


Administrative Matters

- Homework #6
 - Posted
 - Due Friday, 11/30
- Homework #7
 - Posted
 - Due Friday 12/7
- Final Exam
 - Comprehensive
 - Materials from
 - 80% Midterms, quizzes
 - 10% Homeworks
 - 10% Lecture
 - 12/15, 11:30AM-2:30PM

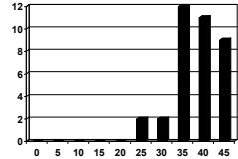


COMPUTER SCIENCE & ENGINEERING

©2004 Morgan Kaufmann Publishers 1

Midterm 2

- 45+: 9
- 40-44: 11
- 35-39: 12
- 30-34: 2
- 0-29: 2
- Average: 40.42 (B-/C+)

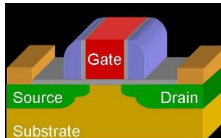


COMPUTER SCIENCE & ENGINEERING

©2004 Morgan Kaufmann Publishers 2

The end of Moore's law?

- Current Trend
 - 45 nm gate length
 - In 2000, gate length is 250nm
- Device scaling
 - Reduce transistor in every dimension
- Gate Oxide
 - Prevent electron from going into channel
 - Currently at 1.2nm
- Size of atom
 - .1 to .5 nanometer!
 - Current gate oxide is about 5 atoms
- How do we scale further!?!?
 - We still need to reduce the gate oxide while maintaining capacitance

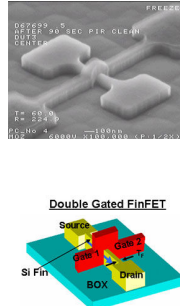


COMPUTER SCIENCE & ENGINEERING

©2004 Morgan Kaufmann Publishers 3

FinFET

- Device engineer will find a way...
- 3-D gate!
- What's to come?
 - 32nm technology!
 - 1.9 billion transistors on a chip

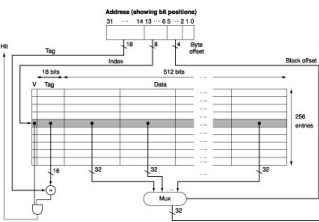


COMPUTER SCIENCE & ENGINEERING

©2004 Morgan Kaufmann Publishers 4

Example Intrinsicity FastMATH processor

- 16 KB cache
- 256 blocks
 - 8 bit index (block offset)
 - Index of blocks in a cache
- 16 words/block
 - 4 bit word offset
 - Index of words in a block
 - Control the Mux
- 2 bit byte offset
 - Index of bytes in a word
- 18 bit tag

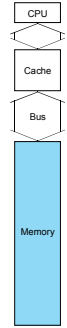


COMPUTER SCIENCE & ENGINEERING

©2004 Morgan Kaufmann Publishers 5

Hardware Issues

- Assume
 - 1 memory bus cycle to send the address
 - 15 memory bus clock cycle to initiate a read
 - 1 memory bus cycle to send a word
 - DRAM does blocks well
 - On a miss of a 4 word block
 - $1 + 4 * 15 + 4 * 1 = 65$

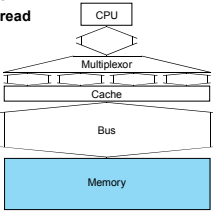


COMPUTER SCIENCE & ENGINEERING

©2004 Morgan Kaufmann Publishers 6

Hardware Issues

- Assume
 - 1 memory bus cycle to send the address
 - 15 memory bus clock cycle to initiate a read
 - 1 memory bus cycle to send a word
 - DRAM does blocks well
- Increase bus bandwidth to 4 words
 - $1 + 1 \times 15 + 1 \times 1 = 17$ cycles
- However, bus may now be slower
 - Lower bus frequency
- Multiplexor may impact hit time
 - Shouldn't make infrequent case (miss) faster by making the frequent case (hit) slower!!!

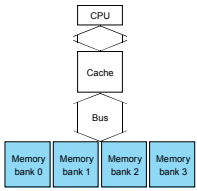


The diagram shows a CPU connected to a Multiplexor, which is connected to a Cache. The Cache is connected to a Bus, which is connected to Memory. The Multiplexor has multiple inputs and outputs, suggesting it can route data to different memory banks.

