Microprocessor & Computer Architecture (µpCA)

UE19CS252

Dr. D. C. Kiran

Department of
Computer Science and Engineering

Microprocessor & Computer Architecture (μpCA)

Unit 4: Cache Optimization

Dr. D. C. Kiran

Department of Computer Science and Engineering

Microprocessor & Computer Architecture (µpCA)

Syllabus

~~Unit 1: Basic Processor Architecture and Design~~

~~Unit 2: Pipelined Processor and Design~~

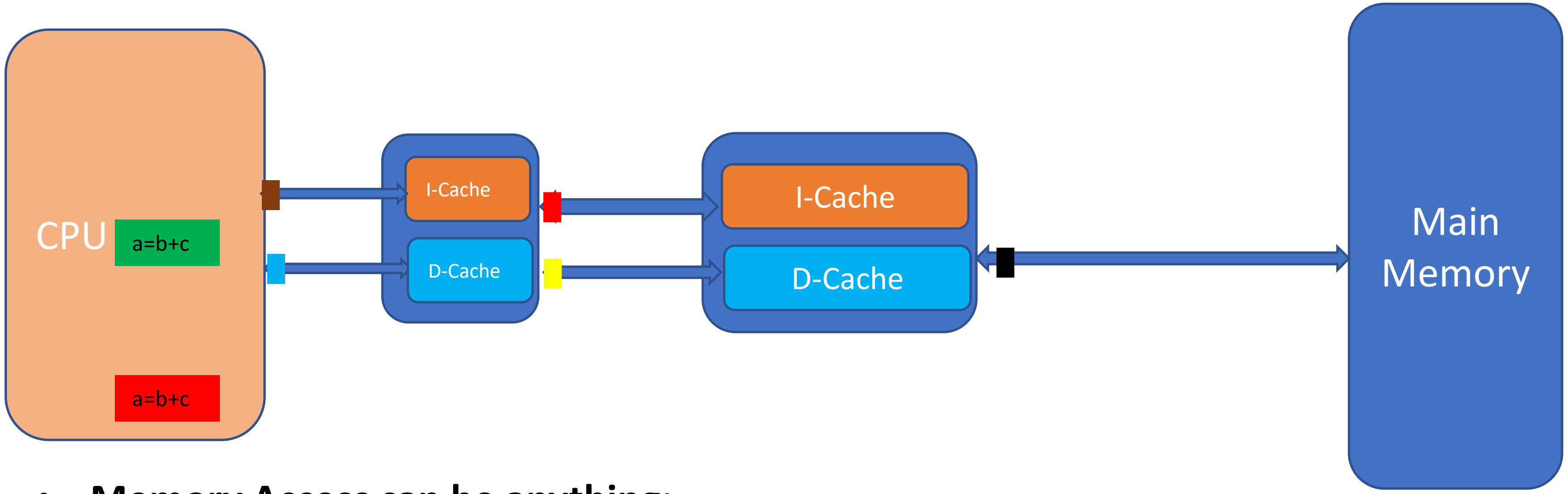
~~Unit 3: Memory~~

Unit 4: Input/Output Device Design

Cache Optimization

Unit 5: Advanced Architecture

Why Average Memory Access Time?



- **Memory Access can be anything:**
  Instruction Fetch form I-Cache in L1
  Instruction Fetch from I-Cache in L2, if Miss in I-Cache of L1
  Data Access from D-Cache in L1.
  Data Access from D-Cache in L2, if Miss in D-Cache of L1
- **Hit time for L1 Cache is different Hit time for L2 Cache**
- **Miss Penalty for L1 Cache is different from Miss Penalty for L2 Cache**
- **Access depend on other factors such as Width of the bus, External & Internal Cache**

Cache Optimization

Aim: To Improve the Performance of the cache

- What is Performance of the Cache?

  Reduced Access time, to keep CPU Busy.

- How to Improve the Performance of the Cache?

?

Average Memory Access Time

AMAT = Hit Time + (Miss Rate x Miss Penalty)

Cache Performance can be improved by

- Reducing the miss rate
- Reducing the miss penalty
- Reducing the time to hit in the cache

AMAT = Hit Time + (Miss Rate x Miss Penalty)

Cache Performance can be improved by

Six basic cache optimizations:

- Larger block size.
- Larger total **cache** capacity to reduce miss rate.
- Higher associativity.
- Higher number of **cache** levels.
- Giving priority to read misses over write.
- Avoiding address translation in **cache** indexing

AMAT = Hit Time + (Miss Rate x Miss Penalty)

Cache Performance can be improved by

- Reducing the miss rate
  - Larger block size, Larger cache size, and higher associativity.
- Reducing the miss penalty
  - Multilevel caches and giving reads priority over the writes.
- Reducing the time to hit in the cache
  - Avoiding address translation when indexing the cache.

Reducing Miss Rate!!!

What is a MISS ☹?

Three Categories Misses

- **Compulsory :**
  - The very first access to block cannot be in the cache. So, the block must be brought into the cache.
  - These are called cold-start misses or first reference misses.
- **Capacity:**
  - If the cache cannot contain all the blocks needed during execution of a program, capacity misses [in addition to compulsory misses] will occur because of blocks being discarded and later retrieved.

Situation: item has been in the cache, but space was tight, and it was forced out

- **Conflict:**
  - If the block placement strategy is set associative or direct mapped, conflict misses [in addition to compulsory and capacity misses] will occur because a block may be discarded and later retrieved if too many blocks map to its set. These misses are also called collision misses.
  - The idea is that hits in a fully associative cache that become misses in an n-way set associative cache, due to more than n requests on some popular sets.

