16.482 / 16.561: Computer Architecture and Design

Spring 2015

Final Exam
April 23, 2015

Name: _____

For this exam, you may use a calculator and two 8.5" x 11" double-sided page of notes. All other electronic devices (e.g., cellular phones, laptops) are prohibited. If you have a cellular phone, please turn it off prior to the start of the exam to avoid distracting other students.

The exam contains 6 questions for a total of 100 points. Please answer the questions in the spaces provided. If you need additional space, use the back of the page on which the question is written and clearly indicate that you have done so.

You will be provided with a single page containing additional copies of information used in Question 1, so that you can more easily refer to this material while answering these questions. You do not have to submit this page when you turn in your exam.

You will have three hours to complete this exam.

Q1: Multiple issue and multithreading	/ 17
Q2: Cache basics	/ 23
Q3: Virtual memory	/ 12
Q4: Cache optimizations	/ 19
Q5: RAID	/ 14
Q6: Multiprocessors	/ 15
TOTAL SCORE	/ 100

1. (17 points) ***Multiple issue and multithreading***

a. (3 points) Explain why simply increasing the number of instructions a processor fetches and issues will not significantly improve its performance.

b. (4 points) You are given a single program containing four separate threads of execution. Two of these threads experience a cache miss every 20 instructions, while the other two threads contain shorter but more frequent stalls. Would this program run faster on a processor using fine-grained or coarse-grained multithreading? Explain your answer for full credit.

1 (continued)
c. (10 points) This problem deals with the 3 threads below:

Thread 1:
```
L.D   F0, 0(R1)
L.D   F2, 8(R1)
ADD.D F4, F0, F8
SUB.D F6, F2, F8
DADDUI R1, R1, 16
BNE   R1, 1600, L1
```

Thread 2:
```
SUB.D F8, F0, F2
ADD.D F6, F4, F8
S.D   F6, 8(R1)
DADDUI R1, R1, 16
L.D   F2, 0(R1)
SUB.D F4, F6, F2
```

Thread 3:
```
DADDUI R3, R1, R2
SUB.D F6, F8, F10
L.D   F4, 0(R3)
S.D   F6, 0(R1)
DADDUI R1, R1, 24
DADDUI R2, R2, -8
BNE   R1, R2, loop
```

The processor executes instructions in order, and you should assume all branches are not taken. Each instruction has the latency given below:

- L.D/S.D: 4 cycles (1 EX, 3 MEM)
- ADD.D/SUB.D: 3 cycles
- DADDUI: 2 cycles
- All other operations: 1 cycle

Determine how long these threads take to execute using simultaneous multithreading on a processor with the following characteristics:

- Five functional units: 2 ALUs, 2 memory ports (load/store), 1 branch
  - Note: The ALUs can handle DADDUI operations
- Thread 1 is the preferred thread, followed by Thread 2 and Thread 3.

Your solution should use the table on the next page, which contains columns to show each cycle and the functional units being used during that cycle. **Clearly indicate which thread contains each instruction when completing the table, but you do not have to write the full instruction—writing the opcode (i.e. L.D, ADD.D) is sufficient.**

Your extra handout contains a copy of the threads and latencies.

SOLUTION TO QUESTION 1c

Cycle	ALU1	ALU2	Mem1	Mem2	Branch
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					

2. (23 points) ***Cache basics***

a. (4 points) Explain the three different cache block placement strategies (in other words, how to determine where a block is located in the cache), listing one benefit of each.

b. (3 points) Explain how and why LRU replacement is approximated—not implemented exactly—in set-associative caches with associativity greater than 2.

2 (continued)

c. (16 points) You are given a system which has a 16-byte, write-back cache with 4-byte blocks. The cache is 2-way set-associative. The system uses 8-bit addresses, and the cache is initially empty. At first, $t0 = 9 and $t1 = 16.

Assume the initial memory state shown below for the first 32 bytes:

Address		Address		Address		Address	
0	27	8	19	16	22	24	13
1	8	9	49	17	5	25	24
2	35	10	9	18	14	26	27
3	10	11	21	19	13	27	7
4	99	12	3	20	49	28	18
5	17	13	0	21	77	29	8
6	64	14	90	22	15	30	55
7	1	15	4	23	44	31	99

For each access in the sequence listed below, fill in the cache state, indicate what register (if any) changes, and indicate if any memory blocks are written back and if so, what addresses and values are written. The cache state should carry over from one access to the next.

