Didactic Cache Memory Simulator Project Using a Structural Modeling

Italo Giovani Abdanur Stefani
Computer Science
Pontifical Catholic University of Minas Gerais
Belo Horizonte, Minas Gerais, Brazil
Email: italogiovani@msn.com

***Abstract*— This document presents the teaching and learning aid tool DCMSim (Didactic Cache Memory Simulator). It is a didactic simulator that uses a structural modeling of cache memories and to be used in undergraduate classes in memory hierarchy topics. The DCMSim was designed to motivate and grow the student participation and reflection through an alternative learning environment. The simulator goal is to represent the cache memory structurally and not only functionally.**

I. Introduction

The study of memory hierarchy systems, including cache memory, may be extremely complex and unmotivated, principally for the hard visualization of the events in the memory positions accesses in the diverse hierarchic levels. To understand the cache memory concepts [Hennessy e Patterson 1996], [Hennessy e Patterson 1997], [Smith 1982] using only the traditional didactic methods, as slides, books or representation in blackboards can be difficult, due to these static resources are not capable to represent simply and dynamically the behavior of memory hierarchy systems.

The use of tools for aid teaching and learning in memory hierarchy systems has become important day by day. This is justified for the facility provided by these tools to create new use cases to illustrate and explain cache memories' architectures and configurations. The complex visual representation of the memory hierarchy system's behavior and functioning is delegated to the tool.

Frebuary 1st, 2005

A. The problem to solve

The teaching inside classroom using conventional methods as drawings, slides, books or blackboards, are usually static representations of the events that involve memory hierarchy, including cache. Is possible to show a dynamic cache memory behavior through a sequence of static representations, but this demand a hard work and time. These methods always limit the teachers to use the same examples unless they spend too much time proposing and creating some use cases to explain and exemplify cache memories' architectures and configurations, and also to compare results obtained by simulating different types and configurations of cache memory.

Thus, the motivation problem of this research is the difficulty and spent time for manual exercises resolution using conventional methods. To understand an exercise is necessary to observe the functioning and to analyze the cache results, that are the main points of the study and learning of memory hierarchy systems' concepts. With the traditional methods is considerably necessary to worry how to make the step-by-step representation of the system processes studied, and this can demand much time and turn the learning an unmotivated activity.

B. Project Goals

This work presents a teaching/learning aid tool for study and understanding cache memories where the main goal is to presents visually, simply and interactively, the memory address access events of a real cache memory.

Another goal is to allow, interactively, the analysis and comparison of some cache memory systems through results generated in the simulations. So, new possibilities of learning are created through experimentations varying cache architectures types and configurations and memory trace (sequence of memory address access). These new possibilities can be analyzed each one separated and, principally, all together [Martins et al. 2002].

It intends, in this project, to create an environment that motivates the students and stimulates their participation and contribution in the studies and tool's development. The use of alternative resources, as simulation, is intended to reach this goal easily [SAGE 2004]. Experiences using simulators, as the first version of **DCMSim**, disclose that the teaching/learning become more efficient because it stimulates the participation and reflection of the student. Then, they start to collaborate more and more, contributing to improvements for the tools developed with this intention.

DCMSim is a simulation tool based in structural modeling of cache memory representation, and it shows each cache's structures and how they interact. Therefore, the **DCMSim** tool shows the cache memory functions, as the most of the correlated works, but it also display the cache memory structurally, on the contrary to the correlated works. The simulator presents a high abstraction level of the cache memory concepts, to be used as a visual explanation of these systems' functioning and performance, turning the tests, analysis and understanding more interactive, efficient, trustworthy and didactic.

C. Project Motivation

The difficulty of visual perception in cache memories' functioning is well-known, due to cache memories, even being a hardware, its architectural composition is an intangible device to be studied. Thus, it has a great difficulty for students in perceiving as a cache memory is organized in the reality and the events that occur for its functioning, as politics of substitution, how the blocks are stored and loaded, how the structure of cache memory interacts with the memory hierarchy.

The motivation for this work is the possibility to supply a teaching/learning aid tool for use in under-graduation disciplines that deals with memory hierarchy. It happened in Computer Architecture classes with the first version of the tool, **DCMSim Crawl** [1] (details are in section II), and now with the **DCMSim V1.0**. Representing the cache memory as real as is possible and providing the possibility to have more practical exercises, the **DCMSim Crawl**, that were developed in a under-graduation discipline and generated publications [Cordeiro et al. 2003], [Stefani et al. 2003], [Stefani et al. 2003], were used in 2003 and 2004 as aid tool to solve exercises and as an example to other simulators in Computer Architecture disciplines in the Pontifical Catholic University of Minas Gerais (PUC Minas). The plan is to use the new **DCMSim** version, presented in this document, in 2005 intending to improve the results of the aid tool's use.

