Power/Performance/Thermal Design-Space Exploration for Multicore Architectures

Matteo Monchiero, *Member*, *IEEE*, Ramon Canal, *Member*, *IEEE*, and Antonio González, *Member*, *IEEE*

Abstract—Multicore architectures have been ruling the recent microprocessor design trend. This is due to different reasons: better performance, thread-level parallelism bounds in modern applications, ILP diminishing returns, better thermal/power scaling (many small cores dissipate less than a large and complex one), and the ease and reuse of design. This paper presents a thorough evaluation of multicore architectures. The architecture that we target is composed of a configurable number of cores, a memory hierarchy consisting of private L1, shared/private L2, and a shared bus interconnect. We consider a benchmark set composed of several parallel shared memory applications. We explore the design space related to the number of cores, L2 cache size, and processor complexity, showing the behavior of the different configurations/applications with respect to performance, energy consumption, and temperature. Design trade-offs are analyzed, stressing the interdependency of the metrics and design factors. In particular, we evaluate several chip floorplans. Their power/thermal characteristics are analyzed, showing the importance of considering thermal effects at the architectural level to achieve the best design choice.

Index Terms—Chip multiprocessor, design-space exploration, thermal-aware microarchitectures, power/performance.

1 Introduction

Main semiconductor companies have recently proposed microprocessor solutions composed of a few cores integrated on a single chip [1], [2], [3]. This approach, named Chip Multiprocessor (CMP), permits one to efficiently deal with power/thermal issues dominating deep-submicron technologies and makes it easy to exploit thread-level parallelism of modern applications.

Power has been recognized as a first-class design constraint [4], and many literature work targets analysis and optimization of power consumption. Issues related to chip thermal behavior have been addressed only recently, but this emerged as one of the most important factors to determine a wide range of architectural decisions [5]. Temperature considerations themselves are one of the main reasons that determined the shift toward multicore architectures. These systems indeed feature more even power density and do not show dramatic temperature peaks, as can be for complex single-processor designs.

High temperatures result in increased static power consumption and transient fault rate [6], [7], affecting the normal execution characteristics of a processor. These situations, known as thermal emergencies, must be handled by specific Dynamic Thermal Management (DTM) techniques, which typically consist of reducing operating voltages or throttling the processor speed, thus harming the processor performance.

To quantitatively illustrate the impact of temperature on leakage energy, we report some numbers obtained from PTM BSIM4 models [8] (see Table 1), showing the increase in leakage current for 65-nm, 45-nm, and 32-nm technology nodes and the sensitivity of the leakage to the temperature at 60, 80, and 100 °C. Although reliability is not the topic in this paper, we furthermore recall some results presented by Srinivasan et al. [6]. They state that temperature variations of 5 °C at 70-75°C correspond to approximately 50 percent more Failures in Time (FITs).

The impact of static power dissipation is increasing in future technologies (see Table 1), approximately 30 percent more per technology node. This makes leakage-aware designs especially important. In addition, future chips are forecast to be much more complex and dense, leading to higher global temperatures and reliability problems. Temperature dependence of the leakage, assumed to be around 2 percent in this paper, is also forecast to be more important in the future, since this is bound to the dramatic degradation of the subthreshold slope [9], [10] foreseen for deep-nanometer-era technologies.

For these reasons, thermal-aware design is important, allowing for early evaluating the possible thermal profile of the chip. Nevertheless, architecting the components of a CMP can be nontrivial if we try to maximize multiple design objectives, like low-energy consumption and high-performance, under temperature/power density constraints. The main goal of this paper is to provide a framework for the evaluation of multicore architectures, taking temperature and power interactions into account.

Our target CMP consists of multiple cores of different complexities. Each processor owns private instruction and data L1 caches, and it has either a private or a shared L2 cache. The private-L2 approach has been proposed for some industrial products so far, for example, the Intel

- *M. Monchiero is with HP Labs, 1501 Page Mill Road, Palo Alto, CA 94304-1100. E-mail: matteo.monchiero@hp.com.*
- *R. Canal is with the Department of Computer Architecture, Universitat Politècnica de Catalunya, Cr. Jordi Girona, 13, 08034 Barcelona, Spain. E-mail: rcanal@ac.upc.edu.*
- *A. González is with the Intel Barcelona Research Center, Intel Labs, Universitat Politècnica de Catalunya, c/ Jordi Girona 29, Edifici Nexus II, 3^a.planta, 08034 Barcelona, Spain. E-mail: antonio.gonzalez@intel.com.*

Manuscript received 31 Aug. 2006; revised 8 Mar. 2007; accepted 25 June 2007; published online 2 Aug. 2007.
Recommended for acceptance by R. Iyer.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-0253-0806.
Digital Object Identifier no. 10.1109/TPDS.2007.70756.

1045-9219/08/$25.00 © 2008 IEEE Published by the IEEE Computer Society

TABLE 1
Leakage Current and Thermal Variations for Future Technologies

	Technology node		
	65nm	45nm	32nm
I_{leak} [$nA/\mu m$]	70	100	150
$\frac{\partial I_{leak}}{\partial T}/I_{leak}$ @60 °C [K^{-1}]	2.20%	2.25%	2.30%
$\frac{\partial I_{leak}}{\partial T}/I_{leak}$ @80 °C [K^{-1}]	2.00%	2.05%	2.10%
$\frac{\partial I_{leak}}{\partial T}/I_{leak}$ @100 °C [K^{-1}]	1.80%	1.85%	1.90%

Montecito [1] and Pentium D [11], the AMD Opteron [12], and the recently announced IBM Power6 [13]. It maximizes design reuse, since this architectural style does not require the redesign of the secondary cache, as it would be for an L2 shared architecture. On the other hand, sharing the L2 cache (for example, the IBM Power5 [3], Sun Niagara [2], and Intel Core Duo [14]), typically offers a lower miss rate and potentially better performance. Nevertheless, the designs of the on-chip interconnect and of the L2 cache itself are more complex in this last scenario.

Unlike much recent work about the design-space explorations of CMPs, we consider parallel shared memory applications, which can be considered the natural workload for a small-scale multiprocessor. We target several scientific programs and multimedia ones. Scientific applications represent the traditional benchmark for multiprocessor, whereas multimedia programs represent a promising way of improving parallelism in everyday computing.

Our experimental framework consists of a detailed microarchitectural simulator [15], integrated with Wattch [16] and CACTI [17] power models. Thermal effects have been modeled inside each functional unit by using HotSpot [5]. We account for leakage energy by integrating an accurate temperature-dependent model [18]. This environment makes a fast and accurate exploration of the target design space possible.

Our main contribution is the analysis of several design energy/performance trade-offs when varying the core complexity, L2 cache size, and number of cores for parallel applications. In particular, we discuss the interdependence of energy/thermal efficiency, performance, and architectural-level chip floorplan. This paper extends our previous work [19] by considering also shared-L2 CMPs, which have not been discussed in [19].

This paper is organized as follows: Section 2 presents the related work. The target design space and the description of the considered architecture are presented in Section 3. The experimental framework used in this paper is introduced in Section 4. Section 5 discusses performance/energy/thermal results for the proposed configurations. Section 6 presents an analysis of the spatial distribution of the temperature (chip thermal maps) for selected chip floorplans. Finally, conclusions are drawn in Section 7.

2 Related Work

Several work has explored the design space of CMPs from the point of view of different metrics and application domains. This paper extends previous work in several ways. In the following paragraphs, we analyze previous work in this area, highlighting the differences with respect to this paper.

2.1 Design-Space Exploration for Chip Multiprocessors

Similarly to our work, several papers have explored the core/memory design space. Huh et al. [20] evaluate the impact of several design factors on performance. The authors discuss the interactions of core complexity, cache hierarchy, and available off-chip bandwidth for a workload composed of single-threaded applications. A similar study was conducted by Ekman and Stenstrom [21] for scientific parallel programs also targeting dynamic power. Nevertheless, these work do not consider leakage power and temperature effects.

In [22] and [23], Li and Martinez study the power/performance implications of parallel computing on CMPs. They use a mixed analytical-experimental model to explore parallel efficiency, the number of processors used, and voltage/frequency scaling. The same authors [24] propose a technique to dynamically adapt the number of active processors and voltage/frequency levels.

Li et al. [25] conduct a thorough exploration of the multidimensional design space for CMPs for single-threaded applications. They show that thermal constraints dominate other physical constraints such as pin bandwidth and power delivery.

These work [22], [23], [24], [25] are orthogonal to ours. Our experimental framework is similar to the one in [22] and [24]. We also consider parallel applications and the benchmarks as in [24]. The power model that we use accounts for thermal effects on leakage energy, but unlike the authors of [22], [23], and [24], we also provide a temperature-dependent model for the whole chip. In addition, unlike Li et al. [25], our leakage model accounts for leakage at smaller granularity inside each processor/memory unit.

Hsu et al. [26] explore "future" large-scale CMPs running server workload. They study a three-level cache hierarchy and the related design space. In this paper, we prefer focusing on two-core to eight-core CMPs, as forecast for the next years.

Finally, in [27], Kumar et al. provide a joint analysis of CMP architectures and on-chip interconnections. The authors show the importance of considering interconnect while designing a CMP, arguing that careful codesign of the interconnection and the other architectural entities is needed. This work served as the basis for the floorplans that we used in this work, suggesting us the basic placement techniques for cores and buses.

2.2 Temperature-Aware Architectures

We have already mentioned several work that account for the chip temperature as an important metric to evaluate multicore architectures from a system perspective [22], [23], [24], [25], [28]. Some papers have also appeared so far, proposing the exploration of temperature control techniques and temperature-aware microarchitectures. For example, Donald and Martonosi [29] focus on thermal-management techniques. They explore global and distributed DVFS, thread migration, and control-theoretic mechanisms. Chaparro et al. [30]. present the organization of a distributed, thus temperature-aware, front end for clustered microarchitectures.

In [31], Sankaranarayanan et al. discuss some issues related to chip floorplanning for single-core processors. To the best of the authors' knowledge, the problem of architectural-level floorplan has not been addressed for what concerns multicore architectures. Several chip floorplans have been proposed as instrumental to thermal/power models, but the interactions of floorplanning issues and CMP power/performance characteristics have not been addressed up to now. Our paper provides the first study on CMP floorplans.

Ku et al. [32] analyze the interdependence of temperature and leakage energy in cache memories, proposing some temperature-aware cache management techniques. In our paper, we account for temperature effects on leakage in the memory hierarchy, giving a quantitative description of its spatial distribution in the caches.

Overall, this paper differentiates from the previous ones, since it combines power/performance exploration for parallel applications and interactions with the chip floorplan. At the same time, our model takes the thermal effects and the temperature dependence of leakage energy into account.

3 Design Space

We consider a shared-memory multiprocessor, composed of several independent out-of-order cores, communicating on a shared bus. Microarchitecture details are shown in Section 4, whereas in this section, we focus on the target design space. This is composed of the following architecture parameters:

- *L2 cache architecture (private or shared).* We account for two different design styles for the L2 cache: private and owned by each core or shared among all the processors.
- *Number of cores (2, 4, or 8).* This is the number of cores present in the system.
- *Issue width (2, 4, 6, or 8).* We modulate the number of instructions that can be issued in parallel to integer/floating-point reservation stations. According to this parameter, many microarchitectural blocks are scaled (see Table 2 for details). The issue width is therefore an index of the core complexity.
- *L2 cache size (1, 2, 4, or 8 Mbytes).* This is the total size of the L2 cache. For private-L2 CMPs, this number represents the aggregate cache size[1]; that is, each processor L2 is actually this number divided by the number of cores.

