

Name: _____

CSE 30321 – Computer Architecture I – Fall 2009
Final Exam
December 18, 2009

Test Guidelines:

1. Place your name on EACH page of the test in the space provided.
2. Answer every question in the space provided. If separate sheets are needed, make sure to include your name and clearly identify the problem being solved.
3. Read each question carefully. Ask questions if anything needs to be clarified.
4. The exam is open book and open notes.
5. All other points of the ND Honor Code are in effect!
6. Upon completion, please turn in the test and any scratch paper that you used.

Suggestion:

- Whenever possible, show your work and your thought process. This will make it easier for us to give you partial credit.

Question	Possible Points	Your Points
1	15	
2	10	
3	20	
4	15	
5	15	
6	10	
7	15	
Total	100	

Name: _____

Problem 1: (15 points)

Question A: (5 points)

Briefly (in 4-5 sentences or a bulleted list) explain why many of the transistors on a modern microprocessor chip are devoted to Level 1, Level 2, and sometimes Level 3 cache. Your answer must fit in the box below!

In HW 8, you saw that some versions of the Pentium 4 microprocessor have two 8 Kbyte, Level 1 caches – one for data and one for instructions. However, a design team is considering another option – a single, 16 Kbyte cache that holds both instructions and data.

Additional specs for the 16 Kbyte cache include:

- Each block will hold 32 bytes of data (not including tag, valid bit, etc.)
- The cache would be 2-way set associative
- Physical addresses are 32 bits
- Data is addressed to the word and words are 32 bits

Question B: (3 points)

How many blocks would be in this cache?

Name: _____

Question C: (3 points)

How many bits of tag are stored with each block entry?

Question D: (4 points)

Each instruction fetch means a reference to the instruction cache and 35% of all instructions reference data memory. With the first implementation:

- The average miss rate in the L1 instruction cache was 2%
- The average miss rate in the L1 data cache was 10%
- In both cases, the miss penalty is 9 CCs

For the new design, the average miss rate is 3% for the cache as a whole, and the miss penalty is again 9 CCs.

Which design is better and by how much?

Name: _____

Problem 2: (10 points)

Question A: (4 points)

Explain the advantages and disadvantages (in 4-5 sentences or a bulleted list) of using a direct mapped cache instead of an 8-way set associative cache. Your answer must fit in the box below!

Assume you have a 2-way set associative cache.

- Words are 4 bytes
- **Addresses are to the byte**
- Each block holds 512 bytes
- There are 1024 blocks in the cache

Question B: (2 points)

If you reference a 32-bit physical address – and the cache is initially empty – how many data words are brought into the cache with this reference?

Question C: (4 points)

Which set does the data that is brought in go to if the physical address F A B 1 2 3 8 9 (in hex) is supplied to the cache?

Name: _____

Problem 3: (20 points)

Question A: (5 points)

Explain (in 4-5 sentences or via a short bulleted list) why there is translation lookaside buffer on the virtual-to-physical address critical path. Your answer must fit in the box below!

For the next question, refer to the snapshot of TLB and page table state shown below.

Initial TLB State:

(Note that '1' = "Most Recently Used and '4' = "Least Recently Used")

Valid	LRU	Tag	Physical Page #
1	3	1111	0001
1	4	0011	0010
1	2	1000	1000
1	1	0100	1010

Initial Page Table State:

	Valid	Physical Page #
0000	0	0011
0001	1	1001
0010	1	0000
0011	1	0010
0100	1	1010
0101	0	0100
0110	1	1011
0111	0	0101
1000	1	1000
1001	1	0110
1010	1	1111
1011	1	1101
1100	1	0111
1101	0	1110
1110	1	1100
1111	1	0001

Also, pages are 4 KB, there is a 4-entry, fully-associative TLB, and the TLB uses a true, least-recently-used replacement policy.

Name: _____

Question B: (5 points)

Assume that the Page Table Register is equal to 0.

