

Modeling For Multicore System Simulator for Computer Architecture

Mr. Shirish Pattalwar¹, Dr. Vilas Thakare²

¹Research Scholar, SGB Amravati University, Amravati, Maharashtra, India

²Professor & Head, Department of Computer Science, SGB Amravati University, Amravati, Maharashtra, India

ABSTRACT

This research discusses the various issues regarding the accurate and fast and automated system architecture which gives specific information about the various parameters and there effects on the simulation of the structure for the efficient processing of the system modeling. As there is a great demand of the simulation of the system architecture this research gives the better idea for the simulation and various components involved and how the process is followed superior quality of design and development components regarding the efficient utilization of the multicore processor. This research also discusses the various components like basic structure for simulation and for the efficient operation of the system using the various components and parameters which are closely related to each other. The detail analysis of these parameters is also done which are so intensely attached to each other that they may affect each other.

Keywords - Multi-core x86 CPU simulator; Emulator; Full- System simulator; Heterogeneous Multi-core systems; Processor Modeling.

I. INTRODUCTION

Now a days there is a great demand of the high end, fast and versatile devices which involves the high end processor and which also leads to different kinds of applications such as hard real time and soft real time. Any processor system before being implemented practically needs much iteration of up-gradations through simulation. The hard real time processors are those in which the deadline for the task assigned has to be completed within the specific and accurate timing constraints. Hence there is a great requirement for the high end processor and the cost of such processor design is very high. As it involves the number of critical issues which includes the

design, development and implementation of such high end processors.

Hence before the actually implementing the hardware in to the hard core processor the various parameters regarding the processor must be studied and analyzed for the proper operation of the system and the overall functionality of the system architecture must be understood for the accurate functionality. Thus there is a great demand of such design which will fulfill all the design requirements which are in continuous demand for the application like military applications.

Another important aspect in the design and development of such system architecture involves the proper memory management. As all the data or information on which the processor is going to

process is present in the memory of the system. In other words we can say that there is a great demand of memory which can be less in cost and saves the most of the necessary information in the processor chip. As by default all the data is stored in the secondary memory of the system. It has to be brought in the main memory or the primary memory for the operation which is then must transferred to the small memory in the processor chip. This small memory has to be on the processor chip. If the data or information which is to be processed by the processor is present in the small memory present on the processor chip then the operation the processor can perform is efficient, fast, accurate and without any delay, otherwise the time required to get the data into the processor is involved in the operation of the processor is going to perform.

Therefore the proper selection of the memory requirement has to be judged accordingly which will satisfy the needs of the application where the processor is to be implemented. The processor has to deal with the various resources and input output devices and needs to be interface with the other integrated devices. The resources may involve the chipset, Dynamic RAM, network interface cards and peripheral devices in addition to this there are number of devices the processor has to deal with.

Thus there is a great requirement for the accurate simulation models which can fulfill the system modeling for the high end processing also with respect to the kind of application involved in the designing phase. The use of single and multicore processors are frequently done in the design, development and implementation of the system architecture which can be very useful for the user for the performing the various operation regarding following the strict deadline of the task which is assigned to the processor.

There is another parameter which plays a very important role in the design, development and implementation. It is the cost of manufacturing. The

designer or the architect of the system must always consider this factor as a high priority which is very essential.

In the development of the system architecture the most essential and important dimension is the use of the proper simulation tool. If the simulation tool is easy to use the designer can develop and use the tool very efficiently and make the system faster and efficient. The simulation tool is very essential for the systematic development and implementation of the parameter which can successfully follow all the steps for the proper operation and systemic mechanism of the development of the processor modeling. Thus the simulation model or simulation tool has to be realistic which involves all the parameters for the proper development of the system architecture.

Such a tool is useful for evaluating and developing products that will use current and emerging single and multicore x86 chips.

This research introduces another dimension in the development of the system architecture which is very useful for the number of application involving the hard real time requirements and soft real time requirements[1].

The other system may use different types of the simulator for the simulation of the various types of the applications involved in the production of the online codes for the IOT industry, such as there are number of system which automatically generates the codes depending upon the system input. Such types of the systems give the better performance with respect to the soft real times as there is not accurate deadline attached or needed for the application to run and hence there is more demand of such type of systems which are useful for the generation of code which are executable for the various applications in different domain.

