CHALMERS

2009-12-10

Exam in DAT 105 (DIT 051) Computer Architecture

Time: December 17, 2009 14 – 18 in V&V

Person in charge of the exam: Mafijul Islam, phone 031-772 1711

Supporting material/tools: Chalmers approved calculator.

Exam Review: On January 5, 2009 10-12 in Room 4128

Grading intervals:

- **Fail**: Result < 24
- **Grade 3**: 24 <= Result < 36
- **Grade 4:** 36 <= Result < 48
- **Grade 5:** 48 <= Result

Important note: Answers must be given in English

GOOD LUCK!
Per Stenström

[General disclaimer: If you feel that sufficient facts are not provided to solve a problem, either 1) ask the teacher when he visits the exam, or 2) make your own additional assumptions. Additional assumptions will be accepted if they are reasonable and required to solve the problem. Always make sure to motivate your answers.]

ASSIGNMENT 1

A)

	Computer A (seconds)	Computer B (seconds)	Reference computer (seconds)
Program 1	1	2	6
Program 2	2	2	1
Program 3	4	2	1

The execution times of three programs on three computers are listed in the table above. Calculate the following

(i) The arithmetic means of the execution times on each computer for the set of programs. Which computer has the highest performance? Mention a potential problem with comparing performance using arithmetic means. **(2 points)**

(ii) For each of the programs, calculate the ratio of the execution time on Computer A and the reference computer for each of the programs and then do the same for Computer B. (**2 points)**

(iii) Now use geometric mean to calculate the ratio of the performance on Computer A and B. Which of the computers has the highest performance using this approach? (**2 points)**

B) We want to reduce the execution time of a program using a multiprocessor system. The execution time of the original sequential program is 10 seconds. Now assume that 80% of the sequential program execution can be distributed evenly across processors in a multiprocessor system. What is the speed-up achieved? What is the maximum speed-up that can be ever achieved? (**3 points)**

C) Assume that a memory instruction has an average CPI=2 and that the average CPI for the rest of the instructions is CPI=1. Given that 20% of the instructions are memory instructions, that a program executes 1,000,000,000 instructions, and the clock frequency is 1 GHz, what is the execution time of the program? **(3 points)**

ASSIGNMENT 2

Assume a MIPS processor with a simple five stage pipeline of the type presented in appendix A of the course book (Instruction fetch, Instruction decode, Execute, Memory access, Write back). Furthermore, assume that all memory accesses complete in a single cycle (cache hit time is one cycle), that pipeline forwarding is used whenever possible to resolve RAW hazards, and that there is two branch delay slots (all branch computations are completed in the EX stage). Integer multiply operations are performed in two steps; the first in the Execute stage, and the second in the Memory Access stage, so multiply operations are fully pipelined like all other arithmetic instructions, but the result is not available until the end of the Memory access stage. You need to write a piece of MIPS assembly code to perform the following computation in as few clock cycles as possible.

```
for(int i=0; i<n; i=i+1) {
            x[i] = x[i]*x[i];
}
```

The x array is an array of long integers (8 bytes). You may assume that register R1 already is loaded with the start address of the x array, i.e. the address of x[0], and that you are allowed to change the value of R1. The address of x[n] is likewise available in register R2. You may assume that n always is greater than zero, and use that fact in your code.

A) The following is a first version of the code:

```
LOOP:     LD        R4, 0(R1)
          DMUL      R5, R4, R4
          SD        R5, 0(R1)
          DADDI     R1, R1, 8
          BNE       R1, R2, LOOP
```

(i) List all true data dependences in the code

(ii) Which true data dependences can be resolved without any RAW hazards and which of them cannot be resolved? Justify your answer.

(iii) How many cycles are lost in each iteration due to RAW and control hazards? Again, justify your answer!

(**3 points)**

B) Now use loop unrolling to get rid of all stall cycles and answer the following questions:

(i) How many times must the loop be unrolled to eliminate all hazards?

