

Sim-PowerCMP: A Detailed Simulator for Energy Consumption Analysis in Future Embedded CMP Architectures

Antonio Flores, Juan L. Aragón and Manuel E. Acacio
Departamento de Ingeniería y Tecnología de Computadores
Facultad de Informática. Campus de Espinardo S/N - 30100. Murcia (Spain)
{aflores, jlaragon, meacacio}@ditec.um.es

Abstract

Continuous improvements in integration scale have made major microprocessor vendors to move to designs that integrate several processor cores on the same chip. Chip-multiprocessors (CMPs) constitute the architecture of choice in the high performance embedded domain for several reasons such as better levels of scalability and performance/energy ratio. On the other hand, higher clock frequencies and increasing transistor density have revealed power dissipation as a critical design issue, especially in embedded systems where reduced energy consumption directly translates into extended battery life. In this work we present Sim-PowerCMP, a detailed architecture-level power-performance simulation tool for CMP architectures that integrates several well-known contemporary simulators (RSIM, HotLeakage and Orion) into a single framework. As a case of use of Sim-PowerCMP, we present a characterization of the energy-efficiency of a CMP for parallel scientific applications, paying special attention to the energy consumed on the interconnect. Results for an 8- and 16-core CMP show that the contribution of the interconnection network to the total power is close to 20%, on average, and that the most consuming messages are replies that carry data (almost 70% of total energy consumed in the interconnect).

1 Introduction

In recent years, high performance processor designs are evolving toward architectures that implement multiple processing cores on a single die, also known as chip-multiprocessors (CMPs). These architectures can provide higher throughput, more scalability and greater energy-efficiency compared to mono-core architectures, and are extensively used in the high performance embedded domain [5]. Furthermore, energy-efficient processor architectures

are currently one of the major goals pursued by designers in embedded computing encouraged by the growing demand of portable systems. Therefore, power dissipation and energy consumption have become two main design concerns in current and future embedded systems. However, unlike the case of mono-core designs where there are some well known and tested design tools able to estimate the power consumption at the architecture level such as *Wattch* [2], in the CMP domain there is a lack of such high level power tools.

In this paper, we propose *Sim-PowerCMP*, a detailed architecture-level power-performance simulation tool that estimates both dynamic and leakage power for CMP architectures based on a Linux x86 port of RSIM [4]. We chose RSIM as performance simulator instead of a full-system simulator such as GEMS/Simics or M5 for several reasons. First, RSIM models the memory hierarchy and the interconnection network in more detail. Second, in the embedded domain, where scientific and multimedia workloads are mostly executed, the influence of the operating system is negligible and it can be ignored. And third, full-system simulators are progressively slower as the number of processors in a CMP increases. The latter is important since the number of processor cores is expected to grow.

Due to difficulty of validating our own power models, *Sim-PowerCMP* incorporates already proposed and validated power models for both dynamic power (from *Wattch* [2]) and leakage power (from *HotLeakage* [10]) of each processing core, as well as the interconnection network (from *Orion* [9]). However, those power models have to be adapted to the peculiarities of CMP architectures. As an example of application of the proposed simulator we also present an evaluation and a characterization of the energy-efficiency of an 8- and 16-core CMP while executing parallel scientific applications. Experimental results show that the internal interconnection network is responsible for about 20% of the overall energy consumption. Most of this energy is spent in the interconnect links due to packet traffic activity. Similar results were previously reported in [7, 8] and

also serve to validate our tool.

The rest of this paper is organized as follows. Section 2 reviews some related work. Section 3 presents the architecture of the proposed CMP power-performance simulator. Section 4 describes and validates the different power models implemented on *Sim-PowerCMP*. The characterization of the energy-efficiency of a CMP is presented in Section 5. Finally, Section 6 summarizes the main conclusions of the work.

2 Related Work

A large body of research has been recently targeted at understanding the performance, energy and thermal efficiency of different CMP organizations in the embedded domain. Concerns about the increasing energy consumption and thermal constrains in modern high performance embedded processors have resulted in the development of architecture-level simulation tools that provide power and energy measurements as well as thermal spatial distribution maps.