The virtual address supplied is:

(MSB) 1100 ■ 0010 ■ 0010 ■ 0100 (LSB)

What physical address is calculated? If you cannot calculate the physical address because of a page fault, please just write "Page Fault".

Question C: (5 points)

Consider the following:

- Virtual address are 32 bits
- Pages have 65,536 (or 2^{16}) addressable entries
- Each page table entry has:
 - o 1 valid bit
 - o 1 dirty bit
 - o The physical frame number
- Physical addresses are 30 bits long

How much memory would we need to simultaneously hold the page tables for two different processes?

Question D: (5 points)

Assume that for a given system, virtual addresses are 40 bits long and physical addresses are 30 bits long. There are 8 Kbytes of addressable entries per page. The TLB in the address translation path has 128 entries. How many virtual addresses can be quickly translated by the TLB? Would changing the page size make your answer better or worse? – **Note that there are 2 questions here!**

Name: _____

Problem 4: (15 points)

This question considers the basic, MIPS, 5-stage pipeline. (For this problem, you may assume that there is full forwarding.)

Question A: (5 points)

Explain how pipelining can improve the performance of a given instruction mix. Answer in the box.

Question B: (6 points)

Show how the instructions will flow through the pipeline:

	1	2	3	4	5	6	7	8	9	10	11	12
lw \$10, 0(\$11)												
add \$9, \$11, \$11												
sub \$8, \$10, \$9												
lw \$7, 0(\$8)												
sw \$7, 4(\$8)												

Question C: (4 points)

Where might the sw instruction get its data from? Be very specific. (i.e. “from the lw instruction” is *not* a good answer!)

Name: _____

Problem 5: (15 points)

This question considers the basic, MIPS, 5-stage pipeline. (For this problem, you may assume that there is full forwarding.)

Question A: (5 points)

Using pipelining as context, explain why very accurate branch prediction is important in advanced computer architectures. Answer in the box below.

Question B: (7 points)

Show how the instructions will flow through the pipeline.

- You should assume that branches are predicted to be taken.
- You should assume that $0(\$2) == 4$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
lw \$1, 0(\$2)																
bneq \$0, \$1, X																
add \$2, \$1, \$1																
X: add \$2, \$1, \$0																
sw \$2, 4(\$2)																

Question C: (3 points)

What is the value that will be written to 4(\$2) in the last instruction?

Name: _____

Problem 6: (10 points)

- Consider an Intel P4 microprocessor with a 16 Kbyte unified L1 cache.
- The miss rate for this cache is 3% and the hit time is 2 CCs.
- The processor also has an 8 Mbyte, on-chip L2 cache.
- 95% of the time, data requests to the L2 cache are found.
- If data is not found in the L2 cache, a request is made to a 4 Gbyte main memory.
- The time to service a memory request is 100,000 CCs.
- On average, it takes 3.5 CCs to find data in the memory hierarchy.
- The L2 hit time is 15 CCs
- The Main Memory hit time is 200 CCs
- What is the miss rate to main memory?

Name: _____

Problem 7: (15 points)

- Assume that for a given problem, N CCs are needed to process each data element.
- The microprocessor that this problem will run on has 4 cores.
- If we want to solve part of the problem on another core, we will need to spend 250 CCs for each instantiation on a new core.
 - o (e.g. if a problem is split up on 2 cores, 1 instantiation is needed)
- Also, if the problem is to run on multiple cores, an overhead of 10 CCs *per data element* is associated with each instantiation.
- We want to process M data elements.

Question A: (8 points)

If N is equal to 500 and M is equal to 100000, what speedup do we get if we run the program on 4 cores instead of 1? (*Hint – start by writing an expression!*)

Question B: (7 points)

If N is equal to 250, M is equal to 120, the communication overhead increases to 100 CC per element, and the instantiation overhead remains the same (at 250 CC), how many cores should we run the problem on? Explain why.