Review: Cache Optimization Summary

Technique	*MR*	*MP*	*HT*	*Complexity*
Larger Block Size	+	–		0
Higher Associativity	+		–	1
Victim Caches	+			2
Pseudo-Associative Caches	+			2
HW Prefetching of Instr/Data	+			2
Compiler Controlled Prefetching	+			3
Compiler Reduce Misses	+			0
Priority to Read Misses		+		1
Subblock Placement		+	+	1
Early Restart & Critical Word 1st		+		2
Non-Blocking Caches		+		3
Second Level Caches		+		2
Small & Simple Caches	–		+	0
Avoiding Address Translation			+	2
Pipelining Writes			+	1

© Patterson 1994

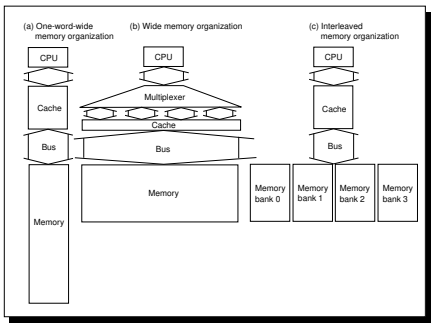
Main Memory Review

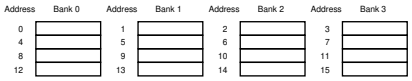
- Wider Memory
- Interleaved Memory: sequential or independent accesses
- Avoiding bank conflicts: SW & HW
- Next: DRAM specific optimizations: page mode & Specialty DRAM

© Patterson 1994

Main Memory Performance

- ***Simple***: CPU, Cache, Bus, Memory same width (32 bits)
- ***Wide***: CPU/Mux 1 word; Mux/Cache, Bus, Memory N words (Alpha: 64 bits & 256 bits)
- ***Interleaved***: CPU, Cache, Bus 1 word: Memory N Modules (4 Modules); example is ***word interleaved***



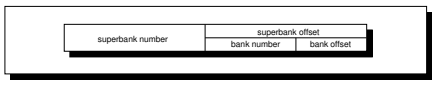


Timing model: 1 to send address, 6 access time, 1 to send data
Cache Block is 4 words
Simple M.P. **= 4 x (1+6+1) = 32**
Wide M.P. **= 1 + 6 + 1 = 8**
Interleaved M.P. **= 1 + 6 + 4x1 = 11**

© Patterson 1994

Review: Independent Memory Banks

- Memory banks for independent accesses vs. faster sequential accesses
  - Multiprocessor
  - I/O
  - Miss under Miss, Non-blocking Cache
- ***Superbank***: all memory active on one block transfer
- ***Bank***: portion within a superbank that is word interleaved



- How many banks?

number banks ≥ number clocks to access word in bank

  - For sequential accesses, otherwise will return to original bank before it has next word ready
- Increasing DRAM => fewer chips => harder to have banks

© Patterson 1994

Review: Fast Memory Systems: DRAM specific

- Multiple RAS accesses: several names (page mode)
  - 64 Mbit DRAM: cycle time = 100 ns, page mode = 20 ns
- New DRAMs to address gap; what will they cost, will they survive?
  - *Synchronous DRAM*: Provide a clock signal to DRAM, transfer synchronous to system clock
  - *RAMBUS*: startup company; reinvent DRAM interface
    - » Each Chip a module vs. slice of memory
    - » Short bus between CPU and chips
    - » does own refresh
    - » variable amount of data returned
    - » 1 byte / 2 ns (500 MB/s per chip)
- Niche memory or main memory?
  - e.g., Video RAM for frame buffers, DRAM + fast serial output

© Patterson 1994

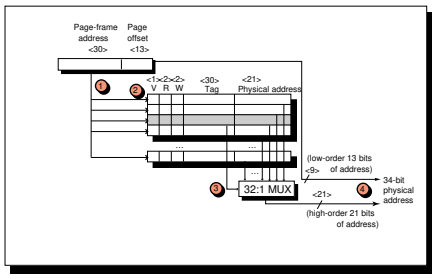
Virtual Memory

- Virtual address (2^{32}, 2^{64}) to Physical Address mapping (2^{28})
- Virtual memory terms of cache terms:
  - Cache block?
  - Cache Miss?
- How is Virtual Memory different from caches?
  - What Controls Replacement
  - Size
  - Lower level use
- 4Qs for VM?
  - Q1: Where can a block be placed in the upper level? Fully Associative, Set Associative, Direct Mapped
  - Q2: How is a block found if it is in the upper level? Tag/Block
  - Q3: Which block should be replaced on a miss? Random, LRU
  - Q4: What happens on a write? Write Back or Write Through (with Write Buffer)