COMPUTER SCIENCE & ENGINEERING
©2004 Morgan Kaufmann Publishers 7

Hardware Issues

- Assume
 - 1 memory bus cycle to send the address
 - 15 memory bus clock cycle to initiate a read
 - 1 memory bus cycle to send a word
 - DRAM does blocks well
- Memory interleaving
 - Increase memory bandwidth
 - $1 + 1 * 15 + 4 * 1 = 20$ cycles
- Does not impact hit time
- Memory banking is a very common practice



The diagram shows a CPU connected to a Cache, which is connected to a Bus. The Bus is connected to four Memory banks (Memory bank 0, Memory bank 1, Memory bank 2, and Memory bank 3).

COMPUTER SCIENCE & ENGINEERING
©2004 Morgan Kaufmann Publishers 8

Performance

- Simplified model: (compute hit cycles and stall cycles separately)
 - execution time = (execution cycles + stall cycles) x cycle time
 - stall cycles = # of instructions x miss/instruction x miss penalty
- E.g. What is the real CPI?
 - instruction miss of 2%, data cache miss of 4%
 - CPI of 2 without memory stalls, miss penalty of 100
 - I.e. A hit take 2 cycle on the average
 - I.e. A miss take 2+100 cycle
 - 36% of instructions are memory access
 - Instruction stall cycles = $1 * 100% * 2% * 100 = 2 * I$
 - Data miss cycles = $1 * 36% * 4% * 100 = 1.44 * I$
 - Total stall cycles $3.44 * I$
 - Total CPI = $2 + 3.44 = 5.44$
- Quick Question: If an instruction miss on IF, could there be a miss on MEM?

COMPUTER SCIENCE & ENGINEERING
©2004 Morgan Kaufmann Publishers 9

Performance (continue)

- E.g. What is the real CPI?
 - instruction miss of 2%, data cache miss of 4%
 - CPI of 2 without memory stalls, miss penalty of 100
 - 36% of instructions are memory access
- 5.44 when it should be 2
- What if we improve CPI by a factor of 2 (to 1)?
 - 4.44 when it should be 1
- What if we improve clock cycle time of the CPU by a factor of 4?
 - Now 400 miss penalty
 - Miss cycles per instruction = $2% * 400 + 36% * 4% * 400 = 13.76$
 - New CPI = 15.76
 - New performance = $15.76 * 1/4 * Cycle_time = 3.94 * Cycle_time$
 - 4 to 1 speed up only result in 5.44 to 3.94 speedup

COMPUTER SCIENCE & ENGINEERING
©2004 Morgan Kaufmann Publishers 10


Performance

- Simplified model:
 - execution time = (execution cycles + stall cycles) x cycle time
 - stall cycles = # of instructions x miss/instruction x miss penalty
- Two ways of improving performance:
 - decreasing the miss ratio
 - associativity
 - decreasing the miss penalty
 - Multi-level cache (cache for cache)

COMPUTER SCIENCE & ENGINEERING
©2004 Morgan Kaufmann Publishers 11

Associativity

- Direct Mapped cache
 - A block can appear only in one place
- Fully associative cache
 - A block can appear in any place
- N way set-associative cache
 - A block can appear in a set of N places



The diagram shows three cache structures: Direct mapped, Set associative, and Fully associative. Each structure shows a set of cache blocks (Data) and their corresponding tags (Tag) and search paths (Search). In Direct mapped, each block has a unique tag. In Set associative, blocks are grouped into sets, and each set has a single tag. In Fully associative, each block has its own tag.

COMPUTER SCIENCE & ENGINEERING
©2004 Morgan Kaufmann Publishers 12

A continuum of associativity

- Given same space for 8 blocks in the cache:

One-way set associative (direct mapped)

Two-way set associative

Four-way set associative

Eight-way set associative (fully associative)

COMPUTER SCIENCE & ENGINEERING

©2004 Morgan Kaufmann Publishers 13

Decreasing miss ratio with associativity

- Given a 4 block cache
- Given access block address of 0 8 0 6 8
- Direct map cache: cache block 0, 1, 2, 3
- 5 misses

Block address	Cache block
0	(0 modulo 4) = 0
8	(8 modulo 4) = 0
0	(0 modulo 4) = 0
6	(6 modulo 4) = 2
8	(8 modulo 4) = 0

