Evaluation of CPU architecture by simulation technologies and benchmark computer systems

Madiha Yousaf[a], Muhammad Haris[b]
[ab]COMSATS Institute of Information Technology Park Road, Tarlai Kalan,
Islamabad-45550, Pakistan
m.haris@comsats.edu.pk

Abstract— The processor architecture designers face major challenges to improve the processor's performance. To measure the performance of the processor there are many parameter like performance of cache, TLB, IO operations, bus speed etc. Different companies launch series of processors with same base configuration and a little change of variations base on cache size, cache levels, share and separate cache and many other parameters like that. There are many simulators available to measure processor's performance theoretically with several parameters. We simulate some models of Intel Pentium 4 and UltraSPARC II processors and analyze the performance of the processors regarding cache and TLB.

Keywords: ISTC (Instruction Simulator Translation Caches), DSTC (Data Simulator Translation Caches)

———————————— ◆ ————————————

1 INTRODUCTION

VLSI design and developer companies like Intel, SUN, AMD and IBM are also working in the field of processor-design. Before moving on we must first discuss that what is processor design and the factors on which it depends. Designing a computer processor in such a way that cycle time, reliability, performance, chip area and many other issues are resolved. The basic task of the processor designer is to optimize all the issues [1]. The companies are designing the processor to achieve the same goal i.e. high performance, low power consumption and less heat generation [2]. In broad-spectrum, design of computer and processors consists of many steps which include accepting applications and workloads that the system will be successively managing as well as innovating probable design along with measuring performance of the design and selecting the best one [3].These are the basic parameters to design the valuable processor.

1.1. Commonly used Technologies

Time by time companies introduce new technologies and architectures for improving their processors performance. Some of these technologies include supper pipelining, super scaling, MMX and Multithreading. Before moving ahead let's have a brief idea about all above mentioned technologies.

1.2. Super Pipelining

Overlapping of many instructions during exection time is called pipelining. These overlapped tasks are executed parallel by different hardware units at the same time. The execution of these tasks is super pipe line. [4]

1.3. Multithreading

Multithreading is the capability of a program or an Operating System process to deal with its use by more than one user at a time and to smoothly manage numerous requests by the same user devoid of having to have multiple copies of the programming running in the computer. [5]

1.4. MMX

It stands for multimedia extensions and was developed by the Intel to increase the speed of multi media operations. [6]

In addition to above mentioned technologies, the architectures like P4 Netburst were very helpful to improve processor performance in the past. Companies used these technologies individually or with other technologies for improve processor performance. But these technologies have some constraints to improve processor performance that is why some new technologies like EPIC [7] and architectures like Multi-Core Architecture introduced. EPIC and Multi-core architecture based on previous technologies like supper pipelining, VLIW processor technologies.

2. Literature Review

The growth of microprocessors, for the most part allowed by new technology progress has shown the way to multifaceted designs that coalesce multiple physical processing units in a single chip. These designs endow with the operating system (OS) the vision of having numerous processors, so that diverse software processes can be planned at the same time. [8] with this sudden change in the physical processors and there working we also need to set high bench marks for their evaluation and working. One of the mostly used bench marks is SPEC as conventional benchmarks have futile to portray the system performance Some of them evaluate component-level performance while others are for system performance. [9]

Multi-core architecture are based on the idea of super scalar architecture with more than one levels of cache. This cache may separate or share among different cores on same chip. To improve the performance of processor with reference to data, processor affinity [10] technique is provided in the multi-core processor. Data tilling [10] technique is also available for improving the cache performance in multi-core processor. This technique improves the processor's performance in the situation where parallel execution is required like mathematical calculations and transaction in multi tire and distributed databases, for network security based on multi-core architecture [11]. To take the benefit from multi-core architecture, companies are designing their products which support multi-core architecture like Compiere Inc introduced ERP solution and CRM solution based on multi-core architecture.

