Final Project Implementation Guide

This document serves as an implementation guide for students who are working on the Cache Simulator Final Project (using LSB indexing scheme). Students are encouraged to develop their own solutions, but may reference this guide if they encounter difficulties.

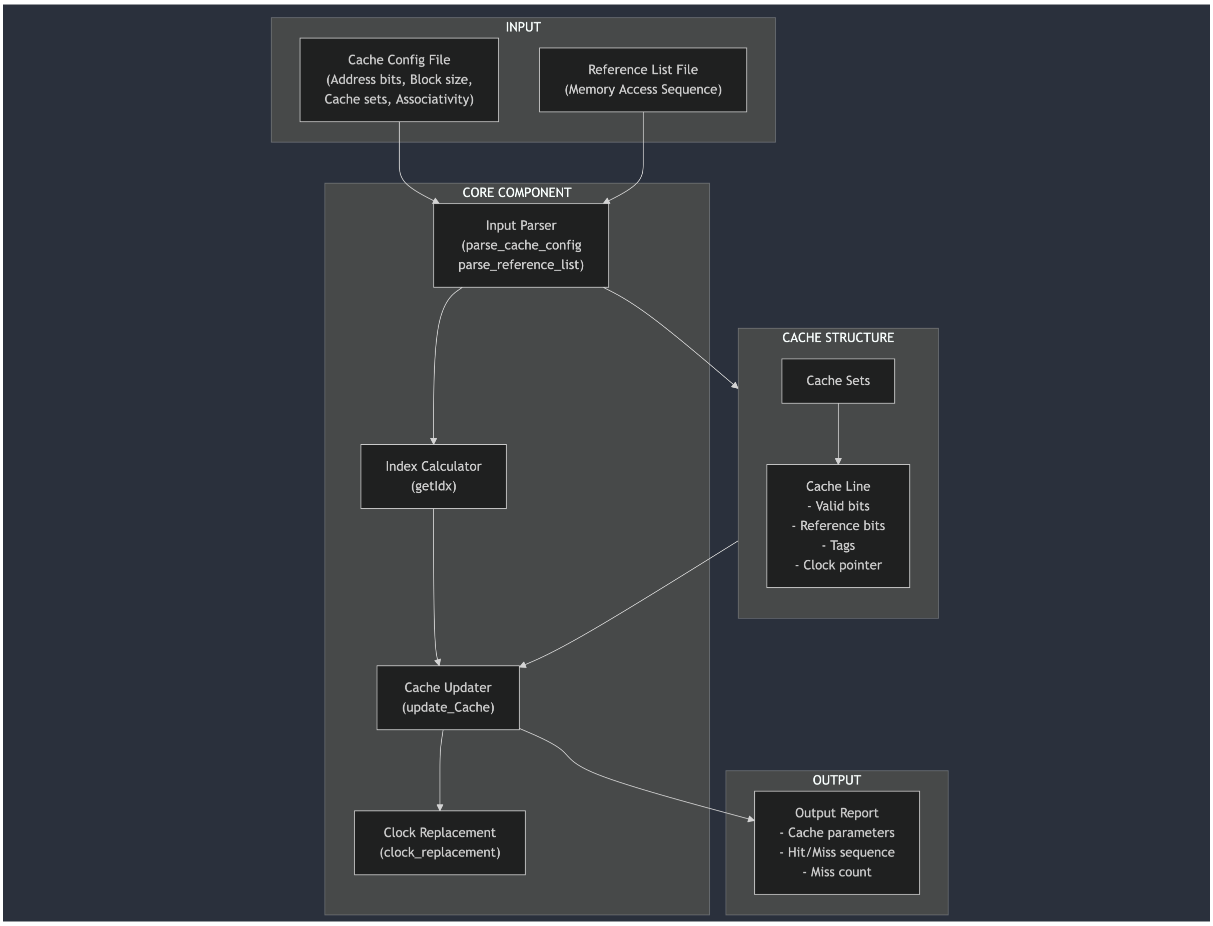
Description

When memory access occurs:

1. Calculate index from address
2. Find cache line using index
3. Check all ways in this line
   If HIT:
   - Found matching tag
   - Set reference bit = 1
   - Report hit (return 1)

   If MISS:
   - No matching tag found
   - Use clock replacement to find way
   - Update way's data
   - Report miss (return 0)

Module Architecture



Code Template

INPUT

Cache Config File Parser

```
void parse_cache_config(const string &path_cache) {
    // TODO: Read config file and initialize cache
    // - Read: Address_bits, Block_size, Cache_sets, Associ
ativity
    // - Calculate: Offset_bits, Indexing_bits
    // - Initialize Cache array
}
```

Reference List File Parser

```
void parse_reference_list(const string &path_ref) {
    // TODO: Process memory access sequence
    // - Read addresses and validate
    // - Store both original and converted forms
}
```

CACHE STRUCTURE

Cache Line Organization

```
struct cache_line {    // Represents a cache line (equivalent to a set)
    // TODO: Define data structures for cache line
    // - Valid bits for each way
    // - Tags for each way
    // - Reference bits for clock policy
    // - Clock pointer for replacement
};
vector<cache_line> Cache;  // All cache lines (sets)
```

CORE COMPONENTS

Index Calculator

```
ull getIdx(ull addr, ull Mask) {
    // TODO: Extract index from address using mask
}
```

Cache Updater

```
int update_Cache(ull addr, int line, ull Mask) {
    // TODO: Handle cache access
    // - Find cache line using index
    // - Check ways in line for hit
    // - Handle hit/miss cases
    // Return: 1 for hit, 0 for miss
}
```

Clock Replacement

```
int clock_replacement(cache_line &cache_set) {
    // TODO: Like a clock hand moving through ways
    // - If current way's ref_bit is 0:
    //   -> Select this way for replacement
    // - If current way's ref_bit is 1:
    //   -> Set it to 0 and move to next way
    // Remember to advance clock pointer
}
```

OUTPUT

Report Generator

```
void output(const string &path_rpt, ull Mask) {
    // TODO: Generate report
    // - Write cache parameters
    // - Write access results and statistics
}
```