Web Memory Hierarchy Learning and Research Environment

José Leandro D. Mendes [1], Luiza M. N. Coutinho [1], Carlos A.P. S. Martins [1]
[1]*Computer Science Department -*
Pontifical Catholic University of Minas Gerais
Av. Dom José Gaspar, 550 – 30.535-500 - Belo Horizonte – MG – Brazil
{joseleandrodm, luizamncoutinho}@yahoo.com.br, capsm@pucminas.br

Abstract

Learning the various structures and levels of memory hierarchy by means of conventional procedures is a complex subject. A memory hierarchy environment (Web-MHE) was proposed and developed as part of our undergraduate research, to serve not only as an auxiliary teaching tool for the Computer's Architecture professor, but also as a learning facility tool for the student carrying out his undergraduate or graduate courses. The environment encompasses the knowledge of the various levels of the memory hierarchy, and can be either used by students in his studies or by senior researchers. In conclusion, the Web-MHE offers the user the possibility to simulate, experiment and learn the various concepts of the memory hierarchy.

1. Introduction

The learning of the memory hierarchy is a complex process mainly due to the difficulties faced when attempting to understand and visualize the process that take place during the accessing of the memory positions in the various hierarchy levels. Teachers are usually face with great difficulties in grabbing the students' attention while using slide shows or other static teaching tools, and, from the student's perspective, the classes gradually become boring. The teachers face difficulties in presenting, in a simple format, the various hierarchy levels using the available conventional teaching tools [1]. The use of simulation didactic tools may well contribute to the learning process steps, as well as to the designing of more objective exercises, enabling the students to compare the results of different configurations and memory traces, and to visualize the various hierarchy levels. The difficulty and complexity faced by the students when solving exercises and verifying questions may be reduced to a minimum with the use of didactic simulators, making the problem solving task a more pleasant one.

The Memory Hierarchy Learning and Research Environment with support the Web (Web-MHE) [2] is composed by a group of didactic materials with which the student can learn alone or to complement his study on the main memory hierarchy concepts, a simulation tool (Web-MHSim - Web Memory Hierarchy Simulator), where the student can test and to increase his knowledge and configurations and memory traces files previously defined, that can be chosen by the user and loaded so that an automatic simulation can be generated.

The Web-MHSim is a didactic memory hierarchy simulation tool of (cache, main and virtual memories), that it has as objective to be an ally of the teacher in the teaching and of the student in the learning, by providing easy ways to configuring and simulating possibilities for different structure options. The simulator is able to handle the different levels of the student's knowledge on the subject, of beginners, in the process of getting familiarized with the theory of memory hierarchy, to the advanced researchers, that want a simulator to verify the results of their research work, in an efficient and quick way. The simulator Web-MHSim is an evolution of the MSCSim [1][3].

For verification of the results were considered some simulators and one environment as reference. The developed environment looked for improvements and new functionalities in relation to these. To facilitate the spread and platform independence, the environment was developed for Web.

In synthesis the Web-MHE has as goal to be an ally of the teacher in the teaching, to enlarge the degree of the students' knowledge and to facilitate the research of the several structures of the memory hierarchy. The users have the possibility to learn, to simulate and to try the several concepts and techniques of the memory

hierarchy, increasing his degree knowledge, to execute practical exercises, to accomplish experimentations and other.

2. The use of Simulators in Computer Architecture

The traditional teaching method, where the teacher presents the various concepts of memory hierarchy by means of static devices such as slides shows and text books is not enough to give to the majority of the students a precise understanding of what is being taught.

The integration of the traditional teaching method with the use of didactic simulators contributes to a better understanding of the subject since they are capable of translating to reality the theoretical concepts by introducing them in a clear and didactic fashion [4]-[9].

The Simulators comprise the creation of dynamic and simplified models of the real world. Simulators, in general, despite being simpler than real systems, often present a reasonable difficulty degree in their use, due to the knowledge level required to handle them.

Any didactic Simulator must be capable of offering to the student a possibility of choice of the simulation complexity according to his knowledge level on the subject, enabling him to gradually increase the simulations complexity degree. Other important features required to a simulator are the platform independence, easy installation, configuration and use of the tool, reducing the time spent in learning its operational requirements and procedures. The tool must offer a great amount of information in a clear and objective manner, therefore contributing to the student's learning process while the student increases his critical evaluation and knowledge degree, through the execution of various exercises and experimentation of different hypothesis, as well as checking solutions for specific problems.