Situation: item was in the cache, but the cache was not associative enough, so it was forced out

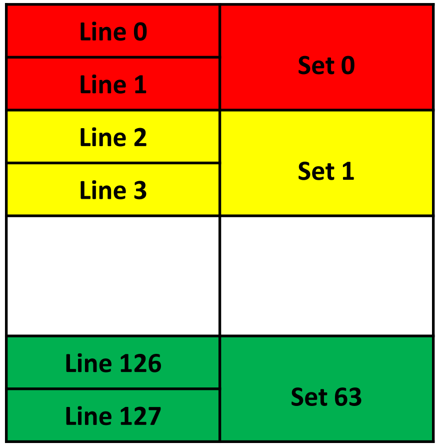
Cache Optimization: Reduce Miss Rates.

How to Categories the Miss

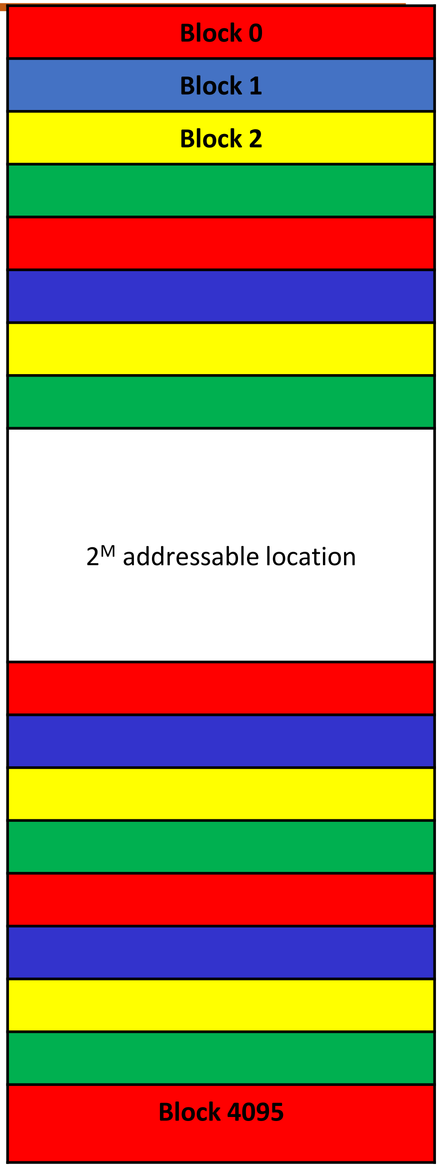
- **Compulsory Misses :**
  - Those that occur in an infinite cache.
- **Capacity Misses:**
  - Those that occur in a fully associative cache.
- **Conflict Misses:**
  - Those that occur going from fully associative to eight-way associative, four–way associative , and so on…

Example: How to Categories the Miss

Consider a 2-way set associative cache with 128 Lines and 64 Sets



Line 0	Set 0
Line 1	
Line 2	Set 1
Line 3	
Line 126	Set 63
Line 127	

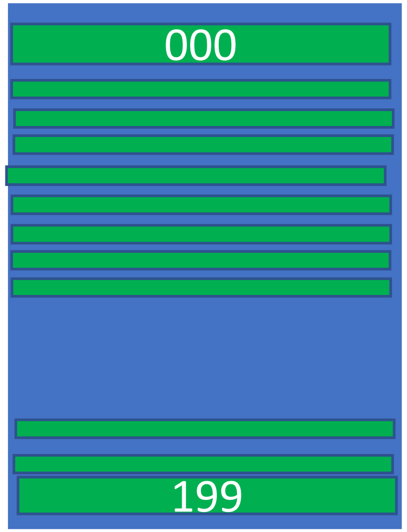


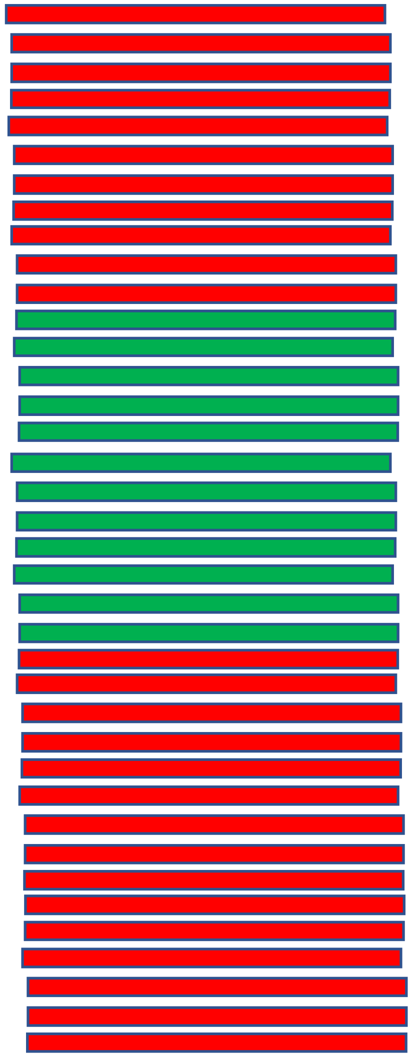
If you come across 1000 Misses, when a program is executed.

How many Compulsory Misses?
How many Capacity Misses?
How many Conflict Misses?

