5008: Computer Architecture
HW#4

- **Limits on Instruction-Level Parallelism**

  Case Study 3.1

- **Review of Memory Hierarchy**

1. You are building a system around a processor with in-order execution that runs at 1.1GHz and has a CPI of 0.7 excluding memory access. The only instructions that read or write data from memory are loads (20% of all instructions) and stores (5% of all instructions).

The memory system for this computer is composed of a split L1 cache that imposes no penalty on hits. Both the I-cache and D-cache are direct mapped and hold 32KB each. The I-cache has a 2% miss rate and 32-byte blocks, and the D-cache is write through with a 5% miss rate and 16-byte blocks. There is a write buffer on the D-cache that eliminates stalls for 95% of all writes.

The 512KB write-back, unified L2 cache has 64-byte blocks and an access time of 15ns. It is connected to the L1 cache by a 128-bit data bus that runs at 266MHz and can transfer one 128-bit word per bus cycle. Of all memory references sent to the L2 cache in this system, 80% are satisfied without going to main memory. Also, 50% of all blocks replaced are dirty.

The 128-bit-wide main memory has an access latency of 60ns, after which any number of bus words may be transferred at the rate of one per cycle on the 128-bit-wide 133MHz main memory bus.

a. What is the average memory access time (AMAT) for instruction accesses?
b. What is the AMAT for data reads?
c. What is the AMAT for data writes?
d. What is the overall CPI, including memory accesses?
e. You are considering replacing the 1.1GHz CPU with one that runs at 2.1GHz, but is otherwise identical. How much faster does the system run with a faster processor? Assume the L1 cache still has no nit penalty, and that the speed of the L2 cache, main memory, and buses remains the same in absolute terms (e.g., the L2 cache still has a 15ns access time and a 266MHz bus connecting it to the CPU and L1 cache).
f. If you want to make your system run faster, which part of the memory system would you improve (CPU @1.1GHz)? Graph the change in overall system performance holding all parameters fixed except the one that you're

improving. Parameters you might consider improving include L2 cache speed, bus speeds, main memory speed, and L1 and L2 hit rates. Based on these graphs, how could you best improve overall system performance with minimal cost?

2. In systems with a write-through L1 cache backed by a write-back L2 cache instead of main memory, a merging write buffer can be simplified. Explain how this can be done. Are there situations where having a full write buffer (instead of the simple version you've just proposed) could be helpful?

3. Smith and Goodman found that for a given small size, a direct-mapped instruction cache consistently outperformed a fully associative instruction cache using LRU replacement.

a. Explain how this would be possible. (*Hint*: You can't explain this with the three C's model because it "ignores" replacement policy.)
b. Explain where replacement policy fits into the three C's model, and explain why this means that misses caused by a replacement policy are "ignored" – or, more precisely, cannot in general be definitively classified – by the three C's model.
c. Are there any replacement policies for the fully associative cache that would outperform the direct-mapped cache? Ignore the policy of "do what a direct-mapped cache would do."

4. As caches increase in size, blocks often increase in size as well.

a. If a large instruction cache has larger data blocks, is there still a need for prefetching? Explain the interaction between prefetching and increased block size in instruction caches.
b. Is there a need for data prefetch instructions when data blocks get larger? Explain.