4 Experimental Setup

The simulation infrastructure that we use in this paper is composed of a microarchitecture simulator modeling a configurable CMP, power models, floorplan models, and a temperature modeling tool.

TABLE 2
Architecture Configuration

Processor				
Frequency	3GHz @70 nm			
Vdd	0.9V			
Core area (w/ L1$) [$mm^2$]	31	39	55	78
Branch penalty	7 cycles			
Branch unit	BTB (1K entries, 2-way) Alpha-style Hybrid Branch Predictor (3.7KB) RAS (32 entries)			
Fetch/**issue**/retire width	2/**2**/4	4/**4**/6	6/**6**/8	8/**8**/10
Fetch Queue size	16	32	48	64
INT Issue Queue size	10	20	30	40
FP Issue Queue size	5	15	25	35
INT registers	40	80	120	160
FP registers	32	72	112	152
ROB size	40	80	120	160
LdSt/Int/FP units	1/2/1	1/4/2	1/6/3	1/8/4
Ld/St queue size	16/16	32/32	48/48	64/64
IL1	64KB, 2-way, 64B block, 2 ports			
Access latency	2 cycles			
ITLB entries	64			
DL1	64KB, 2-way, 64B block write-through for private L2 and write-back for shared L2			
Access latency	2 cycles			
MAF size	8			
DTLB entries	128			
L2 Cache				
	8-way, 64B block, write-back, 2 ports			
Access latency	8 (256KB), 10 (512KB), 15 (1MB), 20 (2MB), 25 (4MB), 30 (8MB)			
MAF size	32			
CMP				
Shared Bus	76B, 1.5GHz, 10 cycles delay			
Bandwidth	57 GB/s			
Memory Bus bandwidth	6 GB/s			
Memory lat	490 cycles			

Issue-width parameters are in boldface. They are used as representatives of a core configuration throughout this paper.

4.1 Performance Simulator and Benchmarks

The CMP simulator is SESC [15]. It models a multiprocessor composed of a configurable number of cores. Table 2 reports the main parameters of simulated architectures. Each core is an out-of-order superscalar processor, with private-L1 caches. The four-issue microarchitecture models the Alpha 21264 [33]. For any different issue width, all the processor structures have been scaled accordingly, as shown in Table 2.

Interprocessor communication develops on a high-bandwidth shared bus (57 gigabyte per second (GB/s)), pipelined and clocked at half of the core clock (see Table 2). Coherence protocol acts directly among L2 (or L1 for shared-L2 CMP) caches, and it is MESI snoopy based. For the private-L2 CMP, the protocol requires additional invalidates/writes between L1 and L2 caches to ensure coherence of the data. Memory ordering is ruled by a weak consistency model.

The latency of each cache and memory access has been considered uniform (we used CACTI estimates to model cache latencies, as shown in Table 2). This is consistent with current commercial implementations. Nevertheless, a Non-uniform Cache Access (NUCA) cache could result in a better design, especially for large shared caches, as proposed by Beckmann and Wood [34]. Accounting for this kind of system is out of the scope of this paper, since we focus on up to 8-Mbyte cache and benchmarks with a working set of a few megabytes; thus, that is not suitable to properly evaluate large caches.

[1] In this paper, we mostly use the *aggregate cache size*, often referred to as the *total cache size* or *cache size*. We sometimes use the cache size of each cache when the single cache needs to be referred.

TABLE 3
Benchmarks

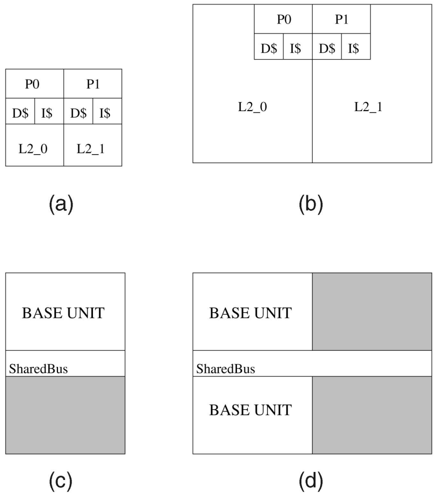
	#graduated instructions (M)	Description – Problem size
FMM	4387–5741	Fast Multipole Method – 16k particles, 10 steps
mpeg2dec	1168	MPEG-2 decoder flowg (Stanford) – 352×240, 10 frames
mpeg2enc	4275	MPEG-2 encoder flowg (Stanford) – 352×240, 10 frames
VOLREND	1425–1888	Volume rendering using ray casting – head, 50 viewpoints
WATER-NS	1780	Forces and potentials of water molecules – 512 molecules, 50 steps

Table 3 lists the benchmarks that we selected. They are three scientific applications from the Splash-2 suite [35] and MPEG2 encoder/decoder from ALPbench [36]. All benchmarks have been run up to completion, and statistics have been collected on the whole program run after skipping initialization. For thermal simulations, we used the power trace related to the whole benchmark simulation (dumped every 10,000 cycles). We used a standard data set for Splash-2, whereas we were limited to 10 frames for the MPEG2. In Table 3, we show the total number of graduated instructions for each application. This number is fairly constant across all the simulated configurations for all the benchmarks, except for FMM and VOLREND. This variation (23 percent to 24 percent) is related to different thread spawning and synchronization overhead as the number of cores is scaled.

4.2 Floorplans

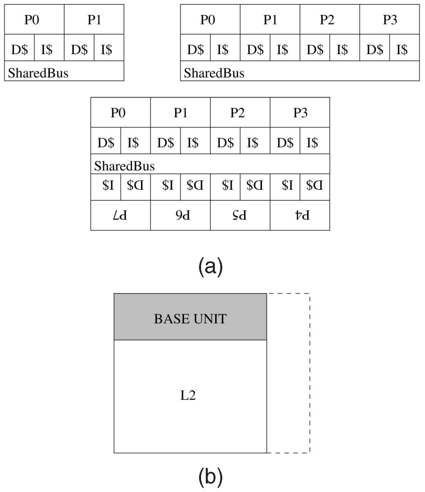
For each configuration, in terms of the number of cores and L2 cache (architecture and size), we consider a different chip floorplan. As in some related work on thermal analysis [5], the floorplan of the core is modeled on the AlphaEv6 (21264) [33]. The area of several units (register files, issue and Ld/St queues, rename units, and FUs) has been scaled according to size and complexity [37]. Each floorplan has been redesigned to minimize dead spaces and not to increase the delay of the critical loops.

Figs. 1 and 2 illustrate the methodology used to build the floorplans that we use for private and shared L2, respectively. For private L2, the two-core layout is used as the base unit to build larger CMPs. To obtain a reasonable aspect ratio, we defined two different layouts: one for small caches, where each L2 is 256 and 512 Kbytes (Fig. 1a), and another one for large caches, where each L2 is 1,024 and 2,048 Kbytes (Fig. 1b). The floorplans for four and eight cores are built by using the two-core floorplan as the base unit. The base unit for small caches is shown in Fig. 1a. Core 0 (P0) and core 1 (P1) are placed side by side. The L2 caches are placed in front of each core. In the base unit for large caches (Fig. 1b), the L2 is split in two pieces: one in front of the core and the other one beside it. This way, each processor+cache unit is roughly squared. For four and eight cores, this floorplan is replicated and possibly mirrored, trying to obtain the aspect ratio, which most approximates 1 (a perfect square).[2] Figs. 1c and 1d show how the four-core and eight-core floorplans have been obtained respectively from the two-core and four-core floorplans.



(a) (b) (c) (d)

Fig. 1. Chip floorplans for private-L2 CMPs: base units for two cores (each L2 is (a) 256 and 512 Kbytes or (b) 1,024 and 2,048 Kbytes) and schemes for (c) four cores and (d) eight cores.



(a) (b)

Fig. 2. Chip floorplans for shared-L2 CMPs. (a) Base units for two, four, and eight cores. (b) Cache placement.

On the other hand, for shared-L2 CMPs, we used different base units for two, four, and eight cores, as shown in Fig. 2a. In this case, the base unit consists of the cores and the bus. The L2 cache is therefore placed on one or two sides of the base unit, as illustrated in Fig. 2b. Notice that a small L2 (with respect to the base unit) will fit on one side of the base unit, whereas a larger one will be placed on two sides, leaving the processors in a corner of the chip. For example, for a four-core four-issue CMP, if the L2 is smaller or equal to 4 Mbytes, it will fit on the bottom side. The 8-Mbyte L2 will be on two sides.

For any cache/core configurations, each floorplan has the shape/organization as outlined in Figs. 1 and 2 but has different size (these have been omitted for the sake of clarity). Table 4 shows the chip area for each design. For each configuration, the shared bus is placed according to the communication needs. The size is derived from [27].

4.3 Power and Thermal Model

The power model integrated in the simulator is based on Wattch [16] for processor structures, CACTI [17] for caches,

[2] This maximizes the number of chips per wafer.

TABLE 4
Chip Area (in Square Millimeters)

		Private L2				Shared L2			
		Issue				Issue			
#cores	Total L2 Size [MB]	2	4	6	8	2	4	6	8
2	1	63	68	80	93	63	68	80	93
2	2	108	113	124	138	99	104	116	129
2	4	187	192	202	216	172	176	189	201
2	8	341	346	356	369	317	322	334	347
4	1	81	90	112	138	90	99	124	149
4	2	117	126	148	174	126	135	160	185
4	4	199	208	228	255	199	208	233	258
4	8	350	359	379	406	344	353	378	403
8	1	126	144	188	239	126	144	188	239
8	2	162	180	224	275	162	180	224	275
8	4	235	253	297	348	235	253	277	348
8	8	398	416	456	511	380	398	442	493

and Orion [38] for buses. The thermal model is based on HotSpot-3.0.2 [5]. HotSpot uses dynamic power traces and a chip floorplan to drive thermal simulation. As outputs, it provides transient behavior and steady-state temperature. According to [5], we chose a standard cooling solution featuring thermal resistance of 1.0 K/W. We chose an ambient air temperature of 45 °C. We carefully select initial temperature values to ensure that the thermal simulation rapidly converges to a stationary state. This is needed, since thermal simulations may take a relatively long time to converge.

In detail, the thermal model allows for numerically solving the heat equation for the chip:

$$\nabla^2 T + \frac{1}{k} q_{gen}(T) = \frac{\rho c}{k} \frac{\partial T}{\partial t}, \tag{1}$$

where k is the thermal diffusivity, ρ is the density, and c is the specific heat. $T = f(t, x, y, z)$ is the time-dependent temperature of the chip, and $q_{gen} = f(T; t, x, y, z)$ is the temperature-dependent internal power generation per unit volume, which is determined by the power-dissipating elements of the chip. In our model, these are modeled as active power source and leakage power source as follows:

$$q_{gen}(t, x, y) = P_D(t, x, y) + P_L(T, x, y), \tag{2}$$

where P_D is the dynamic power dissipation of the unit volume, as obtained from Wattch and CACTI. The leakage model P_L is based on the work of Liao et al. [18]. According to the model, the power per unit volume is

$$P_L = N_i V_{dd} \Big(A_i T^2 e^{\frac{-\alpha_j V_{dd} + \beta_j}{T}} + B_j e^{\gamma_j V_{dd} + \delta_j} \Big), \tag{3}$$

where N_i is the gate density, which depends on the gate density of each microarchitecture block, similar to [39]. V_{dd} is the operating voltage, and A_j, B_j, α_j, β_j, γ_j, and δ_j are the model parameters, whose values differ, depending on the technology (RAM, memory logic, and processor logic) for each microarchitecture block, as in [18].