In [2] Brooks *et al.* introduce *Wattch*, a dynamic power-performance simulator based on *SimpleScalar* [1], that implements dynamic power models for the different structures in a superscalar processor. This simulator was validated with published power numbers for several commercial microprocessors and it has been largely used by the research and academic community in the last years. *HotLeakage* [10] is another simulation tool that extends *Wattch* by adding leakage power models for some processor regular structures (caches and register files) allowing for a more detailed power estimation. In [3] Chen *et al.* present *SimWattch*, a full system energy simulator based on *Simics* (a system level simulator tool) and *Wattch*.

In addition, it can be also found in the literature a large body of research about power efficiency in interconnection networks. The first power model of routers and links in interconnection networks for multiprocessors was introduced by Patel *et al.* [6], where the authors propose a detailed power model based on network topology, transistor count and process technology. In [9] the authors presented *Orion*, an architecture-level power simulator for interconnection networks based on *Wattch*.

Finally, several simulation tools can be currently employed for characterizing the performance of CMP architectures, such as GEMS and M5 full-system simulators or RSIM, but none of them takes power or energy consumption into consideration. Furthermore, RSIM was originally designed for detailed simulation of cc-NUMA multiprocessors, and several changes must be applied to model a CMP.

3 Sim-PowerCMP architecture overview

Sim-PowerCMP is a power-performance simulator derived from a Linux port of RSIM [4]. It models a CMP architecture consisting of arrays of replicated *tiles* connected over a switched network. Each tile contains a superscalar processing core with primary caches (both instruction and data caches), a slice of the L2 cache, and a connection to the on-chip network. The L2 cache is shared among the different processing cores, but it is physically distributed between them. Therefore, some accesses to the L2 cache will be sent to the local slice while the rest will be serviced by remote slices (L2 NUCA architecture). In addition, the L2 cache stores (in its tags' part) the directory information needed to ensure coherence between the L1 caches. The processing cores are connected with a 2D-mesh interconnection network as depicted in Figure 1 (top). These tiled CMPs scale well to larger processor counts and they can easily support families of products with varying numbers of tiles and, thus, performance.

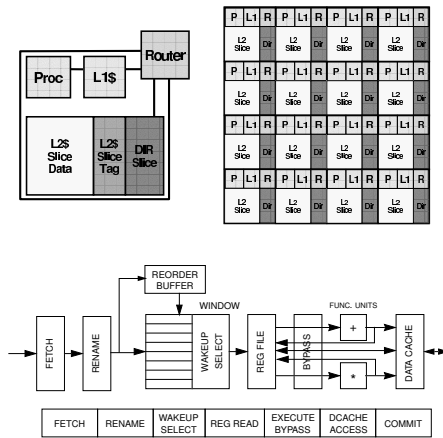


Figure 1. Sim-PowerCMP architecture overview (top); and pipeline organization of each core (bottom).

Each processing core is an out-of-order multiple issue processor (although in-order issue is also supported), modeled according to the pipeline organization shown in Figure 1 (bottom). The MIPS R10000 processor, in which RSIM simulator is based (note that *Sim-PowerCMP* is derived from RSIM) and the Alpha 21264 processor are two examples of this architectural model. On the other hand, *Wattch* and *HotLeakage* simulators, that will be used for validating *Sim-PowerCMP* in Section 4, are based on the RUU architectural model proposed by Sohi. The RUU is a big structure that unifies the instruction window (IW) and the reorder buffer (ROB), at the same time it acts as a phys-

ical register file that temporarily stores the results of non-committed instructions.

There are two major differences between both models. The first one is that in the architectural model of Figure 1 (bottom), all computed values, speculative or not, are stored in the register file. However, in Sohi's model, the RUU is responsible for temporary storing the non-committed output values while a separate register file is responsible for storing the output values of committed instructions. The second major difference is that, in *Sim-PowerCMP* architecture model, computed values are not sent to the IW, only the tags are sent for the tagmatch (or wake-up) process. Computed values are sent to the register file. However, in Sohi's model (used by *HotLeakage*), all computed values are sent to the RUU and, therefore, dependent instructions get their inputs from the RUU. These two architecture differences must be taken into account when validating the proposed power model, as we will show in next section.

