Innovative Research and Applications in Next-Generation High Performance Computing

Qusay F. Hassan
Mansoura University, Egypt

A volume in the Advances in Systems Analysis, Software Engineering, and High Performance Computing (ASASEHPC) Book Series

Information Science REFERENCE
An Imprint of IGI Global

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA, USA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

Copyright © 2016 by IGI Global. All rights reserved. No part of this publication may be reproduced, stored or distributed in any form or by any means, electronic or mechanical, including photocopying, without written permission from the publisher. Product or company names used in this set are for identification purposes only. Inclusion of the names of the products or companies does not indicate a claim of ownership by IGI Global of the trademark or registered trademark.

Library of Congress Cataloging-in-Publication Data

Names: Hassan, Qusay F., 1982- editor.
Title: Innovative research and applications in next-generation high performance computing / Qusay F. Hassan, editor.
Description: Hershey : Information Science Reference, 2016. | Includes bibliographical references and index.
Identifiers: LCCN 2016007158| ISBN 9781522502876 (hardcover) | ISBN 9781522502883 (ebook)
Subjects: LCSH: High performance computing.
Classification: LCC QA76.88 .H38 2016 | DDC 004.1/1--dc23 LC record available at https://lccn.loc.gov/2016007158

This book is published in the IGI Global book series Advances in Systems Analysis, Software Engineering, and High Performance Computing (ASASEHPC) (ISSN: 2327-3453; eISSN: 2327-3461)

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

All work contributed to this book is new, previously-unpublished material. The views expressed in this book are those of the authors, but not necessarily of the publisher.

For electronic access to this publication, please contact: eresources@igi-global.com.

Chapter 4
Design Space Exploration Using Cycle Accurate Simulator

Arsalan Shahid
HITEC University, Pakistan

Muhammad Yasir Qadri
University of Essex, UK

Bilal Khalid
HITEC University, Pakistan

Nadia N. Qadri
COMSATS Institute of Information Technology, Pakistan

Jameel Ahmed
HITEC University, Pakistan

ABSTRACT

Multi-Processor System on Chip (MPSoC) architectures have become a mainstream technology for obtaining performance improvements in computing platforms. With the increase in the number of cores, the role of cache memory has become pivotal. An ideal memory configuration is always desired to be fast and large; but, in fact, striking to balance between the size and access time of the memory hierarchy is considered by processor architect. Design space exploration is used for performance analysis of systems and helps to find the optimal solution for obtaining the desired objectives. In this chapter, we explore two design space parameters, i.e., cache size and number of cores, for obtaining the desired energy consumption. Moreover, previously presented energy models for multilevel cache are evaluated by using cycle accurate full system simulator. Our results show that with the increase in cache sizes, the number of cycles required for application execution decreases, and by increasing number of cores, the throughput improve.

INTRODUCTION

With the evolution of multicore processor architectures and trend shifting towards parallelism, the role of cache memory has become pivotal (Wei, Shao, Huang, 2016; Geer, 2005; Jacob, Ng & Wang, 2010). Cache memory plays a vital role in modern processor architectures as it reduces the gap between main memory and the processor (González, A., Aliagas, C., & Valero, 2014). It serves to reduce the average time taken by each memory access or in other words cache acts as a buffer between Central Processing Unit (CPU) and main memory. Multi-level cache hierarchy is always beneficial as it bridges processor-

DOI: 10.4018/978-1-5225-0287-6.ch004

Copyright © 2016, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

memory gap efficiently (Whitham, J., Audsley, N. C., & Davis, 2014). Moreover, this memory hierarchy can reduce up to 50% of the total energy spent by the microprocessor (Segars, 2001). Therefore, cache size has become a critical parameter for a processor architect to choose (Huang, J., Yeluri, S., Frailong, Libby, 2014). This fact has urged researchers to explore cache hierarchy design in terms of energy optimization.

Energy consumption has always been a key concern and desired objective in multicore processor systems (Hennessy & Patterson, 2011). For the processor architect, choice of components to achieve minimum energy consumption is a very important and critical decision to make. Most of the techniques involve verification methodology based on Transaction Level Modeling (TLM) (Ferro, Pierre, Amor, Lachaize, & Lefftz, 2011) or virtualized platforms (Magnusson, Christensson, Eskilson, & others, 2002) to analyze a proposed configuration several times. But, a single configuration can take several hours for complete evaluation. Moreover, the results obtained out of those techniques to explore the optimum design space are not accurate as the simulators are not cycle accurate. Therefore, an efficient technique to propose search/design space parameters using a cycle accurate simulator is required. The problem of exploring configurable parameters to minimize energy consumption is known as Design Space Exploration (DSE) (Silvano, Fornaciari, & Villar, 2014). DSE is used for system optimization, integration and to explore several design parameters. In this paper, we have explored two such parameters:

1. The optimum sizes of cache at different levels of memory hierarchy.
2. The number of cores for best performance with respect to energy consumption.

We focused on the design space parameters for two level cache memory hierarchies (L1 & L2 cache) and improved the cache energy models presented by M.Y Qadri (Qadri & McDonald-Maier, 2010). Moreover, we evaluated cache energy mathematical models by using different standard benchmarks. These models significantly require less number of parameters which have been estimated by using state-of-the-art cycle accurate Micro Architectural and System Simulator (MARSS) for multicore processors (Patel, Afram, & Ghose, 2011). MARSS gives the exact number of cycles required to execute an instruction. Energy per access of the tag array for the L1 and L2 cache were obtained from HP Labs' CACTI, i.e., an integrated cache timing, power, and area model tool (D. Tarjan & N. P. Jouppi, 2006).

