Write simulator code for an arbitrary n-way set associative cache in C++. The simulator should implement 2 level caches, L1 and L2. Simulator reads trace files and assigns requests to the L1 cache. L1 cache sends read/write requests to the L2 cache. L2 cache interacts with DRAM. L1 and L2 caches keep track of their own counters i.e. reads, writes, misses, hits etc. At the end of the simulation, for a given trace file, your program should print the stats for both caches on the console, as follows:

i. number of L1 reads: _____
ii. number of L1 read misses:_____
iii. number of L1 writes: _____
iv. number of L1 write misses:_____
v. L1 miss rate: _____
vi. number of writebacks from L1 memory: _____
vii. number of L2 reads: _____
viii. number of L2 read misses:_____
ix. number of L2 writes: _____
x. number of L2 write misses:_____
xi. L2 miss rate: _____
xii. number of writebacks from L2 memory:_____

Configurable Parameters

- BLOCKSIZE: Number of bytes in a block/line.
- L1_Size: Total bytes of data storage for L1 Cache.
- L1_Assoc: Associativity of L1 Cache.
- L2_Size: Total bytes of data storage for L2 Cache.
- L2_Assoc: Associativity of L2 Cache.

For example: "./cache_simulate 64 1024 2 65536 8 memory_trace_files/trace1.txt" means the following values for the configurable parameters.

BLOCKSIZE: 64
L1_SIZE: 1024
L1_ASSOC: 2
L2_SIZE: 65536
L2_ASSOC: 8
All parameters should be power of 2.

Non-configurable Parameters

- Cache eviction policy: Simulator uses least replacement policy(LRU) - this means you need to keep track of which block was used least recently

- Cache write policy: Simulator uses Write-back Write-allocate(WBWA) policy - this means you need to keep track of dirty blocks, so that they are written back to the level n+1 of memory hierarchy, before eviction from level n

Running simulator on trace files

If L1 read/write time is 1 ns, L2 read/write time is 20 ns and DRAM read/write time is 200 nanoseconds, add code in the simulator to compute **total access time** for a given trace.

Plot 5 graphs for total access time, as simulation parameters vary. X-axis of each graph will have 5 different values of one configurable parameter, while the other four parameters are kept constant at the default values of 64, 1024, 2, 65536, 8.

- In the first graph, vary block size on x-axis between 8, 16, 32, 64, 128, and keep L1 size, L1 associativity, L2 size and L2 associativity fixed at 1024, 2, 65536, 8.
- In the second graph, vary the second parameter L1 size between 512, 1024, 2048, 4096, 8192, and keep block size, L1 associativity, L2 size and L2 associativity fixed at 64, 2, 65536, 8.
- In the third graph, vary the third parameter L1 associativity between 1, 2, 4, 8, 16, and keep block size, L1 size, L2 size and L2 associativity fixed at 64, 1024, 65536, 8.
- In the fourth graph, vary the fourth parameter L2 size between 16384, 32768, 65536, 131072, 262144, and keep block size, L1 size, L1 associativity and L2 associativity fixed at 64, 1024, 2, 8.
- In the fifth graph, vary the fifth parameter L2 associativity between 1, 2, 4, 8, 16, and keep block size, L1 size, L1 associativity and L2 size fixed at 64, 1024, 2, 65536.

The y-axis of each plot should have total access time. Each plot should contain 8 lines, for the 8 trace files. Plot big graphs, with different line/point types to distinguish among the curves.

List all possible observations you can make from the 5 plots – commenting on the effect of different parameters, comparing across the traces etc.

What to submit

- Folder with source code and Makefile to create **cache_simulate** executable
- For each input trace file, your program should be runnable with ./cache_simulate 64 1024 2 65536 8 memory_trace_files/trace1.txt, where the values of the configurable parameters will be varied by the checker script
- PDF report with plots and observations for the 8 trace files provided.