Step1: Identifying Compulsory Miss

Execute the program on an Infinite Size Cache and Fully Associative



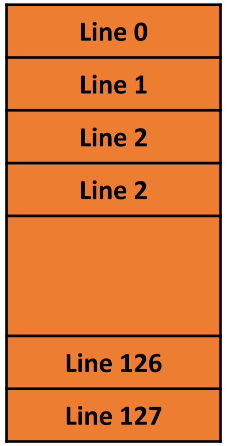


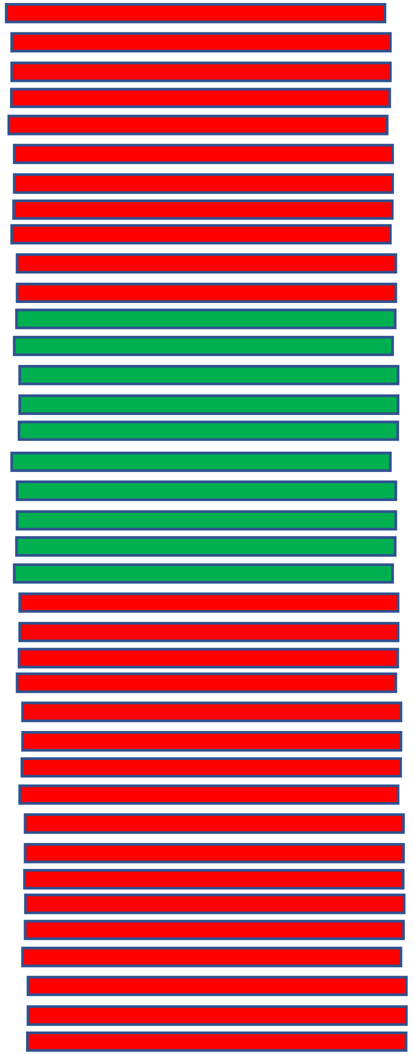
- Suppose there are 200 Misses
- When data is accessed for 1st time, you will have 200 Misses.
- Other Access are Hit.
- So 200 out of 1000 Misses are ***Compulsory Miss***

Step2: Identifying Capacity Miss

Execute the program on a 128 Line Cache and Fully Associative.

128 Line Cache and Fully Associative



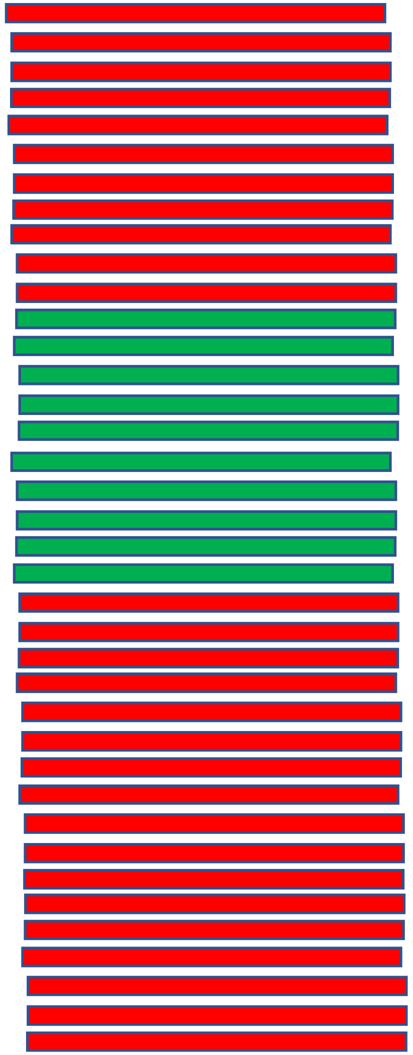


If the number of Misses are 400, 200 out of 1000 are Capacity Misses, as 200 are Compulsory Misses.

Step3: Conflict Miss

Execute the same program on 128 Line, 2-way set Associative cache which will give 1000 Misses.