The cost of such systems is very low as compared to the systems those require the accurate timeliness. If the application requires the accurate timeliness then,

such type of the systems also use the high end processor hence the cost involved is maximum[2].

To the best system architecture it has to go through the various types of stages for that purpose knowledge and use of proper simulation tool is very essential, System architecture is not aware of issues in the public domain, open source simulator that rivals the characteristics of different systems.

This research focuses on the various dimensions of the system implementation and their details regarding the evaluation of the performance of various issues of multi- threaded benchmarks.

II. BACKGROUND

Many research studies have been done on the various simulator designs also to get more perfect simulation results for various parameters.

The out-of-order processing for the core designed is used in various applications which may the issue of the simulation time. In other words it decreases the simulation time to a great extent hence there is a great demand of such type of the simulation tool[1] with the cost of the processing is increased significantly which may hamper the total manufacturing cost of the complete system architecture and evaluation of the performance in some of the application[7].

In order to reduce the cost of the simulation there are many types of the simulation tool which are used such as PTLsim. MPTLsim simulator which may significantly reduce the cost of the simulation and on the other hand the complete system architecture can be developed in the very less cost[3].

The paper is arranged in following manner:

Section I Introduction. Section II describes Background. After that Section III describes previous work done by various researchers in this domain. Section IV describes various existing methodologies. Section V describes analysis & discussion with base

approach of proposed methodology. Section VI is approach for testing. Section VII is proposed methodology. Section VIII is outcome and result Section IX is conclusion. Section X is future scope of this paper.

III. PREVIOUS WORK DONE

In research literature, many simulation technologies have been studied to provide various aspects of this research and improve this technology with maximum output.

The proposed system architecture is useful, integrated and accurate simulation frameworks develop for a broad range of applications[2]. This framework consists of various elements like a internal memory management unit, main CPU emulator and different IO devices for performing input and output, and chipsets operations.

This research gives a very high-level approach of various elements of the proposed model along with the different CPU simulation tools and framework[3]. This work postulates that the coarse-grained migration in existing heterogeneous processor designs may limits their effectiveness and energy savings. This system provides the Composite Cores for different aspects for the emulation of the system components associated with the heart of the computation and different features for the best system development[5][6]. Hence the system discussed can achieve both energy efficiency and high performance. In such work, the system promotes a dual nature of Composite Core using the components for better simulation.

IV. EXISTING METHODOLOGIES

Many techniques have been implemented over the last several decades on Composite Cores, an architecture that brings the concept of heterogeneity into a single core. The performance and energy

implications of several architectural designs of a Composite Core with the goal of maximizing both energy savings and physical layout have been analyzed. System proposed the addition of a small L0 filter cache [8] for the little mEngine, as well as evaluates the effects of various migration techniques. System evaluates proposed Composite Core architecture with cycle accurate full system simulations and integrated power models. Overall, a Composite Core can map an average of 25 percent of the dynamic execution to the little mEngine and reduce energy by 21 percent while bounding performance degradation to at most 5 percent.

gem5-gpu routes most memory accesses through Ruby, which is a highly configurable memory system in gem5. By doing this, it is able to simulate many system configurations, ranging from a system with coherent caches and a single virtual address space across the CPU and GPU to a system that maintains separate GPU and CPU physical address spaces[9].

Single and multicore processors implementing the x86 instruction set architecture (ISA) are deployed within many computing platforms today, starting from high-end servers to desktops and ultimately down to mobile devices (using the Intel Atom and its announced successors), including potential new products that target the smart phone market segment and beyond. The one clear advantage of using the x86 processors in the full range of the product spectrum is to facilitate the rapid deployment of the wide variety of x86 application binaries. It is thus important to have a full system simulation tool that incorporates realistic simulation models for other systems level components such as the chipset, DRAM, network interface cards and peripheral devices in addition to accurate simulation models for single and multicore processors implementing the x86 ISA[30]. Such a tool is useful for evaluating and developing products that will use current and emerging single and multicore x86 chips. MARSS uses a cycle-accurate simulation

models for out-of order and in-order single core and multicore CPUs implementing the x86 ISA[10].

