

1. Assume a cache with the following characteristics:

Word size: 4 bytes
 Block size: 16 words
 Associativity: 4-way
 Replacement policy: least-recently-used (LRU)
 Total cache data capacity: 8192 bytes

a. Assume the CPU uses a 32-bit address. Show how the address is split into **tag**, **index**, **word address**, and **byte address**.

byte address (2 bits): bits 1:0
word address (4 bits): bits 5:2
index: $8192 \text{ bytes} / (4 \text{ bytes/word} * 16 \text{ words/line} * 4 \text{ lines/set}) \text{ bits} \Rightarrow 32 \text{ lines} \Rightarrow 5 \text{ bits} \Rightarrow \text{bits } 10:6$
tag (32-5-4-2): bits 31:11

b. Show the final contents of the cache after the following series of accesses, where the address is shown in decimal. Assume the cache is initially empty.

4300, 10440, 16580, 18628, 6344, 2244

$4300/64=67, 67 \text{ mod } 32=3 \text{ (line 3), tag} = 4300/2048 = 2$
 $10440/64=163, 163 \text{ mod } 32=3 \text{ (line 3), tag} = 10440/2048 = 5$
 $16580/64=259, 259 \text{ mod } 32=3 \text{ (line 3), tag} = 16580/2048 = 8$
 $18628/64=291, 291 \text{ mod } 32=3 \text{ (line 3), tag} = 18628/2048 = 9$
 $6344/64=99, 99 \text{ mod } 32=3 \text{ (line 3), tag} = 6344/2048 = 3$
 $2244/64=35, 35 \text{ mod } 32=3 \text{ (line 3), tag} = 2244/2048 = 1$

All accesses are misses, and all map to line 3. For each access, we show the contents of line 3:

address access	set 0 tag	set 1 tag	set 2 tag	set 3 tag
4300	2			
10440	2	5		
16580	2	5	8	
18628	2	5	8	9
6344	3	5	8	9
2244	3	1	8	9

2. What is the average **memory access time** for the following system, which has a write-back cache:

Hit penalty: 10 ns
 Miss rate: 2%
 Probability that a cache line is dirty: 40%
 Time required to fill a cache line: 100 ns
 Time required to write-back a cache line: 200 ns

$$\text{AMAT} = 10 \text{ ns} + 0.02 * (100 \text{ ns} + 0.4 * 200 \text{ ns}) = 13.6 \text{ ns}$$

3. List and describe three reasons for a cache miss to occur.

Compulsory (cold-start) misses

First access to a block

Capacity misses

Replaced block is needed again

Because... cache capacity isn't sufficient for the program

Conflict (collision) misses

Multiple blocks compete for the same set