International Journal of Electrical and Electronics Engineers
Vol. No. 9, Issue No. 01, January-June 2017
ISSN (O) 2321-2055
ISSN (P) 2321-2045

PERFORMANCE ANALYSIS OF THE DESIGNED CPU USING GEM5 SIMULATOR

Sagar Namdev Sawal

Department of Electronics and Telecommunication, Goa College of Engineering (India)

ABSTRACT

The gem5 is computer system simulator platform. This simulator provides features such as detailed and flexible memory system, including support for multiple cache coherence protocols and interconnects models. The gem5 simulator is the result of the combined effort of many academic and industrial institutions, including AMD, ARM, HP, MIPS, Princeton, MIT, and the Universities of Michigan, Texas, and Wisconsin. Gem5 simulator was created to combine best characteristics of GEMS and M5 simulator. Gem5 is more than just simulator, it is a simulation platform that lets you use as many of its premade component as you want to build your own simulation system.

Keywords: Computer architecture, Computer Simulation, GEM5, syscall emulation

I. INTRODUCTION

The computer researcher commonly uses software simulation in order to evaluate and test performance of their prototype design. With continues advancement in computer industry result in increase in complexity of design hence there is need for simulator which is capable of handling various factors and in turn provide us correct result.

Since introduction of multicore system and various cache hierarchies has resulted in improvement in performance of our CPU's. With entry of this new way of designing CPU's has introduces various new parameter's that designer has to take into consideration. The older simulators were unable to handle these new changes and thus they were replaced with newer simulators.

The gem5 simulator overcomes these limitations by providing a flexible, modular simulation system that is capable of evaluating a broad range of systems and is widely available to all researchers. This infrastructure provides flexibility by offering diverse set of CPU models, system execution modes, and memory system models. A commitment to modularity and clean interfaces allows researchers to focus on a particular aspect of the code without understanding the entire code base. The BSD based license makes the code available to all researchers without awkward legal restrictions.

This present modeling of CPU using gem5 simulator to test its performance with respect various design parameter, and brief validation .

International Journal of Electrical and Electronics Engineers
Vol. No. 9, Issue No. 01, January-June 2017
ISSN (O) 2321-2055
ISSN (P) 2321-2045

II. THE GEM5 SIMULATOR

The goal of the gem5 simulator is to be a Community tool focused on architectural modeling. Three key aspects of this goal are flexible modeling to appeal to a broad range of users, wide availability and utility to the community, and high level of developer interaction to foster collaboration.

2.1 Flexibility

Flexibility is a fundamental requirement of any successful simulation infrastructure. Different types of experiments may require different simulation capabilities.

The gem5 simulator provides a wide variety of capabilities and components which give it a lot of flexibility. These vary in multiple dimensions and cover a wide range of speed/accuracy trade offs.

The gem5 simulator currently provides four different CPU models, each of which lie at a unique point in the speed-vs.-accuracy spectrum. AtomicSimple is a minimal single IPC CPU model, TimingSimple is similar but also simulates the timing of memory references, InOrder is a pipelined, in-order CPU, and O3 is a pipelined, out-of-order CPU model.

Each execution-driven CPU model can operate in either of two modes. System-call Emulation (SE) mode avoids the need to model devices or an operating system (OS) by emulating most system-level services. Meanwhile, Full-System (FS) mode executes both user-level and kernel-level instructions and models a complete system including the OS and devices.

The gem5 simulator includes two different memory system models, Classic and Ruby. The Classic model (from M5) provides a fast and easily configurable memory system, while the Ruby model (from GEMS) provides a flexible infrastructure capable of accurately simulating a wide variety of cache coherent memory systems.

2.2 Availability

There are several types of gem5 user; each has different goals and requirements. These include academic and corporate researchers, engineers in industry, and undergraduate and graduate students.

The gem5 license (based on BSD) is friendly both to corporate users, since businesses need not fear being forced to reveal proprietary information, and to academics, since they retain their copyright and thus get credit for their contributions.

III. EXPERIMENTAL SETUP

The GEM5 simulator can be installed on various platforms such as Linux, BSD, MacOS, and Solaris. It is preferable to have 64-bit system while installing GEM5. The various tools required for building GEM5 are SCons, SWIG, GCC/G++ 4.8, Mercurial, python 2.7, python-dev.

