A Framework for Memory Hierarchies

Associativity schemes

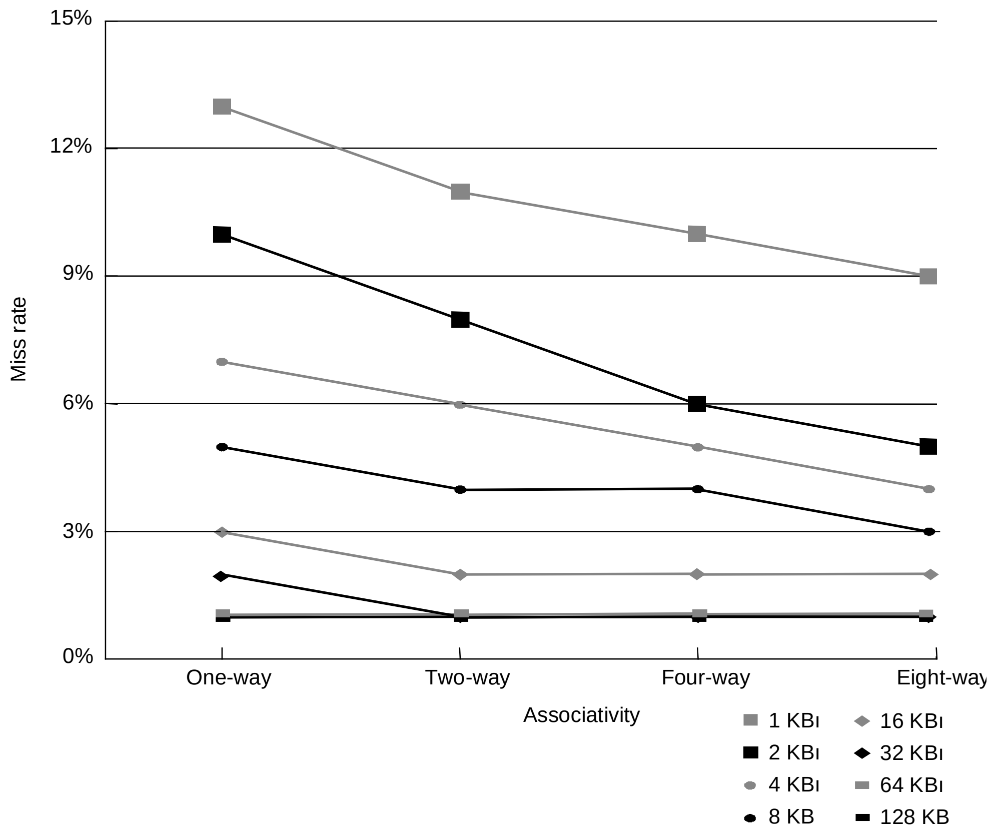
Scheme	Number of sets	Blocks per set
Direct mapped	Number of blocks in cache	1
Set associative	Blocks in cache / Associativity	Associativity (2-8)
Fully associative	1	Number Blocks in cache

Placing blocks

Feature	Typical values for caches	Typical values for VM	Typical values for TLB
Size in blocks	1000 – 100,000	2000 – 250,000	32 – 4,000
Size in KB	8 – 8,000	8000 – 8,000,000	0.25 – 32
Block size in bytes	16 – 256	4000 – 64,000	4 – 32
Miss penalty (clocks)	10 - 100	1,000,000 – 10,000,000	10 – 100
Miss rate	0.1% - 10%	0.00001% - 0.0001%	0.01% - 2%

A Framework for Memory Hierarchies

Miss rates



A Framework for Memory Hierarchies

Finding blocks

Associativity	**Location method**	**Comparisons required**
Direct mapped	Index	1
Set associative	Index the set, search among elements	Degree of associativity
Full	Search all cache entries	Size of the cache
	Separate lookup table	0

Why do we use full associativity and a separate lookup table (page table) in VM:

- Misses are very expensive
- Full associativity allows software to use sophisticated replacement schemes to reduce miss rate
- Full mapping table (all pages have entries) allows indexing with no extra hardware and no searching.
- The large page size means that the page table overhead is relatively small.

A Framework for Memory Hierarchies

Choosing a block to replace

• Random choice: The blocks are randomly selected, possibly using some hardware assistance. Used mainly larger caches

• Least recently used (LRU): The block replaced is the one that has been unused for the longest time. Used in VM (reference bit).

Writing blocks

Write-through:

- Read misses are simpler and cheaper because they do not require writing blocks to the lower level.
- Easier to implement, although normally a write-buffer is required

Write-back:

- Individual words are written at the rate of the cache
- Multiple writes within a block require only one write to the lower level memory
- When blocks are written back a high bandwidth transfer can be used since the entire block is written

