SARC Power Estimation Methodology

Daniele Ludovici and Georgi N. Gaydadjiev
Computer Engineering Laboratory
Electrical Engineering, Mathematics and Computer Science Department
Delft University of Technology
Makelweg 4, 2628 CD, Delft, The Netherlands
{daniele, georgi}@ce.et.tudelft.nl

Abstract— **In modern CMOS technologies, power consumption is becoming a significant challenge for the integrated circuits industry. Accurate estimation of power dissipation is very important during micro-architectural design of every computational structure. This work presents the methodology we intent to use in future investigations regarding power consumption of the SARC architecture. SARC project is targeting next generation scalable computer architectures where multiple cores will be responsible for the performance gains. Proposing new approaches to reduce power consumption and scale for power and performance in the architectural development is one of the main challenges of this project. Therefore, adequate methodology to estimate it is needed. In this paper, we introduce the tools (i.e., Unisim simulator and Cacti 4.0) and the proposed methodology to investigate the problem of power consumption in such architecture.**

Keywords—**Power Estimation, Low Power Design, Power Management, Unisim Simulator, Cacti 4.0**

I. Introduction

The last two decades have seen an inconceivable proliferation of electronic devices in many aspects of daily life, from mobile phones and personal digital assistant (PDAs) to laptops. This phenomenon is stimulating many research efforts from both academical and industrial communities. Electronic equipment is expected to provide both adequate performance and reasonable battery lifetime. Moreover, the performance-power trade-off is a problem not only limited to mobile devices, but is spread out to the whole range of modern electronic appliances. Power related issues represent a design constraint that needs to be addressed by the computer architects. Among these issues, power dissipation constitutes a critical design metric for most of the VLSI circuits. Hence, estimation of power consumption during initial design stages is a required task for the designers. Usually, power estimation is performed using simulation tools. Many high-level tools with the purpose of estimating power and performance characteristics of existing and theoretical system designs were previously proposed [4], [13], [3], [11], [5], [9]. The UNIted SIMulation environment (i.e., Unisim [7]) is the tool that has been selected for our investigation. This paper introduces Unisim, a simulation environment aiming at studying new micro-architecture ideas. Unisim is built upon modules and it has been developed focusing on concepts such as re-usability and interoperability. Unisim comes with several external tools that can to interface to the core of the simulation environment. Such tools allow the researchers to perform different type of evaluations such as area estimation, access time estimation, etc. In this work we propose to use the "power capability" of the simulator to estimate and evaluate the power consumption of the SARC architecture memory subsystem. Due to the fact that a complete Unisim implementation of the SARC architecture is still not available, we refer to the Cell [10] architecture as a model for comparison. Unisim implementation of the Cell architecture (i.e., CellSim [14]) will be presented and some analogies with the SARC architecture will be explained.
The main contributions of this paper are:

- an introduction to the SARC architecture;
- a description of the tools to be used;
- a methodology to estimate and evaluate power consumption for SARC.

The paper is organized as follows. Section II describes the background on power simulators and the use of these tools during development. Section III briefly introduces some insights about the SARC and its similarities with the Cell architecture. Section IV presents the tools we intent to use in our investigation. Section V exposes our ideas and proposed methodology regarding power estimation. Section VI summarizes the conclusions and gives some directions for future investigations.

II. Related Work

Previous work in the field of power consumption estimation has been carried out through several simu-

lation tools developed by different research institutes, universities and companies. XTREM [4] is a power simulator developed for the Intel XSCALE micro-architecture by Princeton University and Intel research center. It reflects accurately a realistic processor pipeline and it offers the possibility to parametrize the cache and TLB sizes. The validation of the obtained results has been done by direct measurements of the power consumption on a real hardware. The evaluation of the performance has been performed using hardware counter sampling techniques [4]. Infrastructures like Wattch [3] and SimplePower [16], [15] were developed aiming at studying energy and performance efficiency of microprocessors. Wattch uses analytical models to estimate the effective capacitance of functional units. Wattch has been deployed for evaluating power consumption of several high-performance microprocessors such as the Pentium Pro, the MIPS R10K and the Alpha 21264 [4]. SimplePower is an execution-driven, cycle-accurate RT level energy estimation tool and it uses transition sensitive energy models. SimplePower has been developed for evaluating the effect of high-level algorithmic, architectural, and compilation trade-offs on energy [16].

