Midterm Exam 2

ECE 741 – Advanced Computer Architecture, Spring 2009

Instructor: Onur Mutlu

TAs: Michael Papamichael, Theodoros Strigkos, Evangelos Vlachos

April 22, 2009

NAME: _____

Problem	Points	Score
1	60	
2	30	
3	20	
4	20	
5	10	
6	15	
7	20	
8	25	
Total	200	

- This is a closed book midterm. You are allowed to have only 3 letter-sized cheat sheets.
- No electronic devices may be used.
- This exam lasts 1 hour 50 minutes.
- If you make a mess, clearly indicate your final answer.
- For questions requiring brief answers, please provide brief answers. Do not write an essay. You can be penalized for verbosity.
- Please show your work. We cannot give you partial credit if you do not clearly show how you arrive at an answer.
- Please do not write on the back of the sheets (if printed in simplex).
- **Please write your name on *every* sheet.**

Name: _____

Problem 1 (Short answers) [60 Points]

Please be concise, clear, and to-the-point in all your answers.

i) DRAM Controllers [7 points]

Some recent chip designs, such as Sun Niagara and AMD Athlon 64 have included the DRAM controller on-chip (i.e., on the processor die) as opposed to off-chip (i.e., in the chipset). Briefly describe two advantages of this design decision. Explain the reasoning clearly for each advantage.

Briefly describe one disadvantage of this design decision. Explain your reasoning clearly.

ii) In-order vs. out-of-order [4 points]

In an in-order processor instructions are dispatched in _____ order.

In an out-of-order processor instructions are dispatched in _____ order.

(Note: the answer should be more sophisticated than in and out-of, which is not grammatically correct anyway.)

iii) Caching instructions [2 points]

In an instruction cache, an instruction can be stored in _____ location(s).

In a trace cache, an instruction can be stored in _____ location(s).

iv) Execution Models [4 points]

Finish the following sentence: The DAE (Decoupled Access/Execute) paradigm provides an opportunity for performance improvement over the VLIW paradigm because

v) SRAM vs. DRAM [2 points]

The main element of storage required to store a single bit of information depends on whether we are talking about DRAM cells or SRAM cells.

For DRAM cells it is:

For SRAM cells it is:

Name: _____

vi) Prefetching [6 points]

Which of the following prefetching methods cannot prefetch compulsory misses? Circle as many as that apply.

1. Markov prefetcher
2. Runahead execution
3. Stream buffers
4. Content directed prefetching
5. Software based pre-execution

Why? Explain very briefly.

vii) Vector processing [4 points]

```
for (i=1; i < 500; i++)
      A[i] = (B[i] + A[i-1])/7919;
```

Is the above loop vectorizable? Circle one: YES NO

Explain why/why not (10 words should be enough).

viii) Runahead execution [8 points]

Suppose a branch is fetched in "runahead mode" of a runahead execution processor. Suppose we know "magically" whether or not the branch is mispredicted. The below questions deal with whether continuing runahead execution is always useless after the fetch of this branch.

Is runahead execution always useless after a mispredicted branch that does not depend on an L2 cache miss? Circle one: YES NO. Why/Why not?

Is runahead execution always useless after a mispredicted branch that depends on an L2 cache miss? Circle one: YES NO. Why/Why not?

Name: _____

ix) Vector processing vs. VLIW [5 points]

A vector processor is fundamentally different from a VLIW processor. Why? Explain in less than 20 words.

x) Predicated vs. dual-path execution [6 points]

What two benefits does predicated execution provide over dual-path execution? Explain, very briefly.

xi) Execution models and branch handling [12 points]

Suppose we define the misprediction penalty of a branch instruction as the number of instruction slots that are wasted when the branch misprediction is detected. Is the branch misprediction penalty constant or variable in the following types of processors? Circle one for each execution paradigm. Briefly explain why.

In-order dispatch: Constant Variable

Out-of-order execution: Constant Variable

VLIW: Constant Variable

Decoupled Access/Execute: Constant Variable

Name: _____

Problem 2 (Branch Prediction) [30 Points]

The first two-level branch predictor was published in Micro-24 in 1991. Since then, it has seen many variations. Each variation has been proposed to provide some additional benefit. Four such variations are described below.

For each variation, describe the intended additional benefit over the baseline two-level predictor that was developed in 1991. It is important to be explicit in your answers without being unnecessarily wordy.

i) Instead of using the global history register to index into the pattern history table, use the XOR of the global history register combined with some bits of the branch's address.

ii) Combine the two-level predictor with a 2-bit counter or a last time taken predictor.