(ii) How many more registers are needed to eliminate name dependences?

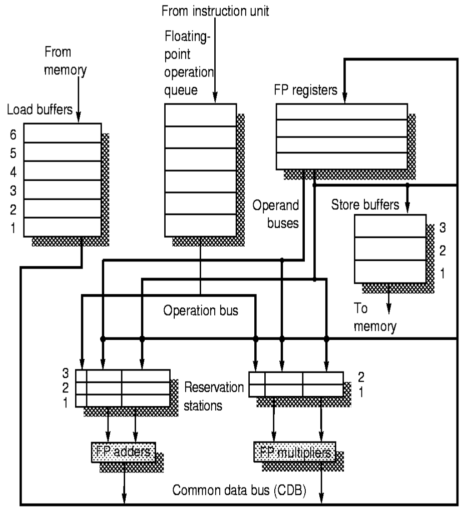
(iii) How many more instructions are needed in each loop iteration?

(3 points)

C) It is possible to remove some of the control hazards by scheduling of the instructions within each iteration. Reschedule the code inside each iteration to minimize the number of control hazards. How many cycles are now lost due to control hazards? (**2 points)**

D) Branch prediction can also be used to eliminate control hazards. In the assumed pipeline the branch condition is computed in the EX stage. Explain how a 2-bit branch predictor can be used to remove control hazards and when it will correctly predict the branch outcome.

(4 points)



The above diagram shows a pipeline with Tomasulo’s algorithm. Consider the following code:

```
SUB  F0, F1,F2   (8 cycles)
DIV   F3,F0,F4    (10 cycles)
ADD F4, F5,F6    (6 cycles)
```

A) Show all name dependences in the code and explain how Tomasulo’s algorithm avoids translating them into hazards by explaining **in detail** what happens when the instructions are (i) issued (ii) reading their operands and (iii) and executed (**4 points**)

B) Assuming that there are two addition functional units and a division functional unit and that a single instruction is issued every cycle. The instruction latencies through the functional units are listed above. How long time does it take until the last instruction has written back the result to the register file? (**2 points**)

C) A reorder buffer and a branch predictor can be added to allow instructions to be speculatively executed. Assume that the three instructions in the code above are speculatively executed. Explain **in detail** what happens when an instruction is speculatively executed in each of the execution stages (instruction issue, execute, write result, commit) (**4 points)**

D) Explain how the content of F0 is supplied to the DIV instruction. (**2 points**)

ASSIGNMENT 4

A) Cache misses in a uniprocessor are of three kinds (the 3C model). Explain **in detail** the reason for the three kinds of misses. **(3 points)**

B) Average memory access time is dictated by cache hit time, miss rate, and miss penalty. Derive an analytical model of the average memory access time based on these concepts. **(3 points)**

C) Way prediction is a technique that can reduce the average memory access time in a cache hierarchy. Explain **in detail** how it does that and what concept in B) that it addresses. (**2 points)**

D) When a pipeline can have multiple outstanding memory requests it is important that the cache can service these requests in parallel. Explain **in detail** how a multibanked cache can increase the number of requests to be serviced in parallel. (**2 points)**

E) Loop interchange is a code rearrangement technique to cut down the number of cache misses. Explain how it works by way of an example. (**2 points)**

ASSIGNMENT 5

A) Time-multiplexed multithreading comes in two flavors: Fine-grain multithreading and coarse-grain multithreading. Explain when a switch to a new thread occurs assuming a simple five-stage pipeline for each of the two approaches. (**4 points**)

B) What structures in a superscalar processor must be replicated to realize a simultaneous multithreaded processor? (**2 points**)

C) Using Flynn’s taxonomy (XIYD) where X;Y={S,M} motivate why symmetric multiprocessors fall into the category MIMD.(**2 points**)

D) Explain the additional structures needed in a copy-back cache to implement a write-invalidate snoopy cache protocol to maintain cache coherence.
(**4 points**)

****** GOOD LUCK! ******