Token3D: Reducing Temperature in 3D Die-Stacked CMPs through Cycle-Level Power Control Mechanisms

Juan M. Cebrián[1], Juan L. Aragón[1], and Stefanos Kaxiras[2]

[1] University of Murcia, Spain
{jcebrian,jlaragon}@ditec.um.es
[2] University of Uppsala, Sweden
kaxiras@it.uu.se

Abstract. Nowadays, chip multiprocessors (CMPs) are the new standard design for a wide range of microprocessors: mobile devices (in the near future almost every smartphone will be governed by a CMP), desktop computers, laptop, servers, GPUs, APUs, etc. This new way of increasing performance by exploiting parallelism has two major drawbacks: off-chip bandwidth and communication latency between cores. 3D die-stacked processors are a recent design trend aimed at overcoming these drawbacks by stacking multiple device layers. However, the increase in packing density also leads to an increase in power density, which translates into thermal problems. Different proposals can be found in the literature to face these thermal problems such as dynamic thermal management (DTM), dynamic voltage and frequency scaling (DVFS), thread migration, etc. In this paper we propose the use of microarchitectural power budget techniques to reduce peak temperature. In particular, we first introduce Token3D, a new power balancing policy that takes into account temperature and layout information to balance the available per core power along other power optimizations for 3D designs. And second, we analyze a wide range of floorplans looking for the optimal temperature configuration. Experimental results show a reduction of the peak temperature of 2-26ºC depending on the selected floorplan.

Keywords: Power budget, power tokens, DVFS, power balancing.

1 Introduction

With the global market dominated by chip multiprocessors and the GHz race over, designers look for ways to increase productivity by increasing the number of available processing cores inside the CMP. The shrinking of transistor's feature size allows the integration of more cores, as the per-core power consumption decreases with each new generation. However, interconnects have not followed the same scaling trend as transistors, becoming a limiting factor in both performance and power consumption. One intuitive solution to reduce wirelength of the interconnection network is to stack structures on top of each other, instead of using a traditional planar distribution.

Introduced by Souri *et al.* in [22], 3D architectures stack together multiple device layers (i.e., cores, memory) with direct vertical interconnects through them

E. Jeannot, R. Namyst, and J. Roman (Eds.): Euro-Par 2011, LNCS 6852, Part I, pp. 295–309, 2011.

(inter-wafer vias or die-to-die vias). A direct consequence of this design is the reduction on the communication delays and power costs between different cores, as well as an increase in packing density that depends on the number of available layers. However, despite of the great benefits of 3D integration, there are several challenges that designers have to face. First, the increase in packing density also leads to an increase in power density that eventually translates into thermal problems. Second, a deeper design space exploration of different floorplan configurations is essential to take advantage of these emerging 3D technologies. Third, chip verification complexity increases with the number of layers.

To face the first challenge there are several proposals that come from the 2D field:

- *Dynamic Voltage and Frequency Scaling* (DVFS) to reduce power consumption, and thus temperature. DVFS-based approaches can be applied either to the whole 3D chip or only to cores that show thermal problems (usually cores away from the edges of the 3D chip) [1][13][20].
- Task/thread migration to move execution threads from internal to external cores whenever possible, or reschedule memory intensive threads to internal cores and CPU intensive threads to external cores [6][24][7].

These mechanisms are usually triggered by a *Dynamic Thermal Management* (DTM) scheme, so whenever a core exceeds a certain temperature, power control or task migration mechanisms take place inside the CMP. However, these mechanisms are not perfect. DVFS is a coarse-grain mechanism usually triggered by the operating system with very long transition times between power modes that leads to a high variability in temperature. On the other hand, task migration, despite the fact that it can be applied at a finer granularity (i.e., faster) than DVFS, has the additional overhead of warming up both the cache and the pipeline of the target core. Moreover, none of these mechanisms affects leakage power consumption. Leakage (or static power) is something that many studies do not take into consideration when dealing with temperature, but it cannot be ignored. For current technologies (32nm and below), even with gate leakage under control by using *high-k* dielectrics, subthreshold leakage has a great impact in the total power consumed by processors. Furthermore, leakage depends on temperature, so it is crucial to add a leakage-temperature loop to update leakage consumption in real time depending on the core/structure's temperature.

Therefore, in order to accurately control peak temperature, which is of special interest in 3D-stacked processors as this integration technology exasperates thermal problems, a much tighter control is necessary to restraint the power consumption of the different cores. Recently, Cebrian *et al.* proposed the use of a hybrid mechanism to match a predefined power budget [4][5]. This mechanism accurately matches a power budget and ensures minimal deviation from the target power and the corresponding temperature, by first using DVFS to lower the average power consumption towards the power budget and then removing power spikes by using microarchitectural mechanisms (e.g., pipeline throttling, confidence estimation on branches, critical path prediction, etc).

