Problem 8: CacheSim (30 points) [File Upload]

By August Ning

Design a cache simulator in C and integrate a fully associative victim cache into the cache hierarchy. **This question is long! Please read the question in its entirety!**

PRINCETON Computer Science
Jane Street
RADIX TRADING AMS-CHI-NYC
CITADEL | CITADEL Securities
Evergreen Statistical Trading
OLD MISSION
PRINCETON GEOSCIENCES
PACM
SIG SUSQUEHANNA
Department of Mathematics
SCM STEVENS CAPITAL MANAGEMENT
ORFE
CENTER FOR STATISTICS AND MACHINE LEARNING
PICSciE
Princeton Physics
HRT hudson river trading

Part 1: To begin, start with creating a cache simulator that will simulate a level 1 cache.

for the first part, you will write a C program called `cachesim` which will take in 4 inputs from the commandline:

```
./cachesim <file.trace> <cache_capacity_bytes> <assoc> <cache_line_bytes>
```

- `file.trace` - a trace file of loads and stores instructions, more details to follow
- `cache_capacity_bytes` - the total capacity of the cache in bytes
- `assoc` - the associativity of the cache
- `cache_line_bytes` - the size of each cache line in bytes

`cachesim` should simulate the cache's behavior of the given configuration, reporting if a load or store is a hit or miss for the cache.

`cachesim` **Details:** The cache should be **write back, write allocate** cache using a **LRU replacement policy** for evicted cache lines. `cachesim` should also keep track of memory. Memory address will range between `0x000000` and `0xFFFFFF`. The memory space is byte addressed and big-endian. Caches will be no larger than 1 MB (1048576 bytes), associativity will be no larger than 32, block sizes will range 8 bytes to 256 bytes. A cache with associativity=1 is a direct mapped cache. The cache capacity, associativity, and cache line size will be a power of 2.

Loads and stores will be either 1, 2, 4, or 8 bytes. No memory accesses will go across multiple cache lines, and no memory accesses will be larger than the configured cache line size.

You do not need to do any error checking on the inputs and the trace file. You can assume that the trace file is properly formatted, the input configuration is a valid cache configuration.

To begin, the memory is all zeros and all cache line entries are invalid.

The implementation of the `cachesim` is up to you. However, it will be beneficial to understand cache fundamentals on how caches are broken down in cache lines, sets depending on associativity, how caches are tagged, and eviction policies.

PRINCETON Computer Science
Jane Street
RADIX TRADING AMS-CHI-NYC
CITADEL | CITADEL Securities
Evergreen Statistical Trading
OLD MISSION
SIG SUSQUEHANNA
Department of Mathematics
SCM STEVENS CAPITAL MANAGEMENT
ORFE
CENTER FOR STATISTICS AND MACHINE LEARNING
PICSciE
Princeton Physics
HRT hudson river trading

Trace File: The trace files will be a plaintext file with each operation on its own line

```
STORE 0x0 0x4 aabbccdd
STORE 0x4 0x4 11223344
STORE 0x1000 0x2 baad
STORE 0x2002 0x2 beef
LOAD 0x0 0x2
LOAD 0x2000 0x4
```

Each line will start with either `LOAD` or `STORE`, followed by the hex value of the memory address, then the hex value of the number of bytes to read/write.

For store instructions, there will be an additional value with the hex value to be stored. The stored value **will not have a leading "0x"**. The stored value will be properly sized to match the number of bytes specified.

Memory addresses and stored values will not have leading zeros. All instructions will be valid for the cache configuration. All tokens will be separated by whitespace.

`cachesim` **Output**: `cachesim` should print the results of each instruction onto the console. For each instruction, print the instruction and memory address and if it's a `HIT` or `MISS` within the cache.

If it's a load instruction, also print the value of data loaded in hex with leading "0x". Always print out the number of bytes specified by the load instruction and include leading zeros.

