Computer Organization Lab06 Report

Author: 109652039 林立倫, 109550003 陳茂祥

1 Implementation

1.1 Direct Mapped

First, we need to analyze how the address is constructed. We calculate the length of tag, index, and block offset in the address by `get_indexing_size`. Second, we construct our cache by a vector of `Block`, which consists of a valid bit and a tag. Then, read the address from file and seperate them into tag, index and block offset by function `get_bits`. If the block is not valid or the tag is not the same as the current one, there is a read miss. Otherwise, it is a read hit. Finally, we calculate the hit rate and return the result.

1.2 Set Associative

First, we also analyze the composition of address. But this time, we construct our cache by vector of `Set`. `Set` contains a vector of `Block` to store valid bit and tags, a `use` vector to store when the block is used last time, and a variable `counter` to represent time. Second, read address from the file and seperate them into tag, index and block offset by function `get_bits`. Then, we call the function `add_block` for further process. In this function, if the tag is found in the corresponding set, we will update the time in `use` and return true. Otherwise, if the set is not full, we simply add it to the set, update the `use` value and return false. For those situations with full set, we replace the block by `LRU` and return false. Finally, we can calculate the hit time and the hit rate by the return value of the `add_block` function.

2 Result

```
❯ ./demo.sh
rm -f main.o cache_utilities.obj direct_mapped_cache.obj main.obj set_associative_cache.obj
g++ -I./include -std=c++17  cache_utilities.cpp -c -o cache_utilities.obj
g++ -I./include -std=c++17  direct_mapped_cache.cpp -c -o direct_mapped_cache.obj
g++ -I./include -std=c++17  main.cpp -c -o main.obj
g++ -I./include -std=c++17  set_associative_cache.cpp -c -o set_associative_cache.obj
g++ -I./include -std=c++17  cache_utilities.obj direct_mapped_cache.obj main.obj set_associative_cache.obj -o main.o
========================== Direct mapped result ==========================

 0.0795226  0.0660363  0.0547202  0.0553402  0.0920787 | 4096
 0.0624709   0.042784   0.031623  0.0244923  0.0398388 | 16384
 0.0570454  0.0356534  0.0234072  0.0159665  0.0124012 | 65536
 0.0565804  0.0350333  0.0227872  0.0151914  0.0114711 | 262144
-----------------------------------------------------
        16         32         64        128        256

==================== N-way set associative result ======================

Block size: 64

  0.110681   0.083553  0.0778174  0.0782824 | 1024
 0.0827779  0.0517749   0.041854  0.0398388 | 2048
 0.0547202  0.0362734  0.0306929  0.0280577 | 4096
 0.0403038  0.0297628  0.0266625  0.0244923 | 8192
  0.031623  0.0237172  0.0234072  0.0229422 | 16384
 0.0254224  0.0232522  0.0227872  0.0227872 | 32768
-------------------------------------------
     1-way      2-way      4-way      8-way
```

3 Problems and Solution

Because we want to seperate the functions from the original cpp files, we add a header file for function declaration and another cpp file for function definition. Thus, we need to modify Makefile to compile with these additional files. Thanks to TAs for answering questions soonly. At the end of this class, thanks to Prof. T.F.Chen for the great class and TAs for helpful assistance.