Adapting the power models of the main hardware structures of each core and the interconnection network to *SimPower-CMP* is not a trivial task. The power modeling infrastructure used in *Wattch* and *HotLeakage* is strongly coupled with the performance simulator code and, although they are mostly parametrized power models, considerable effort and deep understanding of the simulator implementation are needed in order to port the power model infrastructure to *SimPower-CMP*. One major hurdle was the extensive use of global variables to keep track of which units are accessed each cycle in order to account for the total energy consumed for an application. In a power-aware CMP simulator, these counters and statistics must be collected on a per-core basis, and the use of global activity counters is forbidden. On the other hand, the interconnection network power model used in *Orion* is loosely coupled with the Liberty infrastructure, making its integration with another performance simulator easier. However, some additional changes were needed in order to make it fully inter-operative with *SimPower-CMP*.

Finally, we either changed some power models or derived new ones to match several of the particularities of the CMP implemented in *SimPower-CMP*. For instance, we needed to model the impact of the directory. In the CMP architecture modeled in *SimPower-CMP*, the L2 cache stores the directory information needed to ensure coherence between the L1 caches. So we changed the power model of the L2 cache to account for both the extra storage bits as well as the extra accesses to the directory.

4 Validation of the Power Model of *Sim-PowerCMP*

Validating power models is a crucial task to obtain reasonably accurate simulation results. We have used a vali-

ation methodology based on checking our results against those obtained under the same configuration with other power simulators that have already been validated and are widely used by the research community: *HotLeakage* in the case of processor cores and *Orion* for the power modeling of the internal interconnection network.

Core Configuration		
Parameter	HotLeakage	Sim-PowerCMP
Fetch/Issue/Commit width		4
Active List	—	64
Instr. window (RUU)	32	—
Register File	32	64
Functional Units	2 IntALU, 2 FPALU 2 AddrGen, 2 mem ports	
LSQ Entries	64	
L1 I/D-Cache	32K, 4-way	
L2 Cache	256K, 4-way, 10+20 cycles	
Memory	400 cycles	
Branch Pred.	two-level, 4 K-entries	
BTB	4 K-entries	
CMP Parameters		
Technology	70 nm	
Die size	400 mm ²	
Core size	40 mm ²	
Number of cores	8	
Interconnection network	2D mesh	
Router Parameters		
Link length	5 mm	
Flit size	75 Bytes	
Buffer size	64 flits	

Table 1. Configuration of the baseline CMP architecture.

Table 1 shows the configuration used across this paper. It describes an 8-core CMP built in 70 nm technology. The total die area has been fixed to 400 mm² with a core area of 40 mm², including a portion of the second-level cache. With this configuration, links that interconnect routers configuring the 2D mesh topology would measure around one third of the die length. That is, about 5 mm.

	HotLeakage (W)		Sim-PowerCMP (W)	
Total Dynamic Power Consumption:	19.36	19.46		
Branch Predictor Power Consumption:	0.82	4.72%	0.82	4.69%
Rename Logic Power Consumption:	0.08	0.49%	0.09	0.54%
Instruction Decode Power (W):	0.0040		0.0040	
RAT decode _{power} (W):	0.0316		0.0316	
RAT wordline _{power} (W):	0.0085		0.0097	
RAT bitline _{power} (W):	0.0386		0.0463	
DCL Comparators (W):	0.0023		0.0023	
Instruction Window Power Consumption:	0.52	3.01%	0.07	0.39%
tagdrive (W):	0.0354		0.0425	
tagmatch (W):	0.0169		0.0198	
Selection Logic (W):	0.0068		0.0067	
decode _{power} (W):	0.0316		0	
wordline _{power} (W):	0.0205		0	
bitline _{power} (W):	0.4123		0	
Load/Store Queue Power Consumption:	0.64	3.69%	0.64	3.67%
Arch. Register File Power Consumption:	0.46	2.68%	0.82	4.68%
decode _{power} (W):	0.0316		0.0653	
wordline _{power} (W):	0.0205		0.0205	
bitline _{power} (W):	0.4123		0.7308	
Result Bus Power Consumption:	0.77	4.44%	1.02	5.85%
Total Clock Power:	7.30	42.07%	7.24	41.49%
Int ALU Power:	1.55	8.91%	1.55	8.87%
FP ALU Power:	2.37	13.66%	2.37	13.58%
Instruction Cache Power Consumption:	0.67	3.88%	0.67	3.86%
Itlb _{power} (W):	0.05	0.29%	0.05	0.28%
Data Cache Power Consumption:	1.35	7.77%	1.35	7.72%
Dtlb _{power} (W):	0.17	0.97%	0.18	0.96%
Level 2 Cache Power Consumption:	0.59	3.43%	0.59	3.41%
Ambient Power Consumption:	2.00	10.33%	2.00	10.28%