PRINCETON Computer Science · Jane Street · RADIX TRADING AMS-CHI-NYC · CITADEL | CITADEL Securities · Evergreen Statistical Trading · OLD MISSION · Princeton Geosciences · PACM · SIG SUSQUEHANNA · Department of Mathematics · SCM Stevens Capital Management · ORFE · Center for Statistics and Machine Learning · PICSciE · Princeton Physics · HRT hudson river trading

Example Output

```
./cachesim demo.trace 4096 1 64
STORE 0x0 MISS
STORE 0x4 HIT
STORE 0x1000 MISS
STORE 0x2002 MISS
LOAD 0x0 MISS 0xaabb
LOAD 0x2000 MISS 0x0000beef
```

```
./cachesim demo.trace 4096 2 64
STORE 0x0 MISS
STORE 0x4 HIT
STORE 0x1000 MISS
STORE 0x2002 MISS
LOAD 0x0 MISS 0xaabb
LOAD 0x2000 HIT 0x0000beef
```


PRINCETON Computer Science
Jane Street
RADIX TRADING AMS-CHI-NYC
CITADEL | CITADEL Securities
Evergreen Statistical Trading
OLD MISSION
PRINCETON GEOSCIENCES
PACM
SIG SUSQUEHANNA
Department of Mathematics
SCM STEVENS CAPITAL MANAGEMENT
ORFE
CENTER FOR STATISTICS AND MACHINE LEARNING
PICSciE
Princeton Physics
HRT hudson river trading

Part 2: In the next part, modify `cachesim` to add a victim cache. The victim cache has the following behavior:

- A victim cache is a fully associative cache between the cache (L1) and memory.
- Whenever any L1 cache line is evicted, it is first placed in the victim cache.
- If there is a miss in the L1 cache, the system will search the victim cache.
- If there's a hit in the victim cache, the victim cache will swap into the L1 cache.
- When there is a miss in the L1 and victim cache, the data will be provided from memory directly into the L1.

The L1 and victim cache will still follow LRU replacement policy when evicting from L1 into the victim cache, promoting swapping from the victim cache into the L1 cache, and evicting from the victim cache into memory.

When an L1 cache line is evicted into the victim cache, the LRU counts are not reset. When an L1 cache line is evicted and the victim cache is full, evict the LRU from the victim cache.

The victim cache has the same cache line size as the L1. The victim cache starts empty and the victim cache lines are invalid. A a fully associative cache, the victim cache will only evict to memory when it becomes full and a new cache line needs to be placed in the victim cache.

Modify your `cachesim` to add in a victim cache. The victim cache will take an additional command line argument for the victim cache size. If an instruction has a hit in the victim cache, print `VICTIM HIT`.

```
./cachesim_victim <file.trace> <cache_capacity_bytes> <assoc> \
<cache_line_bytes> <victim_cache_size>
```

- `victim_cache_size`: the number of cache lines that the victim cache will hold

PRINCETON Computer Science
Jane Street
RADIX TRADING AMS-CHI-NYC
CITADEL | CITADEL Securities
Evergreen Statistical Trading
OLD MISSION
SIG SUSQUEHANNA
Department of Mathematics
SCM STEVENS CAPITAL MANAGEMENT
ORFE
CENTER FOR STATISTICS AND MACHINE LEARNING
PICSciE
Princeton Physics
HRT hudson river trading

Example Output

```
./cachesim_victim demo.trace 4096 2 64 2
STORE 0x0 MISS
STORE 0x4 HIT
STORE 0x1000 MISS
STORE 0x2002 MISS
LOAD 0x0 VICTIM HIT 0xaabb
LOAD 0x2000 HIT 0x0000beef
```

How To Submit: How to Submit

For both `cachesim` and `cachesim_victim` submit your code as a single C file titled `cachesim_<team_id>.c` and `cachesim_victim_<team_id>.c`. Compile both of these files into a single `.zip` file following the contest naming convention.

Your code will be compiled with `gcc cachesim_{victim}_<team_number>.c` on the Adroit clusters (Adroit runs Springdale Linux 8.8 and has gcc 8.5.0).

Your simulators' print output will be compared to the solution's output.

PRINCETON Computer Science
Jane Street
RADIX TRADING AMS-CHI-NYC
CITADEL | CITADEL Securities
Evergreen Statistical Trading
OLD MISSION
PRINCETON GEOSCIENCES
PACM
SIG SUSQUEHANNA
Department of Mathematics
SCM STEVENS CAPITAL MANAGEMENT
ORFE
CENTER FOR STATISTICS AND MACHINE LEARNING
PICSciE
Princeton Physics
HRT hudson river trading