© Patterson 1994

Fast translation: Translation Buffer

- Cache of translated addresses
- Alpha 21064 TLB: 32 entry fully associative



© Patterson 1994

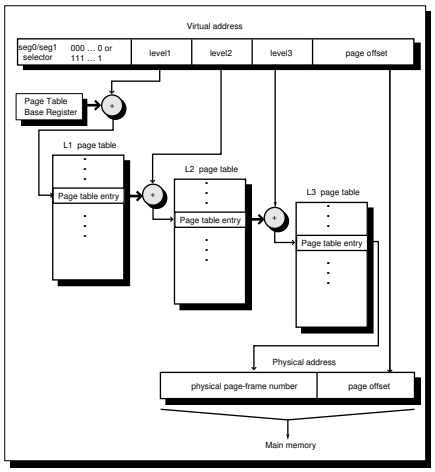
Selecting a Page Size

- Reasons for larger page size
  - Page table size is inversely proportional to the page size; therefore memory saved .
  - fast cache hit time easy when cache ≤ page size; bigger page makes it feasible
  - Transferring larger pages to or from secondary storage, possibly over a network, is more efficient
  - Number of TLB entries are restricted by clock cycle time, so a larger page size maps more memory thereby reducing TLB misses.
- Reasons for a smaller page size
  - don't waste storage; data must be contiguous within page
  - quicker process start for small processes?
- Hybrid solution: multiple page sizes
  Alpha: 8KB, 64KB, 512 KB, 4 MB pages

© Patterson 1994

Alpha VM Mapping

- "64-bit" address divided into 3 segments
  - seg0 (bit 63=0) user code/heap
  - seg1 (bit 63 = 1, 62 = 1) user stack
  - kseg (bit 63 = 1, 62 = 0) kernel segment for OS
- 3 level page table, each one page
  - Alpha only 43 unique bits of VA
  - (future min page size up to 64KB => 55 bits of VA)
- PTE bits; valid, kernel & user read & write enable (No reference, use, or dirty bit)
  - What do you do?



© Patterson 1994

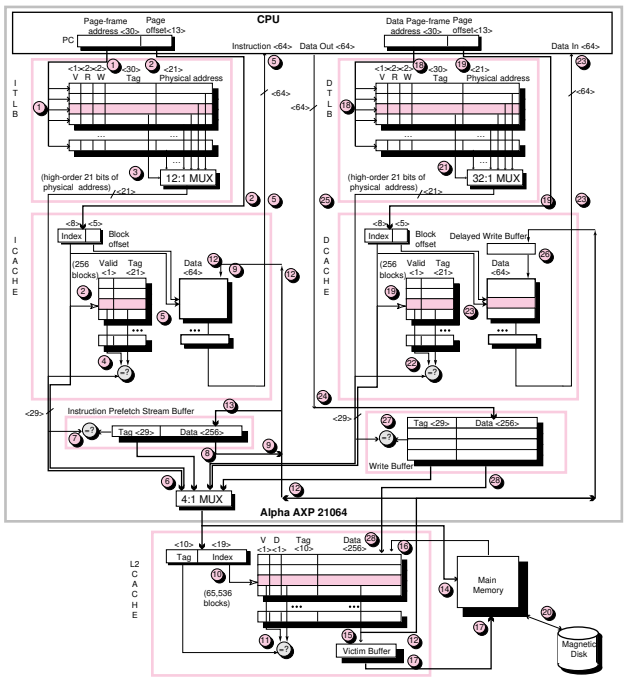
Cross Cutting Issues

- Superscalar CPU & Number Cache Ports
- Speculative Execution and non-faulting option on memory
- Parallel Execution vs. Cache locality
  - Want far separation to find independent operations vs. want reuse of data accesses to avoid misses
- I/O and Consistency of data between cache and memory
  - Caches => multiple copies of data
  - Consistency by HW or by SW?
  - Where connect I/O to computer?