Name: _____

iii) Do not increment/decrement the 2-bit counter in the pattern history register on the basis of T/NT. Instead store a direction bit in the tag store entry for each branch, and increment/decrement on the basis of whether the prediction agrees or does not agree with the stored direction bit.

iv) Instead of using the directions of the last several branches leading up to a branch as the index into the pattern history table, combine bits of the addresses of each of the intervening branches into an index.

Name: _____

Problem 3 (Out-of-order Execution) [20 Points]

Initially, the register file of the Tomasulo-like machine shown below contains valid data in all registers. Four instructions are then fetched, decoded and issued in program order, none are executed. At that point the register file and reservation stations are as shown below:

	V	TAG	VALUE
R0	0	π	4
R1	1	ρ	5
R2	1	σ	6
R3	1	β	7
R4	0	β	8
R5	0	γ	9
R6	1	α	10

Tag	V	Tag	Value	V	Tag	Value
α	1	-	5	1	-	7
β	0	π	-	1	-	8
γ	0	β	-	1	-	10
δ						

+

Tag	V	Tag	Value	V	Tag	Value
π	0	α	-	1	-	4
ρ						
σ						
τ						

*

Note: Some TAG information contained in the register file is left over from earlier instructions.

a. Assuming no additional instructions are fetched, decoded, issued, fill in the state of the register file after the four instructions execute. [10 points]

	V	TAG	VALUE
R0			
R1			
R2			
R3			
R4			
R5			
R6			

b. Using only instructions of the form `ADD Rd, Rs, Rt` and `MUL Rd, Rs, Rt`, where `Rd` is the destination register and `Rs` and `Rt` are the source registers, show the four instructions in program order. [10 points]

1	
2	
3	
4	

Name: _____

Problem 4 (The Magic Processor) [20 Points]

We have a 16 wide issue microprocessor and (magically) a perfect branch predictor (100% accuracy) and perfect caches (100% hit rate). Everything is working correctly on the chip (i.e. no bugs).

a. We measure the IPC on one application and it is less than 16. How is this possible? Explain. [4 points]

b. Can the IPC be less than 1? Explain. If yes, give an example piece of code (in pseudo-assembly language) that will produce this. If no, why not? [8 points]

c. The pipeline consists of 2 stages. If you could redesign the chip, would you change the number of stages? Why or why not? Explain clearly. [8 points]

Name: _____

Problem 5 (Instruction caching) [10 Points]

The Intel Pentium processor had a split-line first-level instruction cache. Why was that a good idea? Be brief, please, but specific.

Name: _____

Problem 6 (VLIW) [15 Points]

a. An IA 64 instruction bundle consists of three 41-bit instructions, packaged in a 128-bit unit. What are the extra five bits used for? What value do they provide over previous VLIW designs? [10 points]

b. What are the disadvantages of this approach compared to traditional VLIW? [5 points]

Name: _____

Problem 7 (Block Structured ISA and Superblocks) [20 Points]

a. The Block-Structured ISA is fundamentally different from the Superblock. How so? [6 points]

b. What are three advantages of Block-Structured ISA caused by this difference? [7 points]

c. What are three disadvantages of Block-Structured ISA caused by this difference? [7 points]

Name: _____

Problem 8 (Register Alias Tables) [25 Points]

a. What is the primary function of the register alias table in an out-of-order execution processor? Explain briefly. [2 points]

b. When does an instruction access the register alias table for that purpose? [2 points]

c. Suppose you froze the execution of a processor and magically observed all entries in the register alias table. You found that all entries in the register alias table are invalid. Is this possible or is this due to an error in the design? If it is possible, explain why are all entries are invalid? If not, why is this not possible? [6 points]

d. Pentium 4 contains a retirement register alias table in addition to a traditional register alias table. What function does the retirement register alias table serve? [2 points]

e. How many entries are there in Pentium 4's retirement register alias table? (We are not looking for an absolute number.) [2 points]

f. Suppose you froze the execution of the Pentium 4 and magically observed all entries in the retirement register alias table. You found that one entry in the retirement register alias table is invalid. Is this possible or is this due to an error in the design? If it is possible, explain why one entry could be invalid? If not, why is this not possible? [6 points]

g. Can the contents of the frontend register alias table and the retirement register alias table be the same? Explain why or why not? [5 points]