CACHE MEMORY

Cache Memory is a high-speed memory used to reduce the average time taken to fetch data from the Main Memory. It stores a copy of the frequently used data fetched from the Main Memory.

1.1 GENERAL INFORMATION

There are three mapping techniques used in the case of Cache Memory –

1. Direct Mapping
2. Associative Memory
3. N – Way Set Associative Memory

All these techniques have their own pros and cons, which directly affect the time taken to fetch the requested data and the number of Cache Hits and Cache Misses encountered.

Cache Hit – When the requested data is found in the Cache Memory, this instance is known as Cache Hit.

Cache Miss – When the requested data is missing in the Cache Memory, this instance is known as Cache Miss.

Cache Latency – When the requested address isn't found in the Cache Memory and the overall time taken to fetch the address is more than the time it would have taken to directly access the address from the Main Memory.

1.2 INPUT

Following is the explanation for the input in the program –

1. First, the user has to choose a mode for the Cache Memory.
2. Then the user has to input the sizes of the Main Memory and Cache Memory and the number of words in each block. The program will automatically compute the rest of the data.
   (Make sure all the parameters are powers of 2)
3. Then the user has to input the value of N for N – Way Set Associative Memory only.
4. Then the user has to repeatedly select the operations he/she wants to perform until he/she selects the "Exit" option.

1.3 EXPLANATION OF THE CODE

These are two modes, namely, Single Cache Memory and Multilevel Cache Memory, one has to choose from. Single Cache Memory deals with just an only Cache Memory of size S, cache lines CL, and block size B. Multilevel Cache Memory deals with two Cache Memories, namely, Level 1 Cache Memory of size S/2 and Level 2 Cache Memory of size S. Both modes have all three mapping techniques. In the case of the latter, the technique used to map the contents between the Main Memory and the Level 1 Cache Memory is also used to map the contents from the Level 1 Cache Memory to the Level 2 Cache Memory.

1.3.1 SINGLE CACHE MEMORY

For the first mode, a user has to input the following parameters. All the values should be powers of 2.

Parameters	Variable
Size of the Main Memory	size_main
Size of the Cache Memory	size_cache
Number of words in each block	words_per_block

Other parameters are automatically computed. After receiving the inputs, the programs create two dictionaries, one for the Main Memory and the other for the Cache Memory. The Main Memory is filled with the calculated number of blocks (size_cache / words_per_block), and in each block, there is words_per_block number of words.

A block is a list of words, and each word is also a list which has the following contents in it -

1. Word
2. Address
3. Block Number
4. Place within the Block
5. Word Number

The Cache Memory is filled with the calculated number of cache lines (size_cache / words_per_block). Each cache line is a list, which is initially filled with None (In python, None stands for null value), that will store a block and an address.

Each mode has three mappings to choose from, namely, Direct Mapping, Associative Memory, and N – Way Set Associative Memory. In the case of N – Way

Set Associative Memory, the user has to input the value of N. Each mapping technique has three operations, namely, Loading, Searching, and Exit. The selection of operations won't stop until one selects "Exit."

1. In Direct Mapping, the program first prints the information about the address to be used, the number of bits representing a word, a cache line, and a tag.

   When loading, the program assigns the requested block to the calculated location in the Cache Memory. If the location is occupied, then the program prints the address, which will be replaced by the requested address.

   When searching, the program searches for the requested address in the Cache Memory. If it is found, then "Cache Hit" and general information about the requested address is printed, which includes the block number, content of the block the word belongs to, and the cache line. If it isn't found, then the program prints "Cache Miss."

2. In Associative Memory, the program first prints the general information about the address and then the list of operations. A counter is also created to iterate over the Cache Memory and is initialized to zero.

   When loading, the program assigns the requested block to the cache line the counter is currently pointing to. The replacement policy used is FIFO. (First In First Out) The counter resets itself to zero whenever the last cache line is filled/altered.

   When searching, the program iterates over the Cache Memory, and if the requested address is found, it prints "Cache Hit" and other information, and if the requested address isn't found, it prints "Cache Miss."

3. In N – Way Set Associative Memory, the user inputs the value of N, and then the program prints the information about the address. It also creates a list of N counters initialized to 0.

   When loading, the program assigns the requested block to the cache line the counter of the required set is currently pointing to. The replacement policy used is FIFO. (First In First Out) The counter of a set resets itself to zero whenever the last cache line of a set is filled/altered.

