

Power-Performance System-Level Exploration of a MicroSPARC2-based Embedded Architecture

Gianluca Palermo^{†‡} Cristina Silvano[†] Vittorio Zaccaria[†]

[†]Dipartimento di Elettronica e Informazione
Politecnico di Milano, Milano, Italy

[‡]STMicroelectronics
Agrate Brianza, Milano, Italy

Abstract

This paper describes the architectural exploration of the system-level parameters for a MicroSPARC2-based embedded system. The overall goal of the exploration task is to quickly identify the best architecture of the embedded system in terms of both energy and delay parameters, avoiding the comprehensive analysis of the architectural design space. The Energy-Delay Product (EDP) has been adopted as the evaluation metric to compare the alternative architectures in terms of different cache memory and bus subsystems. The exploration phase adopts an iterative local-search algorithm based on the sensitivity analysis of the cost function with respect to the tuning parameters of system architecture. The exploration targets the architectural optimisation of the parameters related to the cache memory and the bus sub-systems of an embedded architecture based on the MicroSPARC2 architecture executing the set of Mediabench benchmarks for multimedia applications. The experimental results have shown a reduction up to nine orders of magnitude of the number of design alternatives analyzed during the exploration phase.

Keywords: Design Space Exploration, Embedded Systems, Low-Power

1 Introduction

Decreasing energy consumption without losing performance is a 'must' during the design of a broad range of embedded systems. The evaluation of energy-delay metrics at the system-level is of fundamental importance during the design of embedded applications characterized by low-power and high-performance requirements. The capability to early provide a direct feed-back on the impact of different design architectures at the system level provides the possibility to early re-target the architectural design choices, thus avoiding a shorter development time and costly re-design cycles.

Once the system-level specification expressing the functionality of the embedded system has been defined and validated, the next design phase consists of the design exploration phase to define the best system architecture, mainly

in terms of core processor, number of levels in the memory hierarchy cache-related parameters, system-level bus topology, width of address and data busses, etc.. To perform the design exploration phase of a target embedded architecture, an approach based on the full search of the optimal architectural parameters at the system-level with respect to the energy-delay cost function can be computationally very costly, due to the long simulation time required to explore the wide space of design parameters.

Aim of our work is to describe the system-level exploration of the architectural design space of a MicroSPARC2 embedded system from an energy/delay comprehensive standpoint. For each point of the design space, we dynamically estimate the value of the corresponding Energy-Delay Product (EDP), taking into consideration both performance and energy constraints. The goal is to find the optimal or near-optimal system configuration without performing the exhaustive analysis of the space of the chosen parameters.

The adopted methodology aims at reducing the complexity of the design exploration phase by grouping the design space parameters into clusters. The basic idea is that the optimal value of the parameters in a single cluster has a very little dependence on the value of the parameters of the other clusters, with respect to the energy-delay cost-function.

In our target architecture, the parameters related to cache memories and system buses have been considered independent during the design exploration phase to determine the optimal system configuration from the energy-delay standpoint. Under this assumption, the design parameters related to the cache memories and the system buses have been divided into two clusters to be optimized separately.

In spite of the application of clustering to reduce the duration of the design exploration phase, the design space is still too large to apply the exhaustive approach to find the best system configuration from the energy-delay joint perspective. To further reduce the simulation time required by the design exploration phase, we apply the heuristic methodology we proposed in [1, 2], that is based on sensitivity-based analysis.

In our exploration framework, the simulation phase is based on an accurate profiling of the processor-to-memory

communication based on the dynamic analysis of the memory accesses and the transition activity of the system-level buses. Bus traces, derived from the execution of several application programs by an Instruction Set Simulator and filtered by a behavioral model of the system modules, are then analyzed in terms of the energy-delay metric (EDP) to evaluate the cost associated with different system configurations. In the computation of the EDP function, the energy consumption has been expressed in Joule Per Instruction (JPI), while the execution delay has been expressed in Clock cycle Per Instruction (CPI).

The experimental results have shown that the proposed methodology can speed-up the design exploration phase of up to nine orders of magnitude with respect to the exhaustive analysis.

