
Lab 1

Implementation of a Cache Simulator

Description

◆ Aim:

- Implement a configurable cache simulator in high-level language (C/C++, or Java).

◆ Configuration Requirements (Input Parameters):

- **Cache Size:** Total size for the data in the cache only. For example 32KB.
- **Cacheline Size:** Also aka cache block size. For example, 16 Byte, 32 Byte.
- **Associativity:** Can support direct-mapped (use 1 or DM for parameter), set-associative (use corresponding number for the associativity), and fully-associative (use 0 or FA or parameter).
- **Replacement Policy:** Can support LRU and Random
- **Write Policy (Write-Allocate):** Write Through and Write Back (Dirty bit)

Description

◆ Assumptions:

- Since no cache performance (latency) is simulated in this lab, we assume that each cache access (both hit and miss) will complete before next cache operation.
- The memory address is 32-bit in this lab.

◆ Results Requirements (Output):

- Number of cache access
- Number of cache load
- Number of cache store
- Average cache hit rate
- cache hit rate for loads
- cache hit rate for stores
- Can print the content of the tag array after any cache access (For testing and debugging)

Description

- ◆ **Do your own test first. The test configuration and input will be provided in the future.**

- ◆ **Two weeks for the implementation and one week for testing and report.**