In this paper we make three major contributions. First, we analyze the effects of cycle-level accurate power control mechanisms to control peak temperature in 3D die-stacked processors. Based on this analysis we propose *Token3D*, a novel power

balancing mechanism that takes into account temperature and layout information when balancing power among cores and layers. Second, we analyze a wide range of floorplan configurations looking for the optimal temperature configuration, taking into account both dynamic and leakage power (as well as the leakage-temperature loop). And third, we include some specific power control mechanisms for vertical 3D floorplans. Experimental results show a reduction of the peak temperature of 2-26ºC depending on the selected floorplan when including cycle-level power control mechanisms into the 3D die-stacked design. Summarizing, the main contributions of the present work are the following:

- Reducing the peak temperature through power control mechanisms:
  - Implementation and analysis of power balancing mechanisms on 3D die-stacked architectures to minimize hotspots.
  - Introduction of a new policy to balance power among cores, *Token3D*. This policy will use layout and temperature information to distribute the available power among the different cores and layers, giving more work to cool cores and cores close to edges than to internal cores.
- Temperature analysis of the main 3D design choices:
  - Analysis of different 3D floorplan designs using accurate area, power (both static and dynamic) and heatsink information.
  - Analysis of the effects of ROB resizing [18] on temperature for vertical designs.
  - Temperature analysis when using ALUs with different physical properties (energy-efficient *vs.* low latency ALUs) on the same layout.
  - Implementation and analysis of a hybrid floorplan design (vertical+ horizontal).

The rest of this paper is organized as follows. Section 2 provides some background on power-saving techniques for CMPs and 3D die-stacked multicores. Section 3 describes the proposed Token3D approach. Section 4 describes our simulation methodology and shows the main experimental results. Finally, section 5 shows our concluding remarks.

2 Background and Related Work

In this section we will introduce the main power and thermal control mechanisms as well as an overview on 3D die-stacked processors along with the different floorplan design choices.

2.1 Power and Thermal Control in Microprocessors

2.1.1 Dynamic Voltage Frequency Scaling (DVFS)

Dynamic Voltage and Frequency Scaling (DVFS) has been, for the past 20 years, one of the most common mechanisms to reduce power consumption in microprocessors. Introduced in [13], DVFS takes advantage of transistor quadratical dependence on supply voltage and linear dependence on frequency ($P = V_{DD}^2 \times f$) and downscales

both voltage and frequency to save power. However, as the process technology scales down, the margin between V_{DD} (supply voltage) and V_T (threshold voltage) is reduced, decreasing the processor's reliability among other undesirable effects. Furthermore, the transistor's delay (or switching speed) depends on $\delta \approx 1/ (V_{DD} - V_T)^{\alpha}$, with $\alpha > 1$. That means that V_{DD} can lowered as long as the margin between V_{DD} and V_T is kept constant (i.e., V_T must be lowered accordingly). However, the counterpart of reducing V_T is twofold: a) leakage power increases as it exponentially depends on V_T [8]; and b) processor reliability is further reduced.

In the CMP field, Isci *et al.* [1] and later Sartori *et al.* [20] proposed DVFS-based power control mechanisms specifically designed for single-threaded applications. These proposals switch between different DVFS power modes trying to maximize throughput under certain power constraints. Unfortunately, as they rely on the use of performance counters and/or time estimation, these proposals only work properly for multiprogrammed or single-threaded applications, because in parallel applications synchronization points may increase global execution time although local core performance counters show a performance increase (due to spinning).

2.1.2 Dynamic Thermal Management (DTM)

As mentioned before, temperature is the main drawback in 3D die-stacked designs. In 2001, Brooks and Martonosi [3] introduced *Dynamic Thermal Management* (DTM) mechanisms in microprocessors. In that work they explore performance trade-offs between different DTM mechanisms trying to tune up the thermal profile at runtime. Thread migration [21], fetch throttling [6], clock gating or distributed dynamic voltage scaling [9] are techniques that can be used by DTM mechanisms. For the thermal management of 3D die-stacked processors, most of the prior work has addressed design stage optimization, such as thermal-aware floorplanning (as in [10]). In [24], the authors evaluate several policies for task migration and DVS specifically designed for 3D architectures. Something similar is done in [7], where the authors explore a wide range of different floorplan configurations using clock gating, DVFS and task migration to lower peak temperature.