Table 2. Dynamic power breakdown for the different structures in a processor core.

Table 2 shows an *a priori* comparison of the maximum dynamic power breakdown for a core of the CMP using *Sim-PowerCMP* and *HotLeakage*. Note that there are some differences for structures such as the rename logic, the register file, and mainly, the IW. These differences are due to the different superscalar architectures implemented in both simulators, as mentioned in the previous section. *HotLeakage* simulator implements the RUU model that integrates the IW, ROB, and physical registers in the same hardware structure. So, the power consumption of the IW (RUU for *HotLeakage*) is quite high. On the other hand, the model used in *SimPower-CMP* requires to duplicate the size of the register file because it keeps both speculative and non-speculative result values (logical and physical registers). This explains the higher power consumption in the register file (as shown in Table 2). The higher power consumption in the rename logic is due to the fact that physical register tags have one additional bit in *Sim-PowerCMP* because we double the size of the register file.

	HotLeakage	Sim-PowerCMP
Total Static Power Consumption (W):	0,23812	0,24665
Arch. Register File Power Consumption: (W)	0,00449	0,00898
Instruction Cache Power Consumption: (W)	0,02397	0,02397
Data Cache Power Consumption: (W)	0,02397	0,02397
Level 2 Cache Power Consumption: (W)	0,18974	0,18974

Table 3. Static power consumption for regular structures of a processor core.

Table 3 shows the static power consumption for the main regular hardware structures in a core. The only difference is found in the register file since this structure doubles its size in *Sim-PowerCMP*.

After this *a priori* analysis of the maximum power consumption for a core, the next step in the validation of the power model is to compare the dynamic power consumption when real programs are simulated. However, it is important to note that both simulators use different instruction set architectures (SPARC and PISA ISAs for *Sim-PowerCMP* and *Hotleakage*, respectively) which complicates the comparison. Furthermore, the use of different compilers as well as slightly different optimization flags can appreciably change the instruction mix for a program. Therefore, we decided to use a two-step validation strategy, first using very simple test programs in order to obtain a preliminary validation. Then, we performed the final validation using some of the SPEC2000 applications.

The test programs used for the preliminary power model validation were two minitests written in C and compiled using PISA (*HotLeakage* simulator) and SPARC (*Sim-PowerCMP* simulator) versions of the gcc compiler. The optimization options activated in both cases were `-O3 -funroll-loops`. The first test performs a sequential access to an array of integers, accumulating all the val-

	Test 1		Test 2	
	HotLeakage	PowerCMP	HotLeakage	PowerCMP
Arithmetic-logical	54,10%	54,17%	55,58%	61,92%
Data transfer	41,71%	41,67%	38,86%	33,31%
Unconditional jump	0%	0%	0%	0%
Conditional branch	4,18%	4,17%	4,56%	4,77%

Table 4. Percentage of instructions committed in both test programs.

ues into a global variable, whereas the second one implements the multiplication of two matrices of doubles. Table 4 shows the percentage of instructions that are committed in both cases. Even with these simple codes and using the same compiler with the same optimization options, the obtained instruction percentages are not exactly the same, but are very similar.

