

Aidan Lopez

Assignment: Caches

Instructor: Elaheh Sadredini

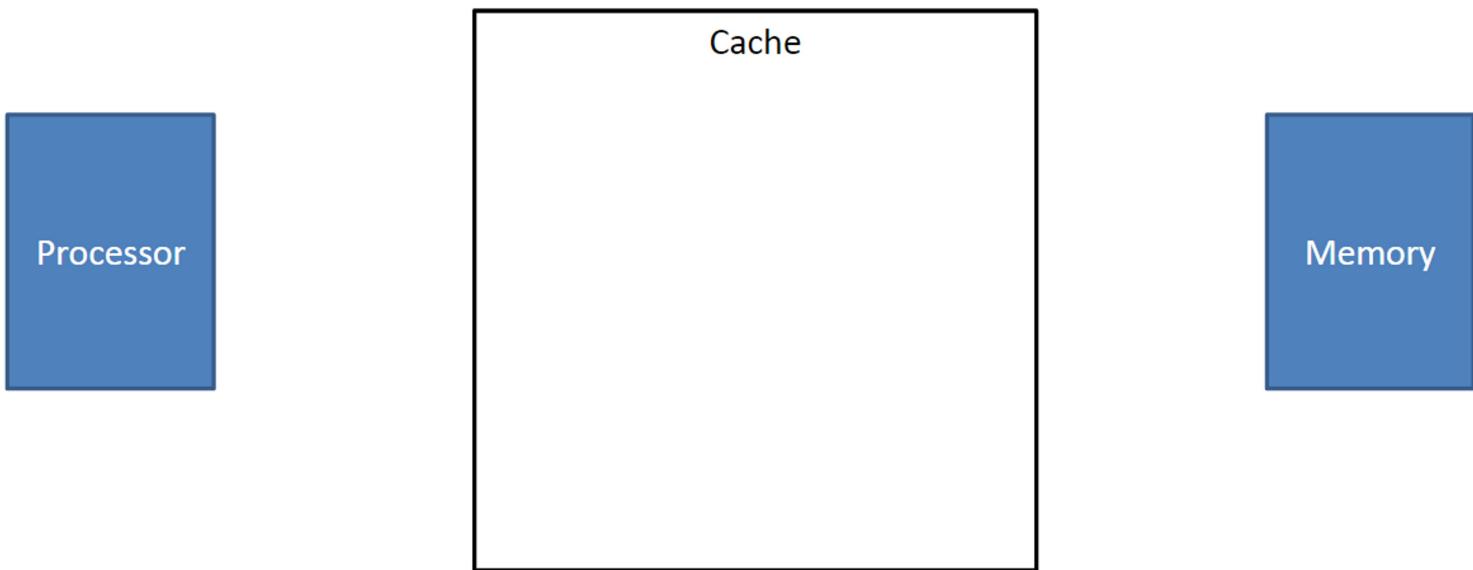
elaheh@cs.ucr.edu

University of California, Riverside

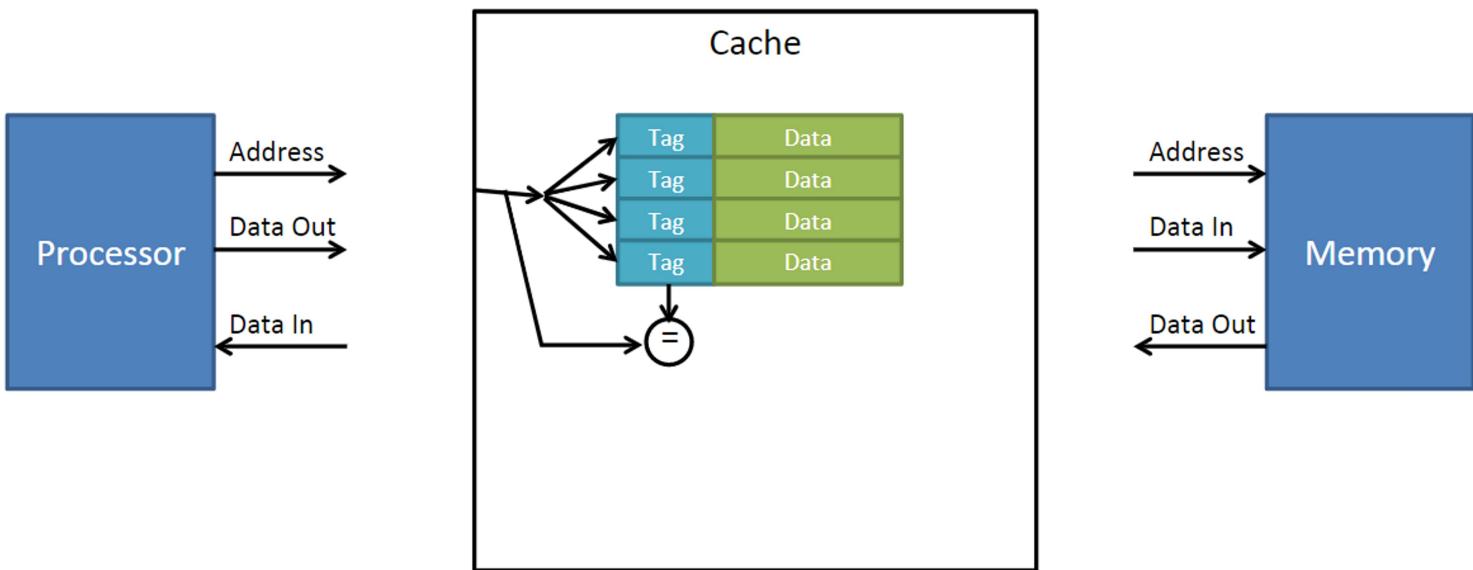
Grading (total 4.2 points)