Nowadays, systems such as chip multi-processor (CMPs) and multi-processor system-on-a-chip (MPSoCs) are emerging. Therefore, high-level power analysis tools are needed [6]. Such systems introduce additional interactions among cores using the on-chip network resulting in additional power dissipation. For such issues, tools like LUNA [5] have been developed. The tool we have selected for our investigation (i.e., Unisim) aims at providing a modular infrastructure to evaluate the complex systems such as CMP and MPSoCs. Furthermore, it has the possibility to interface itself with external tools allowing different kind of estimations (e.g., area, access time, etc.). Later on, more detailed introduction to Unisim will be provided. Next section briefly introduces the SARC architecture.

III. SARC - Scalable Computer Architecture

SARC is an integrated project involving several European institutions and companies. It is targeting next generation scalable computer architectures where multiple cores will be responsible for the performance gains. Devising new approaches to reduce power consumption is one of the most challenging aspects of the project. Moreover, scalability (in terms of power dissipation but not limited to) is a key issue to deal with during architectural development. SARC is a multi-node heterogeneous architecture where each node can present different architectural characteristics. A single node can be comprised of multiple cores (e.g., general purpose processor) and application specific accelerators with different peculiarities. For instance, the design of each accelerator can be optimized for different applications (i.e., bio-informatic, scientific, and multimedia). An accelerator can have a local store memory, a private cache or both of them, etc. The general purpose processor and the accelerators within the same node are interconnected through a intra-node bus to the L2 cache. Furthermore, a node in SARC is interconnected via NIC (i.e., network interface) to the NoC which connects all the nodes to the memory interface controller and the I/O subsystem. Figure 1 shows an example of the SARC architecture. SARC node is somehow similar to the Cell

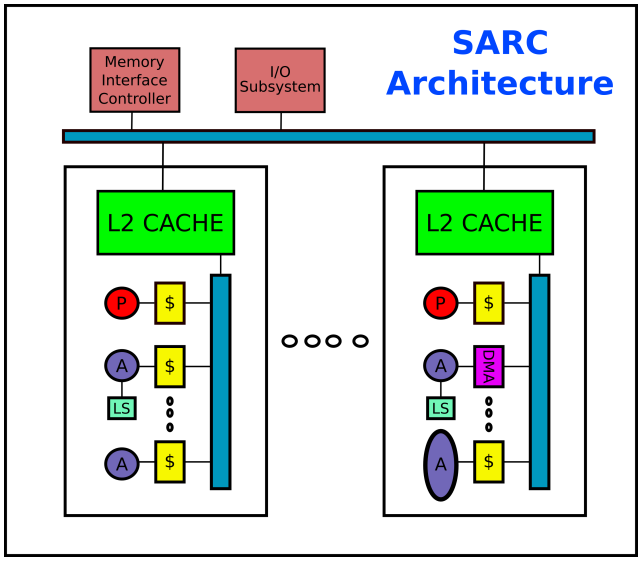

Fig. 1
Overview of the SARC Architecture.

architecture [10] developed by a joint effort of IBM, Toshiba and Sony. Namely, Cell (as SARC) is a chip multi-processor (CMP) system and it has eight SPU "synergistic processor unit" that cooperate together to speed up the execution of computational intensive applications such as multimedia, scientific, and bio-informatic workloads. In SARC, the same functionality is performed by the hardware accelerators. Even though the two architectures are similar, it has to be mentioned that they are not exactly the same but the similarities allow for a preliminary performance comparison. Discussing of detailed architectural differ-

ences between SARC and Cell is out of the scope of this paper.

IV. Tools

This section introduces the tools, Unisim and Cacti, that have been selected to carry out our study. The former is as a structural simulation framework whereas the latter is an integrated cache timing and power model simulator.

