Lab 7 - Project Report

AI24BTECH11031 - Shivram S
AI23BTECH11022 - Dhadheechi Ravva

Contents

1 Introduction

As part of the CS2323 Computer Architecture course, we had to extend the RISC-V simulator developed in Lab 4 to support cache simulation. This report outlines the design of the cache simulator.

2 Usage

The project includes a `Makefile` which can be used to build the project and test it. To build, run:

```
$ make
```

This produces an executable called `riscv_asm` in the project directory.

The simulator can be run using the following command:

```
$ ./riscv_asm
```

3 Design of The Cache

The simulator's cache consists of several **lines**. Each line consists of several **entries**, and each entry holds a block of memory.

The cache has a **replacement policy** which determines the block to be replaced in case of a conflict, and a **write-back policy**, which tells it how to handle writes.

The `Cache` struct holds all data related to the simulator's cache

```
typedef struct Cache {
    // Cache configuration
    size_t num_lines, block_size, associativity;
    enum {WRITEBACK, WRITETHROUGH} write_policy;
    enum {FIFO, LRU, RANDOM} replacement_policy;

    size_t hits, misses, writebacks; // Statistics

    size_t monotime; // For maintaining insert and access times
    uint8_t *mem; // Simulator memory
    CacheLine *lines; // The actual cache data
    FILE *output_file; // File for logging accesses
} Cache;
```