Some memory hierarchy Simulators such as Dinero IV [10], CacheSim [11], VirtualMemory [12], CACTI [13], LDA-Cache [14] and Prima [15] (reference simulators), are known in the academic world but do not possess all the features previously referred.

3. Related Works

HASE Dinero [16] stands out for being an environment learning Web and research of cache memory. It was considered as reference for comparison and definition of new functionalities of the Web-MHE. HASE Dinero was the only environment Web of cache memory with didactic resources found in the study of the state of the art.

The simulators CacheSim and Dinero IV have as main feature the cache simulation and VirtualMemory stands out for the simulation of virtual memory. The most interesting features from these simulators were selected to they be incorporated in the Web-MHSim. For verification of the evolution of the tool MSCSim, was used whose the base aided in the development of the Web-MHSim.

HASE Dinero is a teaching environment of cache memory that was developed at the Institute for Computing Systems Architecture of the University of Edinburgh. The environment is composed by a group of didactic materials, animations that can be visualized in the Web, simulator desktop that possesses all of the functionalities of Dinero IV (except multiple levels and multiple input formats) and previously defined files of configurations that can be used in the simulation. In this tool is not allowed the simulation in the Web, simulation of virtual memory, generation of memory traces files and others.

CacheSim is a tool Web of cache simulation that allows the users to specify a configuration and choose one of the memory trace existent, so that, soon afterwards, they can verify the acting of the cache. With CacheSim is not possible to accompany the simulation step by step, the time of access, performance graphics, some simulation types (only the simulation of unified, instruction or data cache memory it's possible) and it doesn't possess memory trace generator and interaction with the user during the simulation. The memory traces files are previously defined and it doesn't allow the configuration for the user.

Dinero IV was developed at the University of Wisconsin by Mark D. Hill. The results are obtained through the reading of a memory trace, parameters and configurations specified. It is possible to configure several types of caches (direct mapped, set associative and fully associative), that can compose a cache unified, separate and/or multiple levels. Block's size, associative levels and other parameters can also be specified. In this version it is also possible to classify the type of miss (compulsory, conflict and capacity). Parameters such as time and type of access, oriented interface for apprenticeship as well as some types of simulations such as multiple levels and virtual memory are not found in this Simulator.

VirtualMemory was developed by Ngon Tran e Dr. Daniel A. Menasce' at the George Mason University. This tool presents a didactic interface that permits the virtual memory simulation (TLB, main memory, page table and hard disk) and exhibits statistical data during

the simulation. Other type of simulation, configuration and memory trace isn't possible.

MSCSim was designed and developed with the objective of to be a didactic simulator for desktop. In this, it is possible to simulate and to accompany the behavior of several combinations of structures, as unified or split cache, multiple levels and memory hierarchy (cache, main and virtual memories). The simulations offer a great wealth of details, showing step by step as each address allocated. Among its main features stands out memory trace generator, statistics, didactic animation of the hierarchy structures and interface guided to the learning. In this version the users can't enter with data during the simulation, to accomplish simulation with support the separated main memory, to visualize statistical and other graphics.

4. Web-MHE

In computer systems, in order to achieve a high performance level in the handling of information (data and program instructions) to and from the computer memory, various types of memory (memory hierarchy) are required in each single computer. For some specific tasks, it's of the utmost importance that the transfer of the information is to be carried out as fast as possible. It's the case of the tasks carried out in the central processor (internally) where speed is the main requirement, but the amount of bits to be manipulated is very small. This feature characterizes one type of memory known as cache (physically closer to the processor), meaning any memory that, if used in conjunction with a slower one (for instance, the main memory) may reduce the average access time. This memory can be organized as unified, split (separated cache memory) and at various levels (multiple cache levels).

The main memory is the place where the data and instructions are stored and recovered by the processor. This, can be unified (Von Neumann architecture) or separate (Harvard architecture), where exists a main memory of data and another of instructions.

Therefore, accesses that generate hits at the highest hierarchical level can be processed faster. The accesses that generate miss are complied with access lower hierarchical levels, which are slower and bigger. If the hit rate is high, the hierarchy has an average access time very close of the first level component, therefore being faster [17]-[21].