The paper is organized as follows. The next section presents the most relevant approaches for system-level exploration appeared in literature so far, while Section 3 proposes our design space exploration framework. The results derived from the application of the proposed methodology to the Micro-SPARC2 case study have been reported in Section 4. Finally, Section 5 summarizes the main contributions of this work and outlines the future directions of our research.

2 Background

Several system-level estimation and exploration methods have been recently proposed in literature targeting power-performance trade-offs from the system-level standpoint. Among these works, the most significant methods related to our approach can be divided into two main categories: (i) system-level power estimation and exploration in general, and (ii) power estimation and exploration focusing on cache memories.

In the first category, the SimplePower approach [3] can be considered one of the first efforts to evaluate the different contributions to the energy budget at the system-level. The *Avalanche* framework presented in [4] evaluates simultaneously the energy-performance tradeoffs of software, memory and hardware for embedded systems. The work in [5] proposes a system-level technique to find low-power high-performance superscalar processors tailored to specific user applications. More recently, the Wattch architectural-level framework has been proposed in [6] to analyze power vs. performance tradeoffs with a good level of accuracy with respect to lower-level estimation approaches. Low-power design optimization techniques for high-performance processors have been investigated in [7] from the architectural and compiler standpoints. A trade-off analysis of power/performance effects of SOC (System-On-Chip) architectures has been recently presented in [8], where the authors propose a simulation-based approach to configure the parameters related to the caches and the buses.

In the second category of approaches dealing with power estimation and exploration for the memory hierarchy, the authors of [9] propose to sacrifice some performance to save power by filtering memory references through a small cache placed close to the processor (namely *filter cache*). A similar idea has been exploited in [10], where memory locations with the highest access frequencies are mapped onto a small, low-energy, and application-specific memory that is placed close to the core processor. Power and performance tradeoffs in cache architectures have been also investigated in [11]. A model to evaluate the power/performance

tradeoffs in cache design has been proposed in [12], where the authors discuss also the effectiveness of novel cache design techniques targeted for low-power (such as vertical and horizontal cache partitioning). An analytical power model for several cache structures has been proposed in [13]. The model accounts for technological parameters (such as capacitances and power supplies) as well as architectural factors (such as block size, set associativity and capacity). The process models are based on measurements reported in [14] for a 0.8 μ m process technology. The analytical model of energy consumption for the memory hierarchy has been extended in [15] and [16], where the cache energy model is included in a more general approach for the exploration of memory parameters for low-power embedded systems.

3 Design Space Exploration of the MicroSPARC II processor

In this paper, we present the results of a power/performance analysis for a MicroSPARC II processor. The main goal is to quickly evaluate the cost of a given number of system alternatives, driving the designer towards an optimal system configuration with respect to a given cost metric.

The current framework receives as input the description of the *design space* and the target application for which a MicroSPARC II optimal configuration (with respect to the exploration metric) must be found. The framework is mainly composed of an iterative optimization procedure (see figure 1) and it is based on the following modules:

- The *system level executable model* represents the simulatable/emulatable description of the target system. Based on the profiling of the application, the module enables the estimation of the actual values of the cost function used for the exploration. The system level executable model that we considered is shown in figure 2. It is mainly composed of two parts: the simulation tool, called MEX (*Memory Explorer*), and the energy and delay models. The MEX module is based on the Sun's Shade library, that allows a fast instruction set simulation of the target application [17].
- The *optimizer* is the module that chooses the most suitable configuration to be explored and when the search must be stopped. To evaluate the next point to visit during the exploration, the optimizer uses the past values obtained by previous searches.
- The *exploration metric* or *cost function* is used to compare the quality of different configurations. The Energy-Delay Product (EDP) has been selected to compare the alternative system configurations in terms of the best trade-off between the energy dissipated by the system and the performances. In the computation of the EDP function, the energy consumption has been expressed in Joule Per Instruction (JPI), while the execution delay has been expressed in Clock cycle Per Instruction (CPI).

