CS 152 Exam #2

Solutions

Personal Information

<table>
<thead>
<tr>
<th>First and Last Name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

| Your Login         |
| cs152-_____ _____   |

<table>
<thead>
<tr>
<th>Lab/Discussion Section Time &amp; Location you attend</th>
</tr>
</thead>
<tbody>
<tr>
<td>“All the work is my own. I have no prior knowledge of the exam contents nor will I share the contents with others in CS152 who have not taken it yet.” (Please sign)</td>
</tr>
</tbody>
</table>

Instructions

- Partial credit may be given for incomplete answers, so please write down as much of the solution as you can.

- Please write legibly! If we can’t read it from 2 feet away, we won’t grade it!

- Put your login on each page.

- This exam will count for 16% of your grade.

Good luck!

Dave Patterson
John Lazzaro
Doug Densmore
Ted Hong
Brandon Ooi

Grading Results

<table>
<thead>
<tr>
<th>Problem</th>
<th>Time (minutes)</th>
<th>Max. Points</th>
<th>Points Earned</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>25</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>90</td>
<td>125</td>
<td></td>
</tr>
</tbody>
</table>
Problem 1 – Cache Performance (30 points, 20 minutes)
In this problem, we compare two memory system designs for a classic 5-stage pipelined processor. Both memory systems have a 4 KB instruction cache.

- System A has a 4K byte data cache, with a miss rate of 10% and a hit time of 1 cycle.
- System B has an 8K byte data cache, with a miss rate of 5% and a hit time of 2 cycles (the cache is not pipelined).

For both data caches, cache lines hold a single word (4 bytes), and the miss penalty is 10 cycles.

a) What is the average memory access time for data retrieved by load instructions for the two memory system designs, measured in clock cycles? (6 points)

System A: \[ \text{AMAT} = 1 + (0.1 \times 10) = 2 \text{ cycles} \]
System B: \[ \text{AMAT} = 2 + (0.05 \times 10) = 2.5 \text{ cycles}. \]

Cache Short Answers
b) Describe two clear advantages a Harvard architecture (separate instruction and data caches, as in your Lab 4 processor) has over a unified cache architecture (a single cache memory array accessed by a processor to retrieve both instructions and data). Describe one clear advantage a unified cache architecture has over the Harvard architecture. (9 points)

Harvard advantages: (1) No contention if both ports wish to read (2) Faster because each array is smaller.

Unified advantage: Cache provides better performance for a D-cache/I-cache mix other than 50/50. Coherency is also simpler and easier with a unified cache.
c) In the virtual memory lecture, we showed how Translation Lookaside Buffer (TLB) access may be overlapped with the indexing of a physically-address cache, to speed access time. What is the downside of this scheme? (3 points)

Virtual page size locked to the cache tag size.

d) **T**  **F**  Coherence cache misses cannot occur with a memory system attached to a single processor, even if I/O peripherals are able to read and write the memory system independent of the processor. If false, explain your answer. (2 points)

False. I/O may write main memory, but it would not be reflected in cache values.

e) **T**  **F**  In a memory system with a multi-level data cache (small/fast L1 data cache, larger/slower L2 data cache), the miss rate of the L1 cache is usually larger than the miss rate of the L2 cache. If false, explain your answer. (2 points)

False. If L1 sees a hit, L2 never gets accessed. If L1 misses, access often has poor locality and L2’s miss rate is judged on how well it performance on this difficult subset of problems.

f) In the virtual memory lecture, we showed a memory system design in which virtual memory addresses were sent directly to a cache, without using a TLB to convert the virtual addresses to physical addresses. In this scheme, the TLB was only needed if there was a cache miss. What’s a flaw of this scheme? (3 points)

The synonym problem. One physical address may map to 2 virtual addresses and be present in the cache twice.
Problem 2 – RAID (25 points, 15 minutes)

a) T F If false, give a counter example.
   All disks that are being configured with RAID-5 must be of equal size. Hard disks that have different storage capacity can not be used for installing a RAID-5 storage system.
   (3 points)

   False. You can configure the system to use the smallest common size on all disks. (Important when replacing disks. New disks are likely to be bigger.)

b) Match the RAID levels 1, 3, and 5 to the following phrases for the best match. Use each only once. (3 points)
   ______5____ Data and parity striped across multiple disks
   ______1____ Can withstand selective multiple disk failures
   ______3____ Requires only one disk for redundancy

c) RAID systems are either implemented in hardware with a separate RAID controller connected to the disks or in software as part of the operating system. After upgrading a server from a software RAID solution to a hardware RAID solution, the administrator notices a substantial increase in performance. Which of the following are true? (2 point each)