The Web-MHE is inserted in the memory hierarchy learning context, and more precisely in the use of didactic environments for this purpose. The main goal is to be an Web environment with the desirable didactic features (tutorials, didactic interface, Web simulator that has several types of configurations and simulations, inclusion of knowledge levels, configured previously files and animation). Other objectives were the incorporation of the main functionalities of the references and new resources that helps the student and the researcher.

The environment was developed in HTML and JAVA because it's a language free, robust and platform independent that became possible an object oriented development and the use of applet resources. Some software engineering concepts were used as pre-requisites definition, use case and class diagram.

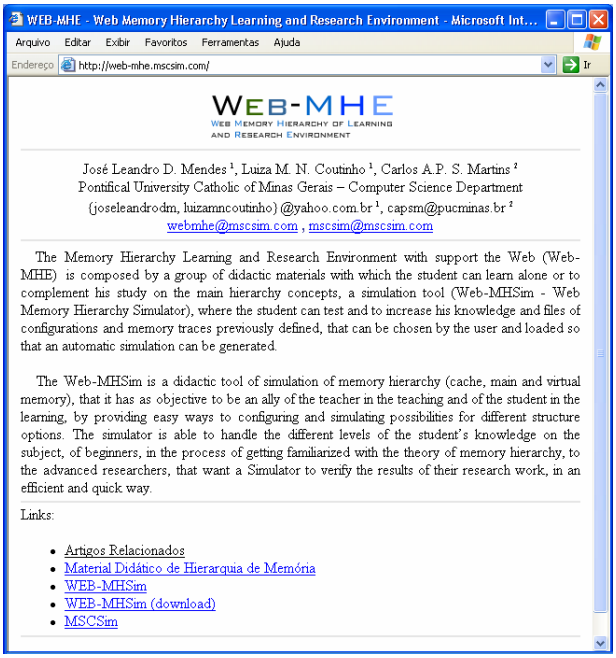

Figure 1. Web-MHE general view.

The Web-MHE is a Web didactic environment that includes a simulation tool, pre-configured files, didactic materials and memory hierarchy animations (Figure 1). The main requirement of the environment is to offer a didactic tool of memory hierarchy simulation.

The main aim of the didactic materials is to offer to the student resources capable to aid him in the study or in the learning alone of memory hierarchy. These materials are composed by cache lectures (direct mapped, set associative, fully associative, multiple levels and split cache), main memory (unified and split), virtual (TLB, pages table and virtual memory) and others. During the lectures are presented pre-configured files where the user can simulate and to visualize the results of the theory. Also are available

animations about these themes, with the objective of helping the student's learning.

One of the main requirements of the Web-MHSim (Figure 2) it is the possibility to simulate and to accompany the behavior of some combinations of structures, as unified cache, split cache, multiple levels and virtual memory. Other requirements considered in the design of the simulator were the possibility to offer a didactic tool, the user's possibility to inform relative data to the accesses during the simulation and support to Web, among others.

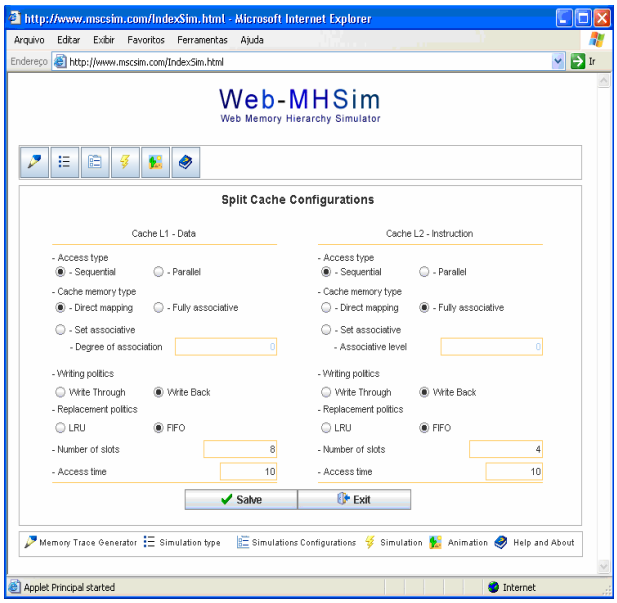

Figure 2. Web-MHSim general view.