II. Correlated Works

The state of the art research disclosed that the majority of the cache memory simulators does not have a didactic approach. These simulators are focused in the study and analysis of the cache memories performance.

The Dinero IV [Hill 2004] is a simulator for cache memory access references. It is toward to multi-level cache simulation and analyzes the performance of the associations of these caches; besides presenting information about cache hit and cache miss (also classifying the miss as compulsory, conflict or capacity). However, it is not propose characteristics as time access counting, functional simulation, structural representation neither didactic resources.

The SimpleScalar tool [Austin 2004] is a system used in construction of performances analyses applications in real programs through a micron-architectures vision or software vision. The tool has been widely used in researches to statistics collection and evaluation of the developed program. It is not a tool that has concerns aid to teaching/learning, although it's strong used in academic field.

The project called Coyote [Vera. 2002] has the objective to evaluate the cache memories performance in compilation time and its possible application. It presents a cache memory simulator to validate this approach in compilation time. This research follows the attempt to improve the cache performance evaluating the cache memory and the applications that are being executed. No concern in teaching/learning is presented in Coyote, but it supports performances researches to cache uses situations.

It is possible to find some simulators that represent the functioning of cache memory systems in a didactic approach. A first example that can be cited is *A Cache Memory Simulator Project and Development: functional and performance analysis* [Costa, Pousa e Martins 2002]. This simulator was developed using the C++ language and with functional implementation approach (functioning model). It allows to simulate and the observe the memory addresses accesses at only one type of cache memory architecture (fully-associative), with the possibility to alter the computational system parameters, as cache memory and main memory access time, size, main memory access type (parallel or sequential), substitution politics FIFO (First in First Out) or LRU (Least Recently Used).

Another relevant work is the **DCMSim Crawl** [Cordeiro et al. 2003], [Stefani et al. 2003], [Stefani et al. 2003]. It presents a didactic simulation tool for analysis and comparison of functioning and performance of different cache memories types and configurations. Its development was considered on 2002/2 in the under-graduation disciplines Computer Architecture as an activity of learning through research [Martins et al. 2002]. It make possible the students to verify and complement the concepts acquired in the discipline [Martins et al. 2002], [Hennessy e Patterson 1997]. For this activity, there was a simulator as example developed by students on 2001/1 [Costa, Pousa e Martins 2002]. The **DCMSim Crawl** was in use until in 2004/2 in Computer Architecture classes in the PUC Minas as a teaching/learning aid tool and as example for new developments of considered similar projects inside the disciplines.

The cache memory simulator KSH [Silva e Pinho 2003], beyond the simulation of three types of cache memory, possess dynamic loading of modules, which is possible to add in execution time a new class contend a type of cache memory architecture, making possible more flexibility and that some characteristics turn independent of the main program. This is a simulator focused only in the results that are generated by the inputs data and there is no functional, neither structural, representation, as necessary to a better teaching/learning of cache memories concepts.

In the scope of didactic tools, is possible to find other examples in different topics, but with intersections of the proposal in **DCMSim** project. The first relevant one is the tool Rin'G, described in [Cordeiro et al. 2004], [Cordeiro et al. 2004]. This is a tool for edition and creation of graphs that provides an environment of algorithms development and animation. There is a non-intrusive characteristic in the algorithm development to generate animation, it presents an intuitive graphical interface and consider about the academic field. These functionalities have the objective to reduce the effort to use tool adaptation and better exploitation of the tool in Graphs and Theory the Complexity disciplines, where it is used since 2004/1.

[1] The first version of the DCMSim, developed in Oct. 2002, is referred in this work as "DCMSim Crawl" or only " Crawl". The new version, presented in this document, is referred as "DCMSim v1.0". The term " DCMSim" will be used in many cases to refer the project, including the two versions.

The work *Simulation and Advanced Gaming Environments (SAGE) for Learning* [SAGE 2004] is a project that explores the potential of simulators and games for support the teaching/learning. This work is based on new ways of education that search greater interaction, contribution and reflection of the student. The research intends to better understand the education using surrounding of simulators and games to be able to take advantage of this better technique. Currently, the work is focused in education in the health area, as public health and professional medical education. This project resembles with **DCMSim v1.0** for uses simulator and an alternative environment, seeking a better teaching/learning method.