A. Introduction to Unisim Simulator

The Unisim environment is composed of a simulation framework and a library of components and it is implemented as a layer on top of SystemC standard. The framework allows easy development of new simulators and it has an engine to run and test them. The library is a set of simulator components and full models that can be used to create a new simulator. Basically, Unisim can be considered a *structural* simulation framework, meaning that the simulator can be decomposed into different modules each of them describing different hardware blocks. Generally, in simulator design the *structural* approach is opposed to the *monolithic* (e.g. SimpleScalar [2]). The former eases a direct mapping between the hardware design (hardware block diagram) into the software simulator (simulator modules). It is well suited for purpose of reusability, modularity and facilitates modification and development of large simulators. In the latter, simulation speed is faster but it is difficult to extract a single component to reuse, share or compare it with others.

Next section introduces the services and the capabilities provided with the unisim simulation framework.

A.1 Services & Capabilities

Unisim comes with a set of *services* and *capabilities* that can be used to collect data from the system they are plugged to. The idea is to have a service connected (plugged) to a module and provide the required functionality whenever requested. A *service* (or a *capability*) in Unisim is a functionality that goes beyond the standard performance evaluation (e.g., number of cycles, etc.). Unisim implements this mechanism using a set of standardized *service APIs* that any simulator module can use as a function call and exploit. For instance, a cache module providing statistics on its activity can get an evaluation of power consumption [1]. Figure 2 shows an example of a simulator with its modules (light) and the relative capabilities (dark) [8].

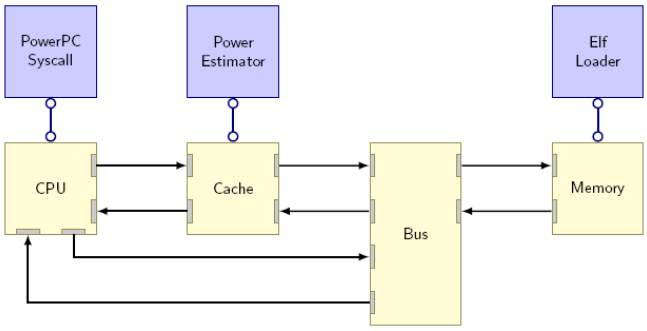

Fig. 2
A simulator with modules (light) and capabilities (dark). Source: https://unisim.org

A.2 Power Estimator Service

The Unisim environment has a specific service for power estimation. This service allows the architecture designer to identify which parts of the system are more power greedy. The system can be divided in functional blocks and each block is built as module. The power estimator tool can be connected to any module (e.g., a cache) and will estimate its power dissipation. Therefore, the most power consuming parts of the chip can be easy identified as primary target for optimization. To evaluate the amount of power consumed by a certain functional block, the power estimator service needs to obtain information from the module it has been connected to. Figure 3 shows an example of the power estimator service connected to a cache module.

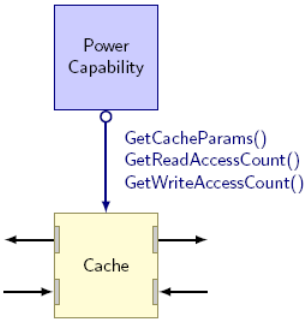

Fig. 3
Power estimator connected to a cache module. Source: https://unisim.org

To calculate the energy dissipated by the cache,

the power estimator needs some parameters from the cache itself. The estimator defines a structure to store these parameters which will be filled in by the cache. The cache module is supposed to implement an interface with the relative function that will be called by the power estimator tool and consequently, the structure will be filled in with the correct parameters. Once the correct parameters are saved in the structure, the power estimator service will be able to calculate the power consumption for that cache module [8]. In Table I a snippet of output relative to the power service is shown. The calculation is based on the number

```
/*********************************/
      Power Service Output
/*********************************/
INPUT PARAMS:
Cache Size = 8192
Cache Line Size = 32
Associativity   = 2
Read/Write Ports= 1
Excl. Read Ports= 0
Excl. WritePorts= 0
Single Ended Read Ports = 0
No. of Banks    = 1
Tech. Node       = 0.07
Output Width     = 8
Specific Tag     = 0
Tag Width        = 0
Access Mode      = 0
Pure SRAM        = 0

Power Service output:
Cache Power Dissipated (W) = 0.0646914
```

TABLE I
POWER SERVICE OUTPUT

of "read" and "write" accesses. The information is maintained by the cache module, therefore it needs to implement an other interface to allow the power service to access these counters. The Power Estimator Capability has been implemented using the Cacti [13] power estimation model developed at HP Labs.