HotSpot has been augmented with the temperature-dependent leakage model, which is sketched above. This way, temperature dependency is modeled by varying the amount of leakage according to the proper distribution. At each iteration of the thermal simulation, the leakage contribution is calculated and "injected" into each HotSpot grid element. In summary, for each grid element, the following illustrates the iterative calculation of the temperature at time $k+1$, which depends on the leakage computed at time k:

$$T(k+1) = f(P_D + P_L(T(k))). \tag{4}$$

5 Performance and Energy Evaluation

This section describes the results obtained for the target design space in terms of system-level metrics. In detail, the metrics that we consider are the following:

- **Delay.** We measure the performance of the system when running a given parallel application by using the *execution time* (or *delay*). This is computed as the time needed to complete the execution of the program.
- **Energy.** We use the system *energy*, that is, the energy needed to run a given application, as our energy-efficiency metric.
- **Temperature.** We account for thermal effects, and we therefore report the *average temperature* across the chip and the *maximum temperature* (the hottest temperature of the chip). In this paper, we refer to temperature numbers as the temporal mean of the physical temperature (time dependent), which may be measured by hardware sensors. Despite that absolute variation of the average temperature is relatively small, at most few degrees, we think that this figure is a proxy for the global heating of the chip.
- **Energy delay product (EDP).** The *EDP* is typically used to evaluate energy delay trade-offs. In addition, we give a discussion of ED^{n}P optimality for the considered design space. This can be useful when considering a different Energy Delay metric, like Energy Delay^2 product (ED2P), typically adopted for high-performance systems, since it gives higher priority to performance over power.

Fig. 3 reports all data for our design space. We base most of our analysis on average values (arithmetic mean) across all the simulated benchmarks. Only when needed to get more insight do we refer to the single benchmark.

5.1 Level-2 Cache

Generally speaking, the shared-L2 architecture achieves better performance (see Figs. 3a and 3b). This is due to the positive effect of the sharing. The gain is approximately 10 percent for two-issue processors, whereas it becomes more appreciable when comparing wider cores (30 percent). Private L2 suffers from coherency misses, as we shall see shortly. This makes also that the shared L2 better exploit the memory-level parallelism. On the other hand, a shared L2 features higher latency. Nevertheless, the impact of the increased latency on the performance seems not appreciable for the considered benchmarks. The impact of the L2 cache size is at a maximum of 4 percent to 5 percent (from 1 Mbyte to 8 Mbytes) when the other parameters are fixed.

Table 5 shows some more insight regarding the L2 cache for each benchmark. The number of misses is overall small: at most a few tens of misses per 10,000 instructions. This is because our benchmarks feature a standard data set fitting in a few megabytes. We report the number of misses per

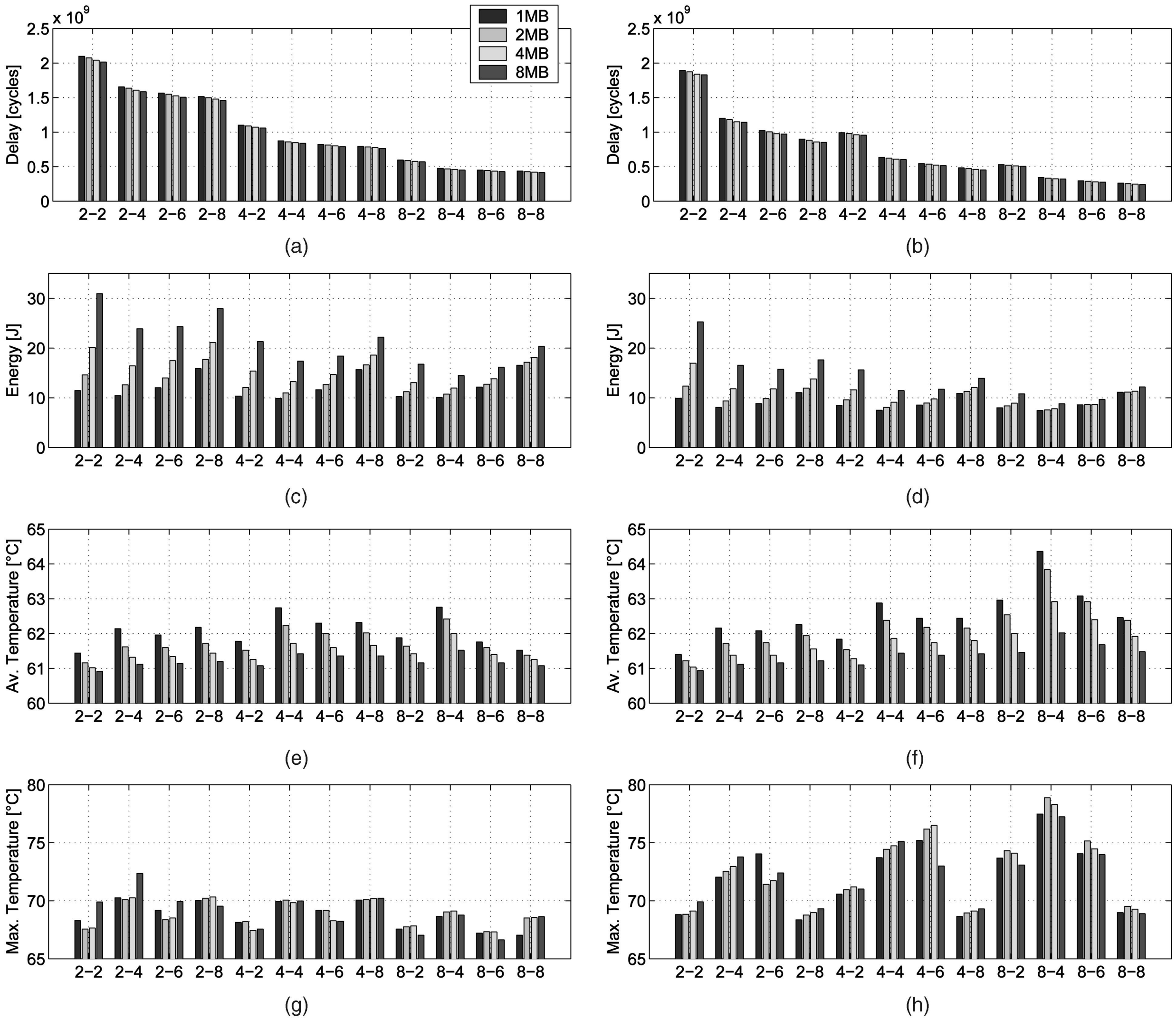


Fig. 3. System-level metrics. Each group of bars (L2 cache varying) is labeled with *<#procs>-<issue width>*. (a) Delay: private L2. (b) Delay: shared L2. (c) Energy: private L2. (d) Energy: shared L2. (e) Average temperature: private L2. (f) Average temperature: shared L2. (g) Maximum temperature: private L2. (h) Maximum temperature: shared L2.

instruction, which we think is a fair metric for comparing two different cache architectures (private and shared) with different L2 accesses count. Consider, for example, that the L1s in the private-L2 architectures must be write through, thus making a large number of writes (hit) to the L2.

The IPC is quite sensitive to the cache size for *VOLREND* and *FMM*, impacting up to 17 percent. Table 5 also reports the average line occupation (that is, the average time that a line is not invalid). The ILP unveiled by the shared-L2 architecture covers the increased misses due to the conflicts of the working sets of the different cores. The shared-L2 architecture permits a larger number of in-flight load/stores in the memory system, which is otherwise forbidden by the private L2. In fact, in this case, every write propagates, causing significant serialization. For *mpeg2dec/enc*, *WATER-NS*, the IPC trend is almost independent of the cache size. In addition, notice that larger private L2 caches cannot be completely filled with useful lines, mostly because of the protocol invalidates (for example, in the case of the 8-Mbyte cache, the line occupation may drop below 10 percent). This is due to the coherency misses for these applications. In fact, the shared-L2 architectures, which does not suffer from coherency misses, always features nearly full occupation with fewer misses.

Regarding energy, when comparing private-L2 and shared-L2 architectures (Figs. 3c and 3d), you can observe that shared-L2 CMPs feature a significant reduction of the energy. This is mostly a consequence of the execution time reduction, which reduces the static energy as well. For two-core architectures, the energy reduction is approximately 20 percent. This becomes larger, scaling the number of processors: 30 percent for four cores and 45 percent for eight cores.

The absolute variation of the average temperature is at most $1.7/3.5\ ^{\circ}\mathrm{C}$ around 62 °C (private/shared L2) (see Figs. 3e and 3f). Since L2 caches occupy most of the chip, it also indicates that the L2 temperature is approximately independent of most parameters. Most differences can be abducted to changes in the core temperature, which can be quite significant and will be discussed in Section 6.

TABLE 5
IPC, L2 Miss/Instruction, and Occupation for Each Benchmark for Four-Core Four-Issue CMPs and Different L2 Architectures and Sizes

		IPC				L2 misses/10k-instr.				L2 average line occupation			
	L2 arch.	1MB	2MB	4MB	8MB	1MB	2MB	4MB	8MB	1MB	2MB	4MB	8MB
FMM	P[1]	2.23	2.27	2.34	2.40	4.44	8.05	10.36	11.71	98.27	97.92	98.79	98.29
	S[2]	4.77	5.03	5.49	5.59	40.7	25.3	5.79	0.99	98.86	97.85	96.94	93.29
mpeg2dec	P	4.33	4.33	4.33	4.33	2.57	10.24	5.12	7.68	52.81	26.67	13.34	6.67
	S	4.67	4.67	4.67	4.67	0.18	0.18	0.18	0.18	93.72	96.86	98.44	99.22
mpeg2enc	P	2.84	2.86	2.87	2.87	4.4	2.09	6.48	8.56	98.44	77.10	44.56	22.31
	S	3.08	3.09	3.09	3.09	2.23	0.13	0.13	5.97	98.33	96.39	98.20	99.10
VOLREND	P	4.07	4.12	4.23	4.31	6.85	12.67	17.64	21.78	97.35	95.62	92.63	84.43
	S	4.13	4.20	4.27	4.39	25.89	19.09	11.06	1.61	97.11	94.84	90.52	91.33
WATER-NS	P	6.31	6.32	6.32	6.32	0.38	0.75	1.12	1.49	73.38	39.46	20.22	10.11
	S	6.32	6.32	6.32	6.32	0.14	0.14	0.14	0.14	98.97	99.48	99.74	99.87

[1] Private L2

[2] Shared L2

Table 6 reports the average and maximum temperature for four-core four-issue 1-Mbyte L2 architectures for every benchmark. For other cache sizes, the temperatures are quite similar. An important maximum temperature difference (up to 12 °C) exists for *FMM* and *VOLREND* between shared-L2 and private-L2 systems. This corresponds to a hotspot in the processor core (Load/Store Queue) for these benchmarks. In the next sections, we shall go into more details about this. *FMM* and *VOLREND* also feature the highest misses per instruction (see Table 5). This means that the Load/Store Unit (and the memory system) is much more stressed. Other benchmarks showing a different memory behavior do not show these temperature differences.

