CS341l Fall 2009 Test #4

Friday, 11 December 2009 10-10:50am

Name: _____**Key**_____

CS 341l Fall 2009, Test #4. 100 points total, number of points each question is worth is indicated in parentheses. Answer all questions. Be as concise as possible while still answering the question. **You have 50 minutes exactly, I'll collect the tests at 10:50am.** A calculator is okay, but it must be just a calculator (no iPhones, *etc*.). Open book and open notes. If a question is ambiguous, state your assumptions before answering it.

#1. (20 points, 4 points each) Circle T for true or F for false.

* Direct Memory Access (DMA) is typically better for high-bandwidth devices than interrupt-driven I/O is.

T

* Strict consistency is a common consistency model in modern multicore CPUs, because relaxed consistency models have poor performance.

F

* GPUs typically allow you to run different code on each stream processor, making them faster than CPUs for things like running operating systems and doing the SPEC benchmarks as long as there are multiple threads to run.

F

* In computer architecture wires are free, so that you can add as many transistors to a chip as you want and worry about wiring them together later.

F

* Honeywell's Kitchen Computer (H316 pedestal model) was a commercial success due to its intuitive user interface.

F

#2. (10 points) Assume cache coherence, but no consistency model is specified. What can you expect will be the output of the following multithreaded program? (Assume that threads 1 and 2 run on different CPUs and only these two threads share variables A, B, and flag, with the initial values being A = 0, B = 0, and flag = 0).

Thread 1:

```
A = 1;
B = 1;
flag = 1;
```

Thread 2:

```
while (flag != 1) {};
print(A);
print(B);
```

a. Both A and B will be printed as equal to 1

b. We cannot assume anything about the values printed for A and B

c. A and B will be printed as equal, but they could either both be 0 or both be 1

d. A will always equal 1, but we can assume nothing about B

#3. (10 points) Repeat question #2, but this time assume that you have been told that the architecture enforces a PRAM consistency model (also known as processor consistency).

a. Both A and B will be printed as equal to 1

b. We cannot assume anything about the values printed for A and B

c. A and B will be printed as equal, but they could either both be 0 or both be 1

d. A will always equal 1, but we can assume nothing about B

#4. (9 points, 3 point each) For the following cache configurations, calculate the number of blocks, the number of sets, and the number of tag bits. Assume 32-bit addresses.

* 128 KB cache, 128-byte cache lines, 2-way set associative:

#Blocks?: ___**1024**___, #Sets?: ___**512**___, #Tag bits?: ___**16**__

* 256 KB cache, 128-byte cache lines, 8-way set associative:

#Blocks?: __**2048**___, #Sets?: ___**256**___, #Tag bits?: __**17**___

* 256 KB cache, 64-byte cache lines, direct mapped:

#Blocks?: ___**4096**____, #Sets?: __**4096**___, #Tag bits?: __**14**___

#5. (20 points, 5 points each) Assume a 32-bit Pentium, with 4 KB pages and a two-level paging scheme where each page table (first or second level) has 1024 entries.

*Suppose that the TLB has just been flushed due to a context switch. If a program's first memory access is to the virtual address 0x0804c7b4, which of these sequences of events is possible?:

a. Hit in the TLB, physical address is 0x258387b4

b. Miss in the TLB, hit in both page tables, physical address is 0x258387b4

c. Miss in the TLB, hit in both page tables, physical address is 0x0804c588

d. Miss in the TLB, page fault, physical address is 0x0804c588

*Suppose the event that is the correct answer to the above occurs, then the very next memory access is to the virtual address 0x0804c700. Which of these sequences of events is possible (assume that no context switches occur)?

a. Hit in the TLB

b. Page fault, but then hit in the TLB

c. Page fault, but then hit in the page tables

d. None of the above

*When you declare a pointer in straight C (not CUDA, ignore CUDA for this question) on a typical modern Linux machine, the value of that pointer if you did a memory dump, or printed it as an integer with printf, is what?

a. A physical address

b. A virtual address

c. A sector number

d. A segment number

*In CUDA, the answer to the previous question is not necessarily true. Why?

a. GPUs have cylinders and tracks, but no sectors

b. In CUDA, the segment number comes *after* the segment offset

c. GPUs typically don't have virtual memory paging

d. GPUs typically don't have any memory

#6. (10 points) Use the following equation for Amdahl's law to answer the question:

O.S. = 1/((1-f)+(f/s))

Suppose that my GPU has 32 streaming processors. I'm considering buying another model that is otherwise identical but has 128 streaming processors. I need my overall performance to at least double to justify the cost. Roughly what fraction of my code must be able to be executed in parallel on the GPU for my overall speedup to be equal to 2.0 by buying the new GPU?

Plug in O.S. = 2 and s = 128/32 = 4, solve to get f = 2/3

#7. (10 points) Suppose an array called `MyNumbers` has exactly 1000 unique, random unsigned integers in it, where `MyNumbers` is declared globally as:

```
unsigned int MyNumbers[1000];
```

Write a function in C called `MyMax()` that takes no arguments (it just uses the globally declared array `MyNumbers`), and returns the index of the element in the array that holds the maximum value, i.e., the largest integer that's in the array. You may declare no additional global variables and no more than three local variables.

```
unsigned int MyMax()
{
    int maxval, maxindex;
    int loop;

    maxval = MyNumbers[0];
    maxindex = 0;
    for (loop = 1; loop < 1000; loop++)
    {
        if (MyNumbers[loop] > maxval)
        {
            maxval = MyNumbers[Loop];
            maxindex = loop;
        }
    }
    return maxindex;

}
```

#8. (4 points) If I increase the block size of my cache, holding the total size and associativity the same, and my miss rate goes down, what is happening (*i.e.*, why did the miss rate go down)?

Because more data is fetched whenever a block is fetched, we can exploit more spatial locality.

#9. (4 points) If I increase the block size of my cache, holding the total size and associativity the same, and my miss rate goes down, what is happening (*i.e.*, why did the miss rate go down)?

Conflict misses have gone up because there are fewer sets to map things into, meaning that each set has twice as many addresses mapping into it.

#10. (1 point) Who popularized the term “debugging”, due to an actual moth found in a computer?

Grace Hopper

#11. (1 point) Who wrote programs for Babbage's computing machine concepts, and is therefore credited with being the first computer programmer (hint: It's not Babbage)?

Ada Lovelace

#12. (1 point) Rewrite the function for #6, but using AS/400 MI code instead of C. Use the back of this page.

#13. (no points, just for fun) What movie is the following quote from?: “I like you, Lloyd. I always liked you. You were always the best of them. Best goddamned bartender from Timbuktu to Portland, Maine. Or Portland, Oregon, for that matter.”

The Shining