However, both thread migration and DVFS-based approaches exhibit really low accuracy when matching a target power budget, and thus a high deviation from the target temperature. So the designers have two choices, either to increase the power constraint to ensure the target temperature or to use a more accurate way to match the desired (if needed) power budget and temperature. In order to do this we first need a way to measure power accurately, because up to now power was estimated by using performance counters, although the new Intel Sandy Bridge processors include some MSRs (machine specific registers) that can be used to retrieve power monitoring information from different processor structures.

2.1.3 Measuring Power in Real-Time

Power tokens were introduced in 2009 [4] as a way to approximate the power being consumed by the processor at a cycle level. The dynamic power consumed by an instruction can be estimated at commit stage by adding, to the base power consumption of the instruction (i.e., all regular accesses to structures done by that instruction which are known *a priori*), a variable component that depends on the time it spends in the pipeline. A *power token* unit is defined as the joules consumed by one

instruction staying in the instruction window for one cycle. The number of *power tokens* consumed by an instruction will be calculated as the addition of its base *power tokens* plus the number of cycles it spends in the instruction window. As in [4][5], the implementation of the *Power Token* approach is done by means of an 8K-entry history table (Power Token History Table – PTHT), accessed by PC, which stores the power cost (in tokens) of each instruction's last execution. The PTHT is updated with the current number of *power tokens* consumed when an instruction commits. Hence, the overall processor power consumption in a given cycle can be easily estimated based on the instructions that are traversing the pipeline without using performance counters just by accumulating the *power tokens* (provided by the PTHT) of each instruction being fetched.

2.1.4 Hybrid Power Control Approaches

Along with *power tokens*, in [4] we introduced a two-level approach that firstly applies DVFS as a coarse-grain approach to reduce power consumption towards a predefined power budget, and secondly chooses between different microarchitectural techniques to remove the remaining and numerous power spikes. The second-level mechanism depends on how far the processor is over the power budget in order to select the most appropriate microarchitectural technique.

However, previous approaches failed to match the target power budget when considering the execution of parallel workloads in a CMP processor. Very recently, we have proposed *Power Token Balancing* (PTB) [5]. This mechanism will balance the power between the different cores of a 2D CMP to ensure a given power constraint (or budget) with minimal energy and performance degradation. Based in *power token* accounting, this proposal uses a PTB *load-balancer* as a centralized structure that receives and sends power information (measured as *power tokens*) from cores under the power budget to cores over the power budget. Tokens are used as a currency to account for power, so it is important to note that they are neither sent nor received, cores just send the number of spare tokens. PTB will benefit from any power unbalance between cores. Note that task migration mechanisms are orthogonal to PTB and can be applied together for further temperature reductions.

2.2 Building a 3D Die-Stacked Processor

In order to build a 3D die-stacked processor we need to decide two things: how we build and put together the different layers and how we establish the communication between them. There are two main approaches to build the layers: the bottom-up and the top-down approaches. The first approach involves a sequential device process. The frontend processing is repeated on a single wafer to build multiple active layers before creating interconnects among them. The second approach processes each layer separately (wafer-to-wafer), using conventional techniques, and then assembles them using wafer-bonding technology. Once we have built the different layers we need to establish communications between them. There are various vertical interconnect technologies that have been explored, including wire bonded, microbump, contactless (capacitive or inductive), and through-via vertical interconnect. A comparison in terms of vertical density and practical limits can be found in [23][24].

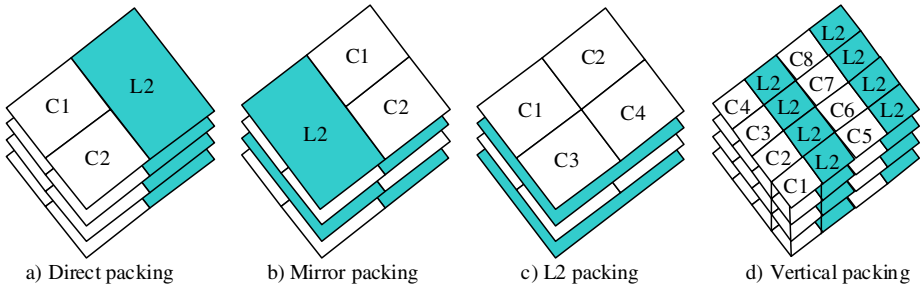

Fig. 1. Core distribution along the layers

2.3 3D Integration Technology

From the previously introduced technologies, wafer-to-wafer bonding appears to be the most promising approach [2] and there are many recent publications that have chosen this type of 3D stacking technology [12][14][16]. Therefore, this is the integration approach we are going to follow in this paper.