The simulations are offered with a wealth of details, showing, step by step each allocated address, also allowing the user to inform the relative data to the current address (block, slot, tag, miss/hit, fault type, access time, virtual page and frame), helping the student in the analysis and performance evaluation of the configured hierarchy.

The tool allows accomplish simulations of unified cache with main memory and unified cache with main and virtual memories. This tool can simulate a separated cache with some configuration type for their caches. In the simulation of multiple levels is possible to configure *n* levels, which can be composed by unified or separated caches. The main memory can be configured as unified or separated, for all of the cases.

Parameters are offered for the configuration of the memory cache (Figure 2), such as access type (sequential or parallel), cache memory type (direct mapped, set associative and fully associative), Write Through (WT) and Write Back (WB) write policy, FIFO e LRU replacement policy, number of slots and cache levels, access time, associative degree and block size. Access time and word size can be defined as main memory parameters. The level of virtual memory allows the configuration of the number of slots and the TLB (Translation-look aside buffer) replacement policy, memory and virtual page size.

With the objective of facilitating the understanding of the theory and the use of the environment, resources were incorporate to the simulator, such as the memory trace generator (memory access sets), advanced statistics, calculations guide (demonstration of all calculations made by the simulator) and memory hierarchy animation.

The generation of the memory trace file can be carried out in manual or random mode. In the first, as it can be seen in the figure 3, it is necessary to specify three fields: address, write (W) or read (R), data (D) or instruction (I). For the random generation the required fields are the total number of addresses and the lower and higher value for random generation of addresses. The memory trace is stored in memory for use in the simulation phase. Another possibility is the use of memory trace files configured.

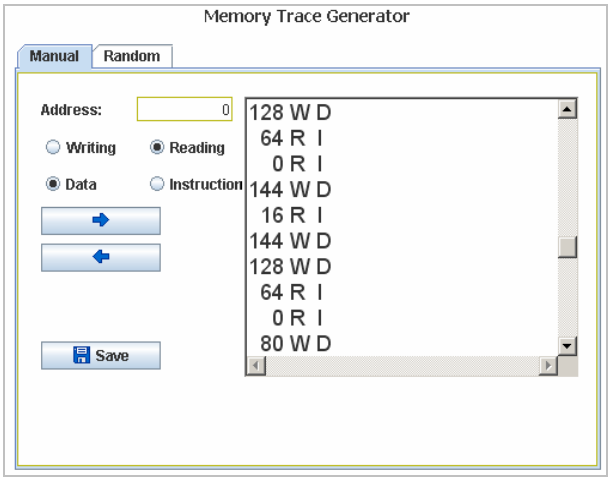

Figure 3. Memory trace generator.

During the simulation is possible to accompany the performance of the structures analyzing the information of the addresses (block number, slot, tag, page and frame, miss/hit, fault type and access time), the remaining addresses in the structure and the statistics (Figure 5, 6, 7 and 8). The main statistical informations offered are: number of distinct blocks, hit rate (percentage of caches hits in relation to the number of accesses), compulsory, conflict and capacity rates (number of faults in relation to the number of occurred cache misses), cache memory, main memory, TLB and page table occupation rate, number of misses

with replacement in the cache and TLB, cache memory, TLB and table pages hit rate, as well as average cache access time. At the same time it is possible to analyze a graphic with the rates obtained during the simulation, aiding the student in the verification of the simulated hierarchy performance.

The user can interact with the simulation, at some moment, informing the values of the block, slot, tag, miss/hit, fault type, access time, fault/hit, virtual page and frame number as it is shown in the figure 5. Later the simulator verify the informed values and it exhibits for the user their mistakes and successes. The student with the interaction can verify and to increase his knowledge level.