The researchers studied different techniques for the better simulation and composition of various units for the better design of the system architecture and also studied and analyzed various parameters regarding the various situation concerning to the various critical situation in which the system has to work for the various application domains.

V. ANALYSIS AND DISCUSSION

The proposed methodology uses advanced simulation tools, which are very useful in judging the performance of the proposed system architecture which is a integration of various processor and various advanced computational units for the combined use of the logical, arithmetic and logical functions of the system for performing the various complicated task associated with computation. Thus the system will provide accurate simulation and analysis of the critical issues occurred in the process of the simulation for better simulation experience, modeling the decomposition is done using the Virtual Machine Framework and is extensively modified to realize some important features of the proposed system architecture. This framework is extremely useful for simulation/emulation and extensive changes are done to make the system applicable to various application domains. The proposed simulation techniques and models are really useful in multicore microprocessors with coherent memory design, Dynamic memory systems and various interconnections including on-chip interconnections and various instructions for execution.

A. Base Approaches for Methodology

There are various priorities and privileges assigned to the different type of task associated with the simulation of the processes that the system architecture is going to perform or in other wards

system is going to perform thus the situation becomes very simple that the task having higher priority or privilege will be executed first depending upon the various resources the process is going to use. There are also many instances of each resource that a system may have and must be allocated in such a way that there should be optimum use of the resources. Hence the priority of the process will help to resolve the conflict if any which is essentially useful for the system in which there are more number of processes and resources are limited. Another approach the system must allow is to use a specific amount of time required for each process and each process is allocated a specific amount of time which is called as quantum time. After completion of this time the resources are taken away from the process and are allocated to another process. This idea or approach is very useful where there is a limited amount of CPU time and that is to be distributed among the various processes associated with the system.

All above discussed scenario is simulated among the different tools for the system architecture simulation. The simulation tool which can perform best or which is of great use in judging the performance of the system architecture modeled is very essential and useful for the actual implementation and therefore, the selection of the simulation tool for the multicore system plays a very essential and important role in the performance evaluation of the system architecture and design.

The basic concept behind the selection of the proper simulation is that the simulator tool should be efficient and effective while judging the performance of the overall system architecture proposed and there should be no errors in the modeling of the system architecture. The system architecture modeled must be accurately modeled by the simulation tool used.

There are various types of simulation tool available in the market, which are particularly working or focusing on different areas of the simulation, but the designer or architect of the system must use some

strategy for the selection of the simulation tool among the various types of the tool available for the simulation depending upon the domain of the applications and the need of the applications. Thus the selection of the simulation for the multicore system has to be done very carefully to enhance the performance of the system in order to achieve the system designed goals.

Another factor which is very important while selecting the simulation tool is the simulation speed. In contrast with the performance judging capacity of the simulation tool, system designer must also concentrate on the various issues of the simulator tool like simulation speed. If the simulation speed is too slow then such simulation is of no use and leads to unnecessary delay in the simulation which the system may not tolerate. In such a case high speed system simulator are very effective and efficient.

In addition to this, the system architecture must use stack registers and allows direct access to most of such registers to improve the speed of the operation. The system must use add-on support for instructions to be executed and register accesses, allowing users to use pre-compiled applications and library functions that utilize system registers.

Communication between Simulator and applications provides a unique feature to system architecture to send and receive instruction from the system simulator and the application where the system to be implemented. This also has a modified mechanism to reflect the simulator cache configuration like memory size, memory line, and shared/private memory to be used. Thus the application enables simulator features to optimize.

VI. APPROACH FOR TESTING

The proposed system architecture uses an event-based memory management for system simulator to increase the simulation speed according to the availability of the memory in the system. Such

simulation models memory and different controllers and management protocols for the simulation, on-chip interconnections and a simple DRAM controller. The proposed system has to be verified and validated against the different characteristics of the desired simulated system which helps in the monitoring the performance of the proposed system against the different parameters which are associated with the various issues of the system simulation process. These parameters are closely linked to each other and may affect the system in different dimension which is finally essential for judging the overall performance of the system architecture.

Hence the parameters for the simulation must be carefully analyzed for their effect on the system simulation.