The aim of this research is to explore the design space parameters, i.e., cache size and number of cores, to estimate the best point where a benchmark takes minimum number of cycles to execute an instruction. For this purpose, we took different sizes of L1 and L2 cache and observed the number of cycles for various number of cores, i.e., 2, 4, 8, 12 and 16. Modern processor architectures have evolved with a lot of complexities that it becomes too difficult to simulate a whole CPU or just even a core. However, MARSS provide simulation results on cycle by cycle basis for microprocessor architectures. For the sake of simplicity, we have evaluated the results by considering homogeneous cores.

The rest of chapter is divided into six sections. Section 2 presents related work in the field of cache memory hierarchy and different performance estimation techniques along with simulation tools. Cache energy models are discussed in section 3. Design space exploration methodology and results have been shown in section 4. In section 5 discussions are presented on energy estimation and number of cycles for various cache sizes and then the chapter concludes.

RELATED WORK

In the recent past years, drive towards low power processing has challenged the designers and researchers to optimize every component of the processor (Wang & Naffziger, 2008). As cache memory is an integral part in designing of microprocessor, cache energy consumption and throughput models have been the focus of research for some time. Some of the previous works related to cache power estimation along with tools such as full system cycle accurate simulators are presented in this section.

Simunic et al. (Šimunić, Benini, & De Micheli, 1999) proposed analytical models for energy estimation in embedded systems. The per cycle energy model presented in their work comprises components such as energy consumption of processor, memory, interconnects and pins, DC-to-DC converters and level two (L2) cache. The model validation was performed using integrated simulator of ARM SDK (Advanced RISC Machines Limited (ARM) Inc., 1996). These models only analyze the embedded system power but do not estimate energy consumption for individual components of a processor. Li et al. (Y. Li & Henkel, 1998) proposed a full system detailed energy model comprising cache, main memory, and software energy components. Their work includes description of an Avalanche framework for estimating and optimizing energy dissipation of embedded systems. M.B. Kamble et al. (Kamble & K. Ghose, 1998) present analytical models for the energy dissipation in low power cache. The power obtained by these models was compared with that obtained by CAPE (Cache Power Estimator). The models for conventional caches are found to be accurate to within 2% error. A. C. borty et al. (Chakraborty et al., 2010) represents a new cache building that is multi-copy cache (MC2) which gives significant reduction in energy consumption of forceful voltage scaling in caches by producing the multiple copies of each data cache. From the experimental results, they obtain that using MC2 60% reduction in energy can be achieved. Johnson Kin et al. (Kin, Gupta, & Mangione-Smith, 1997) proposed an energy efficient memory structure. According to them L2 cache is placed behind the filter cache (a small memory), which is similar in structure and size to L1 cache in order to improve performance of processors.

C. Long. Su et al. (Su & Despain, 1995) have proposed the power trade-offs in designing of caches and energy reduction using Gray code addressing and cache sub banking. And experimental results show that direct-mapped caches consume less energy than set associative caches. Sheng Li et al. (S. Li et al., 2009) presented Multicore Power, Area, and Timing (McPAT) framework that supports comprehensive design space exploration for multicore processor configurations. At micro architectural level, McPAT includes models for the fundamental components of a chip multiprocessor and at circuit and technology level it supports critical-path timing, area, dynamic, short-circuit and leakage power modeling. McPAT help architects to use new standards combining performance with both area and power. As die cost increases with area, hence area is a critical design constraint. Therefore, good trade-off between performances and cost needs careful design of on-chip resources.

Yourst (2007) developed a full system clock accurate simulator (PTLsim) to simulate each component at instruction level. This simulator features the configurable RTL level architecture and pipelines at the speed of host system. Avadh et al. presented MARSS (Patel et al., 2011) based on PTLsim which is a cycle accurate complete system simulator for x86 and x64 based architectures, especially for multi-core hardware's. MARSS extends the functionality and support of PTLsim including complete user space simulations, unmodified software and OS stack and unmodified kernel. Tarjan et al. (D. Tarjan & N. P. Jouppi, 2006) presented CACTI which is an integrated cache access time, cycle time, area, aspect ratio, and power model. CACTI is intended for use by computer architects so they can better understand the performance tradeoffs inherent in different cache sizes and organizations.

Qadri et al. (Qadri, Gujarathi, & McDonald-Maier, 2009) have presented the techniques for power efficiency at processor core level. They proposed that processor's speed can be improved by adding pipeline stages. Clock frequency, supply voltage and cache can be used efficiently for power reduction. They proposed various techniques such as DVFS, Power and Clock gating for power optimization. In (Qadri & McDonald-Maier, 2010), Qadri et al. also presented mathematical models to calculate consumption of energy for multilevel caches using Ultra SPARC-2 and Power PC750 processors for two level cache. Then they extend their work in (Muhammad Yasir, Klaus D, & others, 2010), by considering the concept of battery powered embedded system, i.e., processor only turns on when required, otherwise it remains in sleep mode. Moreover, they proposed improved energy and throughput models of data caches. These models are suitable for design of optimized cache for processors.

In this paper, we evaluated the energy models presented in (Qadri & McDonald-Maier, 2010) for multilevel cache using MARSSx86 simulator. Being cycle accurate, this simulator has been used for multicore implementation of mathematical models and gives more accurate results as compared to other system simulators. We also explored design space to estimate minimum number of cycles a benchmark takes to execute an instruction. The following section illustrates cache energy models and estimation of cache parameters through cache simulator.