A Framework for Memory Hierarchies

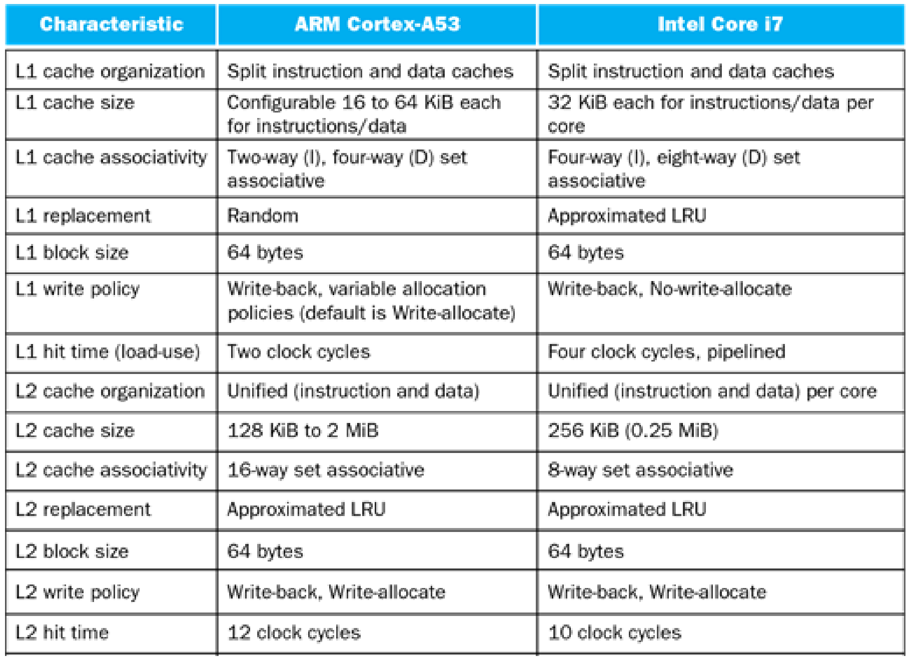
The sources of misses

• *Compulsory misses* (cold-start misses): First time access to a block that has never been in the cache
• *Capacity misses*: The cache cannot contain all the blocks needed during the execution of a program. These misses occur because of blocks being replaced are later retrieved again.
• *Conflict misses* (collision misses): Misses that occur in direct or set associative schemes. Multiple blocks compete for the same set (or entry) in the cache.

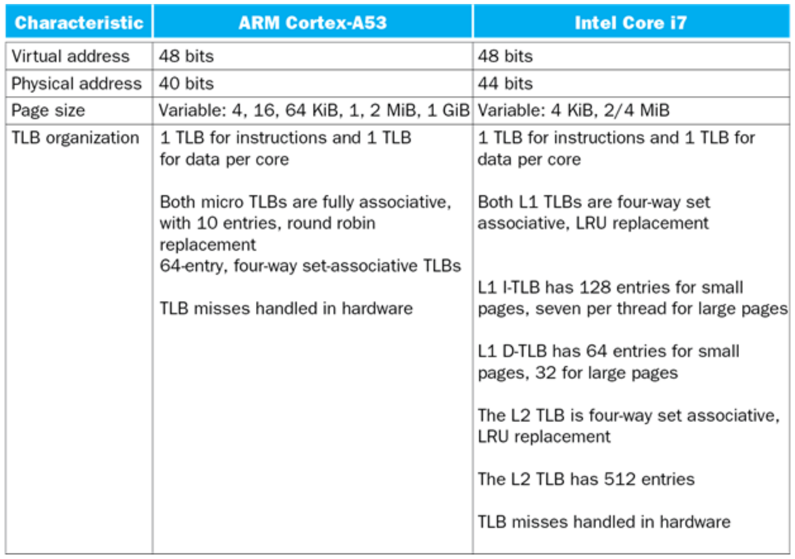
The challenge

Design change	**Effect on miss rate**	**Possible negative effect on performance**
Increase size	Decreases capacity misses	May increase access time
Increase associativity	Decreases conflict misses	May increase access time
Increase block size	Decreases miss rate	May increase miss penalty

A Framework for Memory Hierarchies



Characteristic	ARM Cortex-A53	Intel Core i7
L1 cache organization	Split instruction and data caches	Split instruction and data caches
L1 cache size	Configurable 16 to 64 KiB each for instructions/data	32 KiB each for instructions/data per core
L1 cache associativity	Two-way (I), four-way (D) set associative	Four-way (I), eight-way (D) set associative
L1 replacement	Random	Approximated LRU
L1 block size	64 bytes	64 bytes
L1 write policy	Write-back, variable allocation policies (default is Write-allocate)	Write-back, No-write-allocate
L1 hit time (load-use)	Two clock cycles	Four clock cycles, pipelined
L2 cache organization	Unified (instruction and data)	Unified (instruction and data) per core
L2 cache size	128 KiB to 2 MiB	256 KiB (0.25 MiB)
L2 cache associativity	16-way set associative	8-way set associative
L2 replacement	Approximated LRU	Approximated LRU
L2 block size	64 bytes	64 bytes
L2 write policy	Write-back, Write-allocate	Write-back, Write-allocate
L2 hit time	12 clock cycles	10 clock cycles



Characteristic	ARM Cortex-A53	Intel Core i7
Virtual address	48 bits	48 bits
Physical address	40 bits	44 bits
Page size	Variable: 4, 16, 64 KiB, 1, 2 MiB, 1 GiB	Variable: 4 KiB, 2/4 MiB
TLB organization	1 TLB for instructions and 1 TLB for data per core Both micro TLBs are fully associative, with 10 entries, round robin replacement 64-entry, four-way set-associative TLBs TLB misses handled in hardware	1 TLB for instructions and 1 TLB for data per core Both L1 TLBs are four-way set associative, LRU replacement L1 I-TLB has 128 entries for small pages, seven per thread for large pages L1 D-TLB has 64 entries for small pages, 32 for large pages The L2 TLB is four-way set associative, LRU replacement The L2 TLB has 512 entries TLB misses handled in hardware