III. DCMSIM V1.0 TOOL

DCMSim (*Didactic Cache Memory Simulator*) is a didactic simulator tool to analysis and comparison of functioning and performance of different cache memories' types and configurations. **DCMSim** possesses a structural cache memory modeling; this means that all the cache elements are represented as objects (class) in the software structure. The intention is represents as real as possible a cache memory and not be only a functional simulator.

The simulator intends to serve as a didactic tool to assist in the analysis and comparison of results and performance between different cache memory systems that also could be used to study the functionalities that compose this type of system. The **DCMSim v1.0** version represents an improvement of the early version **Crawl** [Cordeiro et al. 2003], [Stefani et al. 2003], [Stefani et al. 2003], that widely was used by the under-graduation students since the first semester of 2003 in Computer Architecture disciplines. In this time, it was possible to verify new necessities, problems and points that could be improved.

In the new version presented, **DCMSim v1.0** showed in figure 1, the represented cache memory elements are: cache in itself, cache blocks, slots and the cells of each block. The main memory is represented to complement the teaching/learning of cache memory, without deep the memory hierarchy concepts, it is represented by its blocks and cells of each block. The functions represented are: substitution politics and writing politics. As didactic resource, the *MissAnalyst* component is represented, that it analyzes the cache miss and cache hit, classifying the misses between compulsory, capacity or conflict. The *MissAnalyst* is not a presented element in the real cache memory and was enclosed in **DCMSim v1.0** to help understanding cache misses.

The inputs data are represented by the set of memory accesses, called *memory trace*, that simulates the solicitations made by a presumption process in execution. The iteration, or each simulation step, is given by the process solicitation, where all the process on the desired entrance is executed and a result is gotten. This process can be followed in each represented component displayed. The *LogBox* shows what happened with each input, and in the statistician section is possible to verify the expensed access times in the current iteration and the entire

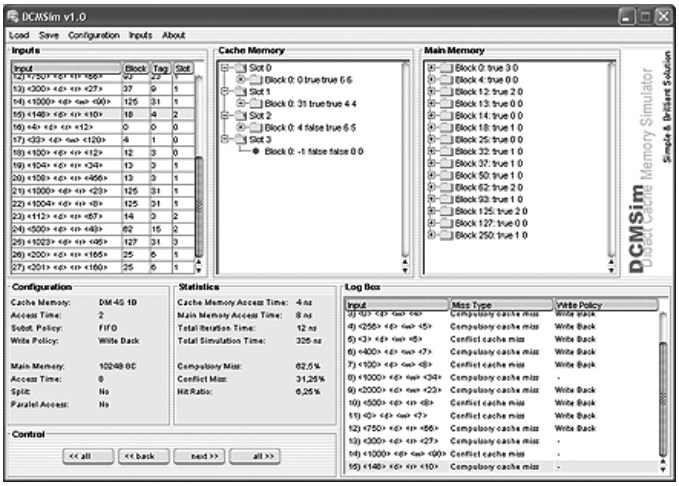

Fig. 1. Interface geral do **DCMSim v1.0**.

simulation. The miss and hit ratio also are computed on each iteration and showed in the statistician section.

The **DCMSim** allows, on the end of each simulation, the creation of an log file that indicates the occurrences sequence in the cache memory for the given *memory traces* and its results, this file still includes information about the cache configuration used and the hit and miss ratios.

IV. USE CASE

Will be presented a short, but complete, use case to better explain the **DCMSim V1.0** functioning. It will execute a *memory trace* in two differents cache configurations and, in the end, compare the results. This case used the *memory trace* showed in figure 2 that was also used for results demonstration in the published essay [Cordeiro et al. 2003] about the **Crawl** version. The first cache configuration is a fully associative cache memory with four slots, substitution politics FIFO and writing politics *Write Back*. The cache memory access time was 2ns and the to main memory was 8ns. To this document, the configuration will be codified as *FA 4S 1B FIFO WB*, referred only as **FA**. The second cache used contained the same configurations, however, it was a direct mapping, its codification is given as *DM 4S 1B FIFO WB*, referred as **DM**.

```
512 d r 1
500 d r 1
500 d w 2
100 d w 3
0 d w 4
512 d r 1
256 d w 5
3 d w 6
400 d w 7
100 d w 8
```

Fig. 2. (*Memory trace* used in simulation).