	Test 1				Test 2			
	Avg. access/c		Avg. power (W)		Avg. access/c		Avg. power (W)	
	HL	SPcmp	HL	SPcmp	HL	SPcmp	HL	SPcmp
Rename Table	2,40	2,30	0,05	0,05	2,86	2,57	0,06	0,06
Branch prediction	0,10	0,10	0,11	0,11	0,16	0,14	0,13	0,13
Instruction window	9,19	4,70	0,52	0,04	10,59	5,57	0,55	0,05
LSQ	1,00	1,00	0,28	0,28	1,27	1,28	0,26	0,25
Register file	3,50	5,80	0,13	0,38	4,25	7,14	0,16	0,46
L1 i-cache	2,40	2,40	0,72	0,72	2,86	3,00	0,70	0,72
L1 d-cache	1,00	1,00	0,79	0,80	0,84	0,86	0,73	0,69
Int + FP ALU	2,40	2,40	1,17	1,17	2,85	1,15	1,73	1,72
Result bus	3,30	2,30	0,64	0,58	3,49	2,57	0,64	0,63
Clock			2,51	2,84			3,24	3,44
Fetch stage			0,84	0,84			0,86	0,85
Dispatch stage			0,05	0,05			0,06	0,06
Issue stage			3,47	2,94			3,93	3,39
Avg. power/cycle			7,24	7,08			8,25	8,21
Avg. power/instr.			3,40	3,42			3,49	3,31
Max power/cycle			9'63	9,26			12,12	11,49

Table 5. Dynamic power consumption for a core after simulating both minitests.

Table 5 shows the results obtained after completing the simulation for both minitests assuming perfect caches. It can be observed that results are almost identical, except for the register file and the instruction window. These differences are related with the particular microarchitecture modeled by each simulator, as explained before.

The second step of our power model validation methodology consisted in comparing the results obtained after running a subset of the SPEC2000. In these simulations we still assume perfect L1 caches in order to avoid interferences due to the different implementations of the memory hierarchy in both simulators. Figure 2 shows the dynamic power as well as the IPC for each application. In general, we obtain the same power distribution among the different hardware structures of a core, although there are some differences that are worth to explain.

Firstly, we observe higher power dissipation in a core for *HotLeakage*, due to the fact that for the same applications the IPC obtained in this simulator is usually higher. The higher the IPC, the higher the number of accesses to the different hardware structures that are modeled. This leads to an increase in the dynamic power of these structures. For

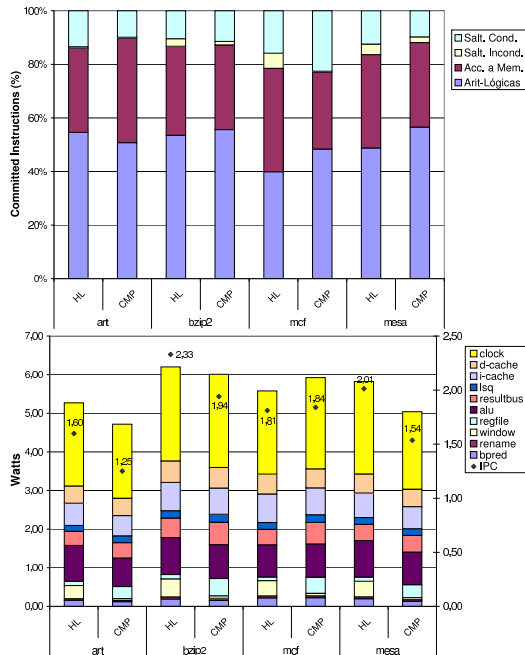


Figure 2. Classification of committed instructions (top); and comparison of dynamic power for the SPEC2000 on a single core of the CMP (bottom).

the *mcf* application, where the IPC obtained in both simulators is similar, power dissipation is very close. Finally, if we analyze the power distribution for *Sim-PowerCMP*, we can appreciate a considerable drop in the power dissipated by the instruction window, partially compensated by the higher power in the register file. The reason to this global dynamic power drop is the different architectural cores modeled in each simulator.