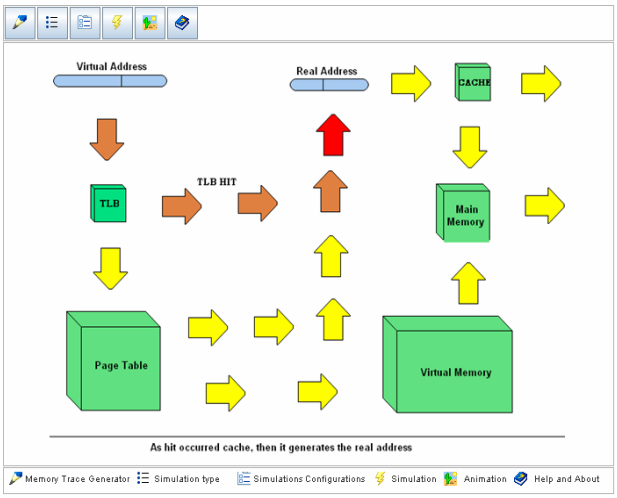

Figure 4. Memory hierarchy animation.

Didactic animations were developed to facilitate the student's understanding of the hierarchy as a completely. In this, it is possible to accompany the path of the addresses to the processor (Figure 4).

5. Results

The Web-MHSim is very flexible allowing the memory trace generation, configurations of parameters and simulation types. The user accompanies and it can interact to each step of the simulation through detailed and easy-to-understand screens.

The simulation begins after the configuration of the parameters and memory trace. After initiate, due to the interface guided to the learning, the user can visualize step by step the input of each address, their information and the actual state of each hierarchy, facilitating the study of the simulated hierarchies.

It is possible to verify the situation and the simulation of the caches with a great wealth of details, what is not found in the reference simulators. The results, statistics and graphics supplied in the screens help to understand if a configuration is the most suitable for the hierarchy.

With a memory trace composed by 100 addresses, were accomplished some simulations, where, in a first moment it is presented a simulation that possesses a basic configuration that can be understood even by a beginner student, in a second moment is accomplished a small one gets better in the simulated architecture and in a third moment a simulation more complex is presented, typical for a student that has got a wide knowledge of the theory or a researcher that wants a fast and correct answer of an architecture.

For all simulations were considered the block size of the cache to 32 bytes, size of the word to 4 bytes, write back policy, FIFO replacement, sequential access, unified main memory and main access time to 500 ns.

5.1 Basic Simulation

With the use of an unified cache memory, fully associative with 4 slots, access time of 15 ns, only 20 cache hits happened, due to the occurrence of various cache miss substitution (Figure 5).

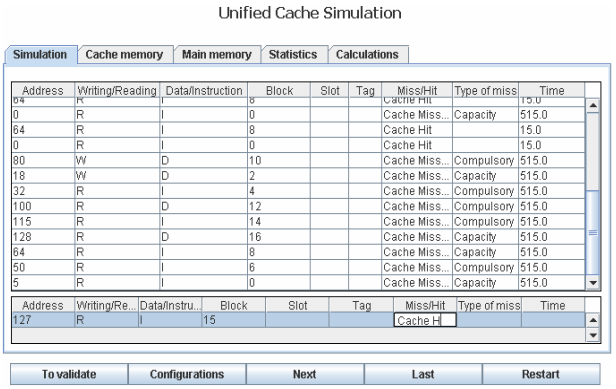

Figure 5. View of an unified fully associative cache simulation.

In this simulation the hit rate (percentage of cache hit) obtained was of 20%, most of the cache miss (address not found in the accessed cache) was of the capacity type (84% approximately), they happened 76 substitutions and the average access time was of 415ns (Figure 8).

5.2 Intermediary Simulation

The intermediary simulation was composed by a split cache (cache L1 – data, L2 – instruction), both fully associative with 4 slots, access time of 15 ns. In

this happened 46 cache hits, due to elimination of the conflicts between data and instructions for the same slots (Figure 6).

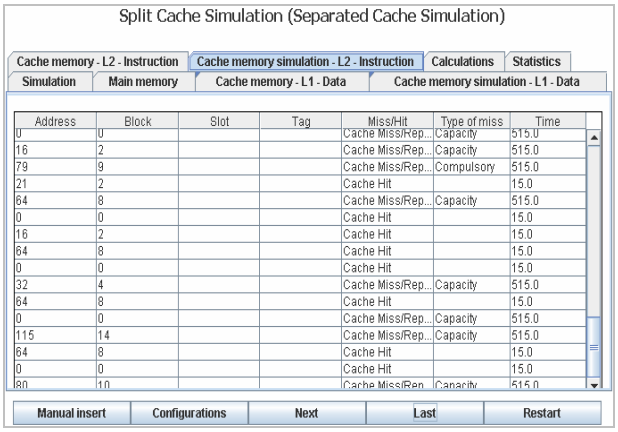

Figure 6. View of a split cache simulation.