The system must maintain event queues for the purpose of the system simulation which is very essential for the testing the various parts of the simulation process. The event queues are modeled in such a way that they does not specify the event compilation order specified within particular simulated cycle.

This proposed system simulation model simulates non- uniform memory accesses and delay in modeling unlimited size of memory accesses within a controller. The system also gives best three models of memory: simple write-back memory, simple write-through memory, and coherent memory implemented using the specified protocol design. Figure indicates a specified flow of diagram of the logic in a very simple memory module. All the different edges between the various processes blocks may indicate quantum of delay to be simulated between various events executed.

VII. PROPOSED METHODOLOGY

Model Refinement through Compact Trace Transformations (CET) methodology is implemented on the basis of various components and related events

like coherent caches, interconnections, chipsets, memory, IO, executions and its timings. This stage of simulation is a beginning step of simulation that uses Trace driven approach combined with flow based approach and full system simulation. To begin with it first focuses on level of abstractions.

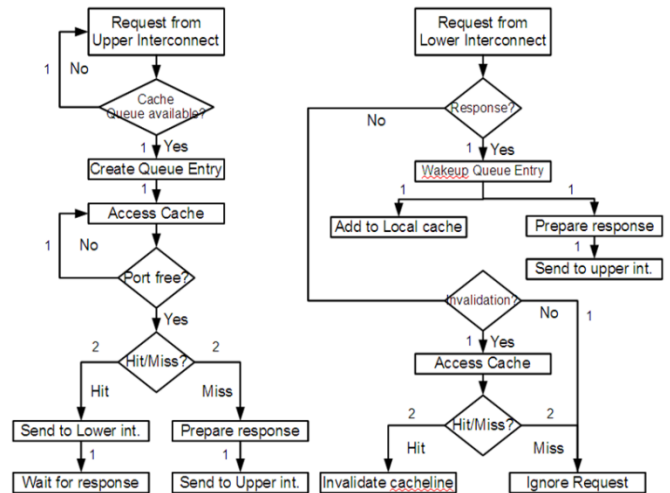


Fig. 1 Flow diagram of CRM logic with delays

A. Level of abstraction

Level of abstraction relates to every subcomponent of the system. So, Refining architecture model components in full architecture system level simulation framework requires that the application events driving these components should also be refined to match the architectural detail.

B. Trace transformations & CET

Refinement of part of system application events is denoted using trace transformations in which the left-hand side contains the coarse-grained application events that need to be refined and the right-hand side the resulting part of architecture-level events.

Furthermore, “!” symbols in trace transformations denote the “followed by” ordering relation. The R & W trace transformations are given below that refines Read and Write application events such that the synchronizations are separated from actual data transfers.

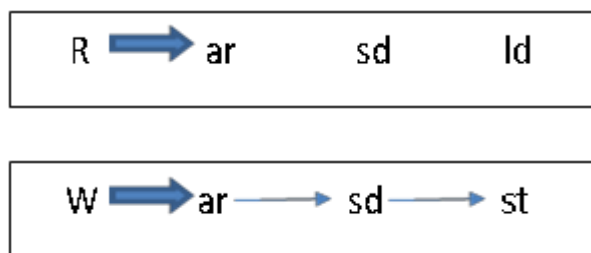


Fig. 2 Flow of shows the logic with delays in processing.

Where, events ar is activate data *
sd is signal data > ld is load data !
st is store data !

The events marked with * refer to synchronizations, while those marked with > refer to data transmissions & ! refers to followed by.

C. Event Refinement Using Dataflow Graphs

In a system, Synchronous Data Flow (SDF) and Integer controlled Data Flow (IDF) actors are deployed to realize trace transformations. As it is shown in this section, the SDF actors perform the actual event refinement while dynamic IDF actors are utilized to model repetitions and branching conditions that are present in the application code. In addition, IDF actors also be used to achieve less complicated (in terms of the number of actors and channels) dataflow graphs. It is assumed the fact that SDF is a subset of IDF.