© Patterson 1994

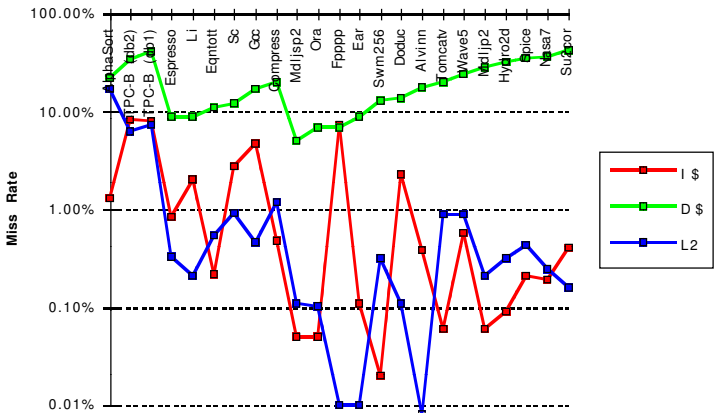
Alpha 21064

- Separate Instr & Data TLB & Caches
- TLBs fully associative
- Caches 8KB direct mapped
- Critical 8 bytes first
- Prefetch instr. stream buffer
- 2 MB L2 cache, direct mapped
- 256 bit path to main memory, 4 64-bit modules



© Patterson 1994

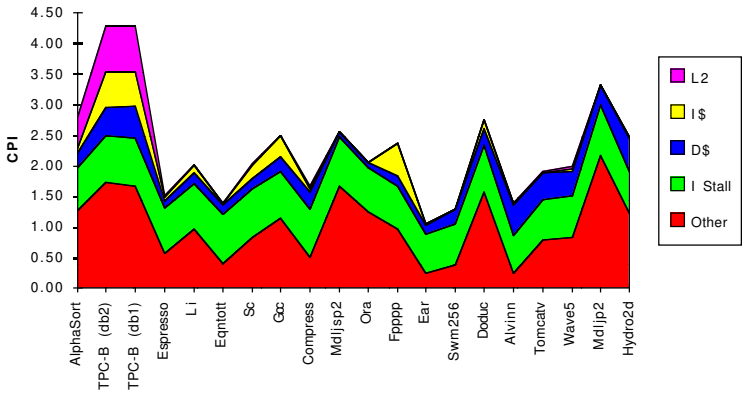
Alpha Memory Performance: Miss Rates



© Patterson 1994

Alpha CPI components

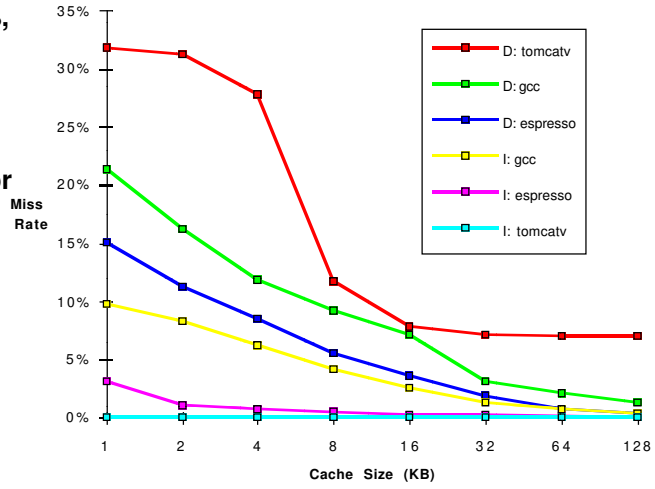
- **Instr stall: branch mispredict;**
  Other: compute + reg conflicts, structural conflicts



© Patterson 1994

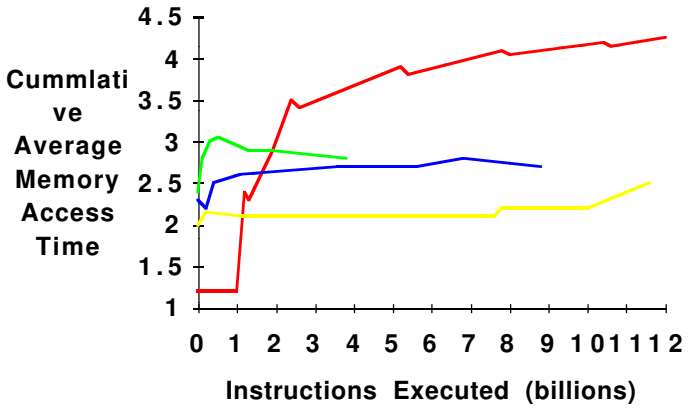
Pitfall: Predicting Cache Performance from Different Program (ISA, compiler, ...)

- **4KB Data cache miss rate 8%,12%, or 28%?**
- **1KB Instr cache miss rate 0%,3%, or 10%?**
- **Alpha vs. MIPS for 8KB Data: 17% vs. 10%**



© Patterson 1994

Pitfall: Simulating Too Small an Address Trace



© Patterson 1994

Highlights from ASPLOS VI

- **Recent conference papers and topics: cache evaluation with different benchmarks, SW branch prediction, memory address disambiguation, OS reducing conflict misses, ...**

© Patterson 1994