CACHE MEMORY ENERGY MODELS

In (Qadri & McDonald-Maier, 2010), Qadri et al. presented mathematical models to calculate consumption of energy for multilevel caches. The models presented by him analyze the energy consumption for multilevel cache using PowerPC750, and UltraSPARC-II processors. These models are used in various publications of authors (Qadri & Klaus, 2008, 2009, 2010); and are also compared against the results reported by Simics (River, W., 2006). According to the proposed models:

$$E_{total} = E_{ic} + E_{dc} + E_{l2c} \tag{1}$$

E_{ic} = Energy consumed by instruction cache.
E_{dc} = Energy consumed by data cache.
E_{l2c} = Energy consumed by L2 cache.

where:

$$E_{ic} = E_{ic-read} + E_{ic-mp} \tag{2}$$

$$E_{dc} = E_{dc-read} + E_{dc-write} + E_{dc-mp} \tag{3}$$

$$E_{l2c} = E_{l2c-read} + E_{l2c-write} + E_{l2c-mp} \tag{4}$$

E_{x-read}, $E_{x-write}$ are read and write energy for instruction, data or L2 cache, E_{x-mp} is miss penalty energy for corresponding cache.

In this paper, we improve the energy models by evaluating them using full system simulator, i.e., MARSS for multicore processor simulations. MARSS provides number of read/write cache hits and misses for evaluation of models presented above. It has been used in hundreds of research papers in order to obtain cycle accurate results. We have compared results of our models with those of MARSS and the comparison shows a small amount of error for various configurations. CACTI is used to calculate read and write energy and energy per cycle, cache access time is obtained by simple formula of energy given below:

$$E = \frac{V * I}{f} \tag{5}$$

Table 1 shows the data obtained from CACTI for various cache sizes such as 4 KB, 8 KB, 16 KB, 32 KB, 64 KB and 128 KB respectively.

Table 1. CACTI simulator data

Parameters	For 4 KB	For 8 KB	For 16 KB	For 32 KB	For 64 KB	For 128 KB
Line Size [Bytes]	64	64	64	64	64	64
Number of lines	1024	1024	1024	1024	1024	1024
R/W ports	1	1	1	1	1	1
Associativity	2	2	2	2	2	2
Read ports	0	0	0	0	0	0
Write ports	1	1	1	1	1	1
Access time [ns]	0.687433	0.774311	0.776892	0.889597	1.21415	1.365661
Cycle time [ns]	0.350633	0.36339	0.367621	0.41209	0.555608	0.6597919
Read Energy [nJ]	0.018092	0.023574	0.031482	0.0359356	0.0421182	0.0421182
Write Energy [nJ]	0.004482	0.009114	0.0091872	0.0095884	0.0118159	0.0179147

Table 2. Benchmark applications from Splash-2

Benchmarks	Description
Barnes	This benchmark simulates the gravitational forces acting on a galactic cluster using the Barnes-Hut n-body algorithm (Singh, Hennessy, & Gupta, 1995).
Fmm	The FMM application implements a parallel adaptive Fast Multipole Method to simulate the interaction of a system of bodies (N-body problem) (Singh, Holt, Hennessy, & Gupta, 1993).
Water-spatial	It imposes a, 3-D spatial data structure on the cubical domain, resulting in a 3-D, grid of boxes.
Water-nsquared	Evaluated forces and potential that occur over time in a system of water & molecules using predictor -corrector method.

DESIGN SPACE EXPLORATION

In this section, we have presented the methodology to explore design space in order to find the optimal point where a program takes minimum number of cycles to run completely. Moreover, results have also been presented.

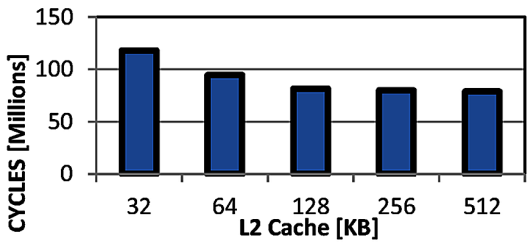
Different standard benchmark applications from SPLASH-2 (Woo, Ohara, Torrie, Singh, & Gupta, 1995) benchmarking suite have been used, i.e., Barnes, Fmm, Water-Spatial and water-nsquared. These benchmark applications are described in Table 2.

We have taken different sizes of L1 and L2 caches for 2, 4, 8, 12 and 16 cores. We changed the sizes of L2 cache to 32kB, 64kB, 128kB, 256kB and 512kB in the machine configuration file of MARSS, keeping L1 cache and number cores constant and observed the number of cycles. Table 3 shows cache parameters, we have kept in cache configuration file of MARSS.

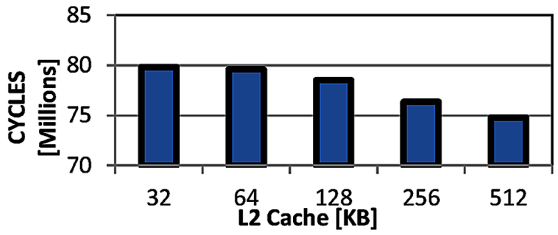
Figure 1 shows the number of cycles consumed by benchmark applications for two cores with different L1 & L2 cache sizes. Similarly results for 4, 8, 12, and 16 cores have been shown in Figure 2, 3, 4, and 5 respectively.

For evaluation of energy we kept L1 (instruction and data) cache as 32kB and L2 cache as 64kB. Table 4 illustrates the energy of calculated for different benchmarks on single core machine. Similarly, we evaluated the energy consumption by changing the number of cores, i.e., 2, 4, 8 and 16 shown in Table 5, 6, 7, and 8 respectively.