   When searching, the program iterates over the required set of the Cache Memory, and if the requested address is found, it prints "Cache Hit" and other information, and if the requested address isn't found, it prints "Cache Miss."

All these mappings have their own advantages and disadvantages, which directly impact the time taken to fetch the data. Sometimes these cache memories are

worthless and result in cache latency (when the requested data isn't found in the Cache Memory). Cache latency increases the overall time taken to search for the requested address, which is more than the time it would have taken to search for address directly in the Main Memory.

1.3.2 MULTILEVEL CACHE MEMORY

For the second mode, a user has to input the following parameters. All the values should be powers of 2. Other parameters are automatically computed by the program.

Parameters	Variable
Size of the Main Memory	size_main
Size of the Level 2 Cache Memory	size_cache_two
Number of words in each block	words_per_block

After receiving the inputs, the programs create three dictionaries, one for the Main Memory, second for the Level 1 Cache Memory, and third for the Level 2 Cache Memory. The Main Memory and the two Cache Memories are precisely the same as constructed in the first mode. (Single Cache Memory)

This mode also has three mappings to choose from, namely, Direct Mapping, Associative Memory, and N – Way Set Associative Memory. In the case of N – Way Set Associative Memory, the user has to input the value of N. Each mapping technique has three operations, namely, Loading, Searching, and Exit. The selection of operations won't stop until one selects "Exit." The mapping used to map the contents between the Main Memory and the Level 1 Cache Memory and the contents between the Level 1 Cache Memory and the Level 2 Cache Memory is the same.

1. In Direct Mapping, the program first prints the information about the address, the number of bits representing a word, a cache line, and a tag in Level 1 Cache Memory address and Level 2 Cache Memory address.

   When loading, the program assigns the requested block to the calculated location in the Level 1 Cache Memory. If the location is occupied, then the program prints the address, which will be replaced by the requested address. The replaced address is added to the Level 2 Cache Memory. The cache line is decided by the number represented by the cache line bits.

   When searching, the program searches for the requested address first in the Level 1 Cache Memory. If it is found, then "Cache Hit" and general

information about the requested address is printed. If it isn't found, the Level 2 Cache Memory is searched. If the requested address is missing in both the Cache Memories, then the program prints "Cache Miss."

2. In Associative Memory, the program first prints the general information about the address and then the list of operations. Two counters are also created and initialized to zero to iterate over the two Cache Memories.

   When loading, the program assigns the requested block to the cache line the counter of the Level 1 Cache Memory is currently pointing to. The replacement policy used is FIFO. (First In First Out) Both the counters reset themselves to zero whenever the last cache line of their respective Cache Memories is filled/altered. During replacement, the replaced address is added to the cache line the counter of the Level 2 Cache Memory is pointing to.

   When searching, the program first iterates over the Level 1 Cache Memory, and if the requested address is found, it prints "Cache Hit" and other information. If the requested address isn't found, Level 2 Cache Memory is searched. If the address is missing in both the Cache Memories, the program prints "Cache Miss."

3. In N – Way Set Associative Memory, the user inputs the value of N, and then the program prints the information about the address. It also creates two lists of N counters initialized to 0.

   When loading, the program assigns the requested block to the cache line the counter of the required set in the Level 1 Cache Memory is currently pointing to. The replacement policy used is FIFO. (First In First Out) The counters of all the sets in both the Cache Memories reset themselves to zero whenever the last cache line of their set is filled/altered. On replacement, the replaced address is added to the cache line the counter of the required set in the Level 2 Cache Memory is currently pointing to.

   When searching, the program first iterates over the required set of Level 1 Cache Memory. If the requested address is found, it prints "Cache Hit" and other information, and if the requested address isn't found, the required set of the Level 2 Cache Memory is searched. If the address is missing in both the Cache Memories, the program prints "Cache Miss."

When replacement occurs in the Level 2 Cache Memory, then the replaced address is completely removed. It has to be added again to the Cache Memory for faster access.

Level 2 Cache Memory is slower than the Level 1 Cache Memory because the size of the Level 2 Cache Memory is twice the size of the Level 1 Cache Memory. This also results in a bigger cache latency in comparison to the Singe Cache Memory.

1.4 ASSUMPTIONS

Following are the assumptions considered for this assignment –

1. All the inputs must be powers of 2.
2. The replacement policy used is FIFO.
3. Words are represented by a string in the following format, "Words" + Word Number.
4. The input address is in binary.
5. Any word should not be added twice.