The hit rate obtained by the data cache was 44% and by the instruction cache was 47%, most of the cache miss was of the capacity type (data cache 74% and instruction cache 68% approximately) and the average access time was 287 ns, obtaining an improvement of 31% in relation to the average time of the basic simulation.

5.3 Advanced Simulation

The advanced simulation (Figure 7) was configured with 3 cache levels. The first level is composed by a split cache (cache L1 – data and cache L2 - instruction), both fully associate with 4 slots and access time of 15 ns, the second level (cache L3) was composed by a unified cache, direct mapped with 16 slots and access time of 30 ns and the third level (cache L4) was composed by set associative (four way) with 8 slots and access time of 60 ns. In this simulation the closest processor level has as objective avoids the conflict between the data and instructions for the same slots and the lower levels reduce the miss penalty.

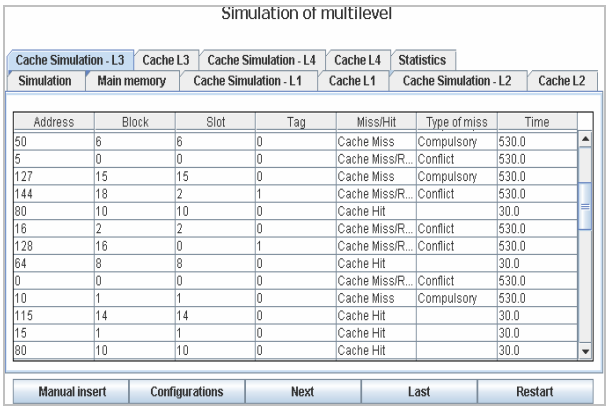

Figure 7. View of the second level of cache simulation.

The hit rate obtained by the data cache of the first level was 44% and by the instruction cache was 47%. The second level also obtained a hit rate of 48%. Already the last level obtained a hit rate of 54%. The data and instruction cache occupation was totally busy. The second level cache was occupation rate of 69% and the third level presented an occupation rate of only 41%. The average access time was 112ns, showing a significant gain in relation to the simple and intermediary simulations (obtaining a reduction of 83% in relation to basic simulation and of 39% in relation to intermediary simulation). That time can be explained due to the low number of accesses (just caches of the compulsory miss type, totaling only 13 accesses) that took place in the main memory, possessing the longer access time.

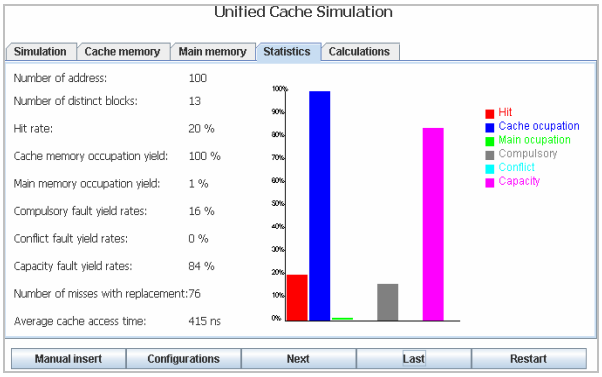

Figure 8. Statistics and graphic of the basic simulation.

With the objective to reduction the faults of the capacity type of cache L2 (Instruction), the number of slots was modified from 4 to 8 and kept all the other parameters of the hierarchy, resulting in an average access time was 105 ns and the hit rate this cache

increased for 75%. This configuration allowed an improvement of 6% in the average access time.

Another possibility of reduction of the average access time is the elimination of the conflicts misses occurred in the second level cache, for this was modified the cache type, from direct mapped to fully associative. This was possible to obtain an average access time of 103 ns, a hit rate of cache L3 (second level) increased of 48 for 76% and occupation increased 69% for 81%.

With both alterations was obtained an average access time of 99 ns. The comparison of the original designs with the three modifications can be visualized in figure 9.