5.2 Number of Cores

When scaling the system from 2 to 4 and from four to eight processors, the delay is reduced by 47 percent each step. This trend is homogeneous across other parameter variations: it is also similar for shared or private L2. It means that high parallel efficiency is achieved by these applications and that the communication overhead is not appreciable.

With regard to energy, it can be seen that when the number of processors is increased, the system energy typically slightly decreases although the area increases. For example, regarding private L2, for a four-issue 4-Mbyte configuration, the energy decrease rate is 20 percent, from two cores to four cores, and eight percent, from four cores to eight cores. For shared L2, the same behavior can be observed. In fact, the leakage energy increase due to the larger chip is dominated by the reduction of the execution time or the *power-on time* (that is, the time that the chip is leaking is shorter). Notice that clock gating[3] has no effect on the leakage. In fact, it affects the sole active power. On the other hand, we do not consider power supply gating.

TABLE 6
Peak/Average Temperature for Each Benchmark for Four-Core Four-Issue 1-Mbyte L2 CMPs

		Temperature [°C]	
	L2 arch.	Peak	Average
FMM	P	66.1	62
	S	78.9	63.7
mpeg2dec	P	71.1	62.7
	S	70.8	62.4
mpeg2enc	P	68.5	62.2
	S	68	62
VOLREND	P	69.9	63
	S	74.9	63.1
WATER-NS	P	74.2	63.8
	S	76	63.2

With regard to temperature, the average temperature highlights some trends for the shared-L2 eight-core CMPs, featuring higher chip temperature (approximately 1 °C). It should be pointed out that the floorplan for eight cores is built with a base unit composed of a tight cluster of eight processors (see Fig. 2). In this case, thermal coupling may exist among the cores.

5.3 Issue Width

By varying the issue width, for a given processor count and L2 size, a larger speedup is observed when moving from two to four issues (21 percent to 32 percent for two-core private/shared L2). If the issue width is furthermore increased, the improvement of the delay saturates for private L2 (5.5 percent from four to six issues and 3.3 percent from six to eight issues), whereas for shared L2, the speedup is 23 percent from four to six issues and 20 percent from six to eight issues. This trend is also seen (but not that dramatically) for the configurations of four and eight cores.

Energy has a minimum at four issues. This occurs for two, four, and eight processors and for private/shared L2. The four-issue cores offer the best balance between complexity, power consumption, and performance. The two-issue microarchitecture cannot efficiently extract available ILP (leakage due to delay dominates). Eight-issue processor cores need much more energy to extract little more parallelism with respect to four-issue and six-issue ones.

The energy of the shared-L2 architectures is less sensitive to the variation of the issue width. For example, for four-core 4-Mbyte L2, the energy variation is at most 25 percent

[3] We assume simple clock gating for each processor structure and caches.

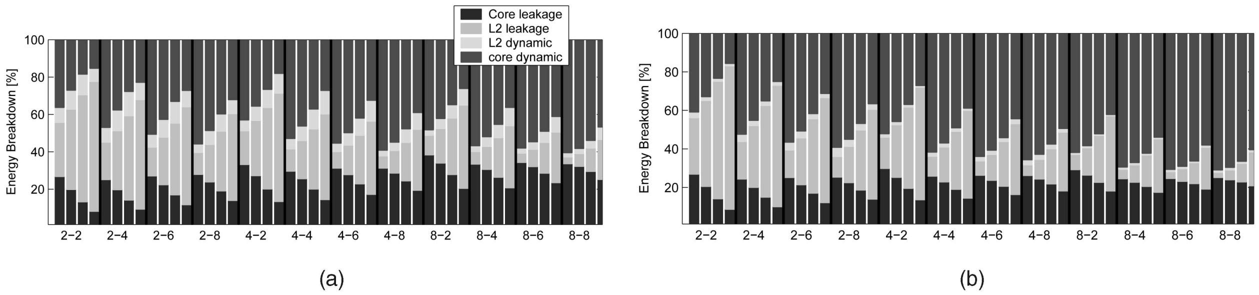


Fig. 4. Energy breakdown (leakage/dynamic/core/L2). Each group of bars (L2 cache varying) is labeled with *<#procs>-<issue width>*. (a) Private L2. (b) Shared L2.

for shared L2, whereas this is 40 percent for private L2. As we have already observed, the private-L2 and the shared-L2 architectures exploit differently the available issue width; that is, shared L2 is typically more efficient. For this reason, for a shared-L2 architecture, an increase in the issue width results in a performance and energy benefits, despite the increased complexity.

Temperature is influenced by the actual floorplan of the core. We shall go into more details in Section 6, but for example, you can easily notice that four-issue cores get much hotter than the others. This is because this layout is more compact than the others.

5.4 Core/Level-2 Dynamic/Static Energy Breakdown

The dynamic/static energy breakdown depends on the several architectural parameters. As shown in Fig. 4, all dimensions of our design space determine the distribution of these energy components. Some interesting points can be summarized as follows:

- The shared-L2 CMPs feature a larger fraction of dynamic energy with respect to the private-L2 ones. This is because the L2 miss stall time (power on) is better hidden by shared-L2 architectures.
- The fraction of dynamic energy for a private-L2 memory system is quite large. This is due to the replicated structures for accessing the caches.
- Increasing the cache size reduces the pressure on several processor units (for example, the Load/Store Queues), making dynamic and static energy in the cores smaller, whereas the cache energy increases, as expected for larger caches.
- Wider processors make dynamic energy in the processors increase because of the complexity of the microarchitecture.

The L2 cache represents the main contributor to chip area, so it affects leakage energy considerably and makes the energy scale linearly with the L2 cache size. The total static energy for our simulations ranges from 80 percent (two-core two-issue 8-Mbyte L2) down to 30 percent (eight-core eight-issue 1-Mbyte L2). See Fig. 4, which shows the energy breakdown in terms of the dynamic/static and core/L2 components. These numbers are congruent with estimates/prediction for 70-nm technologies [40], [41]. Notice that even if the leakage forecast for a given technology node may be more conservative, the data shown in Fig. 4 refers to the *energy consumption* of a specific application/architecture, which can dramatically vary with respect to the nominal leakage breakdown.

5.5 Energy Delay

Fig. 5 shows the Energy Delay scatter plot for the design space. Some curves with constant EDP are reported to better evaluate this metric. Furthermore, all the points of the *energy-efficient families* [42] for the private-L2 (black circles) and shared-L2 (red squares) architecture are connected with lines (one for each architecture). In the Energy Delay plane, these points form a convex hull of all possible configurations for a given design space (private L2, shared L2, or the joint space). Each point of the energy-efficient family is optimal for an ED^{n}P metric for any given n.

The energy-efficient family for the shared L2 is also the energy-efficient family for the whole design space. This means that the shared-L2 architectures are optimal for any ED^{n}P for all the design space.

We can also characterize the two families regarding the *hardware intensity*.[4] In particular, the energy-efficient family for the shared L2 is composed by the configurations with eight cores and four, six, and eight issues. These points lay on a constant EDP curve; that is, they have unitary *hardware intensity*. This means that moving among these configurations, energy and delay are traded off nearly perfectly.

For the private-L2 architectures, the energy-efficient family is split into two pieces: one featuring high *hardware intensity* (eight cores and four, six, and eight issues) and another with low *hardware intensity* (actually, only one configuration), optimal for energy but not for delay. The usage of the L2 cache and issue width in private-L2 CMPs is less efficient,[5] so larger caches cannot be entirely translated into performance gain.

In this case (private L2), the best configuration from the Energy Delay point of view is well defined: the eight-core four-issue 1-Mbyte L2. This is optimal for the ED^{n}P, with $n < 2.5$, as it can be intuitively observed in Fig. 5. If $n \geq 2.5$, the optimum moves to 4-Mbyte L2 and, therefore, to the points of the energy-efficient family with higher *hardware intensity*.

The shared-L2 CMPs better exploit large cache and wide issue: although a clear EDP minimum exists for private L2

[4] *Hardware intensity* is defined as the ratio of the relative increase in energy to the corresponding relative gain in performance achievable by architectural modification. See [42] for a detailed discussion.

[5] This is in terms of hardware intensity; that is, more energy is needed for some performance improvement.

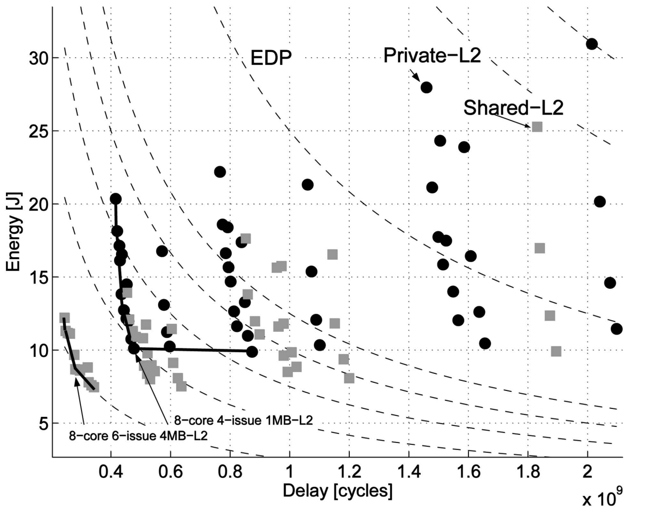


Fig. 5. Energy delay scatter plot for the simulated configurations. Private-L2 configurations are the black circles, and shared-L2 configurations are the red squares. The dashed lines are EDP constant curves.

(eight-core four-issue 1-Mbyte L2), this is not well defined for shared L2. Moreover, larger caches and wider cores seem preferred. In this case, the optimum configuration seems to be eight cores, six issues, and 4 Mbytes, even if all the configurations of the energy-efficient family are close in EDP.

Notice that when considering physical constraints such as power delivery or maximum operating temperature, the design space may come to be pruned. In particular, regarding shared-L2 CMPs, power/temperature considerations may exclude eight cores and four/six/eight issues, and four-core six-issue 1/2/4-Mbyte L2 (consider, for example, that the maximum power is 100 W, and the maximum temperature is 75 °C). In this case, the optimal configuration would shift to the eight-core two-issue 1-Mbyte L2, which, due to the simple core, features better power/temperature characteristics.

6 Temperature Spatial Distribution

This section discusses the spatial distribution of the temperature (that is, the chip thermal map) and its relationship with the floorplan design. We first analyze several different floorplan designs for CMPs. We then discuss the energy and temperature of the microarchitectural units of the processors. Finally, we analyze the temperature distribution while varying the complexity of the processor design (that is, the issue width).

6.1 Floorplan Evaluation

The main goal of this section is to understand how system-level floorplan designs impact on-chip thermal behavior. Several floorplan topologies, featuring different core/cache placements, are analyzed. We reduce the scope of this part of the work to four-core four-issue CMPs and L2 cache size of 1 and 4 Mbytes to model a system composed of Alpha-like processors. Similar results are obtained for the other configurations.

Different floorplans correspond to different circuit delays. We designed the floorplans under evaluation, striving for minimizing the differences. The main issues are related to L2 cache design. Nevertheless, we believe that for the early evaluation phase of a design space, assuming uniform cache access, delay differences could be neglected for the floorplans corresponding to the same architecture.

