Computer Engineering 110: Computer Architecture

Final Examination

Fall 2016

Name: _____Solution_____

Email: _____

1	20	
2	20	
3	25	
4	15	
5	20	
6	20	
Total	120	

This exam is closed book and closed notes. Personal calculators (four-function calculators only) *are* allowed. Show your work on the attached sheets (front and back) and insert your answer in the space(s) provided. **Please provide details on how you reach a result.** Ask for extra paper sheets if necessary.

You have 180 minutes to complete the exam. This exam is worth 120 points. This midterm exam counts for 30% of your course grade.

(20 points) 1. Caches

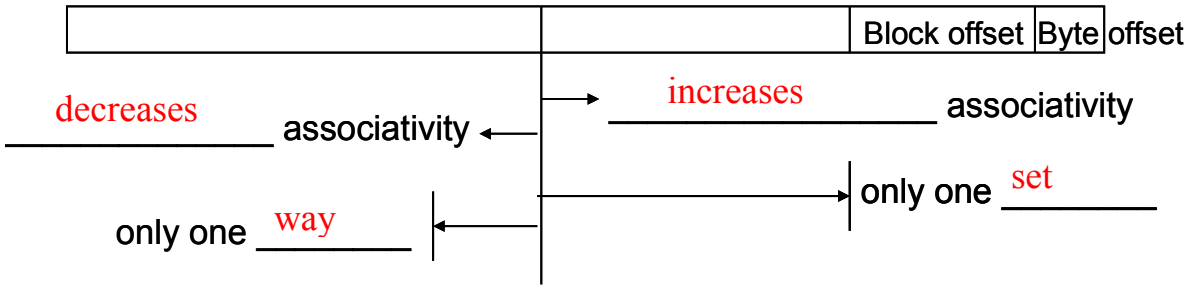
A. Allow cache and memory to be inconsistent, i.e., write the data **only** into the cache block. Is this write-back or write through? _____ (No further explanation needed)

Write-back, 4 points

B. Require cache and memory to be consistent, i.e., always write the data into both the cache block and the next level in the memory hierarchy. Is this write-back or write through? _____ (No further explanation needed)

Write through, 4 points

C. Consider the diagram below that shows the dividing line between the bits used for tag compare and those used to select the cache set. Fill in the lines indicating whether the associativity **increases** or **decreases** and whether the resulting cache has only one **way** or only one **set**.



3 points each, total 12 points

(20 points) 2. Cache Performance

Assuming the base CPI_{base} (no stall) of a pipeline is 1. A program has 25% of load/store instructions. The processor only has one level of cache, i.e., an L1 instruction cache and an L1 data cache. Cache miss rate and penalty are as following:

- L1 instruction cache : $\%_{miss}$ = 2%, t_{miss} = 30 cycles
- L1 data cache: $\%_{miss}$ = 30%, t_{miss} = 30 cycles

If the pipeline has to stall whenever a load/store instruction has a cache miss, what is the CPI?

$CPI = CPI_{base} + (\%_{cache\_access}\%_{miss} * CPI_{miss})$

CPI = 1 + (100%*2%*30 cycles/insn) + (25%*30%*30 cycles/insn)
= 1 + 0.6 + 2.25
= 3.85

Grading instruction:

1. The result 3.85 is worth 4 points
2. Get the instruction cache penalty 0.6, 2 points
3. Get the data cache penalty 2.25, 2 points
4. The equation 100%*2%*30 (or any form of equation with the same meaning), 4 points
5. The equation 25%*30%*30 (or any form of equation with the same meaning), 4 points
6. The equation CPI = 1 + instruction cache penalty + data cache penalty (or any form of equation with the same meaning), 4 points

(25 points) 3. Multicore: Cache Coherence
Assuming write-back caches, private L1 data caches (L1Ds) (no L2), shared memory, and a MESI coherence policy, indicate in the cache tables (*on the next page*) what would be the values (V) of the variables X and Y and their coherency (C) state (either M, E, S or I) in the L1D's for a dual core processor with the following sequence of reads and writes. As shown in the table, the coherency (C) state in Step 0 (the initial coherency state) of both variables in both caches is I (invalid). If a value doesn't change, you may leave that entry blank. You may use either the state transition diagram or the table given below to reason about cache coherence.

Step 0: Initially, X =300, Y = 500 in the shared memory, processor caches are empty

Step 1: Core 1 reads X (from the shared memory)

Step 2: Core 2 reads X

Step 3: Core 1 writes X = 200

Step 4: Core 1 writes Y = 600

Step 5: Core 2 reads Y

Please fill in the cache tables on the next page.