The most frequent loom to guesstimate the performance of a superscalar processor is through edifying a software model and simulating the execution of a set of benchmarks. [7] Regarding the performance measurement of these newly designed processors according to the bench marks available we have many ways by which the performance has been measured in previous years. These methods include Off-chip hardware monitoring, Microprocessor on-chip performance monitoring counters, Micro coded instrumentation and Software monitoring. [12]

EPIC is newly introduced technique in processor design. This technique is based on CISC and RISC architecture and introduce jointly by HP and Intel in 1997. Intel introduces Itanium processor based on this architecture. EPIC technique introduced Plan of Execution (POE) like VLIW processor architecture for instruction execution. EPIC processor reduces several issues like interruption, reparability, branch instruction handling [10]. Order of execution is one of the issues to improve fast instruction execution and EPIC deal with by reducing stall cycles. EPIC has also ability to improve branch prediction. Although EPIC technology has become useful for companies in processor design but some serious problems still need to be solve. The major problems still face in EPIC technology is size compatibility of problem and bundle of instruction. In EPIC instruction bundle size is constant so if the number of instructions of a problem is greater than the bundle size then there is need more than one bundle to solve this problem and if the problem size is smaller than the bundle size the remaining part of bundle will be empty [13].

In this paper our goal is find better processor on the base of cache and TLB performance. For this purpose, we chose some models of Intel Pentium 4 Series and UltraSPARC II processors. We chose Virtutech Simics-3.0.31 [14] for analyzing the processor's performance [15].

3. Methodology:

Quantitative assessment of computer architectures depends profoundly on simulators and simulator infrastructure. [16] Simulators are an enormously important tool for computer architects [17]. They diminish the cost and time of a project by allowing the architect to rapidly evaluate different processor implementations without having to manufacture a chip every time. In addition, a simulator allows the architect to effortlessly determine the performance improvement of a new compiler based or micro architectural mechanism. [18][19]

For this paper, we used academic license of Virtutech Simics for our research. Virtutech Simics provides a simulator which use to simulate full system for single or multi core processors [20]. A system is environment which uses to solve a problem or more of same type. This system may contain one or more nodes and each node contain one or more CPUs. Simics facilitate user to re-configure different parts of system like processor clock speed, cache of different levels and size, I/O etc. Simics is also useful to simulate networks with different protocols.

To simulate any system, we need operating system which must be installed on simulator. For this purpose, some dump operating system images are available on Simics website to simulate the system. These dump image of several versions of Red Hat Linux, Fedora and Suse are using to simulate systems with user's own configuration.

For this purpose, user download the operating system image for specific target machine base on processor's family like for Intel Itanium, UltraSPARC II or any other. Copy the operating system image in image folder of target machine's folder of Virtutech Simics.

User is able to simulate system in three different modes i.e. normal, stall and micro architecture. Micro architecture uses when user wants to observe the processor's performance on micro architecture level. In Start Simics load in normal mode and before a specific operation or run operating system image user can change its mode on micro architecture or Stall.

Stall mode is use to find processor performance regarding time. We use stall mode to simulate processor and analyze cache and TLB performance and for this purpose a session is created. This session based on the target machine and operating system.

Any code in C language or the language which support by dump operating system image can use as a benchmark. This code store in host machine usually and we launch target machine with dump image of operating system and mount the host machine. Benchmark is placed at host machine and executes it on target machine. If benchmark is in the form of C code or any other language, we compile this code on target machine running on Simi's and create application file. If application file is already available user can run it as benchmark.

4. Working Environment

We developed Simics command files called "start. simics" to enable magic breakpoint, disable Instruction Simulator Translation Caches (I-STC), Data Simulator Translation Caches (D-STC), flush out the old data of I-STC, D-STC and define the TLB type.

We create Simics command files for cache configuration also. To simulate the system with reference to cache and analyze the performance of cache Simics provide g-cache class. By using this class, we create L1 and L2 cache according to our configuration. In these files, first we define the name of L1 and L2 cache and their working mode. We define independently values for L1 instruction cache and data cache. we define what CPU attach with newly created cache i.e. for multi-core processors but required also for single core processor, stall time, cache line size, number of lines, virtual index, virtual tag, write back, write allocation, configuration replacement policy, read next penalty, write next penalty, timing model, read miss penalty and write miss penalty. When we have configured instruction and data cache for L1 disjointedly, we join them. We set all above said values for L2 cache also but we do not set these values individually for instruction and data cache for L2. We also configure cache with memory of processor.