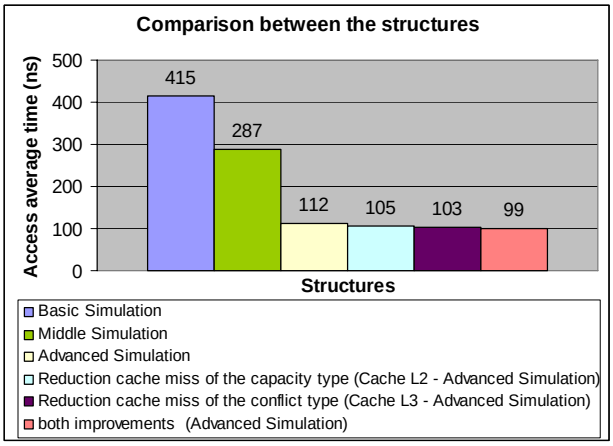

Figure 9. Comparison between the different structures.

All the mentioned metrics can be viewed at any time during the simulation process at the statistics screen (Figure 8), enabling an easy comparison amongst the structures.

As another advanced simulation example is also possible to simulate one complete hierarchy, with cache memories (unified or separate and multiple levels), main (unified or separate), virtual, TLB and page table.

Changing the memory trace and/or some parameters of the simulations as replacement policy, block size, associative degree, hierarchy levels and others, it is possible to obtain an improvement or a worsen in the performance of the simulated hierarchies.

Comparing the Web-MHE and HASE Dinero environments, can be verified that the first includes all of the functionalities presented in the second. The animation found in HASE is more detailed than an available in the Web-MHE, because the animation is the main objective of the reference. The simulator Web-MHSim possesses more features than HASE Dinero's simulator, because the last accomplishes just some simulation types (only allowing two cache levels and split) and it possesses just some configuration types, besides not being a web simulator.

Comparisons were accomplished using the CacheSim simulator, for being the only Web tool of cache simulation found in the state of the art, Dinero IV, for being known thoroughly in the academic field, VirtualMemory, for supporting simulation of virtual memory and MSCSim, that it was used with the purpose of verifying the evolution of the tool (Table 1).

Table 1. Comparison between simulators

Simulator / Features	Web-MHSim	MSCSim	Cache Sim	Dinero IV	Virtual Memory
Simulation of Cache	Yes	Yes	Yes	Yes	No
Simulation of Split Cache	Yes	Yes	Yes	Yes	No
Simulation of Multilevel	Yes	Yes	No	Yes	No
Simulation of Virtual	Yes	Yes	No	No	Yes
Simulation of Cache, Main and Virtual	Yes	No	No	No	No
Harvard architecture	Yes	No	No	No	No
Access Time	Yes	Yes	No	No	No
Write Policy	Yes	Yes	Yes	Yes	No
Multiple input formats	No	No	No	Yes	No
Memory Trace generator	Yes	Yes	No	No	No
Statistics graphics	Yes	No	No	No	No
Didactic interface	Yes	Yes	Yes	No	Yes
Step by step simulation	Yes	Yes	No	No	Yes
Animation	Yes	Yes	No	No	Yes
Web simulation	Yes	No	Yes	No	Yes
User interaction with the simulation	Yes	No	No	No	No

For including some knowledge levels the simulator was implanted at the laboratory of Computer Architecture of Computer Science in the Pontifical University Catholic of Minas Gerais. As one of the results of its use, it was observed on the part of the students, a larger easiness in the understanding of the theory taught at classroom. The tool also aided them in the accomplishment of the exercises and proof of the results. The teachers observed a better learning and a better evaluation with the use of this.

6. Conclusions

The results obtained with the Web-MHE were great, because were selected, improved and implemented the main features of the works in reference and allies to new resources, resulted in a web environment of aid to the learning and the research with new and more functionalities, details and efficiency, thus reaching the main goals. Some of the main features of the references simulators are multiple

levels simulation, split cache simulation, basic statistics and animation.

Among the main contributions is to offer an didactic web environment that possesses all the main features of the references simulators and also new resources as, the possibility to accomplish the simulation of multiple levels with support to split cache and virtual memory, separate main memory, calculations guide, group of didactic materials and possibility of the user's interaction, facilitating the learning and the fixation of the theory, statistics and advanced graphics and memory trace generator.