B. Cacti 4.0

Cacti is a micro-architecture level tool widely used to estimate power dissipation in caches. Cacti accepts several input parameters such as "cache size", "block size", "associativity", etc. and outputs the cache configuration that best approximates the desired optimization function [12]. Cacti assumes a certain cache configuration that is possible to tune up setting six organizational parameters. These parameters correspond to the configuration of both data and tag arrays. Cache access time, power dissipation, and area can be calculated depending on: the cache organization parameters, the cache structure (Figure 4) assumed by Cacti, and the input parameters (mentioned above). This task is performed exhaustively calculating every possible cache configuration and selecting the best in respect of the desired optimization function.

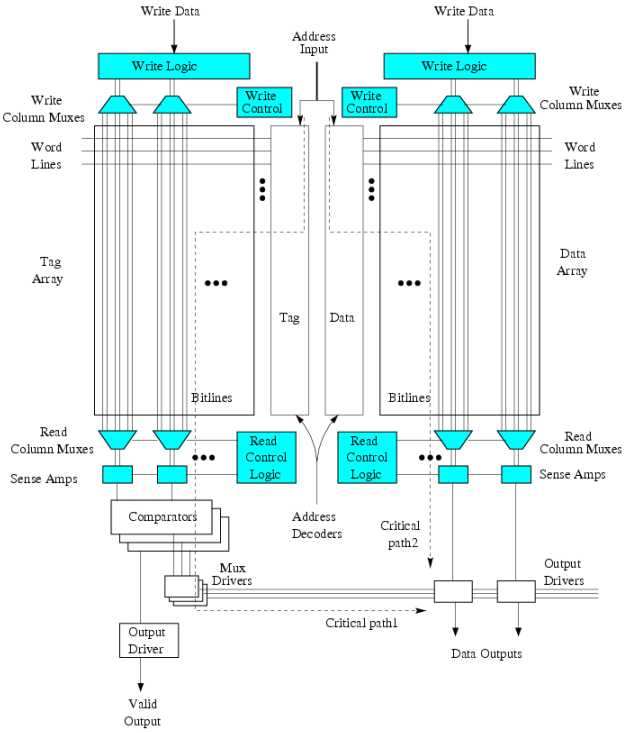

Fig. 4
TYPICAL CACHE STRUCTURE. SOURCE [12]

V. METHODOLOGY

In this section we present our idea regarding the power estimation for the SARC architecture. Due to its similarities with the Cell architecture, we intent to use CellSim (Unisim Cell simulator) to evaluate our methodology and later on repeat the same experiments with SARCsim (Unisim SARC simulator).

CellSim [14] is a source-level compatible modular simulator of the Cell processor architecture implemented using the Unisim framework. It models the Cell architecture almost completely and allows the programmer to run and simulate programs for it. Due to the fact that CellSim has been implemented using the Unisim infrastructure, it is possible to use the same service and capabilities offered by this framework (i.e., area estimator tool, access time estimator, etc.). This is possible because every CellSim component implemented using a separate module has the

same characteristics, i.e., it can inherit a an interface from the service it wants to be connected to. It implements the inherited interface allowing the service to be connected and acquiring data from the module itself. Once the service has the data, the desired estimation can be performed.

Our intention is to set up a simulation scenario using the CellSim, running benchmarks using a representative set of applications such as H.264 for multimedia, Linpack for scientific, and BLAST for bioinformatic. Therefore, collect data relative to power consumption for the memory subsystem of the investigated architecture. The same experiments can be extended connecting more Cell together creating a cluster and studying the power issues of such multi-node architecture. The same set of simulations will be repeated for the multi-node SARC architecture and the obtained results will be compared.