The figures 3 and 4 show the results of the two simulations (fully associative and mapping direct respectively). It is possible to perceive that modifying the type of memory cache the

```
Total Time: 96ns

Compulsory Miss:   60%
Capacity Miss:      0%
Hit:               40%

0) <512> <d> <r> <1>  Compulsory cache miss  -
1) <500> <d> <r> <1>  Compulsory cache miss  -
2) <500> <d> <w> <2>  Cache hit              -
3) <100> <d> <w> <3>  Compulsory cache miss  -
4) <0> <d> <w> <4>    Compulsory cache miss  -
5) <512> <d> <r> <1>  Cache hit              -
6) <256> <d> <w> <5>  Compulsory cache miss  -
7) <3> <d> <w> <6>    Cache hit              -
8) <400> <d> <w> <7>  Compulsory cache miss  Write Back
9) <100> <d> <w> <8>  Cache hit              -
```

Fig. 3. Simulation result of FA 4S 1B FIFO WB

```
Total Time:128ns

Compulsory Miss:   60%
Capacity Miss:     30%
Hit:               10%

0) <512> <d> <r> <1>  Compulsory cache miss  -
1) <500> <d> <r> <1>  Compulsory cache miss  -
2) <500> <d> <w> <2>  Cache hit              -
3) <100> <d> <w> <3>  Compulsory cache miss  -
4) <0> <d> <w> <4>    Compulsory cache miss  Write Back
5) <512> <d> <r> <1>  Conflict cache miss    Write Back
6) <256> <d> <w> <5>  Compulsory cache miss  -
7) <3> <d> <w> <6>    Conflict cache miss    Write Back
8) <400> <d> <w> <7>  Compulsory cache miss  Write Back
9) <100> <d> <w> <8>  Conflict cache miss    Write Back
```

Fig. 4. Simulation result of DM 4S 1B FIFO WB

access time results modifies, as well as percentages of ratio and writing in main memory. It is well-known that the cache functioning intervenes in the access results for one determined *memory trace*. For cache FA, is possible to notice that the way in which the requested data were stored had provided a lesser number of hits diminishing the writing in main memory and reducing the simulation access time. Having written only in the data access shown in line 8 at the figure 3.

For DM cache, it had a great number of conflicts when the blocks that contain the data had been mapped. This resulted in more substitutions, which turned necessary writings and consecutively spend more time to execute the same *memory trace*. This happened due to some slots remained empty while a substitution was done because a conflict. This was the advantage of cache FA, however is important to remember that this happened only to this configuration and *memory trace* used.

The differences results can be noticed varying the cache memories configurations and type, and also they are dependents of the *memory trace*. The presented example, although simple, obtains to cover all variations that occur in the cache functioning, that are of utility for the teaching/learning. In this way, we can prove the efficiency and utility of the new **DCMSim** version.

V. Conclusion

The conclusion is that **DCMSim** had a correct functioning and is intended to surpass the results gotten with the version **Crawl** in the teaching/learning aid.

The objectives defined in this project had been reached, and the non-implemented items are foreseen as future works, as described in the section V-B. These items will be ready to the use in 2005 through the **DCMSim v1.0**.

DCMSim approaches cache memory systems and memory hierarchy concepts. Due to the available resources of experimentation, the tool supplies an objective faster way of teaching/learning aid. **DCMSim** approaches the cache memory in its structural and functional form, on the contrary to the most similar tools.

DCMSim had its first version, **Crawl**, disposed for the Computer Architecture students. Working with the professor, they had been able to confer exercises and to verify the cache memory functioning. This practical activity already turns the disciplines more motivated and efficient in solving exercises and absorption of the given theoretical basement. For the new developed tool is expected to surpass these conquests and improve even more the teaching/learning methods.

A. Contributions

The **DCMSim** contributes to a better understanding and analysis of the cache memory systems functioning process and performance, and its understanding in a structural way. Thus, taking care of the students and professors' necessities and expectations practically and didactically. **DCMSim** is not only a teaching/learning aid tool, but also a efficient method to produce and setting knowledge inside of disciplines.

Exploring the potential use of alternative educations methods, **DCMSim** supplies an environment that stimulates and motivates the students and professors participation. It is a tool that searches to open doors for new ways to educate more efficiently than the conventional methods. It contributes to researches about human behavior front to the diverse technologies with didactic character, not only in cache memory topics, but also as tool of aid to teaching/learning in general.

Beyond the acquired theoretical knowledge, the author of this work could ripen its abilities related with research, elaboration and development of projects. The adopted research characteristic contributes to a motivated work due to the diverse results and achievements, as success published essays and presentations of the tool in scientific congresses.

B. Future Works

The **DCMSim v1.0** already presents an use description with the **Crawl** use, when was possible to evaluate some characteristics. This experience of use indicated the tests to do in the new version and helped to define the **DCMSim v1.0** main goals.

