CSE471—Computer Design and Organization

Prof. Steven Burns *Autumn 1994*

Lecture Notes: Twentyfifth Day

Chapter 7: Memory Hierarchy

- How do you build register files, SRAMs, and DRAMs?
- How do you build direct mapped, set-associative, and fully-associative caches?
- Review of caching techniques.

Memory Hierarchy

- Users want gobs of fast memory.
- But you can't always get what you want (you may not be rich enough.)
- Solution: small amount of expensive fast memory, and a larger amount of cheap slow memory.
- If designed correctly, you have a cheap memory system with good performance.

Speed Comparison

- Let's see why register files are faster than SRAMs, which are in turn faster than DRAMs.
- We'll see how these systems are actually designed in terms of transistors.
- Remember, MOS transistors act like switches. The nMOS type conducts when the gate to source voltage is greater than the threshold voltage (about 1v). When V_{gs} is less than V_t, the switch is off.

Dynamic Random Access Memory

- Destructive read. All reads must be followed by an immediate write.
- Information lost through leaking if not periodically refreshed.

Caches

- What is the cache block (line) size?
- How do we search for the item?
  - Direct-mapped
  - Set-associative
  - Fully-associative
- Which block should be replaced?
  - Random
  - LRU (Least-recently used)

Set Associative Caches

- Look up the entry concurrently in a small number of direct mapped caches. At most one hit should occur.
- Almost as fast and inexpensive as a direct mapped cache, but comparable miss rate with half the memory.

What about Writes?

- Write-through
- Write-back
- Write buffers

Spatial Locality

- To exploit spatial locality, we need wide cache lines.
- This changes the cache hardware by adding a final multiplexor indexed by some low order bits.
- You don't want to increase the line size arbitrarily. Larger line sizes mean increased miss penalties (more time to fill in the line when a miss occurs.)
- You can lower the miss penalty by using *early restart* or *requested word first*.

Interfacing the Memory System to the Caches

- One-word-wide memory organization
- Wide memory organization
- Interleaved memory organization
- Page-mode (static-column) DRAMs

Virtual Memory

- Page table—means of translating virtual addresses to real addresses.
- TLB (Translation Lookaside Buffer)—a special cache for the page table.
- Protection
- Handling page fault exceptions

Virtual Memory

- You can think of physical memory as a cache for a larger virtual memory space.
- We need to translate virtual addresses into physical addresses.
- This is done using a page table, an array of pointers to physical pages, one for each virtual address.
- These page tables can get very large. Other data structures are necessary as the virtual address spaces get larger.
- The TLB (translation lookaside buffer) is a cache for the page table.

Memory Interface to Pipelined Processor

- The memory system must stall the processor on cache misses, cause a page fault exception when the virtual address is not in main memory, and cause a protection violation exception if access to a protected page is attempted.