Address of memory block accessed	Hit or miss	Contents of cache blocks after reference			
		0	1	2	3
0	miss	Memory[0]			
8	miss	Memory[0]	Memory[1]		
0	miss	Memory[0]	Memory[1]	Memory[2]	
6	miss	Memory[0]	Memory[1]	Memory[2]	Memory[3]
8	miss	Memory[0]	Memory[1]	Memory[2]	Memory[3]

COMPUTER SCIENCE & ENGINEERING

©2004 Morgan Kaufmann Publishers 14

Decreasing miss ratio with associativity (continue)

- 2-way set-associative cache
- Replacement policy: Least Recent Used
 - 2-way: 1 bits to tell
 - 4-way: 2 bits to tell
- 4 misses
- Fully associative cache (4-way associative cache)
- 3 misses
 - All cold misses, no conflict misses, the best you can do...

Address of memory block accessed	Hit or miss	Contents of cache blocks after reference			
		Set 0	Set 1	Set 0	Set 1
0	miss	Memory[0]			
8	miss	Memory[0]	Memory[1]		
0	hit	Memory[0]	Memory[1]		
6	miss	Memory[0]	Memory[1]	Memory[2]	
8	miss	Memory[0]	Memory[1]	Memory[2]	Memory[3]

Address of memory block accessed	Hit or miss	Contents of cache blocks after reference			
		Block 0	Block 1	Block 2	Block 3
0	miss	Memory[0]			
8	miss	Memory[0]	Memory[1]		
0	hit	Memory[0]	Memory[1]		
6	miss	Memory[0]	Memory[1]	Memory[2]	
8	hit	Memory[0]	Memory[1]	Memory[2]	Memory[3]

COMPUTER SCIENCE & ENGINEERING

©2004 Morgan Kaufmann Publishers 15

Reducing miss rate with associativity

- SPEC2000 on sample cache
 - Direct map: 10.3%
 - 2-way: 8.6%
 - 4-way: 8.3%
 - 8-way: 8.1%
- 2-way is a good idea: 15% improvement
- Any more does not help much
 - Additional comparator is expensive in area/power
 - May actually increase hit time
- Not so fast:
 - Technology number decrease overhead
 - 16-way, 32-way starting to appear...
 - 2nd level cache are usually higher associativity
 - Off-chip access is too slow!!! (to come...)

COMPUTER SCIENCE & ENGINEERING

©2004 Morgan Kaufmann Publishers 16

An implementation

- What's the block size here?

COMPUTER SCIENCE & ENGINEERING

©2004 Morgan Kaufmann Publishers 17

Size of Tags

- Address is broken down into
 - TAG – INDEX (block offset) – Word Offset – Byte Offset
- Direct map
 - Index correspond to size of cache
- 2-way set associative
 - Index = size of cache – 1 bit
 - Tag size increase by 1 bit
- 4-way set associative
 - Index = size of cache – 2 bits
 - Tag size increase by 2 bits
- Fully associative
 - No index
 - All tag

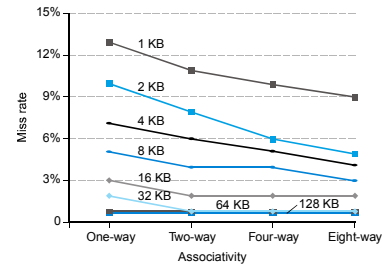
COMPUTER SCIENCE & ENGINEERING

©2004 Morgan Kaufmann Publishers 18

Example for tag size

- If we have a 4 k block cache, 4 word block, 32 bit address
- Direct map
 - Tag + 12 + 2 + 2 = 32, tag = 16 bits
 - 16 bits * 4K = 64Kbits tag total
- 2-way associative
 - Index becomes 11 bits, tag becomes 17 bits
 - 68Kbits Tags total
- 4-way associative
 - 18 * 4K = 72Kbits Tags total
- Fully Associative
 - 28 bits Tag -> 112Kbits total

Performance



What's missing from this picture?

Decreasing miss penalty with multilevel caches

- Add a second level cache:
 - often primary cache is on the same chip as the processor
 - Primary memory (DRAM) is often off-chip
 - Why?
 - use SRAMs to add another cache above primary memory
 - miss penalty goes down if data is in 2nd level cache
- Using multilevel caches:
 - try and optimize the hit time on the 1st level cache
 - Optimize the frequent case
 - Miss penalty of the 1st level cache is o.k. since it is an access to 2nd level
 - try and optimize the miss rate on the 2nd level cache
 - Miss penalty of the 2nd level cache is BAD because it has to go to MM