VI. Conclusions

This work has presented the methodology to investigate power related issues and evaluate power consumption for the SARC architecture. Fist, we introduced related work in the field of high-level power estimation simulators. Consequently, a general description of the SARC architecture has been discussed and the analogies with the Cell architecture have been pointed out. Furthermore, the tools (i.e., Unisim and Cacti) proposed to investigate power concerns in SARC have been presented and described. Moreover, in according to our analysis and due to the similarities among SARC and Cell, a comparison in terms of power consumption between the two architectures is worth. Hence, we propose to investigate power consumption for the Cell architecture using CellSim (Unisim implementation of Cell) and when the Unisim SARC model will be available, the set of simulations will be repeated and a comparison between the results performed. Our methodology will allow accurate estimation and evaluation of the power consumption for SARC. The obtained results will allow to track down the power bottlenecks in the architecture and target such parts for further energy optimizations.

Acknowledgments

This was partially supported by the European Commission in the context of the Scalable computer ARChitectures (SARC) integrated project #27648 (FP6).

References

[1] D. August, J. Chang, S. Girbal, D. Gracia-Perez, G. Mouchard, D. Penry, O. Temam, and N. Vachharajani, *UNISIM: An Open Simulation Environment and Library for Complex Architecture Design and Collaborative Development*, September 2007.

[2] T. Austin, E. Larson, and D. Ernst, *SimpleScalar: an infrastructure for computer system modeling*, IEEE Computer **5** (2002), no. 2, 59–67.

[3] D. Brooks, V. Tiwari, and M. Martonosi, *Wattch: A Framework for Architectural-Level Power Analysis and Optimizations*, Proceedings of the 27th International Symposium on Computer Architecture (Vancouver, Canada), June 2000, pp. 83–94.

[4] G. Contreras, M. Martonosi, J. Peng, R. Ju, and G.-Y. Lueh, *XTREM: A Power Simulator for the Intel XScale Core*, Proceedings of the 2004 ACM SIGPLAN/SIGBED conference on Languages, compilers, and tools for embedded systems (Washington, DC, USA), 2004, pp. 115–125.

[5] N. Eisley and L.-S. Peh, *High-Level Power Analysis for On-Chip Networks*, Proceedings of CASES04 (Washington, DC, USA), September 2004.

[6] N. Eisley, V. Soteriou, and L.-S. Peh., *High-Level Power Analysis for Multi-Core Chips*, Proceedings of CASES'06 (Seoul, Korea), 2006.

[7] S. Girbal, *Unisim Module Repository - Cycle-level Signal & Module Documentation*, `https://unisim.org/`.

[8] ———, *UNISIM: UNIted SIMulation environment*, `https://unisim.org/`.

[9] A. Goldstein and S. Weiss, *ePASS-a software-based POWER simulator Goldstein*, Proceedings of the Seventh Israeli Conference on Computer Systems and Software Engineering, June 1996, pp. 46–54.

[10] J. A. Kahle, M. N. Day, H. P. Hofstee, C. R. Johns, T. R. Maeurer, and D. Shippy, *Introduction to the Cell multiprocessor*, IBM Journal Research & Development **49** (2005), 589–604.

[11] S. Lee, S. Das, V. Bertacco, T. Austin, D. Blaauw, and T. Mudge, *Circuit-Aware Architectural Simulation*, Proceedings of the 41st Design Automation Conference (DAC-2004) (San Diego, California), June 2004, pp. 305–310.

[12] M. Mamidipaka and N. Dutt, *eCACTI: An Enhanced Power Estimation Model for On-chip Caches*, Tech. Report #04-28, Center for Embedded Computer Systems, University of California, Irvine, CA 92697, USA, 2004.

[13] G. Reinman and N. P. Jouppi, *Cacti 2.0: An Integrated Cache Timing and Power Model*, Tech. report, HP Labs, 2007.

[14] Alejandro Rico, *CellSim*, URL reference: `http://pcsostres.ac.upc.edu/cellsim/doku.php`.

[15] N. Vijaykrishnan, M. Kandemir, M. J. Irwin, H. S. Kim, and W. Ye, *Energy-driven integrated hardware-software optimizations using SimplePower*, Proceedings of the 27th Annual International Symposium on Computer Architecture (ISCA-2000) (Vancouver, British Columbia, Canada), 2000, pp. 95–106.

[16] W. Ye, N. Vijaykrishnan, M. Kandemir, and M. J. Irwin, *The Design and Use of SimplePower: A Cycle-Accurate Energy Estimation Tool*, Proceedings of the 37th conference on Design automation (DAC-2000), June 2000.