The improved and added resources contribute to facilitate the learning and the research of memory hierarchy, split cache and multiple levels and to enlarge the critical and analytical thought on the results.

As future expansion will be developed a module of on-line tutor that it will be capable to accompany the level of the student's learning and to inform him which topic is with more difficulty, and also to suggest some student that is well in this topic to aid him. The Web-MHE it will support new resources as, micro architecture, ISA level and multiple input formats. A framework will be developed to facilitate the implementation of new resources for other users and/or developer. Another future resource will be the animation integrated with the simulation.

7. References

[1] Coutinho, L. M. N.; Mendes, J. L. D.; Martins, C. A. P. S. “MSCSim – Multilevel and Split Cache Simulator”, VI Workshop em Sistemas Computacionais de Alto Desempenho (WSCAD), pp. 193-196, 2005 (in portuguese).

[2] J. L. D. Mendes, L. M. N. Coutinho, C. A. P. S. Martins, Web-MHE - Web Memory Hierarchy Learning and Research Environment, 2006, home page http://web-mhe.mscsim.com/.

[3] MSCSim Home Page, http://www.mscsim.com.

[4] Djordjevic, J.; Nikolic, B.; Mitrovic, M. “A Memory System for Education”, The Computer Journal, Vol. 48, No. 6, pp. 630-641, 2005.

[5] Yurcik, W.; Wolffe, G. S.; Holliday, M. A; “A Survey of Simulators Used in Computer Organization/Architecture Courses”, Summer Computer Simulation Conference (SCSC), Society for Computer Simulation (SCS), 2001.

[6] Djordjevic J., Milenkovic A. and Prodanovic S., “A Hierarchical Memory System Environment”, WorkShop On Computer Architecture Education (WCAE), 25th Int Symposium On Computer Architecture, Barcelona, Spain, 1998.

[7] Grünbacher H., “Teaching Computer Architecture/Organisation using simulators”, 28th Frontiers in Education Conference (FIE), pp. 1107-1112, 1998.

[8] Djordjevic, J.; Milenkovic, A.; Grbanovic. N.; "An Integrated Environment for Teaching Computer Architecture" IEEE Micro, Vol. 20, No. 3, pp. 66-74, 2000.

[9] Yurcik, W.; Wolffe, G. S.; Holliday, Mark A.; Osborne, Hugh. “Teaching Computer Organization/Architecture With Limited”, ACM SIGCSE Bulletin, Vol. 34, No. 1, pp. 176-180, 2002.

[10] Hill, M. D., Dinero IV, Home Page, http://www.cs.wisc.edu/~markhill/DineroIV/.

[11] Cache Simulator Versão 1.9, 2003, Home Page http://www.ece.gatech.edu/research/labs/reveng/cachesim/index.html.

[12] Tran, N.; Menasce, D. A.; VirtualMemory, Home Page http://cs.gmu.edu/cne/workbenches/vmsim/vm.html.

[13] Shivakumar, P.; Jouppi, N. J.; Cacti 3.0, “An Integrated Cache Timing, Power, and Area Model”. Technical Report, Compaq Computer Corporation, 2001.

[14] Schintke, F.; Simon, J.; Reinefeld, A.; LDA-Simulator, 2001, Home Page http://www.zib.de/schintke/ldasim/index.en.html.

[15] Prima Cache Simulator, Home Page www.dsi.unimo.it/staff/st36/imagelab/prima.html.

[16] R. N. Ibbett, HASE DLX Simulation Model. IEEE Micro, Vol. 20, No. 3, pp. 38-47, 2000.

[17] Hennessy, J. L.; Patterson, D. A. “Computer Organization and Design: The Hardware/Software Interface”, third edition, Morgan Kaufman, 2004.

[18] Hennessy, J. L.; Patterson, D. A. “Computer Architecture: A Quantitative Approach”, third edition, Morgan Kaufman, 2002.

[19] Smith, A. J.; “Cache memories”, ACM Computing Surveys, Vol. 14, No. 3, pp. 473-530, 1982.

[20] Denning, P. J.; “Virtual Memory”, ACM Computing Surveys, Vol 2, No. 3, 1970.

[21] Denning, P. J.; “Virtual Memory”, ACM Computing Surveys, Vol 28, No. 1, 1996.