Cache Sim Demo

Now that we are more familiar with how caches work, let's get some practice with the cache simulator! First, go to

`https://courses.cs.washington.edu/courses/cse351/cachesim/`

Immediately you'll notice 4 boxes:

- *System Parameters: This lets you play around with the structure/format of the cache
- *Manual Memory Access: This is where you actually make reads and writes to memory
- History: This is an interactive log of previous commands. You can even input commands through this!
- Simulation Message: Describes the most recent execution.

*These include 'explain' toggles that walk you through any execution step-by-step. Highly recommended.

Let's dive in!

Before generating, set:

"Address Width" → 6, "Cache Size" → 16, "Block Size" → 4, "Associativity" → 2 (Leave other settings as is)

A) Determine the following:

i) Highest Address in Memory: _____ **ii)** Number of Sets in Cache: _____

Hit 'Generate' and double check your answer.

B) We want to make a read at address 0x2A. Determine the following:

i) The Set containing the block that was read is number _____. ii) The tag bit in this block is _____.

iii) The full 4 bytes in this block are (in order) 0x_____, 0x_____, 0x_____, 0x_____.

Hit 'Read' and double check your answer.

C) We want to write at address 0x1B the value '0xB1'. Determine the following:

i) The Set containing the block that was read is number _____. ii) The tag bit in this block is _____.

Hit 'Write' and double check your answer.

iii) Notice that the value stored in the cache is now different from the value stored in memory. What, in the cache, indicates this disparity? Given that this was a write miss, what would have happened if our write miss policy were "No Write-Allocate" instead?

D) We want to make a read at address 0x01. Determine the following:

i) The Set containing the block that was read is number _____. ii) The tag bit in this block is _____.

iii) Will this read cause a conflict in the cache? Yes No

iv) If yes, which block will be evicted? Read made in B Write made in C

Hit 'Read' and double check your answer.

E) We want to write at address 0x1C the value '0xE9'. Determine the following:

i) The Set containing the block that was read is number _____. ii) The tag bit in this block is _____.

iii) Will this write cause a conflict in the cache? Yes No

iv) If yes, which block will be evicted? Read made in B Write made in C Read made in D

Hit 'Write' and double check your answer.

As a note, your history should look like this:

```
R(0x2a) = M
W(0x1b, 0xb1) = M
R(0x01) = M
W(0x1c, 0xe9) = M
```

G) Append the following text to the current History:

```
W(0x03, 0xff)
R(0x27)
R(0x10)
W(0x1d, 0x00)
```

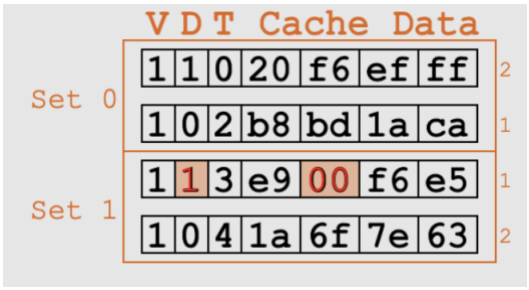
Hit Load. You'll notice that appended to each of these memory accesses is " = ?"

Determine if '?' will resolve to Hit (H) or Miss (M) for each execution.

i) `W(0x03, 0xff) =` _____ ii) `R(0x27) =` _____ iii) `R(0x10) =` _____ iv) `W(0x1d, 0x00) =` _____

Use the down arrow to check your answer for each

H) The cache, after the 8 executions detailed above, should look like this:



The numbers on the right indicate the most recent use of the cache (where 1 was more recent).

i) A LRU replacement policy will evict which block on the next cache conflict? Block 1 Block 2

ii) What is one benefit of using LRU over Random?

iii) What is one benefit of using Random over LRU?

I) If we were to flush the cache right now (don't actually) how many bytes in memory would change? _____ Bytes

How many bytes would change if our "Write Hit" policy were "Write Through" instead of "Write Back"? _____ Bytes

Can you explain why these numbers are the same/different? (if not, try changing the write hit policy and re-running using the same history above).