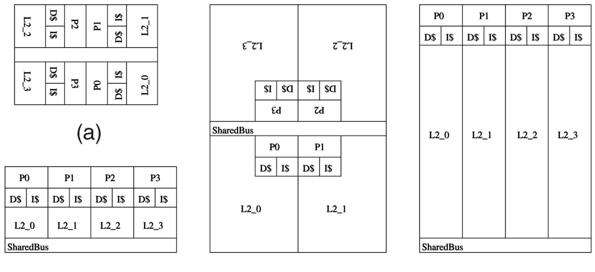


Fig. 6. Private L2. Additional floorplans were taken into account for the evaluation of the spatial distribution of the temperature. (a) 1 Mbyte, centered. (b) 1 Mbyte, lined up. (c) 4 Mbytes, centered. (d) 4 Mbytes, lined up.

The power traces are from the WATER-NS benchmark. We selected this application as representative of the behavior of all the benchmarks. In particular, this benchmark stresses both the integer and floating-point arithmetic of the processor and the memory system. The conclusions drawn in this analysis apply for each simulated application. We accounted for differences in the floorplans while performing the thermal simulation, properly modifying the power distribution in the cache banks.[6]

Figs. 6 and 7 show the additional floorplans that we have considered, in addition to those in Figs. 1 and 2. The floorplans are classified with respect to the processor position in the die:

- *Lined up* (Figs. 6b, 6d, and 2). Cores are lined up on a side of the die. This configuration, along with the following one (*Centered*), is common to private/shared-L2 systems.
- *Centered* (Figs. 6a, 6c, 7b, and 6d). Cores are placed in the middle of the die.
- *Paired* (Fig. 1). Cores are paired and placed on alternate sides of the die.
- *Corner* (Figs. 7a and 7c, *only shared L2*). Cores are placed in a corner of the die. Notice that for the 1-Mbyte L2, the cache is only on one side of the cores.

The area and aspect ratio are roughly equivalent for each cache size across different topologies.

We report the thermal map for each floorplan type and cache size. This is the stationary solution as obtained by HotSpot. We use the *grid model*, making sure that grid spacing is 100 μm for each floorplan. The grid spacing determines the accuracy for the magnitudes of the hotspots, since each grid point has somewhat the average temperature of 100 μm × 100 μm surrounding it. Our constant grid spacing setup ensures that the hotspot magnitude is comparable across designs.

For each thermal map (for example, choose Figs. 8c and 10a), several common phenomena can be observed. Several

[6] Notice that for the layout of any different architectural configuration (any point of the design space in Section 3), we conducted a different power/performance simulation.

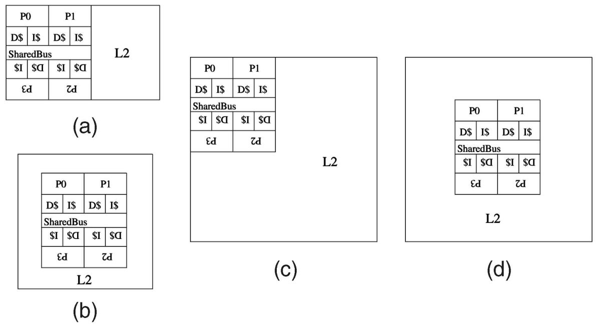


Fig. 7. Shared L2. Additional floorplans were taken into account for the evaluation of the spatial distribution of the temperature. (a) 1 Mbyte, corner. (b) 1 Mbyte, centered. (c) 4 Mbytes, corner. (d) 4 Mbytes, centered.

temperature peaks correspond to each processor. This is due to the hotspots in the core. In particular, for shared-L2 systems (for example, Fig. 10a), a dominant hotspot exists, corresponding to the Load/Store Queue. For private-L2 architectures, several hotspots per processor coexist. Their nature will be analyzed shortly in the next section. Caches, in particular the L2 cache, and the shared bus are cooler, apart from the portions that are heated because of the proximity to the processors.

Private L2. *Paired* floorplans for 1 Mbyte (Fig. 8c) and 4 Mbytes (Fig. 9c) show similar thermal characteristics. The temperature of the two hotspots of each core (the hottest is the FP unit, whereas the other one is the INT Issue Queue coupled with the Ld/St queue) ranges from 73.1 to 74.2 °C for the 1-Mbyte L2. The hottest peaks are the ones closest to the die corners (FP units on the right side), since they dispose of less spreading sides into the silicon. For the 4-Mbyte L2 cache, the hotspot's temperature is between 72.5 and 73.4. In this case, the hottest peaks are between each core pair, due to the thermal coupling between processors.

The same phenomena appear for the *lined-up* floorplans (Figs. 8a and 9a). The rightmost hotspots suffer from corner effect, whereas the inner ones suffer from thermal coupling.

Fig. 8b shows an alternative design. Here, processors are placed at the center of the die (see Fig. 6a). In addition, their orientation is different, since they are back to back.

As can be observed in Table 7, the maximum temperature is lowered (1.6/1.8 °C). In this case, this is due to the increased spreading perimeter of the processors: two sides are on the surrounding silicon, and another side is on the bus (the bus goes across the die; see Fig. 6a). The centered floorplans are the only ones with a fairly hot bus. In this case, leakage in the bus can be significant. Furthermore, for this floorplan, between the backs of the processors, the

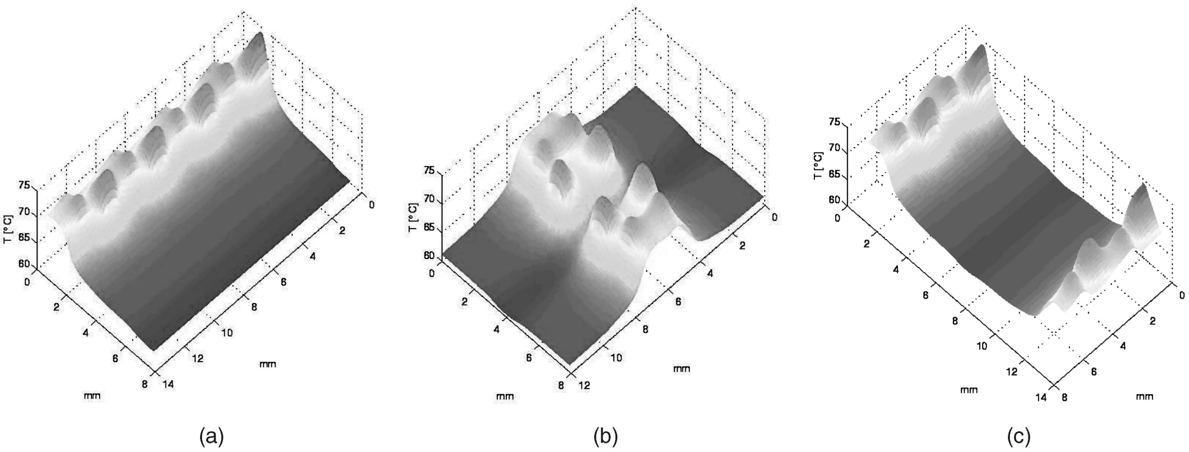


Fig. 8. Thermal maps for private 1-Mbyte L2. (a) Lined up (Fig. 6b). (b) Centered (Fig. 6a). (c) Paired (Fig. 1).

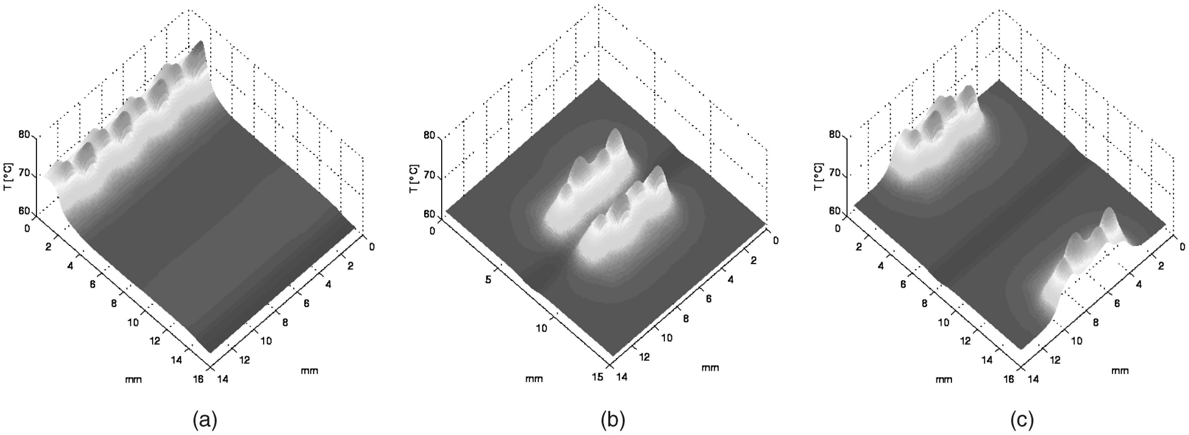


Fig. 9. Thermal maps for private 4-Mbyte L2. (a) Lined up (Fig. 6d). (b) Centered (Fig. 6c). (c) Paired (Fig. 1).

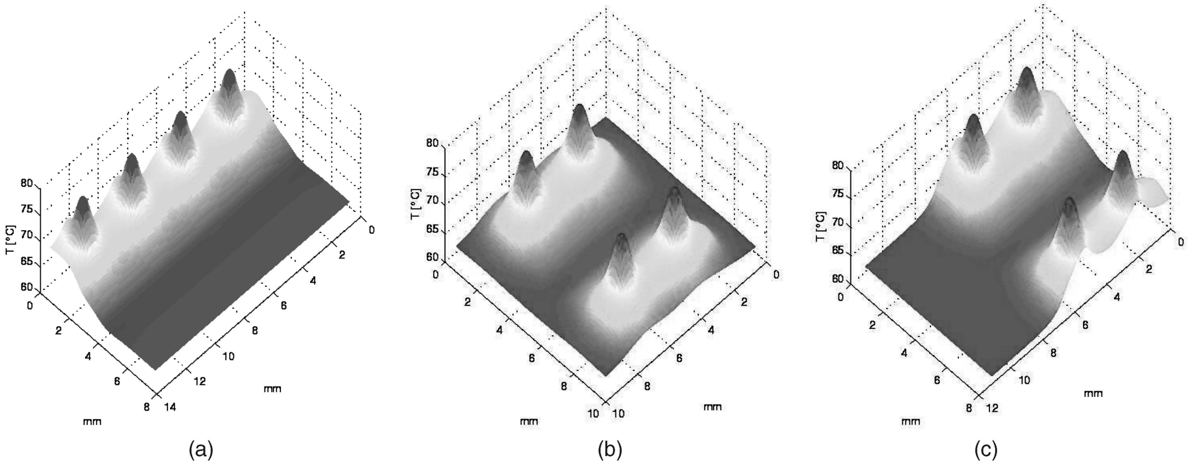


Fig. 10. Thermal maps for shared 1-Mbyte L2. (a) Lined up (Fig. 2). (b) Centered (Fig. 7b). (c) Corner (Fig. 7a).

temperature is quite "high" (69.4 °C), unlike other floorplans featuring relatively "cold" backs (67.9 °C).