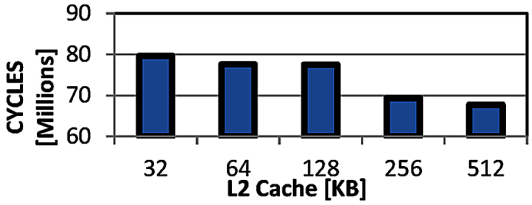
Figure 1. Number of cycles for 2 cores



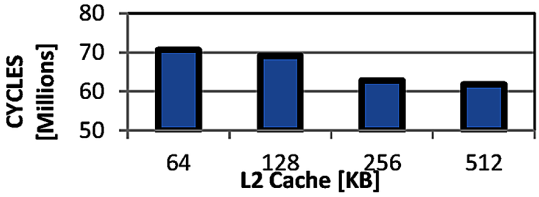
(a) *Number of cycles for L1 = 4 kB*



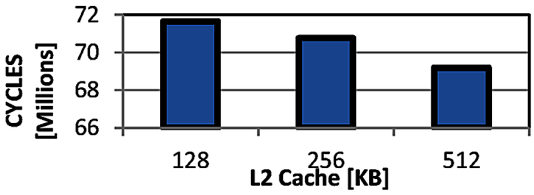
(b) Number of cycles for L1 = 8 KB



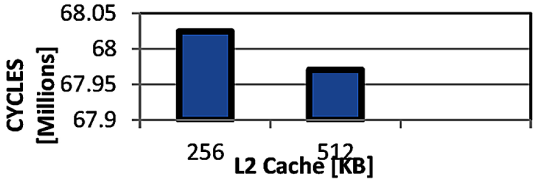
(c) Number of cycles for L1 = 16 KB



(d) Number of cycles for L1 = 32 KB



(e) Number of cycles for L1 = 64 KB



(f) Number of cycles for L1 = 128 KB

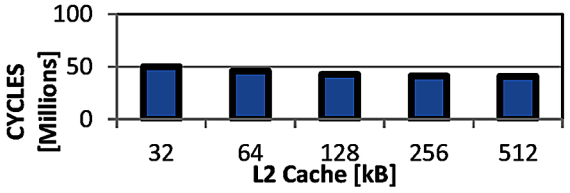
Table 3. Cache configuration data

Parameter	Value
Cache Type	Mesi
Associativity	4 for L1, 8 for L2
Latency	2 for L1, 5 for L2
Read ports	2 for L1, 2 for L2
Write ports	1 for L1, 2 for L2
Line Size (Bytes)	64

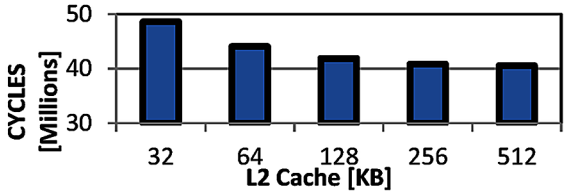
Table 4. Energy evaluation for single core

Benchmarks	E_{ic} [nJ]	E_{dc} [nJ]	E_{l2c} [nJ]	E_{total} [nJ]
Barnes	930034.37	1877528.1277	52741139.298	55548701.298
Fmm	1419665.393	23047857.59	49737310.15	74164833.1344
Water-Spatial	1830823.931	24666379.48	63805772.92	90302976.33
Water-nsquared	844077.9734	18740226.11	62852582.11	82436886.1883