Line 0	Set 0
Line 1	
Line 2	Set 1
Line 3	
Line 126	Set 63
Line 127	

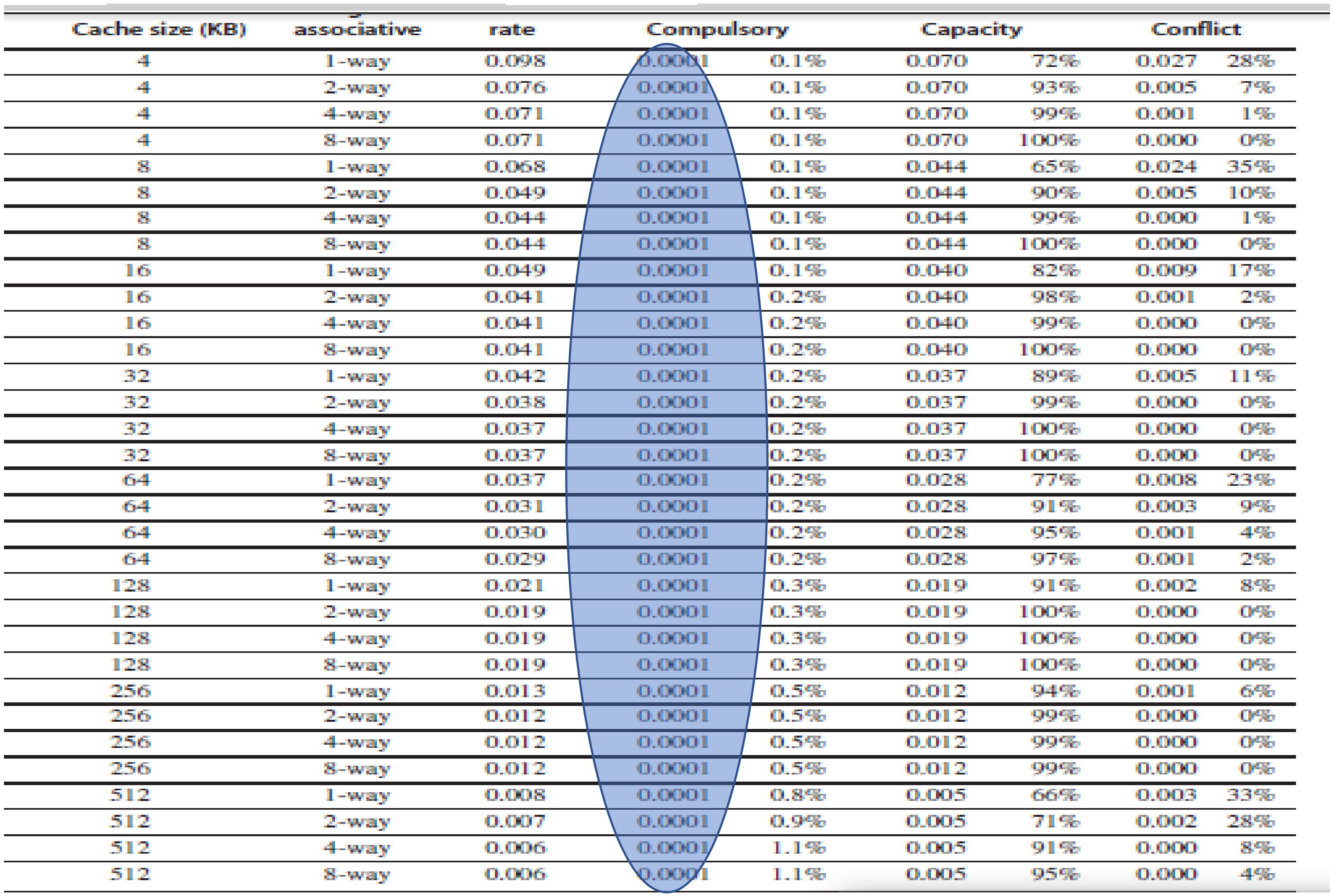


200 Compulsory Misses
200 Capacity Misses
600 Conflict Misses.

Few Facts of 3 C’s

Microprocessor & Computer Architecture (μpCA)

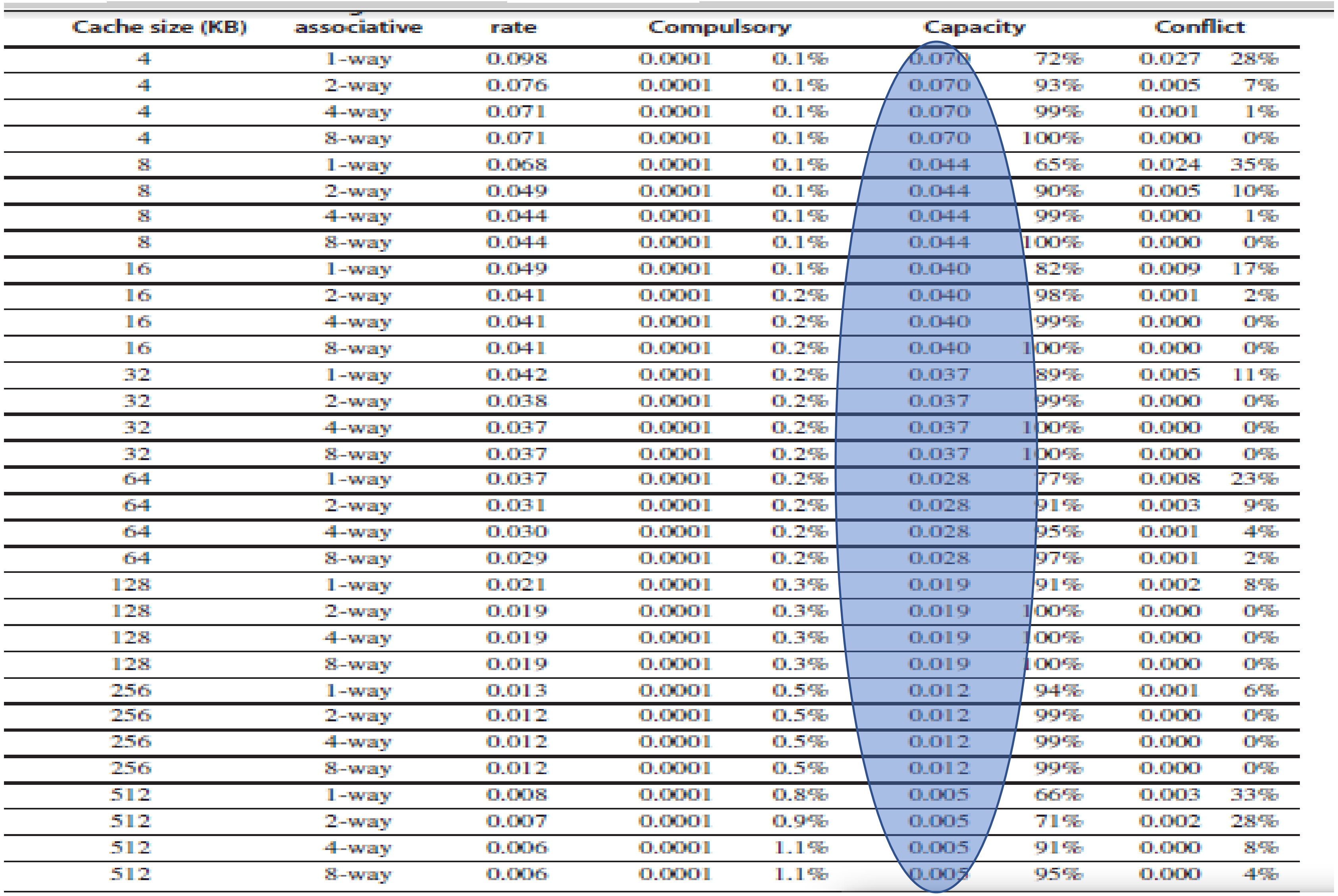
Compulsory Misses are Independent of Cache Size



