CSE 240A Midterm Exam

2011 Fall

Professor Steven Swanson

Please write your name at the top of each page
This is a close book, closed notes exam. No outside material may be used.
You may use a calculator
Show your work. You will get more partial credit that way.
If you have any question, please raise your hand.
Good luck!

Name: ________________________ KEY ________________________

Student ID: ________________________

<table>
<thead>
<tr>
<th>Problem</th>
<th>Possible</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>48</td>
<td></td>
</tr>
</tbody>
</table>
1. Short Answer (8 points)
   a) Give the Amdahl’s Law equation for total speedup, \( S_{tot} \), in terms of the speedup, \( S \), that an optimization achieves on a fraction, \( x \), of a program.

\[
S_{tot} = \frac{1}{(x/S) + (1-x)}
\]

b) List and define the 3 C’s of cache misses that we’ve discussed in class. Give a technique that will help eliminate each kind.

<table>
<thead>
<tr>
<th>C</th>
<th>Description</th>
<th>Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compulsory</td>
<td>Misses on data that has never been accessed.</td>
<td>Prefetching</td>
</tr>
<tr>
<td>Capacity</td>
<td>Misses on data resulting from the application accessing more data than the cache can hold.</td>
<td>Increase cache size.</td>
</tr>
<tr>
<td>Conflict</td>
<td>Misses on data that has previously been in the cache but was evicted to make room for other data.</td>
<td>Increase associativity</td>
</tr>
</tbody>
</table>

c) A common “rule of thumb” is that reducing cache capacity by half while doubling associativity will maintain a roughly constant miss rate. If we apply this guideline repeatedly, it will eventually stop being true. What type of miss (from the 3 C’s) will increase as we apply the rule repeatedly?

   Capacity misses.

d) Why do memories pre-charge bit lines before reading?

   To make reading faster.
2. (2 points) You run your workload on two processors, A and B. Processor A takes 40 seconds to run. The first speeds up 35% of execution by 4x on processor A. The second optimization speeds up a separate 20% of execution by 2x on processor B. What is the speedup of A over processor B?

\[ Sa = \frac{1}{(0.35/4) + (1 - 0.35)} = 1.35 \]
\[ Sb = \frac{1}{(0.2/2) + (1 - 0.2)} = 1.11 \]

\[ \frac{Sa}{Sb} = \frac{1.35}{1.11} = 1.21 \]

3. (14 points) Consider the following snippets of code. What cache line size would you select to maximize the hit rate and why? The total size of the cache is 8KB. Assume ints and pointers are 4 bytes. Long ints are 8 bytes. For the labeled loads in each piece of code, estimate the cache hit rate. Assume both functions will be called repeatedly. Also, would a stream buffer or victim cache help most if applied to the data cache?

a)
```c
struct {
    Link * next;
    int data;
    long int data2;
} Link;

bool Search(Link *node, int k) {
    while(node) {
        if (node->data == k) // Load 1
            if (node->data2 == k) // Load 2
                return true;
    }
    node = node->next; // Load 3
    return false;
}
```

<table>
<thead>
<tr>
<th>Line size</th>
<th>16 bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load 1 hit rate</td>
<td>~0</td>
</tr>
<tr>
<td>Load 2 hit rate</td>
<td>100%</td>
</tr>
<tr>
<td>Load 3 hit rate</td>
<td>100%</td>
</tr>
</tbody>
</table>

Victim cache, to reduce conflict misses. But nothing much will help.
b) struct {
    int data;
    int data2;
} Data;

bool S(Data array[], int p, int k) {
    for (i = 0; i < p; i++) {
        Data * d = &array[i];
        if (d->data == k) // Load 1
            if (d->data2 == k) // Ld 2
                return true;
    }
    return false;
}

4. (6 points) You are designing processors at XYZ Technology Inc. Version 1.0 of the XYZ ISA has 32 architectural registers, and the pipeline for that design is shown below. Your co-worker proposes increasing this number to 1024, even though it will increase the number of cycles required to access the register file from 1 to 2 (adding another pipe stage after decode). She argues that by removing another stage of the pipeline the design can compensate for the increased register file access time.

   a) Why would increasing register access time without making other changes hurt performance? Ignore the increased latency for execution of a single instruction.

      It will increase the branch resolution delay.

   b) Which pipeline stage should be removed and why won’t removing it impact performance?

      Renaming since the we have now have plenty of registers.

   c) Suggest another change to the architecture that could also reduce the impact of increased register access time but does not involve removing anything from the pipeline.

      Improving the branch predictor to limit the impact of the longer branch delay.
5. (6 points) Give a short code snippet that will perform well with the given type of branch predictor
   a) A predictor that uses the PC to select a 2-bit saturating counter to predict the branch

   ```
   for(I = 0; I < 100000; i++) {  if (I > 1000) {} /* this branch */
   ```

   b) A predictor that uses local branch history to select a 2-bit counter

   ```
   for(I = 0; I < 100000; i++) {  if (I % 3 == 0) {} /* this branch */
   ```

   c) A predictor that uses global history to select a 2-bit counter.

   ```
   for(I = 0; I < 100000; i++) {  if (I % 3 == 0) {} if (I % 2 == 0) {} if (I%6 == 0) {}
   /* this branch */
   ```

6. (2 points) Consider a Tomasulo-based floating point unit in which all operations take two cycle to execute, there is one common data bus, and there are 3 functional units. Which would have a larger impact on performance: Doubling the number of functional units from 3 to 6 or doubling the number of busses from 1 to 2? Why?

   Doubling the number of busses will help most because it’s currently bottleneck.
7. (2 points) The WaveScalar discusses “dataflow locality” and remarks that conventional OOO do a lot of extra work in executing each instruction because they do not exploit dataflow locality. What is this extra work and what parts of an OOO processor perform it?

Computing dependences between instructions that are executed repeatedly. The issue queue.

8. (4 points) The Cryptomaniac paper evaluated two ways of providing architectural support for cryptography -- a completely new ISA and adding instructions to an existing ISA. Adding instructions to existing ISAs is more common in practice. Give two reasons why this is the case, and then give two disadvantages of adding to an existing ISA rather than starting from scratch.

<table>
<thead>
<tr>
<th>Reason 1</th>
<th>You can reuse existing hardware designs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reason 2</td>
<td>The resulting processor is still general-purpose</td>
</tr>
<tr>
<td>Disad 1</td>
<td>You have less flexibility in designing the ISA extensions.</td>
</tr>
<tr>
<td>Disad 2</td>
<td>The resulting power and/or performance savings may not be as large.</td>
</tr>
</tbody>
</table>

9. (4 Points) Assume that assembly language instructions have the format “add <dst>, <src1>, <src2>”. Draw the true and false dependencies between the following instructions, labeling each with “R” for RAW, “WW” for WAW, or “WR” for “WAR” and the register involved.

1. add r2, r2, r3
2. sub r3, r2, r4
3. mult r3, r5, r6

1->2 RAW
1->2 RAW
2->3 WAW
1->3 WAR
1->2 WAR
Add

Sub

Multi