Now there are multiple choices on how cores are distributed along the different layers, which are shown in Figure 1. We can clearly indentify two trends; either build the cores vertical or horizontal. Horizontal distributions (a-c) are the most common choices in literature, as they are easier to implement and validate. On the other hand, vertical designs (Figure 1-d), introduced by Puttaswamy *et al.* in [19], offer improved latency and power reduction compared to horizontal designs. However, they supposed an inter-layer communication latency to be in the order of one FO4, and current technologies can do 9-12 FO4 in one cycle. Therefore, in their proposal inter-layer communication could be done in less than one cycle while other papers claim that inter-layer communication takes as long as an off-chip memory access [23]. Furthermore, vertical designs require really accurate layer alignment to match a structure split in different layers, and that is far from the current technology status. However, as a possible future implementation of 3D die-stacked processors we also evaluate these floorplans in this paper, and for comparative purposes, we also assume one FO4 interconnection delay for our evaluation of vertical designs (10µm length wires between layers).

3 Thermal Control in 3D Die-Stacked Processors

3.1 *Token3D*: Balancing Temperature on 3D-Staked Designs

As cited before, *Power Token Balancing* (PTB) is a global balancing mechanism to restrain power consumption up to a preset power budget [5]. One of the main goals of this paper is to analyze the effects of the original PTB approach in 3D die-stacked architectures. We will also propose a novel policy, *Token3D*, aimed at distributing the power among cores and/or dies that are over their local power budget. *Token3D* will give priority to cooler cores, usually located close to the edges/surface of the 3D stack. By prioritizing those cores, *Token3D* balances not only power but also

temperature, as cool cores will work more than the rest of cores, balancing the global CMP temperature. Once a cool core gets to a synchronization point or to a low computation phase (i.e., low IPC due to a misprediction event) it will naturally cool down again, acting like a heatsink to hotter cores located beneath it in the 3D stack.

3.2 *Token3D* Implementation Details

Token3D is a new policy on how PTB splits the available *power tokens*, given by cores under the power budget to the PTB load-balancer, among the cores that are over the power budget (details about *power tokens* and the PTB approach are covered in sections 2.1.3 and 2.1.4). Basically, *Token3D* will create N buckets, where N represents the amount of layers of our 3D die-stacked processor. Then the PTB load-balancer will place the coolest core in bucket *one* and will distribute the rest of the cores between the available buckets in increments of 5% in temperature. So, cores that have a difference between 0 and 5% in temperature with respect to the coolest core will be placed in the same bucket; cores between 5% and 10% will be placed on

Table 1. Simulated CMP configuration

Processor Core	
Process Technology:	32 nanometres
Frequency:	3000 MHz
VDD:	0.9 V
Instruction Window:	128 entries + 64 LsQ
Decode Width:	4 inst/cycle
Issue Width:	4 inst/cycle
Functional Units:	6 Int Alu; 2 Int Mult 4 FP Alu; 4 FP Mult
Pipeline:	14 stages
Branch Predictor:	64KB, 16 bit Gshare
Memory Hierarchy	
Coherence Protocol:	MOESI
Memory Latency:	300 Cycles
L1 I-cache:	64KB, 2-way, 1 cycle lat.
L1 D-cache:	64KB, 2-way, 1 cycle lat.
L2 cache:	2MB/core, 4-way, unified, 12 cycles latency
Network Parameters	
Topology:	2D mesh
Link Latency:	4 cycles
Flit size:	4 bytes
Link Bandwidth:	1 flit / cycle

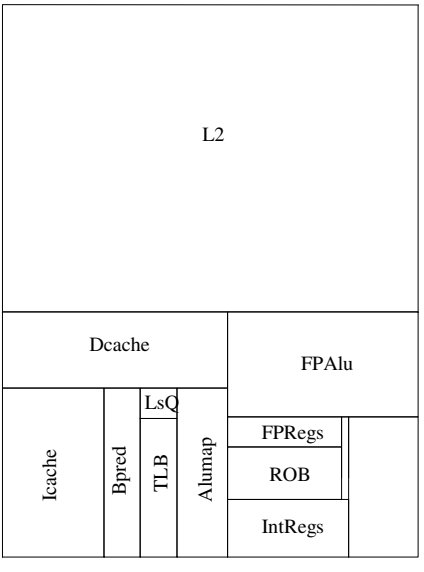

Fig. 2. Core floorplan

Table 2. Evaluated benchmarks and input working sets

	Benchmark	**Size**	**Benchmark**	**Size**
SPLASH-2	Barnes	8192 bodies, 4 time steps	Raytrace	Teapot
	Cholesky	tk16.0	Water-NSQ	512 molecules, 4 time steps
	FFT	256K complex doubles	Water-SP	512 molecules, 4 time steps
	Ocean	258x258 ocean	Tomcatv	256 elements, 5 iterations
	Radix	1M keys, 1024 radix	Unstructured	Mesh.2K, 5 time steps
PARSEC	Blackscholes	simsmall	Swaptions	Simsmall
	Fluidanimate	simsmall	x264	Simsmall