For writing the code in Simics and reset built-in variables, we create python file in Virtutech Simics. We create the python file to reset the value of Instruction replace, Data Replace for small page of TLB and Data miss and Instruction miss for combined statistics of TLB. We create another python file for display the results of instruction replace, Data replace for small page of TLB and Data miss and Instruction miss for combined statistics of TLB.

Simcis make available many operating system dump images for Pentium 4 processors. We use enterprise version of dump image for Pentiume4 Processor. We use g-cache built-in class to achieve our goal. Simics provide 20 MHz as default value of CPU clock speed for Pentium 4 processor in enterprise version of dump image. We set clock speed of Pentium 4 processor in Simics command file enterprise-gcache-common before run the image of operating system according to our requirement.

Virtutech Simcis support UltraSparc II processor for some specific frequencies for simulation [21]. So, we pick only those processors of UltraSPARC II series which support by these frequencies. Virtutech Simics provide default clock speed of 168MHz for UltraSPARC II processor. We set the clock speed in built-in Simics command file cashew-gecache-comman as we done with Pentium 4 processor and run operating system image.

UltraSPARC II uses stall mode both data access and instruction fetches and Pentium 4 processors can stall only for data access.

5. Analysis of Processor Performance

We analyze processor's performance for L1 data cache, L2 cache and TLB. We chose some models of Intel Pentium 4 and SUN UltraSPARC II series of processors and simulate these processors on Virtutech Simics. We chose Gemm as benchmark for study the performance of these processors. When we reach on magic breakpoint i.e. define in benchmark file and reset the L1 data cache, L2 cache and TLB statistics. After this we run 100,000,000 instructions and observe the performance of the processor.

5.1. Analysis of Pentium 4 processor

We simulate four Intel Pentium 4 series of processors i.e. Intel Pentium 4 3.4C, Intel Pentium 4 519K, Intel Pentium 4 631 and Intel Pentium 4 2.0GHz.

We simulate these processors with L2 cache line size of 128. Pentium 4 processor has L2 8-way cache associative. We analyze the L2 cache of processors with read penalty of 10 cycle, write penalty of 10 cycles, virtual Index is 1, virtual Tag is 1, write back is 1 , write allocation use 1 cycle, Replacement policy is LRU, read next use 0 cycles, write next use 0 cycles, Timing model is staller, and stall time is 200.

The configuration of these processors with L1 cache of line size 64 byte Pentium 4 processor has L1 2-way cache associative. Other configurations are read penalty of 0 cycle, write penalty of 3 cycles, virtual Index is 1, virtual Tag is 1, write back is 1 , write allocation use 1 cycle, Replacement policy is LRU, read next use 0 cycles, write next use 0 cycles, Timing model depends on L2 Cache, and stall time is 200. Specifications of these processors are as follows.

5.2. Intel Pentium 4 3.4C

Intel Pentium 4 3.4C processor with code name North Wood. Its clock speed is 3400 MHz with 8 K L1 cache and 512 K L2 cache. FSB is 200 MHZ, processor size 130 nm, transistor count is 125M, Number of cores 1, Multiplier is 17x, voltage 1.5 V, TDP is 110 W, SMP number CPUs are 1, Die size is 146 mm^2 and its features are MMX, SSE, SSE2, HT [6].

5.3. Intel Pentium 4 519K

Intel Pentium 4 519K processor with code name of Prescott. Its clock speed is 3060 MHz, L1 cache is 16 K and L2 cache is 1MB. FSB is 133 MHZ, processor size 90 nm, transistor count is 125M, Number of cores 1, Multiplier is 23, voltage 1.4 V, TDP is 84 W, SMP number CPUs are 1, Die size is 109 mm^2 and its features are MMX, SSE, SSE2, SSE3, NX-Bit, EM64T, C1E. [22]