Access	Modified register	Cache state								Modified mem. block
		V	D	MRU	Tag	Data				
`sb $t0,14($zero)`										
`lb $t1,30($zero)`										
`lb $t0,7($zero)`										
`sb $t0,28($zero)`										

3. (12 points) ***Virtual memory***

a. (3 points) Explain why a translation lookaside buffer (TLB) is implemented as a fully associative structure.

b. (9 points) This problem involves a process using the page table below:

PAGE TABLE STATE:

Virtual page #	Valid bit	Reference bit	Dirty bit	Frame #
0	1	1	0	3
1	0	0	0	--
2	1	0	0	4
3	0	0	0	--
4	1	1	1	0
5	1	0	0	1

Assume the system uses 16-bit addresses and 2 KB pages. The process accesses four addresses: 0x07FE, 0x1978, 0x26AA, and 0x2C33.

Determine (i) which address causes a page fault, (ii) which one sets a previously cleared reference bit for a page that is currently in physical memory, and (iii) which one accesses a page that has been modified since it was brought into memory. **For full credit, show all work.**

4. (19 points) ***Cache optimizations***

a. (9 points) Answer three of the following four questions about cache optimizations:

i. Explain why it is possible to have multiple copies of the same instruction stored in a trace cache

ii. In a multi-banked cache, why should data be sequentially interleaved across the banks?

iii. Why does critical word first offer no performance benefit without being combined with early restart?

iv. Under what circumstances would next sequential prefetching actually increase the miss rate of a cache?

4 (continued)

b. (10 points) A designer is considering two cache designs for a memory subsystem:

- A direct-mapped cache with a 90% hit rate that takes 2 ns to access.
- A set-associative cache with a 93% hit rate that takes 10 ns to access.

To improve the performance of the set-associative cache, the designer considers adding a way predictor to that cache. If the way predictor is correct, cache hits would take only 3 ns, while way misses that still produce cache hits would take 12 ns. Assume that the cache miss penalty is 200 ns, regardless of the type of cache being used.

How accurate must the way predictor be for the system to have the same overall performance with the set-associative cache as it would with the direct-mapped cache? **For full credit, show all work.**

5. (14 points) ***RAID***
a. (6 points) Identify which types of operations—reads and/or writes—can be overlapped in each of the following RAID levels, and briefly explain why:

i. RAID 3

ii. RAID 4

iii. RAID 5

5 (continued)

b. (8 points) This part of the problem involves a five-disk RAID 5 array containing a total of 20 sectors. Sixteen of the twenty sectors (S0-S15) hold data, while the remaining four sectors (P0-P3) hold parity information. You should also assume the following:

- Large reads/writes take 250 ms, small reads take 75 ms, and small writes take 125 ms.
- The number of disks in the array is the only limit on performance—any number of consecutive operations may proceed simultaneously if they do not share any disk within the array. Multiple accesses in the same stripe are allowed.
- All operations must be performed in order.
- If two disks, D_x and D_y, are in use, and the access to D_x finishes before the access to D_y, a new operation may start immediately assuming it does not involve D_y.

Determine the time required for this array to complete the following sequence of accesses. **For full credit, show all work, including the organization of the array**:

1. read S5
2. write S10
3. read S13
4. write S4
5. write S15
6. read S3
7. write S7
8. write S14

The next page contains additional space to solve this problem if necessary.

ADDITIONAL SPACE FOR QUESTION 5b SOLUTION

6. (15 points) ***Coherence protocols***

a. (4 points) In a snooping coherence protocol, why must each processor transmit read or write misses to all other processors? What potential performance problems arise in systems using this type of coherence protocol?

b. (3 points) Explain when a Write Back message would be sent in a directory protocol.

c. (3 points) How can false sharing misses be avoided in multiprocessor systems?

6 (continued)

d. (5 points) Say we have a multi-processor system with three nodes, P0, P1, and P2, which share a block at address *A*. The system uses a write-invalidate, directory coherence protocol. Initially, the directory entry for the block reads:

	P0	P1	P2	Dirty
A	1	1	0	0

If P2 now attempts to write block *A*, what messages are sent between the nodes and directory to ensure P2 gets the most up-to-date block copy and the directory holds the appropriate state? You may want to draw a diagram to support your answer.