For the *4-Mbyte centered* floorplan (Fig. 9b) these characteristics are less evident. The maximum temperature decrease is only 0.4/1.2 °C with respect to *paired* and *lined-up* floorplans. In particular, it is small if compared with the *paired* one. This is reasonable if considering that in the *paired* floorplan, the cache (4 Mbyte) surrounds each pair of cores, therefore providing enough spreading area.

Overall, the choice of the cache size and the relative placement of the processors can be important in determining the chip thermal map. Those layouts, where processors are placed at the center and where L2 caches surround cores, typically feature lower peak temperature. The temperature decrease, with respect to alternative floorplans, is between 1 and 2 °C. The impact on the system energy can be considered negligible, since L2 leakage dominates. Despite this, such a reduction in the hotspot temperature can lead to leakage reduction localized in the hotspot sources.

Shared L2. Shared-L2 systems differ from private-L2 ones, mainly for the hotspot on the Load/Store Queue (see Figs. 10 and 11). This is because the more efficient usage of the L2 reduces stall periods in the processors. Otherwise, L2 miss stalls would reduce activity and cool the chip. According to our model, this hotspot makes the leakage in the Load/Store Queue increase by approximately 50 percent.

In general, this means that peak temperature, for four-issue processors, is 2/3 °C higher than private-L2 systems (see Table 8). On the other hand, the average temperature is basically unchanged. Notice that the behavior of the other benchmarks may emphasize the difference between private and shared L2 (see Table 6).

TABLE 7
Private L2: Average and Maximum Chip Temperature for Different Floorplans

L2	1MB		4MB	
	Max [°C]	Av. [°C]	Max [°C]	Av. [°C]
Lined up	74.0	63.5	74.2	62.2
Centered	72.4	63.6	73.0	62.2
Paired	74.2	63.8	73.4	62.2

When comparing different floorplan topologies, the same considerations for private-L2 systems hold. *Centered* floorplans feature a slightly reduced peak temperature, but in this case, gains are marginal (less than 1 °C). *Lined up* and *Corner* floorplans have the same thermal characteristics. Since the Load/Store Queue hotspot is in the middle of the processor, also, the effects of the proximity to the die edge are not appreciable.

6.2 Processor Energy Breakdown

Table 9 shows some more insight into the core temperature and energy consumption, reporting the energy breakdown for private-L2 and shared-L2 CMPs (four-core 1-Mbyte L2) and the associated maximum temperature for each functional unit.

Similar to what we have already observed, the shared-L2 architecture features an hotspot in the Load/Store Queue, whereas the private-L2 systems show multiple and smaller heating sources. This can be motivated by looking at the energy consumption of the Load/Store Queue, especially for the dynamic energy. The phenomenon is more evident for *FMM* and *VOLREND*. The other benchmarks still have a hotspot in the Load/Store Queue, but there is not so much difference between private and shared L2.

Both floating-point and integer ALUs consume a significant fraction of the energy of the core. These units are indeed quite hot, especially for computationally intensive benchmarks. For example, consider the FPALU in all the Splash-2 benchmarks, and IntALU for *mpeg2*.

Register files and issue queues result also hot, even if our configuration for a four-issue core is not so aggressive regarding these structures (see Table 2). As we shall see shortly (Section 6.3), these become hotter when the issue width is scaled up.

6.3 Processor Complexity

In Fig. 12, several thermal maps for different issue widths of the processors are reported. We selected the WATER-NS benchmark, a floating-point one, since this enables us to discuss heating in both FP and Integer units. The CMP configuration is two-core 1-Mbyte L2. We also chose the private-L2 architecture, since it presents more varied

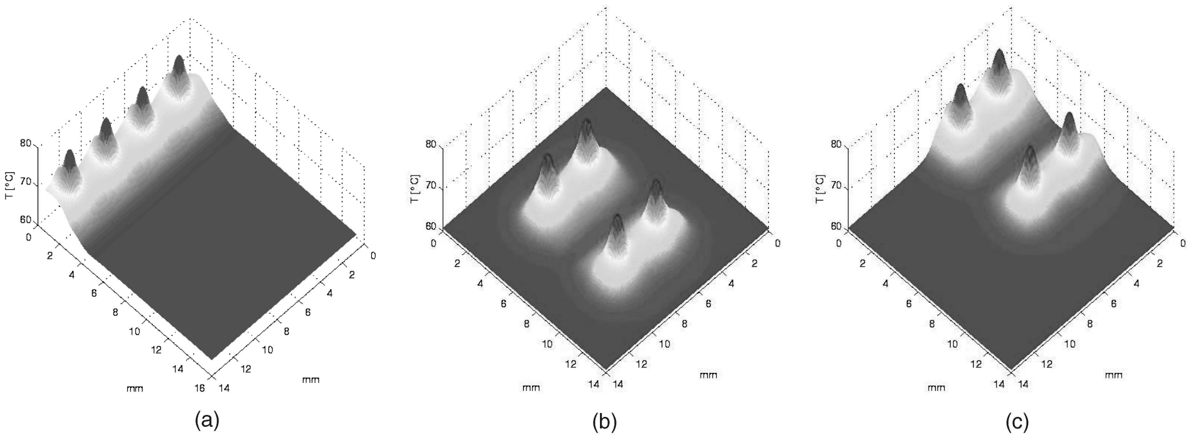


Fig. 11. Thermal maps for shared 4-Mbyte L2. (a) Lined up (Fig. 2). (b) Centered (Fig. 7d). (c) Corner (Fig. 7c).

TABLE 9
Breakdown for the Core Energy and Temperature of a Four-Core 1-Mbyte L2 CMP

		Functional units										
		ICache	DCache	BPred	IntALU	FPALU	IntRF	FPRF	IntQ	FPQ	FetchQ	LdStQ
FMM												
Leakage	P	5.54	5.19	1.52	13.16	13.19	0.84	0.57	1.64	1.33	1.85	1.17
	S	2.62	2.46	0.73	6.40	6.34	0.40	0.27	0.81	0.67	0.89	0.61
Dynamic	P	3.55	3.39	4.12	9.36	8.90	4.77	1.51	7.55	1.49	3.18	6.19
	S	6.83	6.78	5.04	6.39	6.04	3.17	2.63	11.02	7.74	4.17	18.01
Total	P	9.09	8.58	5.64	22.52	22.09	5.61	2.08	9.19	2.82	5.03	7.35
	S	9.44	9.23	5.77	12.79	12.38	3.57	2.90	11.83	8.40	5.06	18.62
Max. Temperature	P	62.15	62.10	63.70	65.80	65.22	66.10	65.32	65.67	64.45	64.71	65.62
	S	64.18	64.43	67.83	70.95	69.00	70.40	69.58	74.00	73.47	69.65	78.90
mpeg2dec												
Leakage	P	5.02	4.72	1.38	12.08	11.84	0.77	0.51	1.51	1.21	1.67	1.08
	S	5.11	4.80	1.41	12.14	12.14	0.77	0.52	1.52	1.25	1.71	1.10
Dynamic	P	3.52	4.50	1.83	12.13	0.39	6.63	0.03	10.75	1.05	3.61	13.76
	S	5.03	5.04	3.72	4.75	4.46	2.34	1.94	8.17	5.71	3.08	13.29
Total	P	8.54	9.21	3.21	24.21	12.23	7.41	0.54	12.26	2.26	5.29	14.84
	S	10.15	9.84	5.13	16.89	16.60	3.11	2.46	9.69	6.96	4.78	14.39
Max. Temperature	P	62.30	62.53	63.30	67.35	64.28	68.00	64.30	67.78	64.90	64.95	68.50
	S	61.98	62.10	63.73	65.50	64.70	65.00	64.90	66.15	65.95	64.60	68.00
mpeg2dec												
Leakage	P	3.77	3.54	1.04	9.22	9.02	0.59	0.39	1.15	0.92	1.27	0.82
	S	4.07	3.82	1.13	9.73	9.71	0.62	0.42	1.22	1.00	1.36	0.89
Dynamic	P	3.81	3.77	2.44	14.84	6.30	6.74	0.82	11.33	1.58	4.19	12.45
	S	5.78	5.77	4.27	5.44	5.12	2.69	2.23	9.36	6.56	3.53	15.27
Total	P	7.59	7.31	3.48	24.06	15.32	7.32	1.21	12.48	2.50	5.46	13.28
	S	9.85	9.59	5.40	15.17	14.83	3.30	2.65	10.59	7.56	4.90	16.16
Max. Temperature	P	63.08	63.20	64.88	69.88	66.60	70.80	66.70	70.58	66.92	67.07	71.10
	S	62.47	62.62	64.70	66.85	65.72	66.35	66.02	68.15	67.80	65.81	70.80
VOLREND												
Leakage	P	3.46	3.23	0.97	8.38	8.49	0.53	0.37	1.04	0.85	1.17	0.74
	S	3.18	2.98	0.88	7.68	7.63	0.49	0.33	0.97	0.80	1.07	0.72
Dynamic	P	3.57	3.34	4.00	11.57	19.92	4.94	2.27	7.96	2.54	3.80	6.87
	S	6.42	6.39	4.75	6.03	5.69	2.98	2.48	10.38	7.28	3.92	16.95
Total	P	7.03	6.57	4.96	19.95	28.41	5.47	2.64	9.00	3.39	4.97	7.61
	S	9.60	9.37	5.63	13.71	13.32	3.47	2.81	11.35	8.08	5.00	17.67
Max. Temperature	P	62.90	62.67	66.20	68.65	69.14	69.12	69.55	68.73	67.50	67.20	68.83
	S	63.35	63.55	66.33	68.93	67.43	68.40	67.88	71.12	70.70	67.80	74.90
WATERNS												
Leakage	P	2.60	2.42	0.73	6.35	6.53	0.40	0.28	0.79	0.65	0.89	0.57
	S	3.01	2.83	0.84	7.30	7.25	0.46	0.31	0.93	0.76	1.02	0.68
Dynamic	P	4.08	3.95	5.18	10.65	22.90	3.91	3.27	8.67	2.97	4.12	8.11
	S	6.54	6.50	4.83	6.14	5.79	3.04	2.52	10.57	7.42	3.99	17.26
Total	P	6.68	6.37	5.91	17.00	29.43	4.31	3.55	9.46	3.61	5.01	8.67
	S	9.55	9.33	5.67	13.44	13.04	3.50	2.83	11.49	8.17	5.01	17.95
Max. Temperature	P	64.25	63.80	69.05	71.00	73.19	71.35	73.93	72.08	70.98	70.38	72.55
	S	63.58	63.85	66.75	69.55	67.89	68.98	68.40	71.93	71.47	68.33	76.00

Standard floorplans (see Figs. 1 and 2).

characteristics in terms of hotspots, whereas the shared-L2 architecture features only one hotspot per core (Load/Store Queue).

The hottest units are the FP ALU, the Ld/St queue, and the INT Issue Queue. Depending on the issue width, the temperature of the INT Queue and the Ld/St vary. For two and four issues, the FP ALU dominates, whereas as the issue width is scaled up, the INT Queue and the Ld/St become the hottest units of the die. This is due to the superlinear increase in power density in these structures.