- Question 1: 1 point
 - Part B: 0.5 points, Part C: 0.5 points
- Question 2: 1.2 point
 - Each part 0.4 point
- Question 3: 1.4 point
 - Part A, B, C each 0.4 point, Part C 0.2 point
- Question 4: 0.6 point
 - Each part 0.2 point

Review: What goes into and out from a cache?

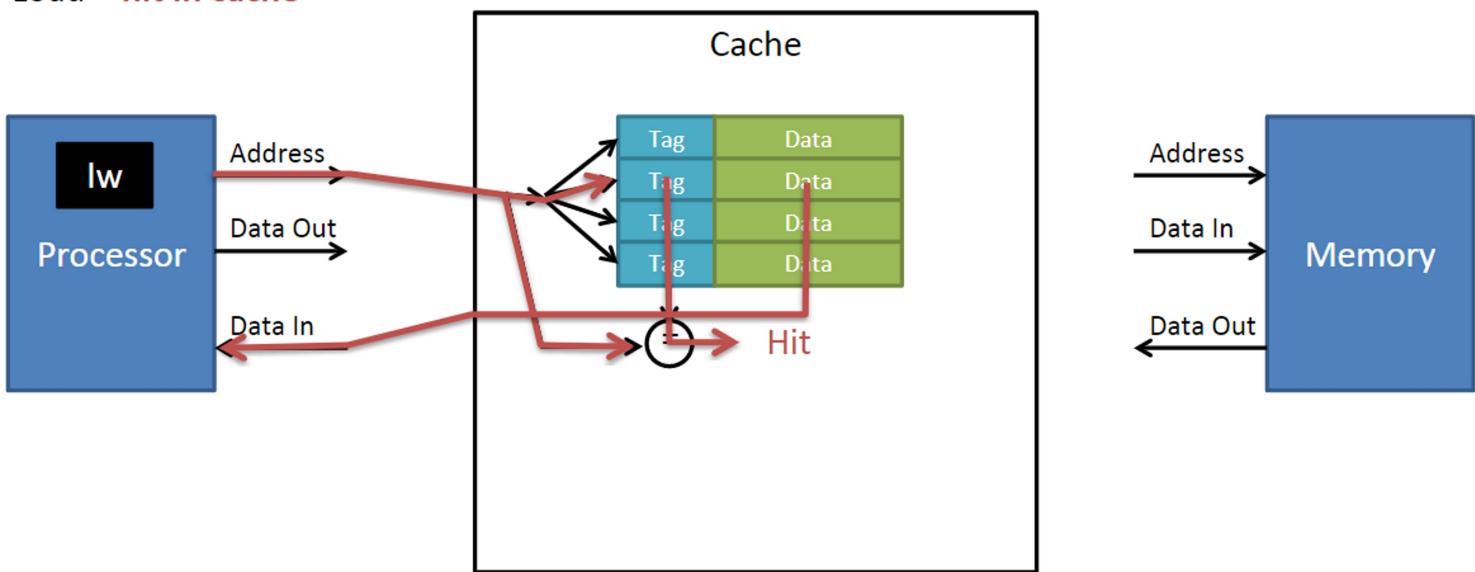


What goes into and out from a cache?