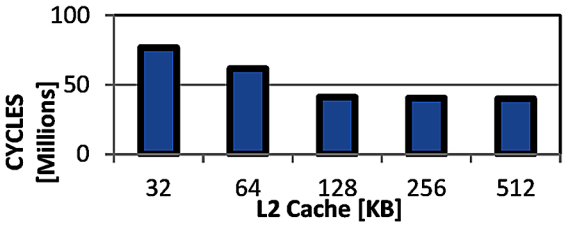
Figure 2. Number of cycles for 4 cores



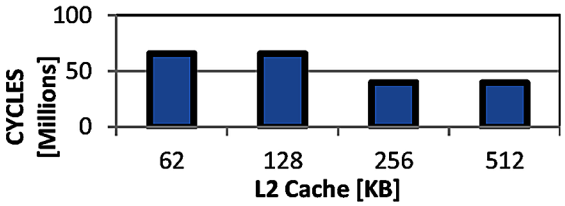
(a) *Number of cycles for L1 = 4 kB*



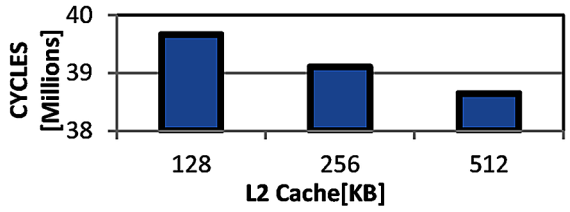
(b) Number of cycles for L1 = 8 kB



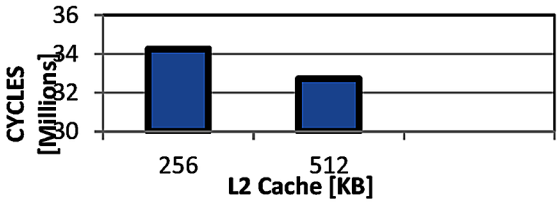
(c) Number of cycles for L1 = 16 kB



(d) Number of cycles for L1 = 32 kB



(e) Number of cycles for L1 = 64 kB



(f) Number of cycles for L1 = 128 kB

Figure 3. Number of cycles for 8 cores

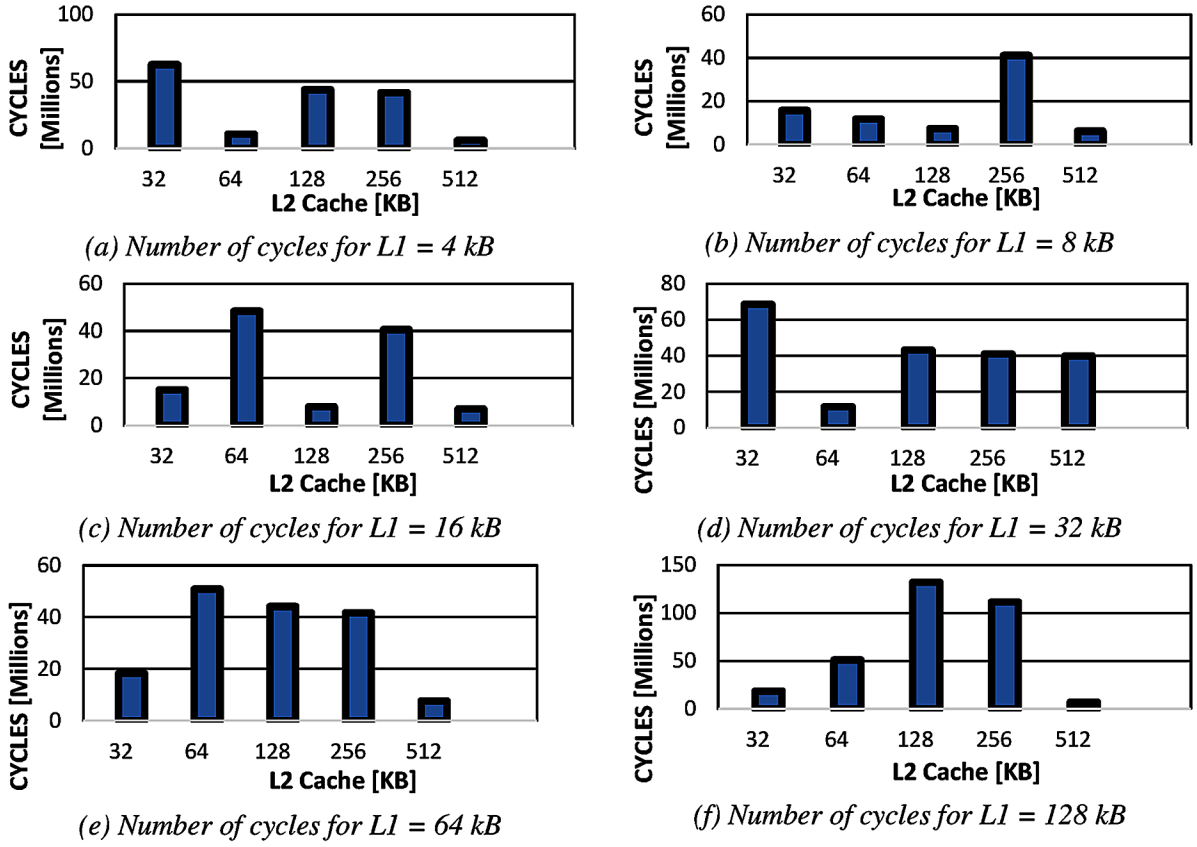


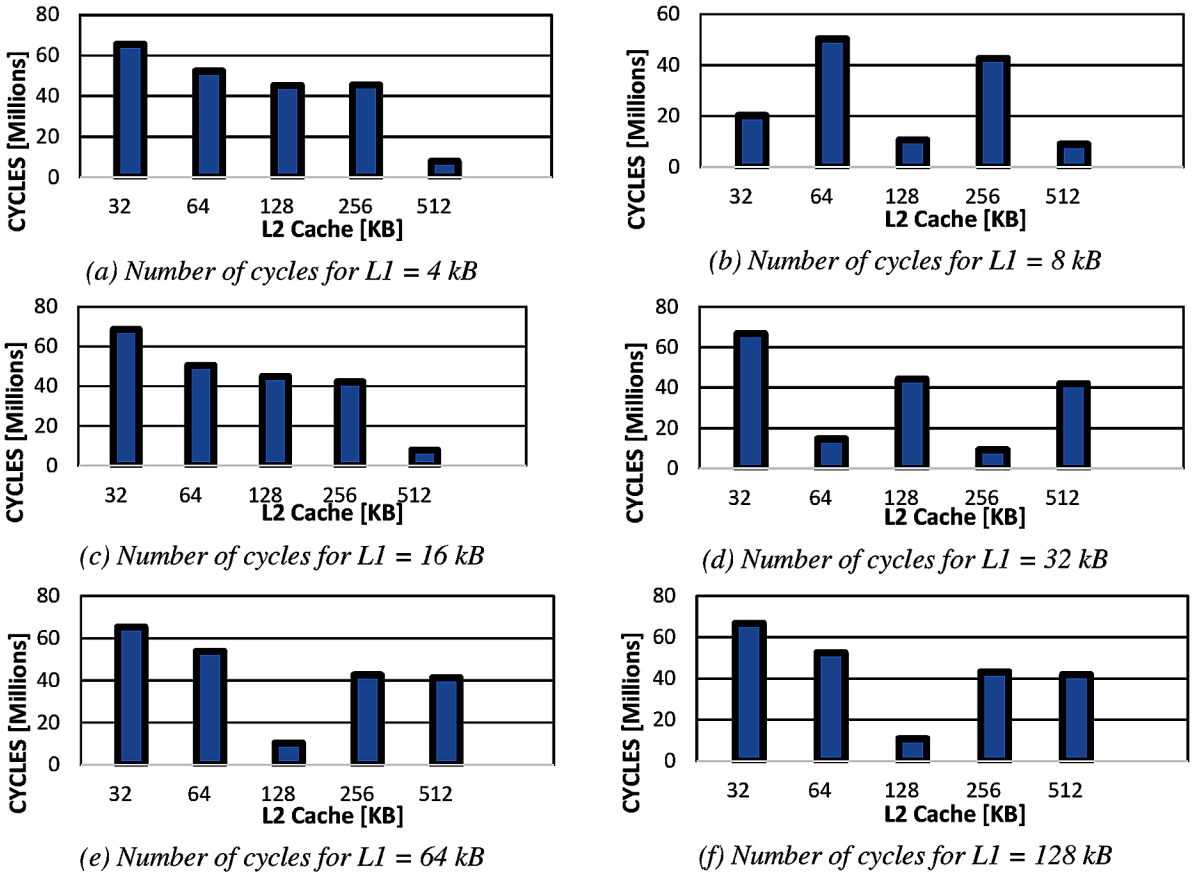
Table 5. Energy evaluation for 2 cores

Benchmarks	E_{ic} (nJ)	E_{dc} (nJ)	E_{l2c} (nJ)	E_{total} (nJ)
Barnes	252064.016	51951.646	728441.298	1032456.96
Fmm	218533.9278	31863.6381	567673.15	818070.7159
Water-Spatial	64791.4625	4337.5904	835757.92	904886.9729
Water-nsquared	1021534.506	6356317.51	796150.11	8174002.126