The first step as future works will be the implementation of the requirements specified and made impracticable until

the moment. These items must be ready for use in the first semester of 2005. Being able to cite:

- dynamic modules loading;
- memory split;
- memory parallel access.

As the previous version, **DCMSim v1.0** will be used in Computer Architecture disciplines and will be possible to verify the efficiency of the functionalities previously consolidated, and also, to evaluate the new enclosed functionalities. The future work includes, to the new version, tests of usability and analysis of functionalities, this last enclosed as an undergraduation work in the **Crawl** version.

As occurred previously with **DCMSim Crawl**, is intended to prepare and to submit academic essays referring to the new features and results gotten with the simulator use as teaching/learning aid tool. Presentations in conferences and congresses also are foreseen.

REFERENCES

[Austin 2004]AUSTIN, T. *SimpleScalar LLC*. 2004. Http://www.simplescalar.com/. ltimo acesso em Novembro de 2004.

[Cordeiro et al. 2003]CORDEIRO, E. S. et al. DCMSim: Didatic Cache Memory Simulator. *33nd ASEE/IEEE Frontiers in Education Conference 2003 - FIE 2003 - Boston*, p. F1C14–F1C19, 2003. ISSN 0190-5848.

[Cordeiro et al. 2004]CORDEIRO, E. S. et al. Rin'G: Um Ambiente No-intrusivo para Animao de Algoritmos em Grafos. *XII WEI - Workshop de Educao em Computao*, p. 1126–1137, 2004. Salvador - BA - Brasil.

[Cordeiro et al. 2004]CORDEIRO, E. S. et al. Uma Proposta Metodolgica para o Aprendizado de Algoritmos em Grafos Via Animao No-intrusiva de Algoritmos. *III WEIMIG - III Workshop de Educao em Computao em Computao e Informtica do Estado de Minas Gerais*, p. 73–86, 2004. Belo Horizonte - MG - Brasil.

[Costa, Pousa e Martins 2002]COSTA, A. S.; POUSA, C. V.; MARTINS, C. A. P. S. Projeto e desenvolvimento de um simulador de memria cache: anlise funcional e de desempenho. *III Workshop em Sistemas Computacionais de Alto Desempenho WSCAD2002*, p. 152–153, 2002.

[Hennessy e Patterson 1996]HENNESSY, J. L.; PATTERSON, D. A. *Computer Architecture: A Quantitative Approach*. second. [S.l.]: Morgan Kaufman, 1996.

[Hennessy e Patterson 1997]HENNESSY, J. L.; PATTERSON, D. A. *Computer Organization e Design: The Hardware/Software Interface*. second. [S.l.]: Morgan Kaufman, 1997.

[Hill 2004]HILL, M. D. *Dinero IV Trace-Driven Uniprocessor Cache Simulator*. 2004. Http://www.cs.wisc.edu/ markhill/DineroIV/. ltimo acesso em Novembro de 2004.

[Martins et al. 2002]MARTINS, C. A. P. S. et al. A New Learning Method of Microprocessor Architecture. *32nd ASEE/IEEE Frontiers in Education Conference 2002 - FIE 2002 - Boston*, p. S1F16–S1F21, 2002. ISSN 0190-5848.

[SAGE 2004]SAGE. *SAGE - Simulation and Advanced Gaming Environments*. 2004. Http://www.sageforlearning.ca/. ltimo acesso em Novembro de 2004.

[Silva e Pinho 2003]SILVA, A. R. da; PINHO, R. D. KSH: Simulador de memria cache com carregamento dinmico de mdulos eexecuo de script de configurao. *IV Workshop em Sistemas Computacionais de Alto Desempenho WSCAD2003*, p. 160–163, 2003.

[Smith 1982]SMITH, A. J. Cache memories. *Computing Surveys*, v. 14, n. 3, p. 473–530, 1982.

[Stefani et al. 2003]STEFANI, I. G. A. et al. DCMSim: Projeto e Desenvolvimento de um Simulador Didtico e Multiplataforma de Memria Cache Usando Modelo Estrutural de Cache. *IV Workshop em Sistemas Computacionais de Alto Desempenho WSCAD2003*, p. 152–155, 2003.

[Stefani et al. 2003]STEFANI, I. G. A. et al. DCMSim: Uma ferramenta para simulao de memria cache. *IV Workshop em Sistemas Computacionais de Alto Desempenho WSCAD2003*, p. 182–183, 2003.

[Vera. 2002]VERA., X. Coyote Project: The Simulator. 2002.