VIII. OUTCOME and RESULTS

Model for Architecture refinement through trace transformations CET overloads:

$$CET = n \times IDFt$$

$$IDFt = SDF1t + SDF2t + SDF3t + \dots + n$$

$$SDFt = Rt + Et + Wt$$

$$Rt = tar + tsd + tld \times Bt \quad Wt = tar + tsd + tst \times Bt$$

Where, t indicates the time required for every event and actor B indicates block operations with n = no of actors

CET Overloads with $Et=0.25, n=1$:

TABLE I. SDFT & CET DETERMINATION (FOR $Et = 0.25, N = 1$)

Sr. No.	tar	tsd	tld	tst	Bt	Rt	Wt	SDFt IDFt CET
1	0.25	0.25	0.25	0.5	1.0	1.0	1.0	3.0
2	0.25	0.25	0.25	0.5	1.0	1.0	1.0	3.0
3	0.25	0.25	0.25	0.5	1.0	1.0	1.0	3.0

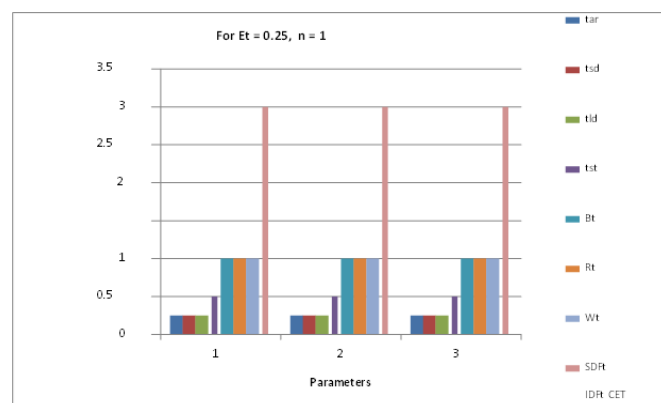


Fig. 3 Chart for Model for Architecture Refinement through Trace Transformation.

CET Overloads with $Et=0.5, n=1$:

TABLE II. SDFT & CET DETERMINATION (FOR $Et = 0.5, N = 1$)

Sr. No.	tar	tsd	tld	tst	Bt	Rt	Wt	SDFt IDFt CET
1	0.25	0.25	0.25	0.5	1.0	1.0	1.0	3.25
2	0.25	0.25	0.25	0.5	1.0	1.0	1.0	3.25
3	0.25	0.25	0.25	0.5	1.0	1.0	1.0	3.25

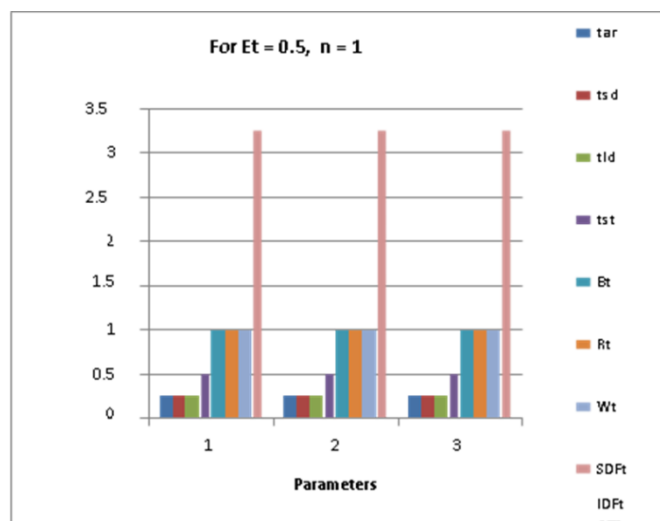


Fig. 4 Chart for Model for Architecture Refinement through Trace Transformation (CET Overloads with Et=0.5, n=1)

CET Overloads with Et=0.75, n=1:

TABLE III. SDFT & CET DETERMINATION (FOR ET = 0.75, N = 1)

Sr. No.	tar	tsd	tld	tst	Bt	Rt	Wt	SDFT	IDFT	CET
1	0.25	0.25	0.25	0.5	1.0	1.0	1.0	3.5		
2	0.25	0.25	0.25	0.5	1.0	1.0	1.0	3.5		
3	0.25	0.25	0.25	0.5	1.0	1.0	1.0	3.5		