Table 6. Energy evaluation for 4 cores

Benchmarks	E_{ic} (nJ)	E_{dc} (nJ)	E_{l2c} (nJ)	E_{total} (nJ)
Barnes	118664.457	30981.096	524411.92	674057.473
Fmm	100933.767	10463.9521	319693.11	431090.8291
Water-Spatial	44790.502	6037.0103	305707.92	356535.4323
Water-nsquared	609532.074	37601.16	602440.12	1249573.354

Figure 4. Number of cycles for 12 cores



(a) Number of cycles for L1 = 4 kB

(b) Number of cycles for L1 = 8 kB

(c) Number of cycles for L1 = 16 kB

(d) Number of cycles for L1 = 32 kB

(e) Number of cycles for L1 = 64 kB

(f) Number of cycles for L1 = 128 kB

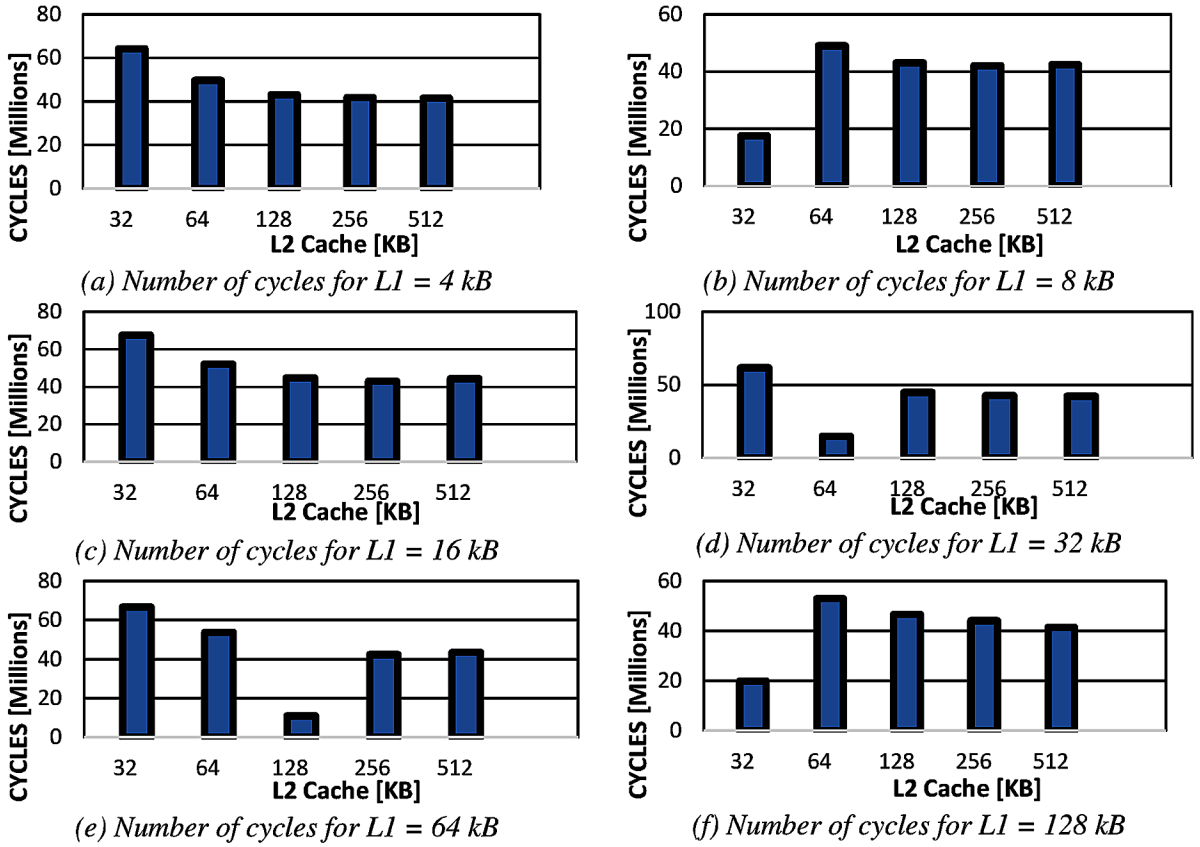
Table 7. Energy evaluation for 8 cores

Benchmarks	E_{ic} (nJ)	E_{dc} (nJ)	E_{l2c} (nJ)	E_{total} (nJ)
Barnes	92014.223	9551.236	40211.32	141776.779
Fmm	8233.919	6263.385	29093.11	43590.414
Water-Spatial	6594.232	927.5934	65047.70	72569.5254
Water-nsquared	50213.205	9317.51	30440.42	89971.135