Cache size (KB)	associative	rate	Compulsory		Capacity		Conflict	
4	1-way	0.098	0.0001	0.1%	0.070	72%	0.027	28%
4	2-way	0.076	0.0001	0.1%	0.070	93%	0.005	7%
4	4-way	0.071	0.0001	0.1%	0.070	99%	0.001	1%
4	8-way	0.071	0.0001	0.1%	0.070	100%	0.000	0%
8	1-way	0.068	0.0001	0.1%	0.044	65%	0.024	35%
8	2-way	0.049	0.0001	0.1%	0.044	90%	0.005	10%
8	4-way	0.044	0.0001	0.1%	0.044	99%	0.000	1%
8	8-way	0.044	0.0001	0.1%	0.044	100%	0.000	0%
16	1-way	0.049	0.0001	0.1%	0.040	82%	0.009	17%
16	2-way	0.041	0.0001	0.2%	0.040	98%	0.001	2%
16	4-way	0.041	0.0001	0.2%	0.040	99%	0.000	0%
16	8-way	0.041	0.0001	0.2%	0.040	100%	0.000	0%
32	1-way	0.042	0.0001	0.2%	0.037	89%	0.005	11%
32	2-way	0.038	0.0001	0.2%	0.037	99%	0.000	0%
32	4-way	0.037	0.0001	0.2%	0.037	100%	0.000	0%
32	8-way	0.037	0.0001	0.2%	0.037	100%	0.000	0%
64	1-way	0.037	0.0001	0.2%	0.028	77%	0.008	23%
64	2-way	0.031	0.0001	0.2%	0.028	91%	0.003	9%
64	4-way	0.030	0.0001	0.2%	0.028	95%	0.001	4%
64	8-way	0.029	0.0001	0.2%	0.028	97%	0.001	2%
128	1-way	0.021	0.0001	0.3%	0.019	91%	0.002	8%
128	2-way	0.019	0.0001	0.3%	0.019	100%	0.000	0%
128	4-way	0.019	0.0001	0.3%	0.019	100%	0.000	0%
128	8-way	0.019	0.0001	0.3%	0.019	100%	0.000	0%
256	1-way	0.013	0.0001	0.5%	0.012	94%	0.001	6%
256	2-way	0.012	0.0001	0.5%	0.012	99%	0.000	0%
256	4-way	0.012	0.0001	0.5%	0.012	99%	0.000	0%
256	8-way	0.012	0.0001	0.5%	0.012	99%	0.000	0%
512	1-way	0.008	0.0001	0.8%	0.005	66%	0.003	33%
512	2-way	0.007	0.0001	0.9%	0.005	71%	0.002	28%
512	4-way	0.006	0.0001	1.1%	0.005	91%	0.000	8%
512	8-way	0.006	0.0001	1.1%	0.005	95%	0.000	4%

Microprocessor & Computer Architecture (μpCA)