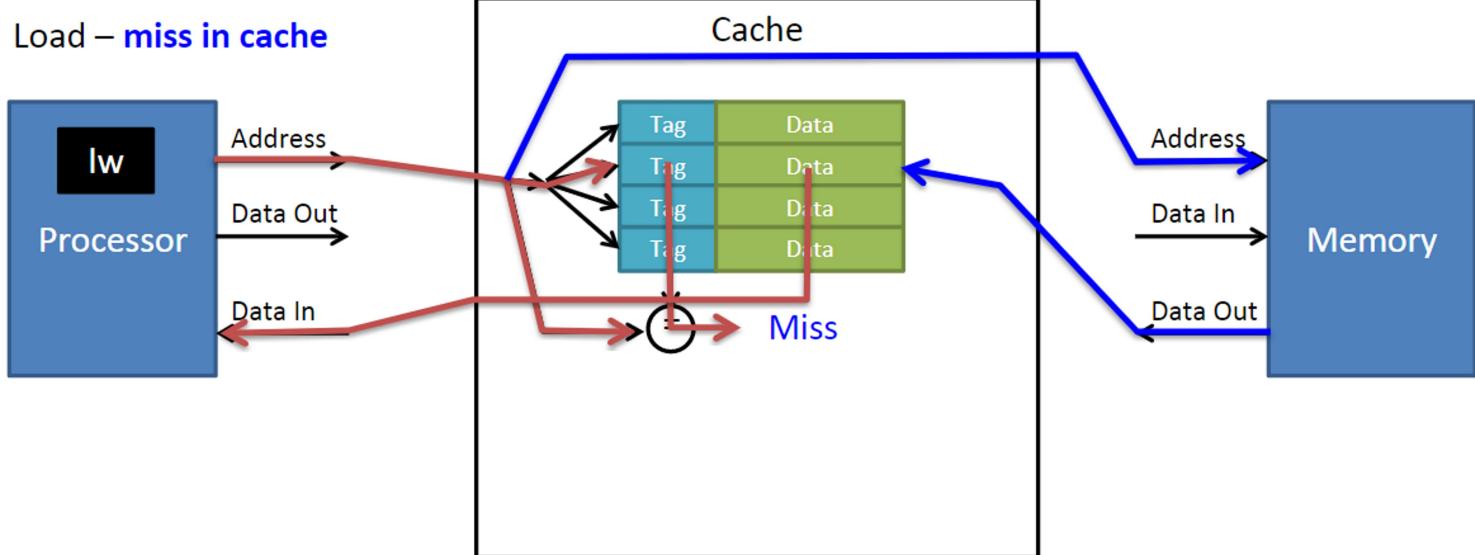
What goes into and out from a cache?

Load – hit in cache



What goes into and out from a cache?

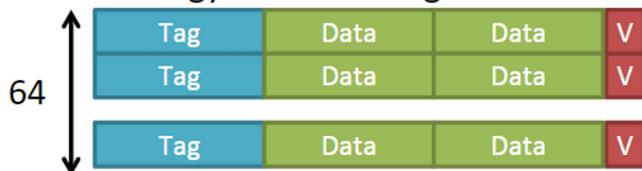
Load – miss in cache



Question 1: Diagram of the Caches

Draw a diagram of the following caches (note: answer for part A is provided. Follow the same structure for part B and C)

- A. Direct-mapped, cache capacity = 64 cache line, cache line size = 8-byte, word size = 4-byte, write strategy: write through



- B. Fully-associative, cache capacity = 256 cache line, cache line size = 16-byte, word size = 4-byte, write strategy: write back

- C. 4-way set-associative, cache capacity = 4096 cache line, cache line size = 64-byte, word size = 4-byte, write strategy: write back

Answer

B.

256

tag	data	data	data	data	v	d
tag	data	data	data	data	v	d
tag	data	data	data	data	v	d

C.

0 ... 16

set 0 {

⋮

set 1023 {

tag	data		data	v	d
tag	data		data	v	d
tag	data		data	v	d
tag	data		data	v	d
tag	data		data	v	d
tag	data		data	v	d
tag	data		data	v	d
tag	data		data	v	d

Question 2: How are the bits used for addressing?