TABLE 8
Shared L2: Average and Maximum Chip Temperature for Different Floorplans

L2	1MB		4MB	
	Max [°C]	Av. [°C]	Max [°C]	Av. [°C]
Lined up	76.0	63.2	77.0	62.0
Centered	75.5	63.6	76.5	62.1
Corner	75.9	63.5	76.9	62.1

Hotspot magnitude depends on issue width, that is, the power density of the microarchitecture blocks, and on the "compactness" of the floorplan. In the four-issue chip, temperature peaks higher than in the six-issue one exist. This is because the floorplan of the six-issue core has many cold units surrounding hotspots (for example, see the FP ALU hotspots). Interleaving hot and cold blocks is an effective method to provide spreading silicon to hot areas.

Thermal coupling of hotspots exists for various units, as shown in Fig. 12, and it is often caused by interprocessor interaction. For example, in the two-issue floorplan, the Ld/St queue of the left processor is thermally coupled with the FP ALU of the right one. In the four-issue, the Integer Execution Unit is warmed by the coupling between the LdSt queue and FP ALU of the two cores. For what concerns intraprocessor coupling, it can be observed for the four-issue design between the FP ALU and the FP register file, and in the eight-issue, it is between the INT Issue Queue and the LdSt queue.

6.4 Discussion

Different factors determine hotspots in the processors. Power density for each microarchitecture unit provides a proxy to temperature but does not suffice in explaining effects due to the thermal coupling and spreading area. It is important to care about the geometric characteristics of the floorplan. We summarize all those impacting on-chip heating as follows:

- *Proximity of hot units.* If two or more hotspots come close, this will produce thermal coupling and therefore locally raise the temperature.
- *Relative positions of hot and cold units.* A floorplan interleaving hot and cold units will result in lower global power density (therefore, lower temperature).
- *Available spreading silicon.* Units placed in such a position, which limits its spreading perimeter, will result in higher temperature, for example, the units placed in a corner of the die.

These principles, all related to the common idea of lowering the global power density, apply to the core microarchitecture and the CMP system architecture. In the first case, they suggest making not-too-close hot units, like register files, instruction queues, etc. For what concerns CMPs, they can be translated as follows:

- *Proximity of the cores.* The interaction between two cores placed side by side can generate thermal coupling between some units of the processors. For example, see Fig. 13a, showing the cross section for the centered layout of Fig. 9b. For *VOLREND*, the proximity of the cores causes approximately 2 °C increase in the hotspot.
- *Relative position of cores and caches.* If L2 caches are placed to surround the cores, this results in better heat spreading and lower chip temperature. For example (see Table 7), a centered floorplan has nearly 2 °C colder hotspots. In addition, caches can be heated by the core. Fig. 13a shows the longitudinal cross section of Fig. 10a. The effects of the hotspot in the core extend to the L1 caches (approximately between 2 and 3 mm), which experience a steep thermal gradient.
- *Position of cores in the die.* Processors placed at the center of the die offer better thermal behavior with respect to processors in the die corners or beside the die edge. For example, the effects of the edges can be observed in Fig. 13b for the backs of the processors

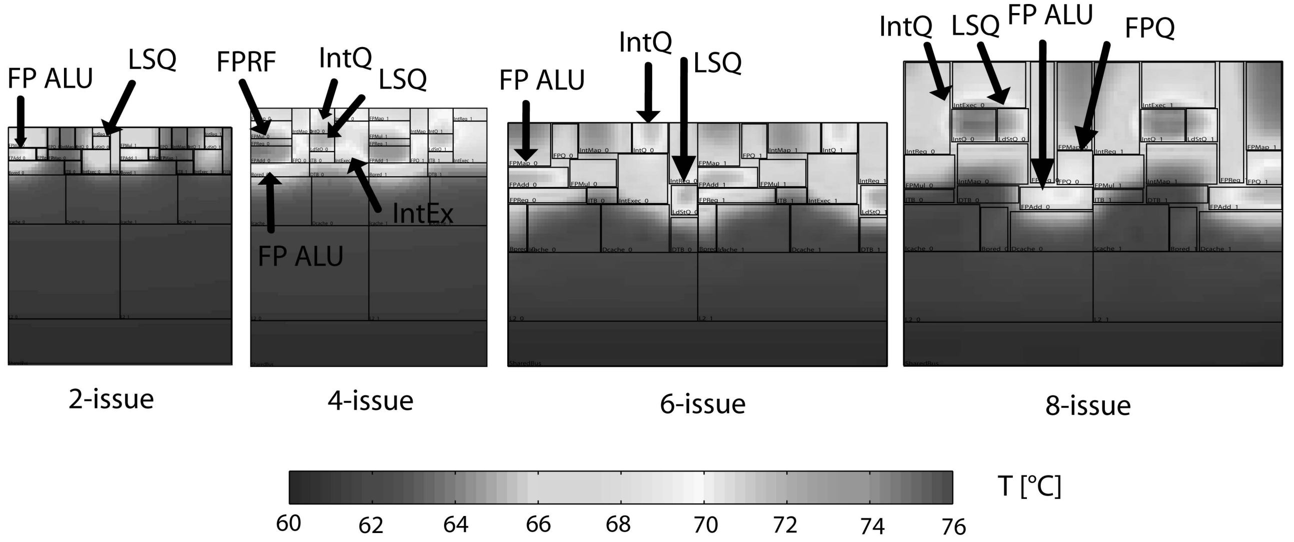
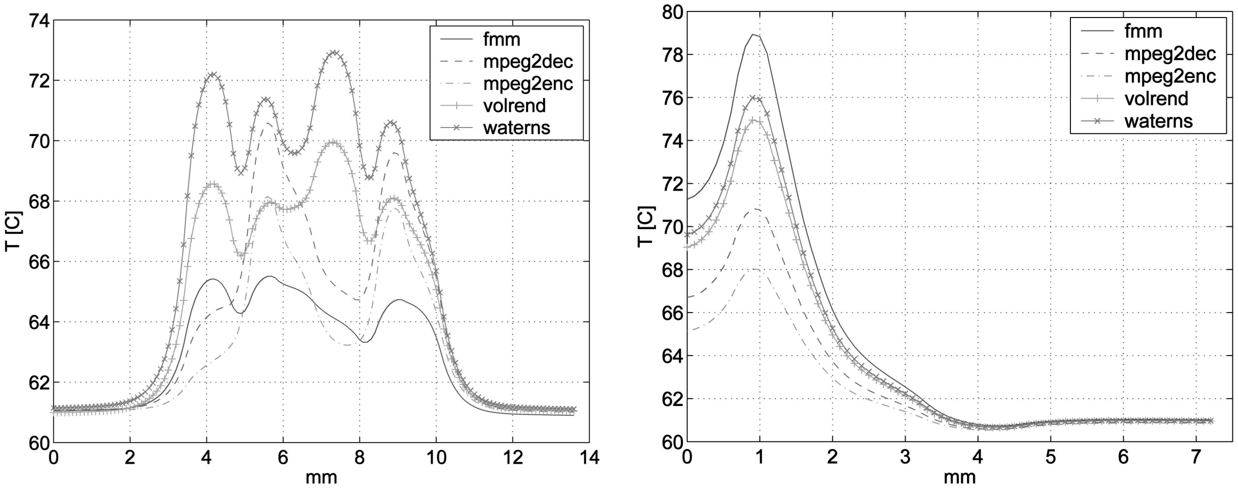


Fig. 12. Thermal maps of two-core 1-Mbyte L2 CMP for WATER-NS, with varying issue widths (from left to right: two, four, six, and eight issues).



(a)

(b)

Fig. 13. Two cross sections of thermal maps. (a) Latitudinal (along the 14-mm side) cross section of Fig. 9b. (b) Longitudinal (along the 8-mm side) cross section of Fig. 10a.

that are at 65-75 °C, instead of reaching the temperature of the L2 caches (like, for example, in Fig. 9b).

7 Conclusions and Future Work

Power/Thermal-aware design is one of the big issues to address in developing future processor architectures. In this paper, we discussed the impact of the choice of several architectural parameters of CMPs on power/performance/thermal metrics. Our conclusions apply to multicore architectures composed of processors with private L2 cache and running parallel applications.

We conducted an experimental evaluation of several shared-L2 and private-L2 CMPs, considering temperature, floorplan, and leakage effects. We showed that shared-L2 CMPs achieve better performance, energy, and EDP with respect to private-L2 architectures. For the explored design space, the optimal configuration (in terms of energy delay) has eight cores, four or six issue width (private L2/shared L2), and 1-Mbyte/4-Mbyte total L2 cache size. Nevertheless, the shared-L2 CMPs feature an important hotspot in the Load/store Queue, whereas this does not happen for private-L2 systems.

We investigated how the design of the floorplan affects the thermal behavior of the chip. We found that alternative floorplan topologies lead to little variation in the chip temperature (few degrees). For example, different placement of the L2 cache may determine 1/2 °C variations of the hotspots in the die. We show that a floorplan, where processors are surrounded by the L2 cache, is typically cooler by 1/2 °C. Efficiently handling large caches and hotspots, as suggested by the data shown in this paper, will be crucial for a competitive design.

Several other factors, not directly addressed in this paper, may affect design choices, such as area, yield, and reliability. These factors may, as well, affect the design choice. At the same time, orthogonal architectural alternatives such as heterogeneous cores, L3 caches, and complex interconnects might be present in future CMPs. Nevertheless, evaluating all these alternatives is left for future research.

References

[1] C. McNairy and R. Bhatia, "Montecito: A Dual-Core, Dual-Thread Itanium Processor," *IEEE Micro,* vol. 25, no. 2, pp. 10-20, 2005.

[2] P. Kongetira, K. Aingaran, and K. Olukotun, "Niagara: A 32-Way Multithreaded Sparc Processor," *IEEE Micro,* vol. 25, no. 2, pp. 21-29, 2005.

[3] B. Sinharoy, R.N. Kalla, J.M. Tendler, R.J. Eickemeyer, and J.B. Joyner, "Power5 System Microarchitecture," *IBM J. Research and Development,* vol. 49, no. 4, pp. 505-521, 2005.

[4] T. Mudge, "Power: A First Class Constraint for Future Architectures," *Proc. Sixth Int'l Symp. High-Performance Computer Architecture (HPCA),* 2000.

[5] K. Skadron, M.R. Stan, K. Sankaranarayanan, W. Huang, S. Velusamy, and D. Tarjan, "Temperature-Aware Microarchitecture: Modeling and Implementation," *ACM Trans. Architecture and Code Optimization,* vol. 1, no. 1, pp. 94-125, 2004.

[6] J. Srinivasan, S.V. Adve, P. Bose, and J.A. Rivers, "The Impact of Technology Scaling on Lifetime Reliability," *Proc. Int'l Conf. Dependable Systems and Networks (DSN '04),* p. 177, 2004.

[7] J. Srinivasan, S.V. Adve, P. Bose, and J.A. Rivers, "Exploiting Structural Duplication for Lifetime Reliability Enhancement," *Proc. 32nd Ann. Int'l Symp. Computer Architecture (ISCA '05),* pp. 520-531, 2005.

[8] W. Zhao and Y. Cao, "New Generation of Predictive Technology Model for Sub-45 nm Design Exploration," *Proc. Seventh Int'l Symp. Quality Electronic Design (ISQED '06),* pp. 585-590, 2006.

[9] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, "Leakage Current Mechanisms and Leakage Reduction Techniques in Deep-Submicrometer CMOS Circuits," *Proc. IEEE,* vol. 91, no. 2, pp. 305-327, Feb. 2003.