Table 8. Energy evaluation for 16 cores

Benchmarks	E_{ic} (nJ)	E_{dc} (nJ)	E_{l2c} (nJ)	E_{total} (nJ)
Barnes	8064.25	981.6	8169.33	17215.18
Fmm	433.9278	863.8	33410.78	34708.5078
Water-Spatial	491.450	74.5	5773.34	6339.29
Water-nsquared	9534.506	657.49	5582.109	15774.105

Figure 5. Number of cycles for 16 cores



(a) Number of cycles for L1 = 4 kB

(b) Number of cycles for L1 = 8 kB

(c) Number of cycles for L1 = 16 kB

(d) Number of cycles for L1 = 32 kB

(e) Number of cycles for L1 = 64 kB

(f) Number of cycles for L1 = 128 kB

DISCUSSIONS

In order to find out the energy out of proposed model, we used MARSS-x86 (Patel et al., 2011) to run a number of benchmark applications from SPLASH-2 (Woo et al., 1995) benchmarking suite. Simulations have been run on Linux kernel 3.5.7. It is to be noted that MARSS provide cycle accurate simulations for various benchmark applications but parameters such as E_{x-read}, $E_{x-write}$ were obtained using HP labs' CACTI that is an integrated cache timing, power, and area model tool (Tarjan & Jouppi, 2006) (see Table 1.)

Following are the observations made during design space exploration for finding optimum conditions where a program takes minimum number of cycles to run completely.

1. Benchmark takes less number of cycles to execute an instruction, if value of L1 cache increases.
2. Large value of L2 requires less number of cycles to execute an application.
3. Number of cores and energy consumption have inverse relationship. Greater number of cores in a processor architecture will take less energy to complete the task.

CONCLUSION

In this chapter, we had proposed a technique for the design space exploration of multicore processors to explore the optimum search space solutions in the form of number of cores and cache sizes. Furthermore, we evaluated cache memory energy models for various benchmark applications from SPLASH-2 benchmarking suite, i.e., Barnes, Fmm, Water-nsquared and Water-Spatial. For this purpose, a full system cycle accurate simulator had been used. The benchmarks had been run on various number of cores by changing cache sizes as well. For evaluation of mathematical models, parameters such as cache access time were obtained by using cache simulator. For dual core machine, Barnes performed 25.1%, 32.61%, and 32.61% better than Fmm, Water-nsquared and Water-spatial respectively. We observed that with the increase in cache sizes, number of cycles in benchmarks execution decreases. By increasing number of cores, the throughput improved.

We had evaluated the results by considering homogeneous cores. In the future work the presented technique will be improved to explore more design space solutions and will be extended for heterogeneous cores. The design solutions will be applied in real time adaptive memory systems, where an accurate estimate of energy consumption for cache is required for reconfiguration purpose. Moreover, Cache models had been previously used to analyze the multilevel cache using PowerPC750, and UltraSPARC-II processors. In the future, these models would be used to evaluate state-of-the-art Intel Xeon processors.

ACKNOWLEDGMENT

This work was supported by the National ICT R&D Fund, Pakistan through grant numbered: ICTRDF/TR&D/2012/65

REFERENCES

Advanced RISC Machines Limited (ARM) Inc. (1996). *ARM Software Development Toolkit*.

Chakraborty, A., Homayoun, H., Khajeh, A., Dutt, N., Eltawil, A., & Kurdahi, F. (2010). E< MC2: less energy through multi-copy cache.*Proceedings of the 2010 international conference on Compilers, architectures and synthesis for embedded systems* (pp. 237–246). ACM.

Ferro, L., Pierre, L., Amor, Z. B. H., Lachaize, J., & Lefftz, V. (2011). Runtime Verification of Typical Requirements for a Space Critical SoC Platform. In *Formal Methods for Industrial Critical Systems* (pp. 21–36). Springer. doi:10.1007/978-3-642-24431-5_4

Geer, D. (2005). Chip makers turn to multicore processors. *Computer*, *38*(5), 11–13. doi:10.1109/MC.2005.160

González, A., Aliagas, C., & Valero, M. (2014, June). A data cache with multiple caching strategies tuned to different types of locality. *Proceedings of the25th Anniversary International Conference on Supercomputing Anniversary* (pp. 217-226). ACM. doi:doi:10.1145/2591635.2667170 doi:10.1145/2591635.2667170

Hennessy, J. L., & Patterson, D. A. (2011). *Computer architecture: a quantitative approach*. Elsevier.

Huang, J., Yeluri, S., Frailong, J. M., Libby, J. G., Gupta, A. P., & Coelho, P. (2014). *U.S. Patent No. 8,627,007*. Washington, DC: U.S. Patent and Trademark Office.

Jacob, B., Ng, S., & Wang, D. (2010). *Memory systems: cache, DRAM, disk*. Morgan Kaufmann.

Kamble, M. B., & Ghose, K. (1998). Modeling energy dissipation in low power caches. *Proceedings of theInternational Symposium on Low Power Electronics and Design* (pp. 143–148).