3.1 Exploration Methodology

One of the most challenging tasks of the design flow of embedded systems is the exploration phase, that aims at finding the optimal configuration of the design parameters for an

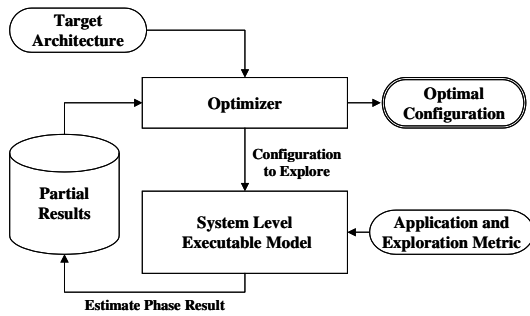


Figure 1: Proposed design space exploration flow

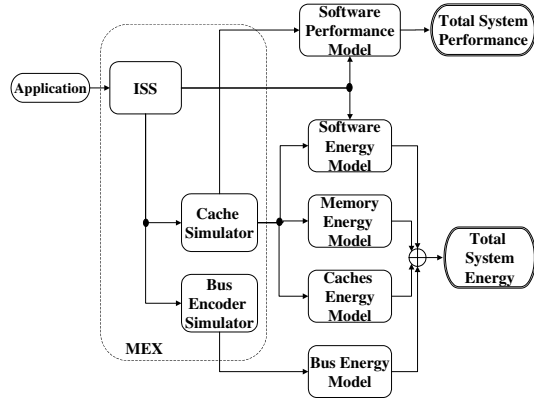


Figure 2: System-level executable model

embedded system. The main problem is that even a simple microprocessor-based architecture has a very large number of possible implementations, making the exhaustive analysis of each configuration practically impossible.

The design space of an architecture takes into account all the possible combinations of the configurable parameters:

$$\mathcal{A} = S_{p_1} \times \dots \times S_{p_l} \times \dots \times S_{p_n}$$

where \mathcal{A} is the design space, S_{p_i} is the set of the possible configurations for parameter p_i and " \times " is the cartesian product. As an example, if we consider a simple architecture with four configurable parameters and only five possible different values for each one, the resulting number of possible configurations to be simulated and analyzed is over 500 for each target application to be considered for the system.

The application of an exhaustive approach to this problem implies the exploration of power and performance values for all the configurations in the space. This solution implies a very large simulation time, due to the large number of configurations to be visited, making this approach impractical. Therefore some heuristic methods must be applied to the problem to obtain an acceptable solution in terms of simulation time while preserving accuracy.

For a simple microprocessor-based embedded system (as those shown in Figure 3), the design space can be considered as composed of the parameters related to the cache and bus sub-systems.

The memory sub-system of the target architecture consists of a multi-level memory hierarchy: L1 On-chip and L2 Off-chip caches and the main memory. Each cache can be organized in several configurations in terms of cache and block

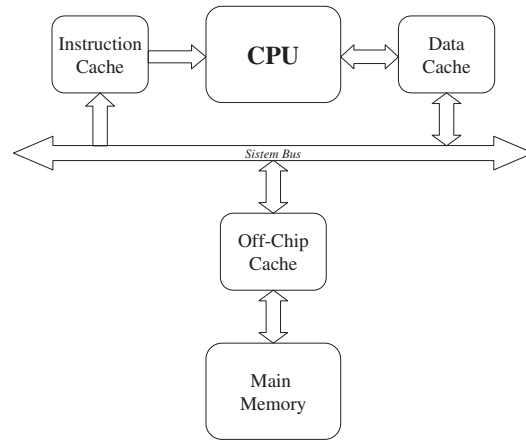


Figure 3: Target system architecture

size, degree of set-associativity, write strategy and replacement policy. Other degrees of freedom for the memory hierarchy configuration are related to the width of data and address buses and their encoding strategy.

The methodology used for the Micro-SPARCII architecture based on the assumption of the parameters independence, as previously noted in [18]. The methodology reduces the complexity of the exploration by grouping the design parameters into clusters. The basic idea is that the optimal value of the parameters in one cluster has a very little dependence on the value of the parameters of the other clusters, when considering the energy-delay cost-function. In part, we assume that cache parameters are independent with respect to the system bus parameters in determining the optimal configuration from the energy-delay standpoint. Thus, cache parameters and system bus parameters has been divided into two clusters to be optimized separately.

