
         J b2
t0 :     ADDI R6, R6, #1
b2 :     BNEZ R1, loop
...
```

Note: The instruction DSLL performs a logical left shift. Please refer to the textbook if you need more information on the instruction set.

Suppose the code executes on a computer architecture with the following simplifying assumptions:

- a single-issue, in-order processor that can fetch, decode and issue one instruction per clock cycle;
- when a data dependency is present, the processor stalls and does not issue the instruction;
- ALU and branch instructions take one cycle to execute and the result can be used in the next cycle;
- the processor has an ideal branch predictor and there is no delay penalty for conditional branches; the processor doesn't have a data cache; each memory operation takes 60 CPU cycles;
- the load/store unit is non-blocking and fully pipelined: after issuing a memory operation the processor continues issuing and executing the following instructions until it reaches an instruction that is dependent on an outstanding memory operation.

Do the following:

a) Compute how many clock cycles are necessary to complete one iteration of the loop with the described architecture.

b) Assume that the code above represents a thread and that you can use a multithreaded version of the previous architecture in order to run multiple instances of this code as *N* separate threads processing N distinct 64-bit integer arrays $X_1, X_2, \cdots X_N$. Assume coarse-grain multi-threading: the processor switches to a different thread only when it is forced to stall the present thread. Assume three cycles of penalty for context-switch. What is the minimum number *N* of threads necessary to fully utilize the processor, i.e., to avoid any stall cycle?

c) Make the same assumptions as in (b) but assume fine-grain multithreading instead of coarse-grain multithreading: at each clock cycle the processor switches to a different thread following a statically determined round-robin schedule. In other words, for each thread the processor executes one instruction every *N* cycles. Assume no context-switch penalty. What is the minimum number *N* of threads necessary to fully utilize the processor in this case?

4. *Storage Systems* (25 points). Suppose you were given 8 identical disks and were asked to set them up in a RAID configuration. Further suppose that you could only choose between two RAID configurations: mirrored striped (RAID 0+1) [apply striping first at lower level, then apply mirroring at higher level] or striped mirrored (RAID 1+0) [mirroring at lower level, striping at higher level]. Assume that: (i) striping in RAID 0 is 4-way, and (ii) RAID 1 duplicates information onto two sets.

    a) Draw pictures illustrating the two configurations, and identify which is which.

    b) If the probability of a given physical disk failing on any given day is p, calculate for each of the two configurations the probability of a non-recoverable failure of the storage system on a given day. Assume that p is small (i.e., $p << 1$), and simplify your expressions by ignoring higher powers of p.

    c) If your primary objective was achieving reliable storage, which of the two configurations would you choose, and why? By what factor is your chosen configuration more reliable than the other?

    d) If your primary objective was achieving high data transfer rates, which of the two configurations would you choose, and why?