Kin, J., Gupta, M., & Mangione-Smith, W. H. (1997). The filter cache: an energy efficient memory structure.*Proceedings of the 30th annual ACM/IEEE international symposium on Microarchitecture* (pp. 184–193). IEEE Computer Society. doi:10.1109/MICRO.1997.645809

Li, S., Ahn, J. H., Strong, R. D., Brockman, J. B., Tullsen, D. M., & Jouppi, N. P. (2009). McPAT: an integrated power, area, and timing modeling framework for multicore and manycore architectures. *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture MICRO-42* (pp. 469–480). IEEE. doi:doi:10.1145/1669112.1669172 doi:10.1145/1669112.1669172

Li, Y., & Henkel, J. (1998). A framework for estimation and minimizing energy dissipation of embedded HW/SW systems.*Proceedings of the 35th annual Design Automation Conference* (pp. 188–193). ACM. doi:10.1145/277044.277097

Magnusson, P. S., Christensson, M., Eskilson, J., Forsgren, D., Hallberg, G., Hogberg, J., & Werner, B. et al. (2002). Simics: A full system simulation platform. *IEEE Computer*, *35*(2), 50–58. doi:10.1109/2.982916

Mittal, M. (1998). *Computer system and method of allocating cache memories in a multilevel cache hierarchy utilizing a locality hint within an instruction*. Google Patents.

Muhammad Yasir, Q., Klaus D, M.-M. (2010). Data cache-energy and throughput models: design exploration for embedded processors. *EURASIP Journal on Embedded Systems*. EURASIP.

Patel, A., Afram, F., & Ghose, K. (2011). Marss-x86: A qemu-based micro-architectural and systems simulator for x86 multicore processors. Proceedings of the 1st International Qemu Users' Forum (pp. 29–30).

Przybylski, S. A. (1990). *Cache and memory hierarchy design: a performance-directed approach*. Morgan Kaufmann.

Qadri, M. Y., Gujarathi, H. S., & McDonald-Maier, K. D. (2009). Low Power Processor Architectures and Contemporary Techniques for Power Optimization–A Review. *Journal of Computers*, *4*(10), 927–942. doi:10.4304/jcp.4.10.927-942

Qadri, M. Y., & McDonald-Maier, K. D. (2010). Analytical Evaluation of Energy and Throughput for Multilevel Caches. *Proceedings of the 2010 12th International Conference on Computer Modelling and Simulation (UKSim)* (pp. 598–603). IEEE. doi:doi:10.1109/UKSIM.2010.114 doi:10.1109/UKSIM.2010.114

River, W. (2006). Simics Full System Simulator.

Segars, S. (2001). *Low power design techniques for microprocessors*. Tutorial Note of the ISSCC.

Silvano, C., Fornaciari, W., & Villar, E. (2014). *Multi-objective Design Space Exploration of Multiprocessor SoC Architectures*. Springer.

Šimunić, T., Benini, L., & De Micheli, G. (1999). Cycle-accurate simulation of energy consumption in embedded systems.*Proceedings of the 36th annual ACM/IEEE Design Automation Conference* (pp. 867–872). ACM. doi:10.1109/DAC.1999.782199

Singh, J. P., Hennessy, J. L., & Gupta, A. (1995). Implications of hierarchical N-body methods for multiprocessor architectures.*ACM Transactions on Computer Systems*, *13*(2), 141–202. doi:10.1145/201045.201050

Singh, J. P., Holt, C., Hennessy, J. L., & Gupta, A. (1993). A parallel adaptive fast multipole method.*Proceedings of the 1993 ACM/IEEE conference on Supercomputing* (pp. 54–65). ACM. doi:10.1145/169627.169651

Smith, A. J. (1982). Cache memories.[CSUR]. *ACM Computing Surveys*, *14*(3), 473–530. doi:10.1145/356887.356892

Su, C.-L., & Despain, A. M. (1995). Cache design trade-offs for power and performance optimization: a case study.*Proceedings of the 1995 international symposium on Low power design* (pp. 63–68). ACM. doi:10.1145/224081.224093

D. Tarjan, S. Thoziyoor, & N. P. Jouppi. (2006). CACTI 4.0. *HP Laboratories Palo Alto*.

Wang, A., & Naffziger, S. (2008). *Adaptive techniques for dynamic processor optimization: theory and practice*. Springer Science & Business Media. Springer. doi:10.1007/978-0-387-76472-6

Wei, H., Shao, Z., Huang, Z., Chen, R., Guan, Y., Tan, J., & Shao, Z. (2016). RT-ROS: A real-time ROS architecture on multi-core processors. *Future Generation Computer Systems*, *56*, 171–178. doi:10.1016/j.future.2015.05.008

Whitham, J., Audsley, N. C., & Davis, R. I. (2014). Explicit reservation of cache memory in a predictable, preemptive multitasking real-time system. *ACM Transactions on Embedded Computing Systems*, *13*(4s), 120. doi:10.1145/2523070

Wilson, A. W. Jr, & Frank, S. J. (1988). *Hierarchical cache memory system and method*. Google Patents.

Woo, S. C., Ohara, M., Torrie, E., Singh, J. P., & Gupta, A. (1995). The SPLASH-2 programs: Characterization and methodological considerations. In ACM SIGARCH Computer Architecture News (Vol. 23, pp. 24–36). ACM.

Yourst, M. T. (2007). PTLsim: A cycle accurate full system x86-64 microarchitectural simulator. *Proceedings of the IEEE International Symposium on Performance Analysis of Systems & Software ISPASS '07* (pp. 23–34). IEEE. doi:doi:10.1109/ISPASS.2007.363733 doi:10.1109/ISPASS.2007.363733

KEY TERMS AND DEFINITIONS

Benchmarks: Standard tools for performance evaluation and comparison.

Cache Memory: A memory used by central processing unit of a computer to reduce the average time to access data from the main memory, i.e., it serves as a bridge between processor and main memory.

Cycle Accurate Simulator: A computer program that simulates a microarchitecture on a cycle-by-cycle basis.

Design Space Exploration: Finding the optimal search space solutions for desired objective.

Multicore Processors: A multicore processor is a processor with multiple processor cores on a single die or multiple dice inside a multi-chip module.