Core 1's L1D Cache (private)												
	Step 0		Step 1		Step 2		Step 3		Step 4		Step 5	
	V	C	V	C	V	C	V	C	V	C	V	C
X	--	I	300	E	300	S	200	M	200	M	200	M
Y	--	I	--	I	--	I	--	I	600	M	600	S

Core 2's L1D Cache (private)												
	Step 0		Step 1		Step 2		Step 3		Step 4		Step 5	
	V	C	V	C	V	C	V	C	V	C	V	C
X	--	I	--	I	300	S	300	I	300	I	300	I
Y	--	I	--	I	--	I	--	I	--	I	600	S

Shared Memory						
	Step 0	Step1	Step2	Step3	Step4	Step5
	V	V	V	V	V	V
X	300	300	300	300	300	300
Y	500	500	500	500	500	600

Grading instruction:

1. Each cell worth 0.5 points
2. Shared memory Step 3 can also be 200. Both are correct.

(15 points) 4. Memory Consistency

The memory consistency model provides a specification of how the memory system will appear to the programmer. Consider the following code segment, where the initial values are A = 0, B= 0, and writes propagate immediately.

Thread 1 (on P1)

```
A = 1;
If (B == 0)
   print "Santa";
```

Thread 2 (on P2)

```
B = 1;
If (A == 0)
   print "Cruz";
```

What will be printed out if we maintain sequential consistency? (You do not need to exhaustively show all permutations. But please show an example permutation for each output you get.)

Grading instruction:

1. Santa (2 points) or Cruz (2 points) or nothing (2 points)
2. Need to show at least one instruction permutation example for each print out, each worth 3 points.
3. If "Santa Cruz", take 5 points away

(20 points) 5. Main Memory

Given a system with 2 memory channels and 4 DRAM DIMMs (2 DIMMs per channel), each DIMM has:

- 1 rank
- 8 chips per rank
- 8 bits per column
- 4 banks per chip
- 32,768 rows per bank
- 1,024 columns per bank

1) What is the total amount (bytes) of physical memory in the system?

Memory per chip = column size * banks * rows * columns = 8 * 4 * 32768 * 1024 = 1Gbit

Memory per DIMM = memory per chip * chips per rank * ranks per DIMM =1 Gbit * 8 * 1= 8 Gbit

Total physical memory = memory per DIMM * DIMMs in the system = 8 Gbit * 4 = 32 Gbits = 4 GBytes

2) What is the minimum number of physical address bits needed to address this much memory?

4 GBytes = $4 * 2^{30}$ Bytes = 2^{32} Bytes

Physical address bits = $\log_2 2^{32}$ = 32 bits

3) With the number of physical address bits obtained in 2), also assume

- The physical address space has 1M (i.e., 1048576) pages
- Virtual addresses have 64 bits

What is the maximum number of pages in the virtual address space?

The statement of "physical address space has 1M pages" may seem vague to some of students. So we have to solutions -- both are correct.

Solution 1: assume the number of physical pages is 1048576

Page size = 4GBytes / 1048576 = 4096 Bytes

Thus, bits in page offset = $\log_2 4096$ = 12 bits

Bits in virtual page number = 64 – 12 = 52 bits

Maximum pages in virtual address space = 2^{52} = 4503599627370496 Bytes

Solution 2: assume the physical page size is 1048576 bytes

Thus, bits in page offset = $\log_2 1048576$ = 20 bits

Bits in virtual page number = 64 – 20 = 34 bits

Maximum pages in virtual address space = 2^{34} = 17179869184 = 16GBytes

Grading instruction:

1. Each result worth 3 points
2. Show their works with equations: 3 points for Q2, and 4 points for Q1 and Q3

(20 points) 6. GPUs

(1) Which ones of the following are the basic ideas of GPU processor architecture design? (May have multiple choices)

A. Interleave processing of many fragments on a single core to avoid stalls caused by high latency operations
B. Make common case fast
C. Remove components that help a single instruction stream fast
D. Amortize cost/complexity of managing an instruction across many ALUs

ACD, 9 points (each choice is worth 3 points, take 3 points away if choose B)

(2) Assume a hypothetical GPU with the following characteristics:

Clock rate 1 GHz

Contains 8 SIMD processors, each containing 32 single-precision floating-point units, each instruction performs one single-precision floating-point operation.

What is the peak single-precision floating-point throughput for this GPU in GFLOP/sec, assuming that all memory latencies can be hidden?

1 GHz * 8 *32 = 256 GFLOPS

Grading instruction:

1. the result 256 GFLOPS is worth 5 points
2. 4 points for the 8*32 part of the equation
3. 2 points for the 1GHz in the equation