5.4. Intel Pentium 4 631

Intel Pentium 4 631processor with code name Cedar Mills. Its clock speed is 3000 MHz with 28 K L1 cache and 2MB L2 Cache. FSB is 200 MHZ, processor size 65 nm, transistor count is 188M, Number of cores 1, Multiplier is 15x, voltage 1.325 V, TDP is 86 W, SMP number CPUs are 1, Die size is 81 mm^2 and its features are MMX, SSE, SEE2, SSE3, HT and EM64T. [22]

5.5. Intel Pentium 4 2.0GHz

Intel Pentium 4 2.0GHz processor with code name of Willamette. Its clock speed is 2000 MHz with 8k L1 cache and 256k L2 cache. FSB is 100 MHZ, processor size 180 nm, transistor count is 42M, Number of cores 1, Multiplier is 20x, voltage 1.7 V, TDP is 52 W, SMP number CPUs are 1, Die size is 217 mm^2 and its features are MMX, SSE, SEE2. [22]

5.6. Analysis of SPARC processors

We simulate four different versions of UltraSPARC II processors with the benchmark. These processors are UltraSPARC II STP 1031, UltraSPARC II STP 1032, UltraSPARC IIi SME 1430 and UltraSPARC IIe SME 1701. For these processors, we analyze the performance of L1 data cache and L2 cache. The cache lines size for L1 is 64 and for L2 cache is 128. L2 cache has 8-way associative and L1 has 2-way cache associative. We define processor read and write miss penalty 10 cycles.

5.6.1. UltraSPARC II STP 1031

UltraSPARC II STP 1031 with code name is Black Bird and It's clock speed is 336 MHz, Board Frequency is 84MHz Clock Multiplier 4.0, Data Bus (ext) 64 Bit, Address Bus 64 Bit, Transistors 5.4 M, Circuit Size 0.3 micro, voltage is 2.5, Die Size is 265 mm^2, L1 cache is 16 k and L2 cache is 4MB [22].

5.6.2. UltraSPARC II STP 1032

UltraSPARC II STP 1032 with code name is Sapphire Black and It's clock speed is 400 MHz, Board Frequency is 100MHz Clock Multiplier 4.0, Data Bus (ext) 64 Bit, Address Bus 64 Bit, Transistors 5.4 M, Circuit Size 0.25 micro, voltage is 1.9, Die Size is 126 mm^2, L1 cache is 16 k and L2 cache is 8MB [22].

5.6.3. UltraSPARC IIi SME 1430

UltraSPARC IIi SME 1430 with code name is Sapphire Red and It's clock speed is 360 MHz, Board Frequency is 120MHz Clock Multiplier 3.0, Data Bus (ext) 64 Bit, Address Bus 64 Bit, Transistors 5.4 M, Circuit Size 0.25 micro, voltage is 1.7, Die Size is 126 mm^2, L1 cache is 16 k and L2 cache is 8MB [22].

5.6.4. UltraSPARC IIe SME 1701

UltraSPARC IIi SME 1701 with code name is Humingbird and Its clock speed is 400 MHz, Board Frequency is 66MHz Clock Multiplier 6.0, Data Bus (ext) 64 Bit, Address Bus 64 Bit, Circuit Size 0.18 micro, voltage is 1.7, Die Size is 126 mm^2, L1 cache is 16 k and L2 cache is 256KB [22]

6. Result for Pentium 4 Processor

We executed the benchmark and got the results for series of Pentium 4 processors. We use gemm.c as benchmark to analyze the processor with reference to L1 Data cache, L2 cache, and TLB small page and combined TLB statistics. We use enterprise3-rh3.craff as disk dump of Pentium 4 processors. This disk dump contains RedHat 7.3 Linux with Linux kernel 2.4.18. it also includes SMP support.

6.1. L1 Data Cache

We analyze L1 data cache for Pentium 4 for Intel Pentium 4 3.4C, Intel Pentium 4 519K, Intel Pentium 4 631 and Intel Pentium 4 2.0GHz processor and find the result for Read Hit Ratio and Write Hit Ratio as shown in table 1. We find that there is minor difference in results in Read and Write Hit Ratio.