The GEM5 is downloaded from gem5.org with help of mercurial. We then build various architectures such as ARM, ALPHA, X86 etc. using the build command in gem5.After building various architectures in gem5 we then download cross compilers to compile the c program to a particular architecture. GEM5 has two mode of operation System Emulation (SE) mode and Full System (FS) mode. By default System Emulation (SE) mode is enabled for all architectures without the requirement of any new packages or image files. System Emulation

International Journal of Electrical and Electronics Engineers
Vol. No. 9, Issue No. 01, January-June 2017
ISSN (O) 2321-2055
ISSN (P) 2321-2045

mode usually used for running individual application or set of application. The Full System mode is used for booting operating system, to study behavior interrupts, exceptions, fault handlers, and privileged instruction.

The cross compiled programs can be run on respective architectures in System Emulation mode. Statistics files are generated for each run.

IV. SYSTEM DESIGN AND MODELLING

The gem5 simulator has a wide range of simulation capabilities ranging from the selection of ISA, CPU model, and coherence protocol to the instantiation of interconnection networks, devices and multiple systems.

4.1Simple system modelling

To set up a simple simulation script for gem5 and to run gem5 for the first time. We have to first successfully built gem5 with an executable "build/X86/gem5.opt". Our configuration script is going to model a very simple system. We'll have just one simple CPU core. This CPU core will be connected to a system-wide memory bus. And we'll have a single DDR3 memory channel, also connected to the memory bus. Gem5 takes, as a parameter, a python script which sets up and executes the simulation. In this script, we can create a system to simulate, create all of the components of the system, and specify all of the parameters for the system components as shown in figure 1.

4.2 System modelling with two-level cache hierarchy

Using the previous configuration script as a starting point in this we will add a cache hierarchy to the system as shown in figure3.

FIGURES

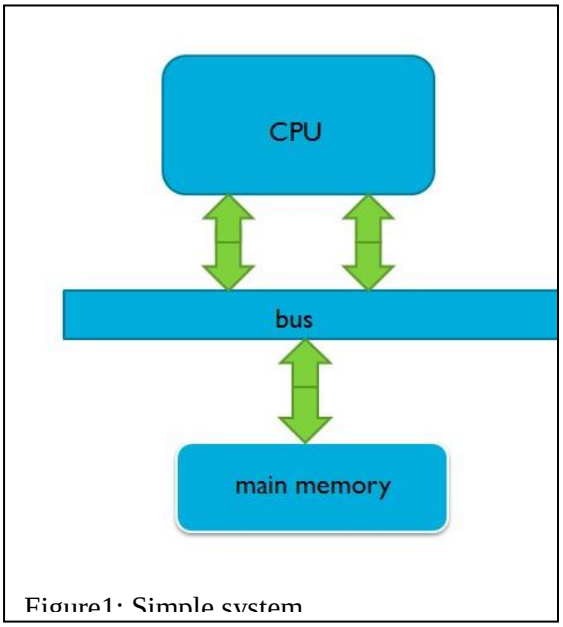

Figure1: Simple system

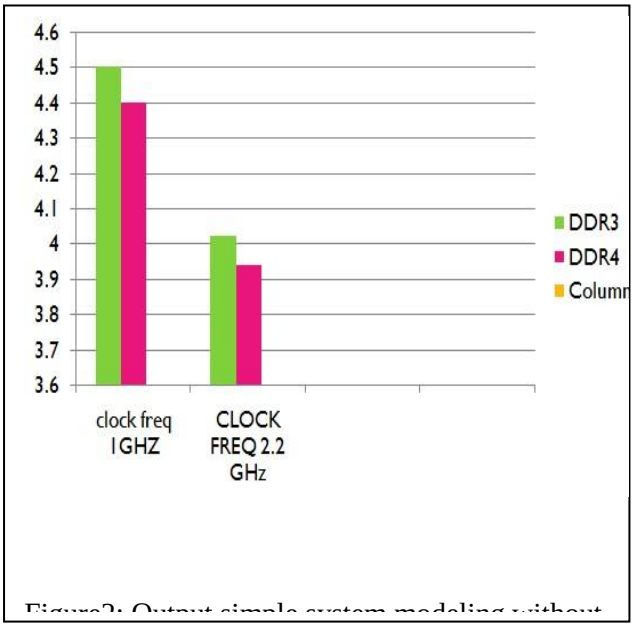

Figure2: Output simple system modeling without

International Journal of Electrical and Electronics Engineers
Vol. No. 9, Issue No. 01, January-June 2017
ISSN (O) 2321-2055
ISSN (P) 2321-2045