[10] R. Chau, S. Datta, M. Doczy, B. Doyle, B. Jin, J. Kavalieros, A. Majumdar, M. Metz, and M. Radosavljevic, "Benchmarking Nanotechnology for High-Performance and Low-Power Logic Transistor Applications," *IEEE Trans. Nanotechnology,* vol. 4, no. 2, pp. 153-158, Mar. 2005.

[11] "Superior Performance with Dual-Core," white paper, Intel, ftp://download.intel.com/products/processor/xeon/srvrplatformbrief.pdf, 2005.

[12] K. Quinn, J. Yang, and V. Turner, "The Next Evolution in Enterprise Computing: The Convergence of Multicore X86 Processing and 64-bit Operating Systems," white paper, Advanced Micro Devices Inc., Apr. 2005.

[13] B. McCredie, *POWER Roadmap.* IBM Corp., http://www2.hursley.ibm.com/decimal/ibm-power-roadmap-mccredie.pdf, 2006.

[14] S. Gochman, A. Mendelson, A. Naveh, and E. Rotem, "Introduction to Intel Core Duo Processor Architecture," *Intel Technology J.,* vol. 10, no. 2, pp. 89-98, 2006.

[15] J. Renau, B. Fraguela, J. Tuck, W. Liu, M. Prvulovic, L. Ceze, S. Sarangi, P. Sack, K. Strauss, and P. Montesinos, *SESC Simulator,* http://sesc.sourceforge.net, Jan. 2005.

[16] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A Framework for Architectural-Level Power Analysis and Optimizations," *Proc. 27th Int'l Symp. Computer Architecture (ISCA '00),* pp. 83-94, 2000.

[17] P. Shivakumar and N.P. Jouppi, "CACTI 3.0: An Integrated Cache Timing, Power, and Area Model," Compaq Technical Report 2001/2, Western Research Laboratory, 2001.

[18] W. Liao, L. He, and K. Lepak, "Temperature and Supply Voltage Aware Performance and Power Modeling at Microarchitecture Level," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems,* vol. 24, no. 7, pp. 1042-1053, July 2005.

[19] M. Monchiero, R. Canal, and A. Gonzalez, "Design Space Exploration for Multicore Architectures: A Power/Performance/Thermal View," *Proc. 20th Ann. Int'l Conf. Supercomputing (ICS '06),* 2006.

[20] J. Huh, D. Burger, and S. Keckler, "Exploring the Design Space of Future CMPs," *Proc. 10th Int'l Conf. Parallel Architectures and Compilation Techniques (PACT '01),* pp. 199-210, 2001.

[21] M. Ekman and P. Stenstrom, "Performance and Power Impact of Issue-Width in Chip-Multiprocessor Cores," *Proc. 2003 Int'l Conf. Parallel Processing (ICPP '03),* pp. 359-369, 2003.

[22] J. Li and J. Martinez, "Power-Performance Implications of Thread-Level Parallelism on Chip Multiprocessors," *Proc. Int'l Symp. Performance Analysis of Systems and Software (ISPASS '05),* pp. 124-134, 2005.

[23] J. Li and J.F. Martnez, "Power-Performance Considerations of Parallel Computing on Chip Multiprocessors," *ACM Trans. Architecture and Code Optimization,* vol. 2, no. 4, pp. 397-422, 2005.

[24] J. Li and J. Martinez, "Dynamic Power-Performance Adaptation of Parallel Computation On-Chip Multiprocessors," *Proc. 12th Int'l Symp. High Performance Computer Architecture (HPCA),* 2006.

[25] Y. Li, B. Lee, D. Brooks, Z. Hu, and K. Skadron, "CMP Design Space Exploration Subject to Physical Constraints," *Proc. 12th Int'l Symp. High Performance Computer Architecture (HPCA),* 2006.

[26] L. Hsu, R. Iyer, S. Makineni, S. Reinhardt, and D. Newell, "Exploring the Cache Design Space for Large-Scale CMPs," *SIGARCH Computer Architecture News,* vol. 33, no. 4, pp. 24-33, 2005.

[27] R. Kumar, V. Zyuban, and D.M. Tullsen, "Interconnections in Multi-Core Architectures: Understanding Mechanisms, Overheads and Scaling," *Proc. 32nd Ann. Int'l Symp. Computer Architecture (ISCA '05),* pp. 408-419, 2005.

[28] Y. Li, D. Brooks, Z. Hu, and K. Skadron, "Performance, Energy, and Thermal Considerations for SMT and CMP Architectures," *Proc. 11th Int'l Symp. High-Performance Computer Architecture (HPCA '05),* pp. 71-82, 2005.

[29] J. Donald and M. Martonosi, "Techniques for Multicore Thermal Management: Classification and New Exploration," *Proc. 33rd Int'l Symp. Computer Architecture (ISCA '06),* pp. 78-88, 2006.

[30] P. Chaparro, G. Magklis, J. Gonzalez, and A. Gonzalez, "Distributing the Front End for Temperature Reduction," *Proc. 11th Int'l Symp. High-Performance Computer Architecture (HPCA '05),* pp. 61-70, 2005.

[31] K. Sankaranarayanan, S. Velusamy, M. Stan, and K. Skadron, "A Case for Thermal-Aware Floorplanning at the Microarchitectural Level," *J. Instruction-Level Parallelism,* http://www.jilp.org/vol7/v7paper15.pdf, Oct. 2005.

[32] J.C. Ku, S. Ozdemir, G. Memik, and Y. Ismail, "Thermal Management of On-Chip Caches through Power Density Minimization," *Proc. 38th Ann. IEEE/ACM Int'l Symp. Microarchitecture (MICRO '05),* pp. 283-293, 2005.

[33] R.E. Kessler, "The Alpha 21264 Microprocessor," *IEEE Micro,* vol. 19, no. 2, pp. 24-36, Mar./Apr. 1999.

[34] B.M. Beckmann and D.A. Wood, "Managing Wire Delay in Large Chip-Multiprocessor Caches," *Proc. 37th Ann. IEEE/ACM Int'l Symp. Microarchitecture (MICRO '04),* pp. 319-330, 2004.

[35] S.C. Woo, M. Ohara, E. Torrie, J.P. Singh, and A. Gupta, "The SPLASH-2 Programs: Characterization and Methodological Considerations," *Proc. 22nd Ann. Int'l Symp. Computer Architecture (ISCA '95),* pp. 24-36, 1995.

[36] M.-L. Li, R. Sasanka, S.V. Adve, Y.-K. Chen, and E. Debes, "The ALPBench Benchmark Suite for Complex Multimedia Applications," *Proc. IEEE Int'l Symp. Workload Characterization (IISWC '05),* 2005.

[37] S. Palacharla, N.P. Jouppi, and J.E. Smith, "Complexity-Effective Superscalar Processors," *Proc. 24th Ann. Int'l Symp. Computer Architecture (ISCA '97),* pp. 206-218, 1997.

[38] H.-S. Wang, X. Zhu, L.-S. Peh, and S. Malik, "Orion: A Power-Performance Simulator for Interconnection Networks," *Proc. 35th Ann. ACM/IEEE Int'l Symp. Microarchitecture (MICRO '02),* pp. 294-305, 2002.

[39] J.A. Butts and G.S. Sohi, "A Static Power Model for Architects," *Proc. 33rd Ann. ACM/IEEE Int'l Symp. Microarchitecture (MICRO '00),* pp. 191-201, 2000.

[40] A. Grove, "Changing Vectors of Moore's Law," keynote speech, *Proc. Int'l Electron Devices Meeting (IEDM '02),* http://www.intel.com/pressroom/archive/speeches/grove_20021210.htm, 2002.

[41] *The International Technology Roadmap for Semiconductors,* http://www.itrs.net/Common/2005ITRS/Home2005.htm, 2005.

[42] V. Zyuban and P.N. Strenski, "Balancing Hardware Intensity in Microprocessor Pipelines," *IBM J. Research and Development,* vol. 47, nos. 5-6, pp. 585-598, 2003.

Matteo Monchiero received the MS and PhD degrees from the Politecnico di Milano, Milano, Italy, in 2003 and 2007, respectively. During the academic year 2005-2006, he was a visiting student at the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain. Since April 2007, he has been a postdoctoral research associate in the Advanced Architecture Laboratory, Hewlett-Packard Laboratories, Palo Alto, California. He served as a reviewer for international symposia and journals, including the Design Automation Conference (DAC), Conference on Design, Automation and Test in Europe (DATE), International Symposium on Computer Architecture (ISCA), and various IEEE Transactions. His research interests include multiprocessor/multicore architectures, simulation technologies, and low-power/thermal-aware microarchitectures. He has an extensive list of publications on computer architecture and VLSI design. He is a member of the IEEE.

Ramon Canal received the MS and PhD degrees from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, and the MS degree from the University of Bath, United Kingdom. He was with Sun Microsystems in 2000. He joined the faculty of the Computer Architecture Department, UPC, in 2003. During the school year 2006-2007, he was a Fulbright visiting scholar at Harvard University. His research interests are power-aware and thermal-aware architectures, and reliability. He has an extensive list of publications and several invited talks. He was a reviewer for more than 60 program committees of international symposia and journals on computer architecture, including the International Symposium on Computer Architecture (ISCA), the Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), the Symposium on High-Performance Computer Architecture (HPCA), the International Conference on Parallel Architectures and Compilation Techniques (PACT), the International Conference on Supercomputing (ICS), the International Conference on Computer Design (ICCD), the IEEE International Symposium on Performance Analysis of Software and Systems (ISPASS), the International Conference on Compilers, Architecture, and Synthesis for Embedded Systems (CASES), the International Parallel and Distributed Processing Symposium (IPDPS), various IEEE Transactions, *IEEE Micro*, and *the ACM Transactions on Architecture and Code Optimization*, among others. He was a member of the program committee for ICCD 2007, the 13th International Conference on Parallel and Distributed Systems (ICPADS 2007), CF 2007, ICCD 2006, and ICPADS 2006. He is a member of the IEEE and the IEEE Computer Society.

Antonio González received the MS and PhD degrees from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain. He joined the faculty of the Computer Architecture Department, UPC, in 1986 and became a full professor in 2002. He is the founding director of the Intel-UPC Barcelona Research Center, which started in 2002 and whose research focuses on new microarchitecture paradigms and code generation techniques for future microprocessors. He has published more than 200 papers, has given more than 80 invited talks, is the holder of more than 20 pending patents, and has advised 13 PhD dissertations on computer architecture and compilers. He is an associate editor for the *IEEE Transactions on Computers*, the *IEEE Transactions on Parallel and Distributed Systems*, the *ACM Transactions on Architecture and Code Optimization*, and the *Journal of Embedded Computing*. He has served on more than 100 program committees for international symposia on computer architecture, including the International Symposium on Computer Architecture (ISCA), the Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), the Symposium on High-Performance Computer Architecture (HPCA), the International Conference on Parallel Architectures and Compilation Techniques (PACT), the International Conference on Supercomputing (ICS), the International Conference on Computer Design (ICCD), the IEEE International Symposium on Performance Analysis of Software and Systems (ISPASS), the International Conference on Compilers, Architecture, and Synthesis for Embedded Systems (CASES), and the International Parallel and Distributed Processing Symposium (IPDPS). He has been a program chair or cochair of ICS 2003, ISPASS 2003, MICRO 2004, and HPCA 2008, among other symposia. He is a member of the IEEE and the IEEE Computer Society.