Processor	DC Read Hit Ratio	DC Write Hit Ratio
Intel Pentium 4 3.4C	93.52	99.77
Intel Pentium 4 519K	93.55	99.80
Intel Pentium 4 631	93.54	99.80
Intel Pentium 4 2.0GHz	93.52	99.79

Table 1. L1 Data Cache for Pentium 4

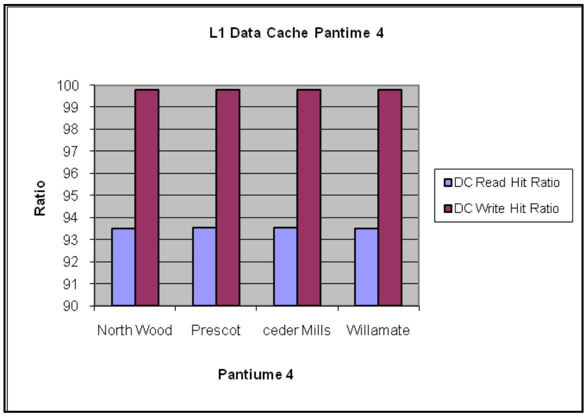

Figure 1. Graphical result of L1 Data Cache

6.2. L2 Cache for Pentium 4

We analyze the performance of these processors also for L2 cache for Read and Write Hit Ratio. We find that write Hit ratio is same for all processors and Read hit ratio is also have very little variation but Intel Pentium 4 631 processor has very high Read Hit Ratio. This is because of L2 cache size of which is 2 MB.

Processor	L2 Read Hit Ratio	L2 Write Hit Ratio
Intel Pentium 4 3.4C	6.59	99.9
Intel Pentium 4 519K	6.26	99.90
Intel Pentium 4 631	95.06	99.90
Intel Pentium 4 2.0GHz	6.51	99.90

Table 2. L2 Cache for Pentium 4

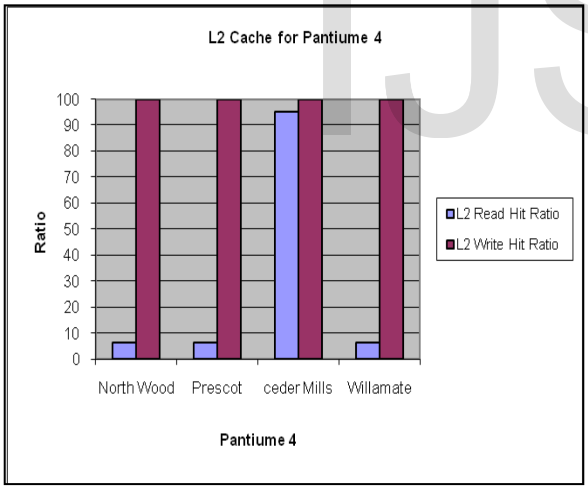

Figure 2. Graphical result of L2 Cache

6.3. TLB for Small Page for Pentium 4

Statistics of TLB for small page size is given in table 3. Only Intel Pentium 4 3.4C processor replaces nearly double data as compare to other processors and replaces six instructions. That shows the low performance of this processor with reference to data and instruction replace in small page.

Processor	Data Replace	Instruction Replace
Intel Pentium 4 3.4C	9389602	6
Intel Pentium 4 519K	4696742	0.00
Intel Pentium 4 631	4697592	0.00
Intel Pentium 4 2.0GHz	4696881	0.00