<table>
<thead>
<tr>
<th></th>
<th>Hardware solutions can support more RAID levels than software solutions, resulting in improved performance.</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>Processing is now performed by the RAID controller, resulting in less system CPU usage.</td>
</tr>
</tbody>
</table>

d) Why would RAID level 5 have a higher write data bandwidth than level 4 for the same total number of disks? (3 points)

For small writes, RAID 4 has one parity disk which is a bottleneck since one parity and one data block must be updated. RAID 5 does not have all parity blocks on one disk so it has higher small write bandwidth.
e) Why would RAID level 5 have a higher read data bandwidth than level 4 for the same total number of disks? (3 points)

small reads to not need to check parity, so small reads can be read from all disks in RAID 5. N-1 disks in RAID 4 as one raid 5 disk only has parity.

f) One simple way to get redundancy to improve reliability of data transmission is to simply triplicate each bit. Thus, each 0 becomes 000 and each 1 becomes 111.

   a. How many errors can this scheme correct? (2 points)

      1

   b. How many errors can this scheme detect? (2 points)

      1 (Hamming distance is 3, so cannot detect more than 1)

   c. How can you improve the error correction / detection of this scheme while sending fewer bits? Explain in 1 or 2 sentences, giving a concrete example. (5 points)

      Divide each transfer into a fixed size packet, and then add SEC/DED using a hamming code. For example, 64 bit data + 7 bit of ECC + 1 bit of parity offers SEC/DED with only 12.5% overhead.
Problem 3 – Virtual Address Translation (25 points, 25 minutes)
There is a diagram provided of a flow chart detailing the process of translating a virtual address into a physical memory reference and cache accesses. Assume that you have a hardware loaded TLB. Along with the diagram are the entries you can select from. Please put into the diagram the number which belongs to each flow chart location. There are more numbers than locations so be careful! You will not use the same number twice. (13 points)

A-14, B-1, C-11, D-10, E-13, F-12, G-5, H-4, I-7, J-1, K-19, L-17, M-21, N-8, O-15, P-2
Login: ______________________

Simply decide if the following statements are true or false. **In the event that they are false** please provide an explanation in the box provided. (2 points each)

<table>
<thead>
<tr>
<th></th>
<th>You can have a hit in the TLB, a miss in the page table, and a hit in the cache for a single memory access.</th>
</tr>
</thead>
</table>
| T | **Explanation:**  
|   | False – cannot have a hit in TLB and miss in page table since all TLB entries have to have pages in memory |
| F | You can have a miss in the TLB, a hit in the page table, and a miss in the cache for a single memory access. |
| T | **Explanation:**  
|   | True. |
| F | Page faults cannot be handled by software because the overhead is too large. |
| T | **Explanation:**  
|   | False. Page faults take large cycle counts to resolve so software routines will easily finish (be amortized or hidden) |
| F | A “write through” policy is used for virtual memory systems. |
| T | **Explanation:**  
|   | False. Writing to lower levels always effectively undoes the speedup of TLBs etc. since Disk and memory are so slow. |
| F | Virtual memory space has to be bigger than the physical memory space. |
| T | **Explanation:**  
|   | False. VM is just a translation. If your virtual address is smaller than memory you can have it go to a larger physical address. |
| F | The operating system is a key component in a virtual memory system. |
| T | **Explanation:**  
|   | True |
Login: ______________________

**Problem 4 – Tomasulo Out-of-Order Processor** (15 points, 10 minutes)

a) **T   F** In an out-of-order architecture, a physical register may be written several times during its lifetime. If false, explain. (3 points)

False. Only 1 write allowed – this lets us execute instructions out-of-order w/o WAW, WAR hazards

b) **T   F** In an out-of-order architecture that implements precise exceptions, exception handling occurs after the result of the instruction that causes the exception is committed to an architected register. If false, explain. (3 points)

We use the reorder buffer as a history buffer of program execution and copy it to an architected state once we know an exception or interrupt has not affected.

c) Below is the block diagram of the single-threaded Power 4. Three blocks were duplicated in order to support multi-threading in the Power 5. Circle all three. (6 points)

**Legend**

- IF = instruction fetch
- IC = instruction cache
- BP = branch predict
- D0 = decode stage 0
- Xfer = transfer
- GD = group dispatch
- MP = mapping
- ISS = instruction issue
- RF = register file read
- EX = execute
- EA = compute address
- DC = data caches
- F6 = six-cycle FP execution pipe
- Fmt = data format
- WB = write back
- CP = group commit.

IF, D0 and CP
d)  T  F  The performance data on the Power 5 shown in class indicated that two threads sharing a single core, run with equal scheduling priority, each ran about as fast (within 5%) as it would if it ran on the core without the second thread present. If false, explain why. (3 points)

False. A thread might need a resource the second thread is using. This contention slows the thread down relative to how it runs and it owns the machine.