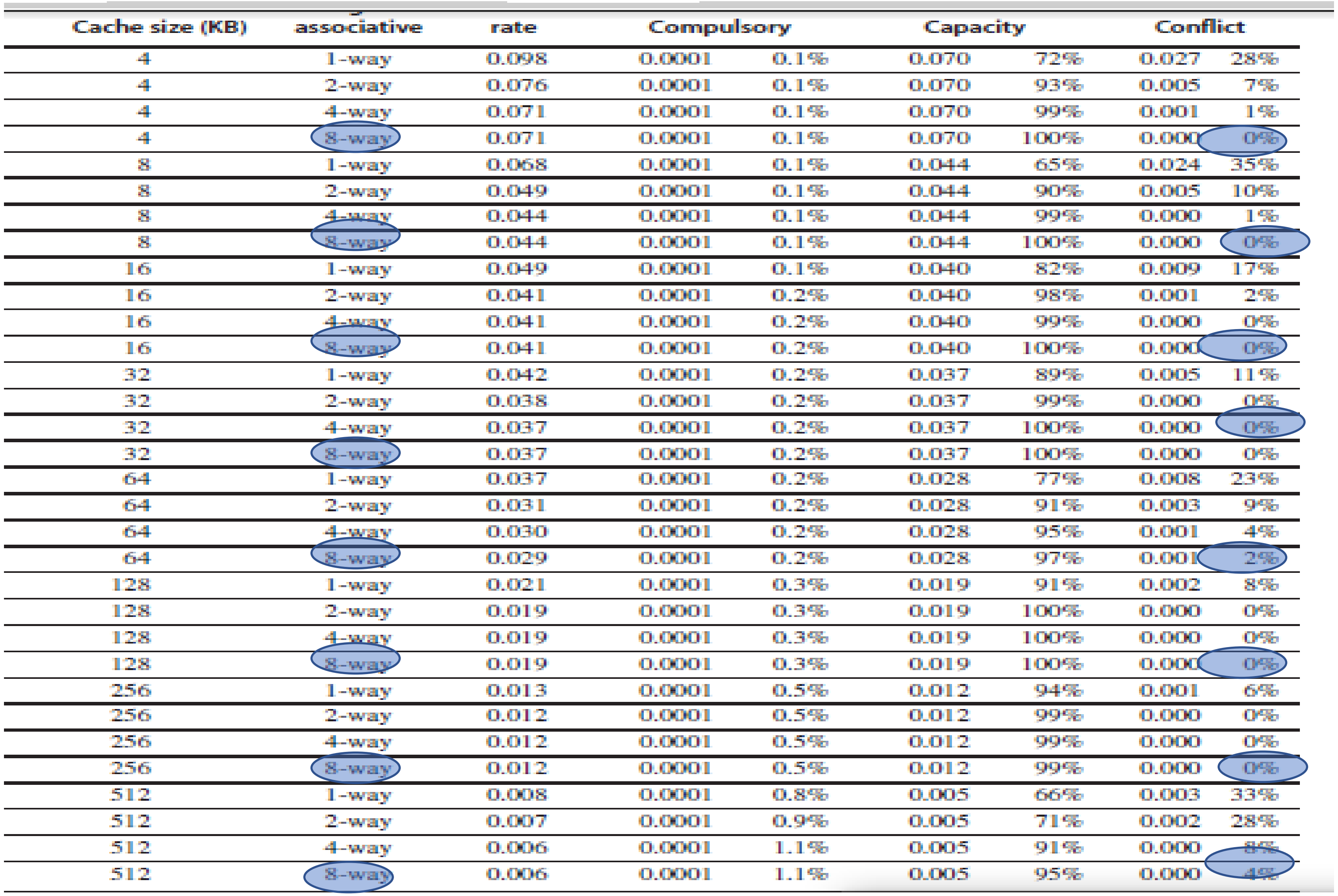
Capacity Misses Decrease as Cache Size Increases



Cache size (KB)	associative	rate	Compulsory		Capacity		Conflict	
4	1-way	0.098	0.0001	0.1%	0.070	72%	0.027	28%
4	2-way	0.076	0.0001	0.1%	0.070	93%	0.005	7%
4	4-way	0.071	0.0001	0.1%	0.070	99%	0.001	1%
4	8-way	0.071	0.0001	0.1%	0.070	100%	0.000	0%
8	1-way	0.068	0.0001	0.1%	0.044	65%	0.024	35%
8	2-way	0.049	0.0001	0.1%	0.044	90%	0.005	10%
8	4-way	0.044	0.0001	0.1%	0.044	99%	0.000	1%
8	8-way	0.044	0.0001	0.1%	0.044	100%	0.000	0%
16	1-way	0.049	0.0001	0.1%	0.040	82%	0.009	17%
16	2-way	0.041	0.0001	0.2%	0.040	98%	0.001	2%
16	4-way	0.041	0.0001	0.2%	0.040	99%	0.000	0%
16	8-way	0.041	0.0001	0.2%	0.040	100%	0.000	0%
32	1-way	0.042	0.0001	0.2%	0.037	89%	0.005	11%
32	2-way	0.038	0.0001	0.2%	0.037	99%	0.000	0%
32	4-way	0.037	0.0001	0.2%	0.037	100%	0.000	0%
32	8-way	0.037	0.0001	0.2%	0.037	100%	0.000	0%
64	1-way	0.037	0.0001	0.2%	0.028	77%	0.008	23%
64	2-way	0.031	0.0001	0.2%	0.028	91%	0.003	9%
64	4-way	0.030	0.0001	0.2%	0.028	95%	0.001	4%
64	8-way	0.029	0.0001	0.2%	0.028	97%	0.001	2%
128	1-way	0.021	0.0001	0.3%	0.019	91%	0.002	8%
128	2-way	0.019	0.0001	0.3%	0.019	100%	0.000	0%
128	4-way	0.019	0.0001	0.3%	0.019	100%	0.000	0%
128	8-way	0.019	0.0001	0.3%	0.019	100%	0.000	0%
256	1-way	0.013	0.0001	0.5%	0.012	94%	0.001	6%
256	2-way	0.012	0.0001	0.5%	0.012	99%	0.000	0%
256	4-way	0.012	0.0001	0.5%	0.012	99%	0.000	0%
256	8-way	0.012	0.0001	0.5%	0.012	99%	0.000	0%
512	1-way	0.008	0.0001	0.8%	0.005	66%	0.003	33%
512	2-way	0.007	0.0001	0.9%	0.005	71%	0.002	28%
512	4-way	0.006	0.0001	1.1%	0.005	91%	0.000	8%
512	8-way	0.006	0.0001	1.1%	0.005	95%	0.000	4%