Table 3. TLB Small Page for Pentium 4

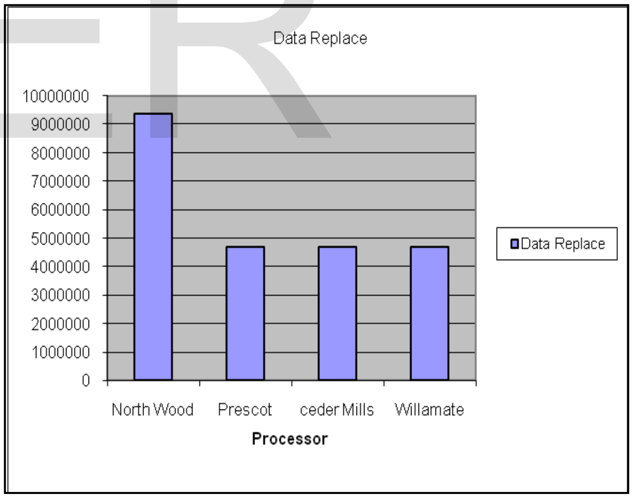

Figure 3. Graphical result of TLB Small Page

6.4. TLB Combined Statistics for Pentium 4

If we see overall statistics of TLB we see only Intel Pentium 4 3.4C processor is only face data and instruction. This shows for TLB this processor has very poor performance.

Processor	Data Miss	Instruction Miss
Intel Pentium 4 3.4C	**9535560**	**260**
Intel Pentium 4 519K	**4769626**	**0.00**
Intel Pentium 4 631	**4770488**	**0.00**
Intel Pentium 4 2.0GHz	**4769766**	**0.00**

Table 4. TLB Combined Statistics for Pentium

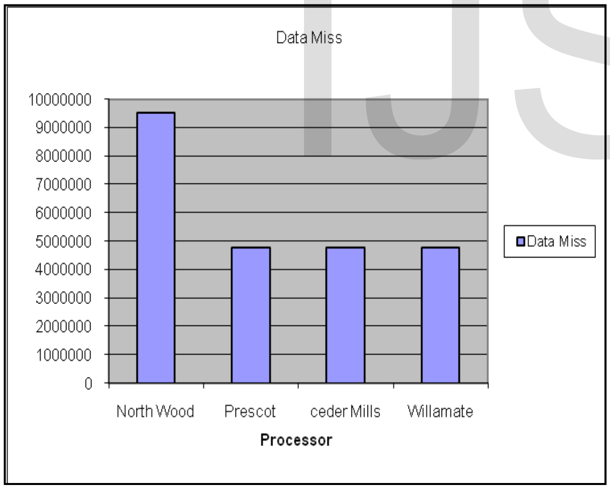

Figure 4. Graphical result of TLB Combined Statistics

7. Result for UltraSPARC II processors

We use dump image of Aurora 2.0 (Fedora Core 3) Linux as target machine for Ultra SPARC II processors. The image name is cashew1-aurora2.0. craff. Here Linux Kernel is 2.6.15 with SMP support. We run the benchmark on target machine with different configurations of Ultra SPARC II processors and analyze their performance with regard to L1 Data cache and L2 cache.

7.1. L1 Data Cache for UltraSPARC II

Now we analyze L1 Data Cache for UltraSPARC II processor. There is no big change observe in L1 Data Cache Read Hit Ratio and Write Hit Ratio. Because all of the UltraSPARC II processors have same L1 cache size.

Processor	DC Read Hit Ratio	DC Write Hit Ratio
UltraSPARC II STP 1031	**94.57**	**99.84**
UltraSPARC II STP 1032	**94.57**	**99.75**
UltraSPARC IIi SME 1430	**94.58**	**99.85**
UltraSPARC IIe SME 1701	**94.58**	**99.74**

Table 5. L1 Data Cache for UltraSPARC II

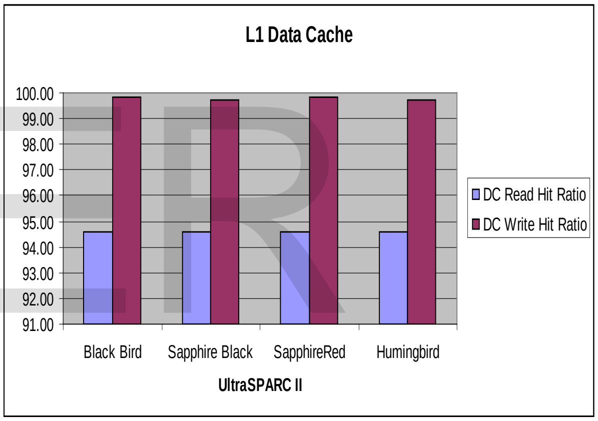

Figure 5. Graphical result of L1 Data Cache

7.2. L2 Cache for UltraSPARC II

We simulate L2 cache and observed there is no big change in Read and Write Hit Ratio in three out of four processors. UltraSPARC II STP 1031 processor of 4 MB size of L2 cache i.e. one of these three processors which have same Write Hit Ratio but there is minor difference in Read Hit Ratio as compare to other two processors. Big change observed in UltraSPARC IIe SME 1701 because of its L2 cache size i.e. 256 KB.