Problem 5 – Theory (20 points, 10 minutes)

a) Computational theory predicts that the Sort A algorithm is a more efficient algorithm than the Sort B algorithm in terms of the number of integer comparisons required. In practice however, Sort B tends to run faster than Sort A for large data sets. Based on your knowledge of caches, speculate as to why this may be? (5 points)

Like the radix vs. quick sort phenomena. Quicksort takes advantage of caching through spatial and temporal locality more so than radix which is a scanning styled algorithm.

High-Level Cache Optimization
Problems (b) – (c) refer to the following code segment. Assume you are given a multi-dimensional array M[128][128][128].

```c
int total = 0;
for(int x = 0; x < 128; x++)
    for(int y = 0; y < 128; y++)
        for(int z = 0; z < 128; z++)
            total += M[y][x][z];
```

The above code finds the total of all the numbers in the array. Multi-dimensional arrays are usually organized in memory in row-major order. For example:

```
[ [(0,0,0)…(n,0,0)] [(0,1,0)…(n,1,0)]…[(0,n,0)…(n,n,0)] ],
[(0,0,1)…(n,0,1)] [(0,1,1)…(n,1,1)]…[(0,n,1)…(n,n,1)] ],
...
[(0,0,n)…(n,0,n)] [(0,1,n)…(n,1,n)]…[(0,n,n)…(n,n,n)] ]
```
b) Rewrite the code in two different ways to more effectively take advantage of caching. Give a one sentence explanation for each. (10 points)

Based on the memory drawing given, we expected these two optimal solutions.

```c
int total = 0;
for(int x = 0; x < 128; x++)
    for(int y = 0; y < 128; y++)
        for(int z = 0; z < 128; z++)
            total += M[z][y][x];

int total = 0;
for(int z = 0; x < 128; x++)
    for(int x = 0; y < 128; y++)
        for(int y = 0; z < 128; z++)
            total += M[y][x][z];
```

Super Pipelining

c) Explain why branch mispredictions are especially costly in processors with long pipelines. Where does this cost come from? Give a concrete example. (5 points)

In a long pipeline, the operation to determine whether a branch is taken could happen fairly late in the pipeline. Mispredicted branches cause a flush of the pipeline after the branch. However, the cost of the misprediction comes from the large bubble in the pipeline which must be refilled to restore the efficiency of the CPU.
Problem 6 – Super Scalar Processors (15 points, 10 minutes)

a) Given the Multicycle Processor (not pipelined) from chapter 5, we are now creating a dual-issue superscalar processor based off this. If the original multicycle processor has an optimal CPI of N, what is the optimal CPI of the superscalar processor? (3 points)

N/2

b) Consider the data hazards for both of the processors, for each of the following data hazards explain whether or not the hazard exists. If it does exit, give an explanation of how it is solved. (7 points)

<table>
<thead>
<tr>
<th></th>
<th>Multi-cycle</th>
<th>Super-scalar Multi-cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAW</td>
<td>None. Only one instruction is being executed at a time and completely finishes before the next one begins.</td>
<td>Yes. Two instruction which are issued together may depend on the first. e.g. Add $1, $0, $0 Add $2, $1, $1 Solved by issuing only independent instructions.</td>
</tr>
<tr>
<td>WAW</td>
<td>None.</td>
<td>Yes. Instructions can save at different times and second issued instruction write may occur after the first. Solved by FSM, force the write of one before the other.</td>
</tr>
<tr>
<td>WAR</td>
<td>None.</td>
<td>No, the two issued instructions read from the register file at the same time before any has had a chance to write.</td>
</tr>
</tbody>
</table>

c) What changes are needed to the datapath and other components to support the dual issue multicycle processor? (5 points)

Dual port inst-memory to fetch 2 instructions.
Dual port register file to read 4 operands.
Control to handle hazards
Duplicate multicycle datapath.
Scratch Copy
Do NOT write your solution on this copy.

1. Cache miss stall while read block
2. Page Fault Routine
3. Clear out old TLB values
4. Try to read data from cache
5. Write?
6. TLB Hit?
7. Cache hit?
8. No
9. Blue Screen of Death
10. Page in main memory?
11. Access page table
12. Generate Physical Address
13. Update TLB
14. TLB Access
15. Write data into the cache
16. Write to Disk
17. Try to write data to cache
18. Write protection exception
19. Write Access bit on?
20. Deliver Data to the CPU

A.

B.

C.

D.

E.

F. 12

G. 5

H.

I.

J.

K.

L.

M.

N.

O.

P.

Q.