In spite of the application of clustering to reduce the duration of the design exploration phase, the design space is still too large to apply the exhaustive approach to find the best system configuration from the energy-delay joint perspective. To further reduce the simulation time required by the design exploration phase, in our exploration we applied the heuristic methodology we proposed in [1, 2], that is based on sensitivity-based analysis.

The *sensitivity analysis* is an exploration methodology based on the idea that the EDP exploration metric is not equally sensitive to the variation of all the design parameters. In fact, after a specific tuning phase, the parameters of the system can be characterized by a degree of influence (or sensitivity) on the cost function and consequently the search can be prioritized on the most sensitive ones. In fact, during the exploration phase, the sensitivity optimizer evaluates the next configuration to visit, changing only one parameter at time, in order of sensitivity.

In this work, we propose a unique analysis for instruction and data caches parameters in order to find a sub-optimal first level configuration in the memory hierarchy, followed by a separated analysis of bus encodings.

4 Case Study

In this section, we present the experimental results obtained by applying our methodology on a real microprocessor-based embedded system, running a set of multimedia applications selected from the Mediabench suite [19].

4.1 TargetSystem Architecture

The target system architecture is composed of the 32-bit MicroSPARC-II [20] high-performance RISC processor core, without D- and I- caches, operating at low voltage to optimize the power consumption. The base architecture has a separate on-chip L1 instruction and data caches, the external memory and the bus encoders/decoders. The L1 instruction and data caches, implemented in CMOS technology with a 1 CPU clock cycle hit-time, are configurable in terms of cache size, block size and degree of associativity, with fixed replacement policy (random) and write strategy (write-back write-allocate). The power model for these caches has been derived from [13].

The external memory is a 32MByte DRAM characterized by a 7 CPU cycle latency with power model derived from [21]. Further details on the selected memory are provided in [22].

4.2 TargetDesign Space

In our work we reduce the design space to L1 data and instruction caches parameters and bus encodings as follows:

- $S_{ics}, S_{dcs} = \{2\text{KB}, 4\text{KB}, 8\text{KB}, 16\text{KB}, 32\text{KB}, 64\text{KB}\}$
- $S_{ibs}, S_{dbs} = \{4\text{B}, 8\text{B}, 16\text{B}, 32\text{B}\}$
- $S_{iv}, S_{dv} = \{1, 2, 4, 8\}$
- $S_{ibe}, S_{dbe}, S_{mbe} = \{\text{Binary}, \text{Gray}, \text{Gray4}, \text{Offset}, \text{OffsetXor}, \text{T0}, \text{T0Xor}\}$

where ics and dcs are the I- and D-cache size, ibs and dbs are the I- and D-block size, iv and dv are the I- and D-associativity and ibe , dbe and mbe are the encodings on the instruction, data and off-chip address busses [23, 24, 25].

As can be noted, the total size of the exploration parameters is of over two order-billion configurations, a design space practically impossible to simulate and to explore comprehensively.

4.3 Exploration of Cache Parameters

The first step of our analysis consists of considering the cache parameters independently from the bus encodings and thus by optimizing them separately.

We chose the *sensitivity analysis* to find the sub-optimal configuration of the target system because this exploration algorithm enables a fast search convergence. In Figure 4 are shown the results of the tuning phase. We noticed how the exploration metric EDP, for this target architecture, is more sensible to the variation of instruction cache parameters, size and associativity. We can see a minor sensitivity value for the data cache size and associativity, and I- and D-cache block size.

The sensitivity values have been used within a sensitivity optimizer algorithm to optimize a sub-set of the Mediabench

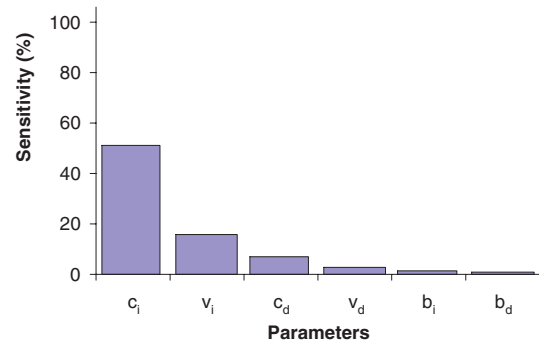


Figure 4: Sensitivity of the EDP cost function with respect to the I-cache and D-cache parameters (size, block size and associativity).

