1. Objectives

The main objective of this project is the evaluation of the effect of replacement algorithm and block size on miss rate.

2. Requirements

The only requirement for this project is to learn how to use the software, that is, the SMPCache simulator. The getting started manual of the simulator will help.

3. Development

3.1. Cache block replacement algorithms: LRU, random, FIFO, LFU

Follow these steps:

1. Start SMPCache
2. Configure – main memory – word wide 32 bits, words by block 64, blocks in main memory 1024, block size 256 bytes, main memory size 256 KB
3. Configure – multiprocessor – number of processors 1, coherence protocol MESI, bus arbitration LRU
4. Configure – caches - blocks in cache 32, cache size 8 KB, mapping set-associative, number of cache sets 2, replacement policy LRU, cache levels 1, writing strategy Write-Back
5. File – Open memory traces – Use the WAVE.prg trace included in the SMPCache installation, load it in processor 1 (check the box), click OK
6. View, cache evolution, text, click OK, click Execute

The above sequence of steps sets up an LRU replacement algorithm.

Please record the miss rate for this cache configuration with LRU, random, FIFO, and LFU replacement algorithms. Which replacement algorithm is best, second best, etc?

View the cache evolution in graphics mode once. Does the miss rate remain constant? Why?

3.2. The effect of block size on miss rate

Copy the five ParmFile.cfg files (provided by the professor) into your directory. Then, run the 5 cache configurations – all with a direct mapped 2 KB cache with the following block sizes 4 bytes, 8 bytes, 16 bytes, 32 bytes, and 64 bytes.

The following table shows the setup configurations in the different ParmFile.cfg files. The rest of parameters are equal for all the 5 cache configurations and equal to the parameters in section 3.1:

	Config1	Config2	Config3	Config4	Config5	
L1dcBlocks	512	256	128	64	32	← total blocks in cache
L1dcBlockEntries	1	2	4	8	16	← block size (data/block)
L1dcEntryBytes	4	4	4	4	4	← datum size
L1dcAssoc	1	1	1	1	1	← associativity
L1dcFullyAssoc	0	0	0	0	0	← fully associative or not

Record the cache miss rate for each of the 5 cache configurations – and answer the following questions:

- Which cache configuration provides the best performance?
- What type of locality is exploited with increasing larger block sizes?
- Why doesn't the miss rate continue to decrease with larger block sizes?

4. Reports

The conclusions of this project must be typed. Basically, it is necessary to answer all the questions in the Development section. All your answers should be justified with the help of tables and graphs.