Universidad del País Vasco Euskal Herriko Unibertsitatea
FACULTY OF INFORMATICS
UNIVERSITY OF THE BASQUE COUNTRY

(individual work)

Optional practical work: simulator of a simple Cache M.

The more frequently used data blocks in the processor are loaded in Cache Memory (CM). Each memory access (`rd` / `wr`) is a hit if the required data block is in CM, and a miss if the data block is not in CM. On misses, the data block needs to be transferred from Main Memory (MM).

As the CM is smaller, a decision on where to place each concrete data block must be made. The space the CM has for data is divided in sets; blocks are directly assigned to the sets; and within the sets any position can be chosen. When the sets have a single line, the correspondence is direct mapped: a MM block can be placed in a single cache line. On the other end, if there is a single set with all the space in the CM, MM blocks can be placed in any position in CM: the correspondence is fully associative. When placing a MM block in CM, if one between several CM lines can be selected to be deleted to load a new one, a replacement policy is implemented.

Write a program that simulates the behaviour of a CM (use the programming language you want). The CM has 8 lines, and the sets can have a single line (direct), 2 lines, 4 lines or 8 lines (fully associative). The write strategy is *write-back*.

The program will have (for instance) the following **input** data**:**

Size of the words	4 or 8 (bytes)
Size of the block/line	32 or 64 (bytes)
Size of the sets	1 (direct), 2, 4, or 8 (fully associative)
Replacement policy	FIFO or LRU
Address	a Memory address (byte)
Operation	read (`ld`, 0) or write (`st`, 1)

and will provide the following **results**:

Interpretation of the addresses: word, block, cache set, tag…
If the access was hit or miss
If a replacement has happened or not
Access-time

Consider for example the following access times, Tcm = 2 cycles; Tmm = 21 cycles (1 cycle from the interleaving buffer Tbuff = 1 cycle). For instance, a hit in CM: 2 cycles, a miss in cache that requires transferring a block of 8 words: 2 + 21 + 7 = 30 cycles.

Suggestion: you can use a 8×5 matrix to represent the CM. Each row in the matrix will be used to represent the information of a line and will contain the following 5 columns: Busy (1/0), Dirty (1/0), Tag (for searches), Repl. (information for replacement), Data (the block). There is an example at the end (this is only an option).

1 point is the maximum mark that can be obtained with the program. The given mark will depend on the achievements or the levels reached. For instance,

(a) Interpreting the address. This is very simple and the minimum to do if you want to obtain something: 0,2 points
(b) Functioning of the CM and updating data and control information:
  - structure of the CM, one or more options (direct, associative...): up to 0,35 points
  - replacement algorithms (no, one, the two): up to 0,3 points
  - hit rates, times ….: 0,15 points

The program will be assessed in a demo performed to the lecturer.

An example of the program input/output

```
$ cachesim

Word size (bytes):   4 - 8         > 4
Block/Line size (bytes):  32 - 64  > 32
Set size (lines):  1-2-4-8         > 2
Replacement pol.: 0(FIFO) - 1(LRU) > 1

Mem. address (byte) (-1 to finish) > 1123
Load (0) / Store (1)               > 1



   > Address: 1123 - Word: 280 - Block: 35 (words 280 - 287)
   > Set: 3 - Tag: 8
   > Cache miss

   > Access time:  cache fetch, 2 -- block transf. (MM>CM or CM>MM), 21+7
   > T_access: 30 cycles


    Busy dirty tag  repl. ||  data
   --------------------- --------------
      0     0    0    0   ||   ---
      0     0    0    0   ||   ---
   -------------------------------
      0     0    0    0   ||   ---
      0     0    0    0   ||   ---
   -------------------------------
      0     0    0    0   ||   ---
      0     0    0    0   ||   ---
   -------------------------------
      1     1    8    0   ||   B35
      0     0    0    0   ||   ---
   -------------------------------

Mem. address (byte) (-1 to finish)  >
```

Addresses can be specified using the keyboard or from a specific file. Finally, the global hit rate and access time must be provided, for all the simulated accesses. For instance,

```
Ref: 4 -- Hits: 2 -- Hit rate, h = 0.50
Total access time = 64 cycles
```