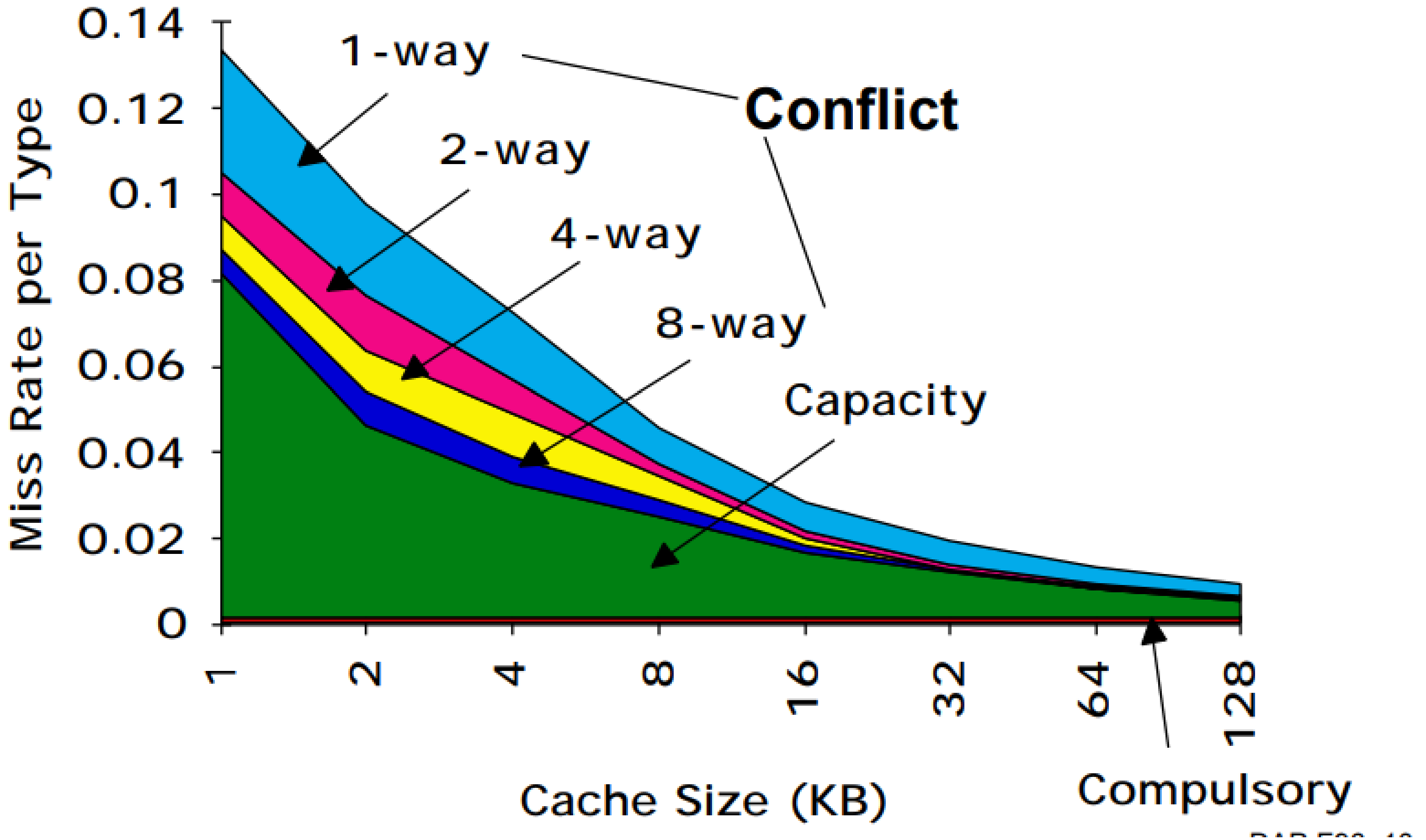
Microprocessor & Computer Architecture (μpCA)