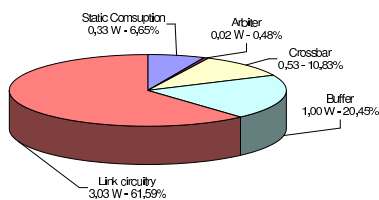


Figure 3. Distribution of the power consumption inside a router.

Once we finished the validation of the power model associated with each core of the CMP, the next step was to validate the power model of the interconnection network. Figure 3 shows the distribution of the power consumption for routers that implement the 2D-mesh. For our modeled

routers, 62% of the total power comes from the link circuitry. This value is similar to the 60% dissipated by links in the Alpha 21364 routers and a little lower than the 82% reported in [7]. The power dissipated by the links strongly depends on the amount of buffer space assigned to the router compared with channel bandwidth. Our results also agree with the results reported in [8], with a maximum power consumption of 2-3 W per router inside a CMP (excluding link circuitry). With this power data, the interconnection network takes about 20% of the total CMP power budget, as published in different works [7, 8].

5 Characterization of the energy-efficiency of a CMP for parallel scientific applications

As an example of application of *Sim-PowerCMP*, we present in this section a characterization of the energy-efficiency of an 8- and 16-core CMP executing parallel scientific applications. The configuration of the simulated CMP architecture is shown in Table 1.

Application	Problem size
Barnes-Hut	16K bodies, 4 timesteps
FFT	256K complex doubles
LU-cont	256 × 256, B=8
LU-noncont	256 × 256, B=8
MP3D	50000 nodes, 2 timesteps
Ocean-cont	258 × 258 grid
Ocean-noncont	258 × 258 grid
Radix	2M keys
Unstructured	mesh.2K, 5 timesteps
Water-nsq	512 molecules, 4 timesteps

Table 6. Applications and problem sizes used in this characterization.

Table 6 shows the applications used in our experiments from the SPLASH and SPLASH-2 benchmark suites. The problem sizes have been chosen commensurate with the size of L1 caches and the number of cores. All experimental results reported in this work are for the parallel phase of these applications. Data placement in our programs is either done explicitly by the programmer or by RSIM which uses a first-touch policy on a cache-line granularity. Thus, initial data-placement is quite effective in terms of reducing traffic in the interconnection network.

Figure 4 presents a breakdown of the power dissipated in an 8- and 16-core CMP. Total power dissipation is split among the most important structures of the CMP (for the sake of legibility we have omitted the contribution of the clock). As expected, it can be observed that most of the power is dissipated in the processor cores of the CMP. In particular, the ALU reveals as one of the most consuming structures. Regarding the caches (private L1 caches and the shared multibanked L2 cache) we can see that their fraction of the total power is quite significant. Additionally, we see that most of the power is dissipated in the L1 I- and

D-caches. Figure 4 shows that the contribution of the interconnection network to the total CMP power is close to 20%, on average, with several applications reaching up to 30%. In this case, we have found that most of this power is dissipated in the point-to-point links used to configure the interconnect and, therefore, message size plays a major role. In particular, reply messages, which are 75-byte long, are the most power consuming ones (almost 70% on average of the energy consumed in the interconnect) although they represent 30% of the total number of messages.

