

COMP 222 Computer Organization

Assignment #4—Direct Mapping Cache Memory

Objective:

To simulate reading and writing to a custom-sized direct-mapped cache, involving a custom-sized main memory.

Inputs:

- The total size of accessible main memory (in words)
- The total size of the cache (in words) and the block size (words/block)
- A signal to read (0) or write (1) to the cache
- The main memory address to read from/write to
- The contents of the address for writing to the cache (data)

Outputs:

- The corresponding cache tag, block, and word for a main memory address
- The contents of the address resulting from reading/writing to the cache
- A message indicating either a hit or a miss to the cache

Specification:

- The program simulates reading from and writing to a direct-mapped cache based on choosing from a menu of choices, where each choice calls the appropriate procedure, where the choices are:
 - 1) Enter parameters
 - 2) Access cache for reading/writing and transfer data
 - 3) Quit program

What to do:

- Use a structure (struct) to represent a cache line consisting of a tag (integer) and a block (integer pointer). Define the cache to be a pointer to the struct.
- Upon entering the parameters, the main memory and cache are to be dynamically allocated based on their respective total sizes. Each word i of main memory is initialized with the value $M+i$, where M is the size of main memory in words. For example, if the memory size is 16384, then word 10 will initially contain the value 16394 (which is $16384+10$).
- Reading/writing from/to a new block in the cache results in dynamically allocating the block based on the block size.

What NOT to do (any violation will result in an automatic score of 0 on the assignment):

- Do NOT modify the choice values (1, 2, 3, 4, 5) or input characters and then try to convert them to integers--the test script used for grading your assignment will not work correctly.
- Do NOT turn in an outdated version of the assignment downloaded from the Internet (coursehero, github, etc.) or a version that was coded by someone else (former student, tutor, etc.)
- Do NOT use any self-created or external libraries that cannot be located/utilized by zylabs
- Do NOT turn in your assignment coded in another programming language (C++, C#, Java, Python, Perl, etc.)—it will NOT compile under zyLabs C compiler.

What to turn in:

The source code as a single C file uploaded to Canvas (<http://canvas.csun.edu>) by the deadline of 11:59pm PST (-20% per consecutive day for late submissions, up to the 4th day—note 1 minute late counts as a day late, 1 day and 1 minute late counts as 2 days late, etc.).

Sample test run (Inputs: 1 65536 1024 16 2 1 65535 14 2 0 65535 2 1 65534 512 2 0 1023 3):

Direct-Mapped Cache Memory:

- 1) Enter parameters
- 2) Access cache
- 3) Quit program

Enter option: 1

Enter main memory size (words): 65536

Enter cache size (words): 1024

Enter block size (words/blocks): 16

Direct-Mapped Cache Memory:

- 1) Enter parameters
- 2) Access cache
- 3) Quit program

Enter option: 2

Select read (0) or write (1): 1

Enter main memory address to write to: 65535

Enter value to write: 14

Access result from address 65535:

Write miss!

Tag: 63

Block: 63

Word: 15

Data: 14

Direct-Mapped Cache Memory:

- 1) Enter parameters
- 2) Access cache
- 3) Quit program

Enter option: 2

Select read (0) or write (1): 0

Enter main memory address to read from: 65535

Access result from address 65535:

Read Hit!

Tag: 63

Block: 63

Word: 15

Data: 14

Direct-Mapped Cache Memory:

- 1) Enter parameters
- 2) Access cache
- 3) Quit program

Enter option: 2

Select read (0) or write (1): 1

Enter main memory address to write to: 65534

Enter value to write: 512

Access result from address 65534:

Write Hit!

Tag: 63

Block: 63

Word: 14

Data: 512

Direct-Mapped Cache Memory:

- 1) Enter parameters
- 2) Access cache
- 3) Quit program

Enter option: 2

Select read (0) or write (1): 0

Enter main memory address to read from: 1023

Access result from address 1023:

Read miss!

Tag: 0

Block: 63

Word: 15

Data: 66559

Direct Mapping Cache Memory:

- 1) Enter parameters
- 2) Access cache
- 3) Quit program

Enter selection: 3