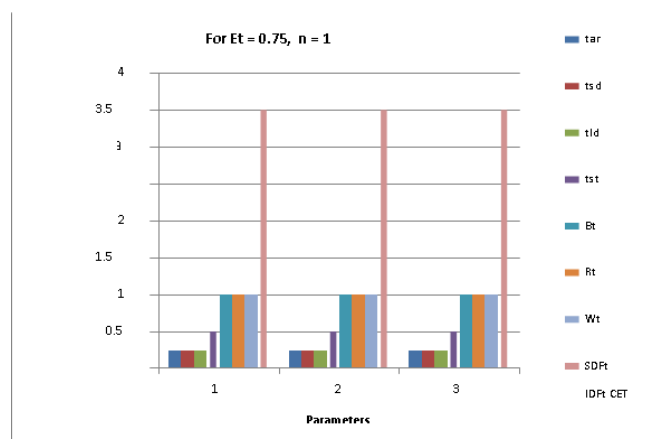


Fig. 5 Chart for Model for Architecture Refinement Through Trace Transformation (CET Overloads with Et=0.75, n=1)

IX. CONCLUSION

This research analyzed the various factors for the better simulation techniques for the best system architecture development scenario considering the various parameters such as CPU time, memory, performance, user convenience and efficiency and effectiveness with respect to different platforms specially for the multicore system development and provides the unique model for the simulation of the complete system development with high efficiency and less cost.

X. FUTURE SCOPE

It is expected that the continuous research and development will eventually result in new design of multicore and multithread processor simulator with more simulation speed and accuracy. These strategies will also improve the effectiveness and efficiency of simulation.

XI. REFERENCES

- [1]. B. Christopher, P. Vaidya, and J. L. Jaehwan, "An XML-Based ADL Framework for Automatic Generation of Multithreaded Computer Architecture Simulators", IEEE Computer Architecture Letters, Vol: 8 Issue No: 1, 13-16, January 2009.
- [2]. H. Morteza, "Flow-Based Simulation Methodology", IEEE Computer Architecture Letter Vol: 17, Issue No: 1, 51-54, January 2018.
- [3]. J. K. Archibald, "An Innovative Simulation Approach for Labs in Computer Architecture", 3rd ASEE/IEEE Frontiers in Education Conference, 19-24, November 9. 2002.
- [4]. G. Braun, A. Nohl, A. Hoffmann, O. Schliebusch, R. Leupers, and H. Meyr, "A Universal Technique for Fast and Flexible Instruction-Set Architecture Simulation", IEEE Transaction on

Computer Aided Design of Integrated Circuits and Systems, Vols.: 23 No.: 12, 1625- 1639, December 2004.

- [5]. E. Schneider and H. J. Wunderlich, "SWIFT: Switch-Level Fault Simulation on GPUs", IEEE Transaction on Computer Aided Design of Integrated Circuits and Systems, Vol. 38, No. 1, 122-135, January 2019.
- [6]. A. Zjajo, M. Eijk, R. Leuken, C. Strydis, "A Real-Time Reconfigurable Multichip Architecture for Large-Scale Biophysically Accurate Neuron Simulation", IEEE Transactions on Biomedical Circuits and Systems, Vols. 12 No.: 2, 326-337, April 2018.
- [7]. S. Lee and W. W. Ro, "Parallel GPU Architecture Simulation Framework Exploiting Architectural-Level Parallelism with Timing Error Prediction", IEEE Transactions on Computers, Vols. 65, No. 4, 1253-1265, April 2016.
- [8]. A. Lukefahr, S. Padmanabha, R. Das, F. M. Sleiman, R. G. Dreslinski, T. F. Wenisch, and S. Mahlke, "Exploring Fine-Grained Heterogeneity with Composite Cores", IEEE Transactions on Computers, Vols. 65 No. 2, 535-547, February 2016.
- [9]. J. Power, J. Hestness, M. S. Orr, M. D. Hill, and D. A. Wood, "gem5- gpu: A Heterogeneous CPU-GPU Simulator", IEEE Computer Architecture Letters, Vols. 14, No. 1, 34-36, June 2015.
- [10]. A. Patel, F. Afram, S. Chen, and K. Ghose, "MARSS: A Full System Simulator for Multicore x86 CPUs", DAC'11, June 5-10, 2011, San Diego, California, USA ,1050-1055, June 5-10, 2011.