Processor	Read Hit Ratio	Write Hit Ratio
UltraSPARC II STP 1031	99.80	100.00
UltraSPARC II STP 1032	99.98	100.00
UltraSPARC IIi SME 1430	99.98	100.00
UltraSPARC IIe SME 1701	0.35	99.93

Table 6. L2 Cache for UltraSPARC II

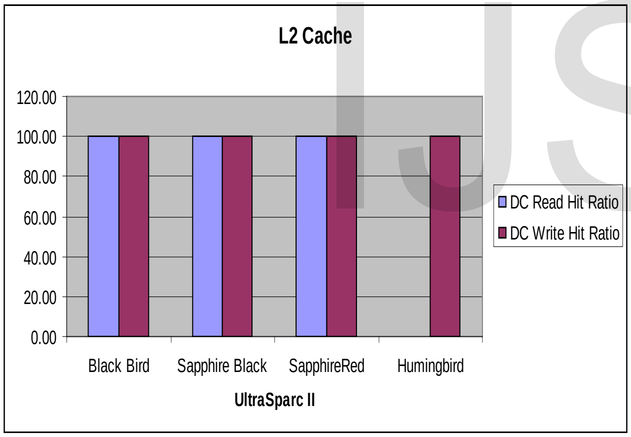

Figure 6. Graphical result of L2 Cache

8. Conclusion and Future Work

We simulate the series of Intel Pentium 4 regarding L1 data cache, L2 cache and TLB and UltraSPARC II processors regarding L1 data cache and L2 cache. We analyzed processor performance is based on cache size. These caches are interdependent for good performance. We also find that there is a big difference in TLB statistic of same series of processors. There is still a big room for work in simulation and analysis the performance of processor. In future we can work to find out the performance of processors regarding execution time, I/O and processor performance on a specific temperature. We simulate the processors for one benchmark but for affirmation of the results we can use other available benchmarks for cache and TLB performance analysis.

REFERENCES

[1] Tuo Li, Jude Angelo Ambrose, Roshan Ragel, Sri Parameswaran "Processor Design for Soft Errors: Challenges and State of the Art" Journal ACM Computing Surveys (CSUR) Article No. 57 Volume 49 Issue 3, November 2016

[2] William Lloyd Bircher, Lizy K. John "Complete System Power Estimation Using Processor Performance Events" IEEE Transactions on Computers pp. 563 - 577 ,Volume: 61, Issue: 4, April 2012

[3] Aldeida Aleti, Barbora Buhnova, Lars Grunske, Anne Koziolek, Indika Meedeniya "Software Architecture Optimization Methods: A Systematic Literature Review" IEEE Transactions on Software Engineering pp. 658 – 683 Volume: 39, Issue: 5, May 2013

[4] Dongsuk Jeon, Mingoo Seok, Chaitali Chakrabarti, David Blaauw, Dennis Sylvester "A Super-Pipelined Energy Efficient Subthreshold 240 MS/s FFT Core in 65 nm CMOS" IEEE Journal of Solid-State Circuits pp. 22-34 Volume: 47, Issue: 1, Jan. 2012

[5] Kazutoshi Suito, Rikuhei Ueda, Kei Fujii, Takuma Kogo, Hiroki Matsutani, Nobuyuki Yamasaki "The Dependable Responsive Multithreaded Processor for Distributed Real-Time Systems" IEEE Micro pp. 52-61, Volume: 32, Issue: 6, Nov.-Dec. 2012

[6] Cheong Ghil Kim, Jeom Goo Kim, Do Hyeon Lee "Optimizing image processing on multi-core CPUs with Intel parallel programming technologies" Springer Multimedia Tools and Applications pp 237–251, Volume 68, Issue 2, Jan 2014

[7] W.W.S Chu, R.G. Dimond, S. Perrott, S.P. Seng, and W .Luk, "customizable EPIC Processor: Architecture and Tools" Page 30236 Proceedings of the conference on Design, automation and test in Europe - Volume 3,Feb 2004

[8] Nicholas FitzRoy-Dale, "The VLIW and EPIC processor architectures," Master Degree Thesis, Department of Computer Science and Engineering, New South Wales University, July 2005.