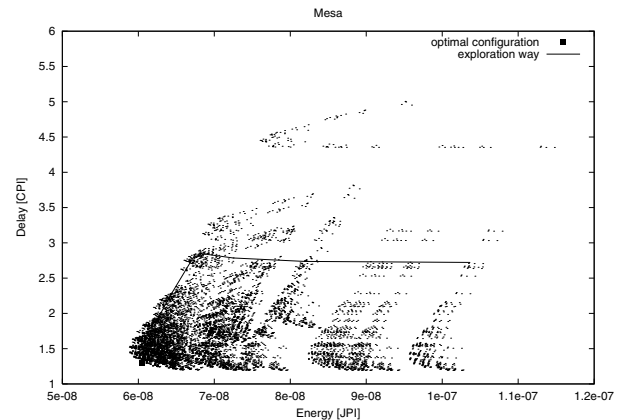


Figure 5: Exploration path of the cache design space for the Mesa benchmark

benchmarks that we use as validation benchmarks for our methodology. In Figure 5 we can see the fast convergence of the algorithm in the cache design space (bold line).

The small number of simulations in the exploration phase is confirmed by Table 2, where we can notice a reduction of over two order in magnitude with respect to the exhaustive search (characterized by $N_{sim} = 9216$) with only a very small average error on the Energy-Delay optimal configuration.

4.4 Exploration of Bus Encoding Techniques

The second step of our methodology consists of the exploration of the bus encoding techniques. The best technique has been found by simulating all the selected bus encoding techniques for each benchmark with the cache configurations found in the previous step. This has been done for each single on-chip and off-chip bus, since we assume that also the system buses are independent among each other in terms of energy and delay. In Table 3 we report the best encoding technique found for each benchmark and the correspond-

Benc hmark	Instruction Cache			Data Cache			Off-Chip Bus	Vs μ S-II Standard		
	Size[B]	Block Size[B]	Way	Size[B]	Block Size[B]	Way	Optimal Encoding	Δ Energy [%]	Δ Delay [%]	Δ EDP [%]
Adpcm Dec	4096	4	4	8192	4	2	Gray4	-14.36	-13.20	-25.66
Adpcm Enc	4096	4	2	8192	4	2	Gray4	-12.74	-11.40	-22.69
Epic	2048	8	2	32768	4	4	T0Xor	-11.82	-19.17	-28.73
Gsm Dec	8192	8	8	16384	32	2	Gray4	-11.71	-13.67	-23.79
Gsm Enc	16384	16	2	8192	16	1	T0Xor	-0.53	-2.02	-2.54
Jpeg Dec	4096	4	4	32768	16	4	Gray4	-25.14	-50.93	-63.27
Jpeg Enc	4096	8	2	65536	8	1	T0Xor	-17.51	-50.37	-59.06
Mesa	16384	16	4	16384	4	4	T0	-8.41	-23.24	-29.70
Mpeg Enc	2048	4	4	4096	4	4	Gray4	-27.23	-47.98	-62.15
Pegwit	8192	16	2	65536	4	1	T0	-33.75	-73.19	-82.24
Unepic	2048	8	2	65536	8	4	T0Xor	-7.53	-17.95	-24.13

Table 1: Results of the proposed methodology to explore the cache and bus encoding parameters

Benc hmark	Error[%]	N _{sim}
Adpcm Dec	0	28
Adpcm Enc	0	30
Epic	0,02	27
Gsm Dec	1,83	29
Gsm Enc	0	26
Jpeg Dec	0	36
Jpeg Enc	0	29
Mesa	0,45	36
Mpeg Enc	0,62	26
Pegwit	0	21
Unepic	0,11	25
Mean	0,28	28,5

Table 2: Results of the cache parameters exploration phase obtained by applying the *sensitivity* analysis for the selected benc hmarks.

ing energy reduction of the off-chip bus subsystem since, for what concerns the on-chip buses, the results have shown that they do not impact significantly the overall power budget.