- Determine which bits in a 32-bit address are used for selecting the **byte (B)**, selecting the **word (W)**, indexing the **cache (I)**, and the cache **tag (T)**, for each of the following caches:
- Direct-mapped, cache capacity = 64 cache line, cache line size = 8-byte, word size = 4-byte
 - Fully-associative, cache capacity = 256 cache line, cache line size = 16-byte, word size = 4-byte
 - 4-way set-associative, cache capacity = 4096 cache line, cache line size = 64-byte, word size = 4-byte

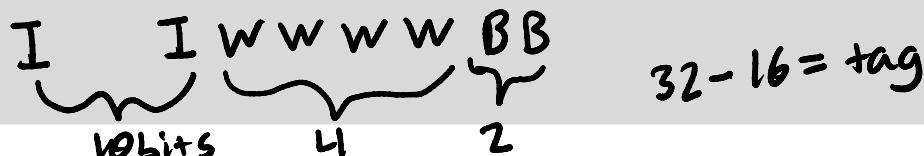
Note: cache capacity represents the maximum number of cache blocks (or cache lines) that can fit in the cache

Answer

$$A. B=2, W=1, I=0, T=32-3=29$$

$$B. B=2, W=2, I=0, T=32-4=28$$

$$C. \frac{64}{4} = 16 \text{ words} \quad \frac{4096}{64} = 1024 \quad B=2, W=4, I=10, T=16$$



Question 3: How much overhead for the caches?

- Calculate the number of bits used for actual data and overhead (i.e., tag, valid bit, dirty

Question 3: How much overhead for the caches?

- Calculate the number of bits used for actual data and overhead (i.e., tag, valid bit, dirty bit) for each of the following caches
- Direct-mapped, cache capacity = 64 cache line, cache line size = 8-byte, word size = 4-byte, **write strategy: write through**
 - Fully-associative, cache capacity = 256 cache line, cache line size = 16-byte, word size = 4-byte, **write strategy: write back**
 - 4-way set-associative, cache capacity = 4096 cache line, cache line size = 64-byte, word size = 4-byte, **write strategy: write back**
 - How the cache line size affect the overhead?

Answer

A. $V=1$ bit, $tag=29$, 30 meta/line

$$64 \times 30 = 1920 \text{ bits (overhead)}$$

$$64 \times 8 \times 8 = 4096 \text{ bits (data)}$$

B. $V=1$, $d=1$, $tag=28$, 30 meta/line

$$256 \times 30 = 7680 \text{ (overhead)}$$

$$256 \times 16 \times 8 = 32,768 \text{ (data)}$$

C. $V=1$, $d=1$, $tag=16$, 18 meta/line

$$4096 \times 18 = 73,728 \text{ (overhead)}$$

$$4096 \times 64 \times 8 = 2,097,152 \text{ (data)}$$

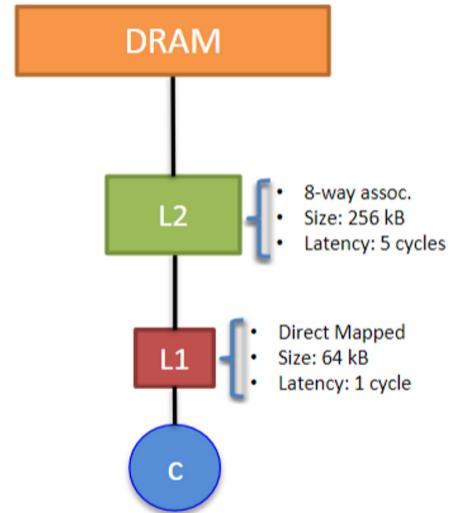
D. As the line size increases, so does the overhead

Question 4: Average Memory Access Time

Consider the processor (on right) with two-level cache hierarchy.

Suppose that **65%** of *benchmark A*'s instructions are accessed from the **L1 cache**, whereas the rest come from the **L2 cache**.

- What is the average access time for memory instructions in the benchmark
- If we change the L1 cache to be 4-way set associative we increase its flexibility and find that 85% of the instructions are now accessed from the L1. What is the average access time for memory instructions in the benchmark now?
- What is the speedup for changing the L1?



Answer

A. $65\% \times 1 + 35\% (1+5) = 2.46 \approx 3 \text{ cycles}$

B. $85\% \times 1 + 15\% (1+5) = 1.75 \approx 2 \text{ cycles}$

C. $\frac{2}{3} \approx 0.67$