**REPORT ASSEMBLY AND COMPUTER ARCHITECHTURE LAB WEEK 12**

**STUDENT NAME: PHI QUOC HUNG**

**STUDENT ID: 20184270**

**ASSIGNMENT 1:**

Here are the results observed when we choose the number of blocks is 16 and cache block size is 4:

Sample code 1:

Graphical user interface, application, Word

Description automatically generated

Sample code 2:

Graphical user interface, application

Description automatically generated

**ASSIGNMENT 2:**

By definition, Cache memory works by taking data or instructions at certain memory addresses in RAM and copying them into the cache memory, along with a record of the original address of those instructions or data.

In this lesson, specifically in the 2 sample codes above, after every main memory referrences (sw or lw instruction), the word at that address is cached, but because the block size is 4 words, those 4 4-bit words are stored to build up a block. The block size here is 4, so the 4 least significant bits in address is used to offset the bits. The first memory reference is this sw instructions to save fp to address 0x7fffeff8, or in binary:

0b0111 1111 1111 1111 1110 1111 1111 1110

Since the offset bits are ignored, the next 4 bits is where the block is mapped to. In this sample code, 11112 = 1510. This results in that the word at address 0x7ffeff0, 0x7fffeff4, 0x7fffeff8, 0x7fffeffc to block 15.

Solution for the questions:

- How is the 32-bit address used in the cache memory: Hash function

- What happens when there is a cache miss: Cache miss occur when we try to refer to a memory that is not in cache memory. Solution: Write code to entry the cache for block data, then continue.

- What happens when there is a cache hit: The data is stored into cache memory

- What is the block size: Block size is the unit of chace memory. All volumes on the storage system share the same cache space. Therefore, the volume can have only 1 blocksize

- Function of the tags: Distinguish between multiple address in the memory

**ASSIGNMENT 3:**

Affection of paramenters:

- Cache size: Bigger cache size -> can hold more data -> better performance in case program need to access large amount data from memory

- Block size: If the data needed is large, such as an array -> bigger block size is better. If the data needed to access is small -> small block is better

- Set size: Bigger set size -> smaller conflict -> more performance

- Write and replacement policy: I don’t know :<<

Question 2: Cache miss rate is smaller because we can reduce conflict in the program, and this is good

Question 3: After a sequence of instructions we should try to refer memory as near as possible

I’m so sorry because I have been late for more than a day. I hope that you will subtract just a little point because of this delay! Thank you very much and have a nice day, Sir!