Benc hmark	Encoding	Ebus _{saved} [%]
Adpcm Dec	Gray4	30,4
Adpcm Enc	Gray4	34,1
Epic	T0Xor	49,9
Gsm Dec	Gray4	34,1
Gsm Enc	T0Xor	54,1
Jpeg Dec	Gray4	35,5
Jpeg Enc	T0Xor	47,7
Mesa	T0	52,0
Mpeg Enc	Gray4	39,2
Pegwit	T0	55,6
Unepic	T0Xor	48,5
Mean		46,3

Table 3: Results of the bus encoding exploration phase in terms of energy savings with respect to the binary encoding for the selected benchmarks.

As can be seen from the table, bus encoding techniques enable an energy reduction of up to 50% on the bus subsystem, with an average energy saving of 46%.

4.5 Final results of the exploration phase

Once the two clusters of parameters have been optimized independently, the estimated optimal system configurations of each cluster are joined together to define the estimated

global optimal configuration. We summarize the results obtained in Table 1, showing at the same time the cache optimal configurations and the optimal encodings for the system bus. The table shows also the average energy and delay savings with respect to the reference configuration of the MicroSPARC-II that is characterized by the following values:

- Instruction cache: 16KB total size, direct mapped, 32B block size.
- Data cache: 8KB total size, direct mapped, 16B block size.

As can be seen, we reach always a system configuration that is better with respect to the reference configuration in terms of both energy and delay. We notice also that the optimal configuration found depends strongly on the data and instruction access pattern of the specific program. For example, the Unepic benchmark shows a high instruction locality with poor data locality, thus requiring a small I-cache and a very big D-cache. On the contrary, other programs (such as the Mpeg Encoder) show a high data locality that is exploitable both in terms of energy by minimizing the size of the caches. Finally, by comparing the total number of simulations that we would have needed for the full search (over twenty-billion) with the sum of the simulations performed by our methodology, we reached a speed-up of up to 9 orders of magnitude.

5 Conclusions

In this paper the design space exploration methodology for the cache and bus subsystems of an embedded system based on the Micro-SparcII processor has been presented. The experimental results have shown that the proposed methodology applied to the selected architecture is able to speed-up the design exploration phase of up to 9 orders of magnitude. As a future direction of our work, we can envision the possibility to further reduce the dimension of the space of parameters based on the results of the sensitivity analysis.

References

- [1] W. Fornaciari, D. Sciuto, C. Silvano, and V. Zaccaria. A design framework to efficiently explore energy-delay tradeoffs. In *Proceedings of CODES-2001: Ninth International Symposium on Hardware/Software Codesign*, pages 260–265, April 25–27 2001.