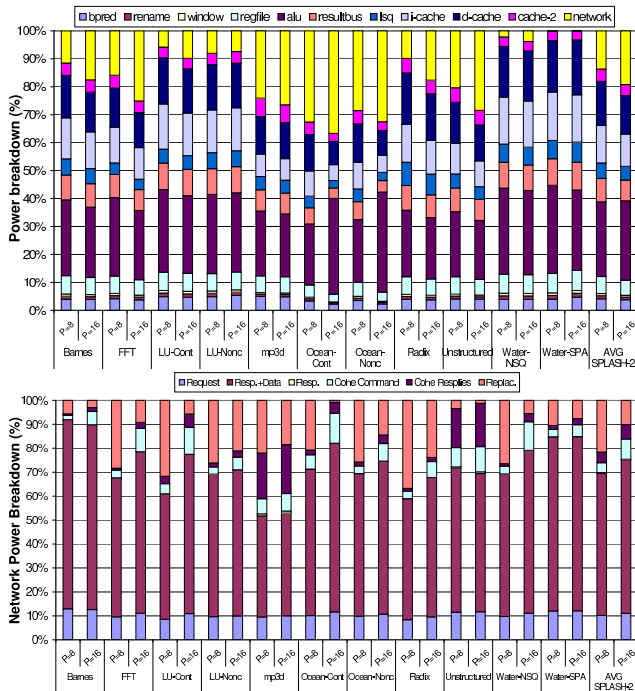


Figure 4. Overall CMP power dissipation breakdown (top); and percentage of the power dissipated in the interconnection by each type of message (bottom).

6 Conclusions and Future Work

In this work we present *Sim-PowerCMP*, a detailed power-performance simulation tool for CMP architectures that allows precise analysis of power and energy consumption (both dynamic and static) taking into account performance. Through experimentation we demonstrate that the power models used in *Sim-PowerCMP* give results that are comparable with those found in simulators currently used for characterizing the energy consumption of processor cores and interconnection networks individually. Additionally, as a case of use of *Sim-PowerCMP*, we present a

characterization of the energy-efficiency of a CMP executing several parallel scientific applications. Results for an 8- and 16-core CMP show that the contribution of the interconnection network to the total CMP power is close to 20% on average and that the most consuming messages are the replies that carry data (almost 70% of overall energy consumed in the interconnect).

As part of our future work, we plan to develop new techniques aimed at reducing the energy consumed by reply messages. Our proposal is based on the observation that when a load or store misses at the L1 cache, not all the memory block is needed to allow the load or store to proceed, just the requested word. In this way, dynamically adjusting the size of the memory blocks would lead to reductions in energy consumption without degrading performance. Additionally, we plan to extend the functionality of *Sim-PowerCMP* with thermal models.

References

- [1] T. Austin, E. Larson, and D. Ernst. SimpleScalar: An Infrastructure for Computer System Modeling. *Computer*, 35(2):59–67, 2002.
- [2] D. Brooks, V. Tiwari, and M. Martonosi. Wattch: a framework for architectural-level power analysis and optimizations. In *Proc. of the 27th Int'l Symp. on Computer architecture (ISCA-27)*, pages 83–94, 2000.
- [3] J. Chen, M. Dubois, and P. Stenström. Integrating Complete-System and User-level Performance/Power Simulators: The SimWattch Approach. In *Proc. of the 2003 IEEE Int'l Symp. on Performance Analysis of Systems and Software*, 2003.
- [4] C. J. Hughes, V. S. Pai, P. Ranganathan, and S. V. Adve. RSIM: Simulating Shared-Memory Multiprocessors with ILP Processors. *IEEE Computer*, 35(2):40–49, 2002.
- [5] A. Jerraya and W. Wolf. *Multiprocessor Systems-on-Chips*. Morgan Kaufman Publishers, Inc., 2004.
- [6] C. S. Patel. Power constrained design of multiprocessor interconnection networks. In *Proc. of the 1997 Int'l Conf. on Computer Design (ICCD '97)*, pages 408–416, 1997.
- [7] L. Shang, L. Peh, and N. Jha. Dynamic voltage scaling with links for power optimization of interconnection networks. In *Proc. of the 9th Int'l Symp. on High-Performance Computer Architecture (HPCA-9)*, pages 91–102, 2003.
- [8] H.-S. Wang, L.-S. Peh, and S. Malik. A Power Model for Routers: Modeling Alpha 21364 and InfiniBand Routers. *IEEE Micro*, 23(1):26–35, 2003.
- [9] H.-S. Wang, X. Zhu, L.-S. Peh, and S. Malik. Orion: a power-performance simulator for interconnection networks. In *Proc. of the 35th Int'l Symp. on Microarchitecture (MICRO-35)*, pages 294–305, 2002.
- [10] Y. Zhang, D. Parikh, K. Sankaranarayanan, K. Skadron, and M. Stan. HotLeakage: A Temperature-Aware Model of Subthreshold and Gate Leakage for Architects. Technical report, University of Virginia, 2003.