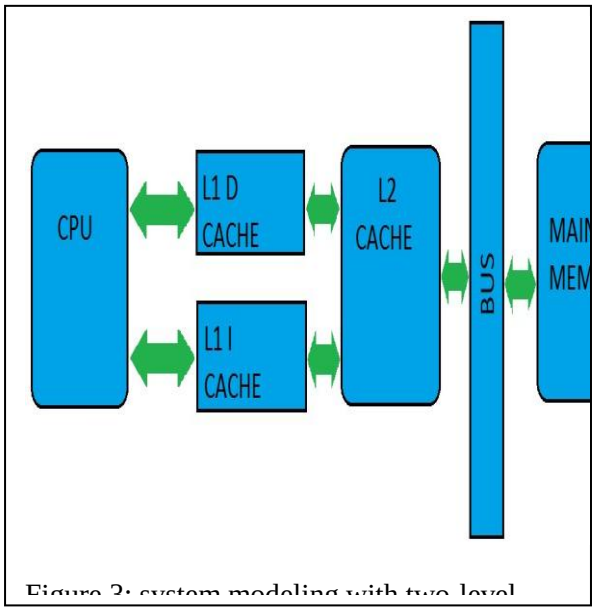

Figure 3: system modeling with two level

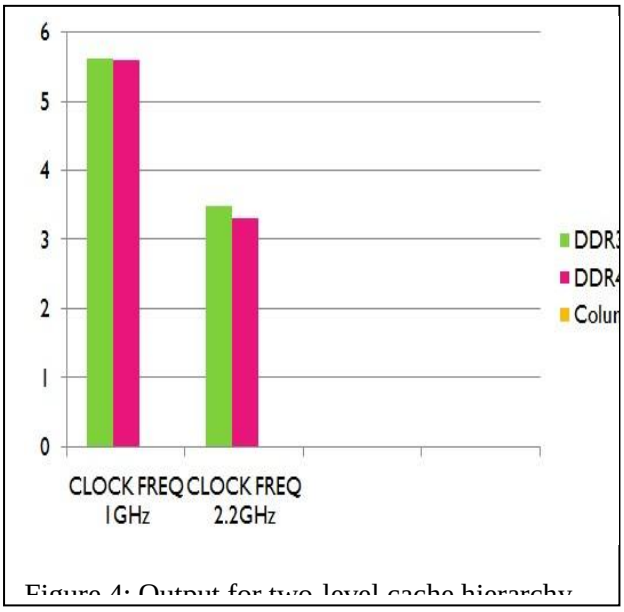

Figure 4: Output for two level cache hierarchy

V. CONCLUSION

In this paper we studied cache behavior on X86 processor in Gem5 syscall emulation simulator. Because cache plays important part in computer performance it's necessary to study the cache behavior.

REFERENCES

[1] Marty, M. R., Bingham, J. D., Hill, M. D., Hu, A. J., Martin, M. M. K., and Wood, D. A. Improving multiple-CMP systems using token coherence. In Proceedings of the 11th Annual International Symposium on High-Performance Computer Architecture (HPCA) (2005), pp. 328–339

[2] Barnes, B., and Slice, J. SimNow: A fast and functionally accurate AMD X86-64 system simulator. Tutorial at the IEEE International WorkloadCharacterization Symposium, 2005

[3] Bienia, C.; Kumar, S.; Li, K., "PARSEC vs. SPLASH-2: A quantitative comparison of two multithreaded benchmark suites on Chip-Multiprocessors," Workload Characterization, 2008. IISWC 2008. IEEE International Symposium on , vol., no., pp.47,56, 14-16 Sept. 2008

[4] Black, G., Binkert, N., Reinhardt, S. K.,and Saidi, A. Processor and System-on-Chip Simulation. Springer, 2010, ch. 5, "Modular ISAIndependent Full-System Simulation".

[5] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi,A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, and D. A. Wood, "The gem5 simulator," ACM SIGARCH Computer Architecture News, vol. 39, no. 2, 2011.

[6] Vikas B, Basavaraj Talawar "On the Cache Behavior of SPLASH-2 Benchmarks on ARM and ALPHA processors in Gem5 Full System Simulator" 2014 3rd International Conference on Eco-friendly Computing and Communication Systems.

[7] Avani Sharma; Anay Jain "using gem5 simulator to analyse TLB and Cache Statistic with multi threaded PARSEC benchmark" 2015 international Conference of green computing and internet of things.