Conflict Miss Decreases as the Associativity Increases



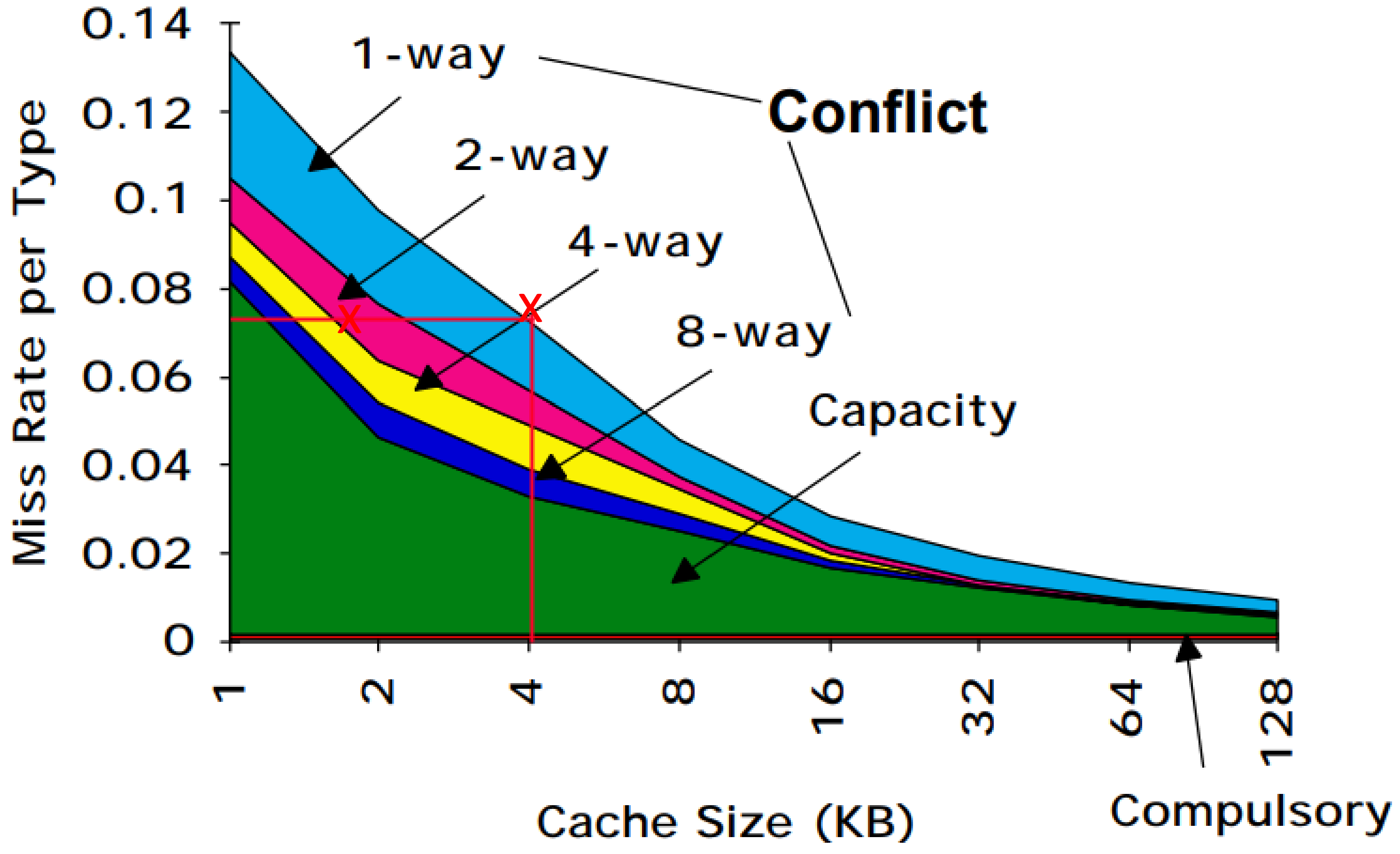
Cache size (KB)	associative	rate	Compulsory		Capacity		Conflict	
4	1-way	0.098	0.0001	0.1%	0.070	72%	0.027	28%
4	2-way	0.076	0.0001	0.1%	0.070	93%	0.005	7%
4	4-way	0.071	0.0001	0.1%	0.070	99%	0.001	1%
4	8-way	0.071	0.0001	0.1%	0.070	100%	0.000	0%
8	1-way	0.068	0.0001	0.1%	0.044	65%	0.024	35%
8	2-way	0.049	0.0001	0.1%	0.044	90%	0.005	10%
8	4-way	0.044	0.0001	0.1%	0.044	99%	0.000	1%
8	8-way	0.044	0.0001	0.1%	0.044	100%	0.000	0%
16	1-way	0.049	0.0001	0.1%	0.040	82%	0.009	17%
16	2-way	0.041	0.0001	0.2%	0.040	98%	0.001	2%
16	4-way	0.041	0.0001	0.2%	0.040	99%	0.000	0%
16	8-way	0.041	0.0001	0.2%	0.040	100%	0.000	0%
32	1-way	0.042	0.0001	0.2%	0.037	89%	0.005	11%
32	2-way	0.038	0.0001	0.2%	0.037	99%	0.000	0%
32	4-way	0.037	0.0001	0.2%	0.037	100%	0.000	0%
32	8-way	0.037	0.0001	0.2%	0.037	100%	0.000	0%
64	1-way	0.037	0.0001	0.2%	0.028	77%	0.008	23%
64	2-way	0.031	0.0001	0.2%	0.028	91%	0.003	9%
64	4-way	0.030	0.0001	0.2%	0.028	95%	0.001	4%
64	8-way	0.029	0.0001	0.2%	0.028	97%	0.001	2%
128	1-way	0.021	0.0001	0.3%	0.019	91%	0.002	8%
128	2-way	0.019	0.0001	0.3%	0.019	100%	0.000	0%
128	4-way	0.019	0.0001	0.3%	0.019	100%	0.000	0%
128	8-way	0.019	0.0001	0.3%	0.019	100%	0.000	0%
256	1-way	0.013	0.0001	0.5%	0.012	94%	0.001	6%
256	2-way	0.012	0.0001	0.5%	0.012	99%	0.000	0%
256	4-way	0.012	0.0001	0.5%	0.012	99%	0.000	0%
256	8-way	0.012	0.0001	0.5%	0.012	99%	0.000	0%
512	1-way	0.008	0.0001	0.8%	0.005	66%	0.003	33%
512	2-way	0.007	0.0001	0.9%	0.005	71%	0.002	28%
512	4-way	0.006	0.0001	1.1%	0.005	91%	0.000	8%
512	8-way	0.006	0.0001	1.1%	0.005	95%	0.000	4%

Microprocessor & Computer Architecture (μpCA)



2:1 Rule

miss rate 1-way associative cache size X = miss rate 2-way associative cache size X/2



Reducing Miss Rate

Larger block size (to Reduce Compulsory Miss)

•Larger total **cache** capacity to reduce miss rate. (To Reduce Miss Capacity Miss)

•Higher associativity. (To Reduce Conflict Miss)

PES UNIVERSITY ONLINE

THANK YOU

Dr. D. C. Kiran

Department of Computer Science and Engineering

dckiran@pes.edu

9829935135