- [2] W. Fornaciari, D. Sciuto, C. Silvano, and V. Zaccaria. Fast system-level exploration of memory architectures driven by energy-delay metrics. In *Proceedings of ISCAS-2001: International Symposium on Circuits and Systems*, volume IV, pages 502–506, May 6–9 2001.
- [3] N. Vijaykrishnan, M. Kandemir, M.J. Irwin, H.S. Kim, and W. Ye. Energy-driven integrated hardware-software optimizations using simplepower. In *ISCA 2000: 2000 International Symposium on Computer Architecture*, Vancouver BC, Canada, June 2000.
- [4] Y. Li and J. Henkel. A framework for estimating and minimizing energy dissipation of embedded hw/sw systems. In *DA-C35: A CM/IEEE Design Automation Conference*, June 1998.
- [5] T. M. Conte, K. N. Menezes, S. W. Sathaye, and M. C. Toburen. System-level power consumption modeling and tradeoff analysis techniques for superscalar processor design. *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, 8(2):129–137, Apr. 2000.
- [6] David Brooks, Vivek Tiwari, and Margaret Martonosi. Wttch: a framework for architectural-level power analysis and optimizations. In *Proceedings ISCA 2000*, pages 83–94, 2000.
- [7] N. Bellas, I. N. Hajj, D. Polychronopoulos, and G. Stamoulis. Architectural and compiler techniques for energy reduction in high-performance microprocessors. *IEEE Transactions on Very Large Scale of Integration (VLSI) Systems*, 8(3), June 2000.
- [8] Tony D. Givargis, Frank Vahid, and Jörg Henkel. Evaluating power consumption of parameterized cache and bus architectures in system-on-a-chip designs. *IEEE Transactions on Very Large Scale of Integration (VLSI) Systems*, 9(4), August 2001.
- [9] J. K. Kin, M. Gupta, and W. H. Mangione-Smith. Filtering Memory References to Increase Energy Efficiency. *IEEE Trans. on Computers*, 49(1), Jan. 2000.
- [10] L. Benini, A. Macii, E. Macii, and M. Poncino. Increasing energy efficiency of embedded systems by application-specific memory hierarchy generation. *Design and Test of Computers*, 17(2):74–85, April–June 2000.
- [11] R. I. Bahar, G. Albera, and S. Manne. Power and performance tradeoffs using various caching strategies. In *ISLPED-98: A CM/IEEE Int. Symposium on Low Power Electronics and Design*, Monterey, CA, 1998.
- [12] C. L. Su and A. M. Despain. Cache design trade-offs for power and performance optimization: A case study. In *ISLPED-95: A CM/IEEE Int. Symposium on Low Power Electronics and Design*, 1995.
- [13] M. B. Kamble and K. Ghose. Analytical energy dissipation models for low power caches. In *ISLPED-97: ACM/IEEE Int. Symposium on Low Power Electronics and Design*, 1997.
- [14] S. E. Wilton and N. Jouppi. An enhanced access and cycle time model for on-chip caches. Technical Report 93/5, Digital Equipment Corporation Western Research Lab., 1994.
- [15] P. Hicks, M. Walnock, and R. M. Owens. Analysis of power consumption in memory hierarchies. In *ISLPED-97: A CM/IEEE Int. Symposium on Low Power Electronics and Design*, pages 239–242, Monterey, CA, 1997.
- [16] W.-T. Shiue and C. Chakrabarti. Power estimation of system-level buses for microprocessor-based architectures: A case study. In *Proc. DA C99: Design Automation Conference*, New Orleans, LU, 1999.
- [17] Bob Cmelik and David Keppel. Shade: A fast instruction-set simulator for execution profiling. *ACM SIGMETRICS Performance Evaluation Review*, 22(1):128–137, May 1994.
- [18] J. Henkel, T. Givargis, F. Vahid. System-level exploration for pareto-optimal configurations in parameterized systems-on-a-chip. In *ICCAD 2001: IEEE/ACM International Conference on Computer Aided Design*, pages 25–30, 2001.
- [19] C. Lee, M. Potkonjak, and W. H. Mangione-Smith. Mediabench: A tool for evaluating multimedia and communication systems. In *Proceedings of Micro 30*, 1997.
- [20] Sun Microelectronics. Microsparc-ii: Sparc v8 32-bit microprocessor with dram interface, doc. no. stp1012, July 1997.
- [21] Itoh Sasaki Nakagome. Trends in low-power ram circuit technologies. *Proceedings of the IEEE*, 83(4), April 1995.
- [22] NEC. 16m-bit synchronous dram, doc. no. m12939ej3v0ds00. Data Sheet, April 1998.
- [23] Luca Benini, Giovanni de Micheli, Enrico Macii, Massimo Poncino, and Stefano Quer. Power Optimization of Core-Based Systems by Address Bus Encoding. *IEEE Transactions on VLSI Systems*, 6(4):554–562, Dec. 1998.
- [24] M. Stan and W. Burleson. Bus-Invert Coding for Low-Power I/O. *IEEE Trans. on VLSI Systems*, pages 49–58, Mar. 1995.
- [25] W. Fornaciari, M. Polentarutti, D. Sciuto, and C. Silvano. Power optimization of system-level address buses based on software profiling. In *CODES-2000: 8th Int. Workshop on Hardware/Software Co-Design* San Diego, CA, May 2000.