Computer Science Comprehensive Exam

Architecture

June 10, 1996

This is a closed-book exam. Answer all of the questions. Be concise in your answers. You will receive partial credit for partially correct answers, but extraneous remarks may count against you.

1. When adding two's complement numbers, you discard the carry out from the most significant bit. Show that in one's complement, you must add the carry back into the least significant bit.

2. What is a data hazard? Provide an example. How are data hazards handled in hardware? and in software? How do the solutions compare?

3. Consider the following fragment of C code. An `int` is 4 bytes.

```
int array_[1024][1024];

...

{
    ...

    for (i = 0; i < 1024; i++)
        for (j = 0; j < 1024; j++)
            array_[j][i] += 1;

    ...
}
```

   Consider a memory hierarchy that includes a TLB and a cache. The page size is 4Kbytes. The TLB is fully associative with 64 entries. The cache is direct mapped with 4K 16-byte blocks.

   (a) If the cache is physically addressed, can `array_[i][0]` and `array_[i + 1][0]` collide? Explain your answer.

   (b) What characteristic of the memory hierarchy hurts the performance of this code?

4. Today's high-performance disks incorporate at least 128KB of DRAM on the disk in order to cache disk blocks. Based on what you've learned of processor caches and virtual memory, what explanation can you offer for this fact? Is there anything about the physical nature of a disk that suggests a different policy for bringing data into the cache than the on-demand policy used by processor caches? What possible danger can arise if this cache uses a write-back

rather than write-thru policy? (If the cache uses a write-thru policy, assume that the disk controller will not acknowledge the completion of the disk write until the bits are recorded on the media.)