[9] Didona,D., Quaglia,F., Romano,P. & Torre, E. (2015). Enhancing performance prediction robustness by combining analytical modeling and machine learning.In ICPE 2015-Proceedings of the 6th ACM/SPEC International Conference on Performance Engineering. (pp.145-156).Association for Computing Machinery, Inc. DOI: 10.1145/2668930.2688047 Nikrouz Faroughi,"Profiling of Parallel Processing Programs on Shared Memory Multiprocessors Using Simics"

[10] Hyoseung Kim, Arvind Kandhalu, Ragunathan Rajkumar "A Coordinated Approach for Practical OS-Level Cache Management in Multi-core Real-Time Systems" IEEE 25th Euromicro Conference on Real-Time Systems (ECRTS),Sep 2013

[11] Ahmed Saeed, Ali Ahmadinia,Mike Just "Secure On-Chip Communication Architecture for Reconfigurable Multi-Core Systems" World Scientific Journal of Circuits, Systems and Computers Volume 25, Issue 08, Aug 2016

[12] Alexander Wert, Henning Schulz, Christoph Heger "AIM: Adaptable Instrumentation and Monitoring for Automated Software Performance Analysis" IEEE/ACM 10th International Workshop on Automation of Software Test (AST), May 2015

[13] Chaturvedi, Nitin; Gurunarayanan, S. "Study Of Various Factors Affecting Performance Of Multi-Core Processors"International Journal of Distributed and Parallel Systems pp. 37-45 4.4, Jul 2013

[14] http://www.windriver.com/products/simics [AccessedNovember 2016]

[15] P.S. Magnusson, M. Christensson, J. Eskilson, D. Forsgren, G. Hallberg, J. Hogberg, F. Larsson, A. Moestedt, B. Werner "Simics: A full system simulation platform" IEEE Computer pp. 50-58 Volume: 35, Issue: 2, Feb 2002 https://www.simics.net/mwf/vh?31

[16] Teodor Sommestad, Mathias Ekstedt, Hannes Holm "The Cyber Security Modeling Language: A Tool for Assessing the Vulnerability of Enterprise System Architectures" IEEE Systems Journal pp. 363 - 373 , Volume: 7, Issue: 3, Sept. 2013

[17] Lourenço A. Pereira ; Edwin L.C. Mamani ; Marcos J. Santana ; Regina H.C. Santana ; Pedro Northon Nobile ; Francisco José Monaco "Non-stationary Simulation of Computer Systems and Dynamic Performance Evaluation: A Concern-Based Approach and Case Study on Cloud Computing" 27th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD),Jan 2016

[18] Daniele Bortolotti, Christian Pinto, Andrea Marongiu, Martino Ruggiero, Luca Benini "VirtualSoC: A Full-System Simulation Environment for Massively Parallel Heterogeneous System-on-Chip" IEEE 27th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW),Oct 2013

[19] Trevor E. Carlson, Wim Heirman, Lieven Eeckhout "Sampled simulation of multi-threaded applications" IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), July 2013

[20] Renato Mancuso, Roman Dudko, Emiliano Betti,Marco Cesati, Marco Caccamo, Rodolfo Pellizzoni "Real-time cache management framework for multi-core architectures" IEEE 19th Real-Time and Embedded Technology and Applications Symposium (RTAS), June 2013

[21] Maximilien Breughe, Stijn Eyerman, Lieven Eeckhout "A mechanistic performance model for superscalar in-order processors" IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS),April 2012

[22] Collection of processors http://www.cpu-collection.de [Accessed November 2016]