**Cache Project**

**Student 1: 2171052 허채린**

**Student 2: 2176123 Jimin Nam**

**Index**

1. **Introduction and Background Information -- pg3**
2. **Overall Flow ------------------------------------ pgx**
3. **Description of Given Functions --------------- pgx**
4. **Explanation of retrieve\_data() ---------------- pgx**
5. **Explanation of check\_cache\_data\_hit() ------- pgx**
6. **Explanation of access\_memory() ------------- pgx**
7. **Explanation of find\_entry\_index\_in\_set() ----- pgx**
8. **Comparison of the result of the function with the actual result -------------------------------- pgx**
9. **Conclusion -------------------------------------- pgx**
10. **Introduction and Background Information**

This project conducts to understand the role of cache in a high-level language.

The cache consists of SRAM and is located at the top of the memory hierarchy. The reason why caches are designed is to use memory efficiently according to the principle of locality. The principle of locality can be exploited by implementing a computer's memory as a memory hierarchy. The memory hierarchy consists of several tiers of memory of different speeds and sizes. Cache is expensive and small compared to the main memory and the disk. In addition, the cache is closer to the CPU than the main memory and disk, so access time is faster. The unit in which the data is stored in the cache is called a block. Cache stores memory corresponding to a portion of main memory and when the data requested by the processor is in a block of cache, it is called a hit. Conversely, when the data requested by the processor is not found in a certain block in the cache, it is called a miss. Usually, the hit ratio is used to evaluate the performance of the memory hierarchy. In this case, the hit ratio is composed of (number of hits)/(number of hits + number of misses).

To explain this project, we implement a simulator of n-way set associative cache by using C language. The cache size is set to 32 bytes and one block is set to 8 bytes. Therefore, the number of blocks is 4, so we can implement direct mapped cache, 2-way set associative cache, and fully associative cache(4-way set). This cache simulator writes the data returned to the output file when given an input file. At this time, in the input file, the byte address and data type are given. If there is data corresponding to the cache according to the value given in the input file, a hit event occurs and the data is fetched. Otherwise, the main memory is accessed to receive the data.