

Cache Simulation in C++: Implementation and Testing Report

Jash Jhatakia - CS22BTECH11028

23 Nov, 2023

1 Introduction

This report outlines the development and testing process for a cache simulator implemented in C++. The simulator models basic cache operations, including read and write functionalities, and incorporates cache replacement policies like Least Recently Used (LRU) and First-In-First-Out (FIFO).

2 Coding Approach

2.1 Data Structures and Initialization

- Utilized arrays for storing cache data and managing replacement policies.
- Defined constants for maximum cache lines and associativity.
- Implemented file reading for configuration and access patterns.

2.2 Cache Operations

- Developed `handleRead` and `handleWrite` functions for processing cache operations.
- `handleMissRead` and `handleMissWrite` manage cache misses and update cache data accordingly.
- Cache indexing and tag extraction are dynamically computed based on the configuration.

2.3 Replacement Policy

- Incorporated LRU and FIFO replacement strategies.
- Implemented logic for direct mapped, fully associative and set associative caches.

2.4 Reporting

- The simulator outputs detailed information for each cache access.
- It calculates and displays the total number of hits and misses.

3 Testing Methodology

3.1 Test Cases

- Generated various test cases, and tried the results.
- Validated the answers.