the next bucket; and so on until N. Note that this process does not need to be done at a cycle level, as temperature does not change so quickly. In our case, this process is performed every 100K-cycles. For example, in a four layer 3D-stacked processor, if the coolest core has an average temperature of 70ºC, bucket *one* will hold cores with temperatures between 70ºC and 73.5ºC, bucket *two* will hold cores with temperate between 73.5ºC and 77ºC, bucket *three* 77ºC to 80.5ºC and bucket *four* any core over 80.5ºC.

Once we have identified the cores that are over the power budget (those that did not provide any tokens to the PTB load-balancer), the load balancer will distribute the *power tokens* between the active buckets (i.e., the buckets that have cores over the power budget) in an iterative way, giving extra tokens depending on the bucket the core is in. For a 4-layer design, the bucket that holds the hottest core will have a *x1* multiplier on the number of received tokens, while the coolest bucket will have a *x4* multiplier on the amount of received tokens. For example, if buckets 1, 2 and 3 are active (being 1 the one that holds the coolest cores), all the cores will receive one token, cores in buckets 2 and 1 will receive a second token and, finally, cores in bucket 1 will receive a third token. If there are any *power tokens* left, we repeat the process.

4 Experimental Results

In this section we will evaluate both the original PTB and the novel *Token3D* approaches as mechanisms to control temperature in a 3D die-stacked CMP. In addition, we will evaluate some specific optimizations for a vertical design that uses a custom floorplan where hotspot structures have been placed in the upper (cooler) layers whereas cooler structures are placed in lower layers. We will also analyze the different floorplan organizations in order to minimize peak temperature in the 3D die-stacked architecture. For our evaluation the selected power budget is 50% of the original power consumption of the processor.

4.1 Simulation Environment

For evaluating the proposed approache we have used the Virtutech Simics platform extended with Wisconsin GEMS v2.1 [17]. GEMS provides both detailed memory simulation through a module called Ruby and a cycle-level pipeline simulation through a module called Opal. We have extended both Opal and Ruby with all the studied mechanisms that will be explained next. The simulated system is a homogeneous CMP consisting of a number of replicated cores connected by a switched 2D-mesh direct network. Table 1 shows the most relevant parameters of the simulated system. Power scaling factors for a 32nm technology were obtained from McPAT [13]. To evaluate the performance and power consumption of the different mechanisms we used scientific applications from the SPLASH-2 benchmark suite in addition to some PARSEC applications (the ones that finished execution in less than 5 days in our cluster). Results have been extracted from the parallel phase of each benchmark. Benchmark sizes are specified in Table 2.

3D thermal modeling can be accomplished using an automated model that forms the RC circuit for given grid dimensions. For this work we have ported HotSpot 5.0 [21] thermal models into Opal and have built our tiled CMP by replicating N times our customized floorplan, where N is the number of cores. Figure 2 shows the base floorplan design we have chosen. This floorplan was obtained from Hotfloorplaner (provided by the Hotspot 5.0). Our resulting CMP will be composed of a varying number of these cores (from 2 to 16). As cited before, we will assume an interconnection delay between layers of one FO4 (10μm length wires, as in [19]).

Moreover, thermal hotspots increase cooling costs and have a negative impact on reliability and performance. The significant increase in cooling costs requires designs for temperature margins lower than the worst-case. Leakage power is exponentially dependent on temperature, and an incremental feedback loop exists between temperature and leakage, which may turn small structures into hotspots and potentially damage the circuit. High temperatures also adversely affect performance, as the effective operating speed of transistors decreases as they heat up. In this paper we model both leakage (through McPAT) and the leakage/temperature loop in Opal, so leakage will be updated on every Hotspot exploration window (10K cycles). Leakage power is translated into power tokens and updated according to the formula $L_{new} = L_{Base}\ x\ e^{Leak\_Beta\ x\ (TCurrent\ -\ TBase)}$ where *Leak_Beta* depends on technology scaling factor and is provided by HotSpot 5.0, L_{new} is the updated leakage, L_{Base} is the base leakage (obtained using McPAT), *TCurrent* is the current temperature and *TBase* is the base temperature. Once leakage is updated, it is translated back to *power tokens*.

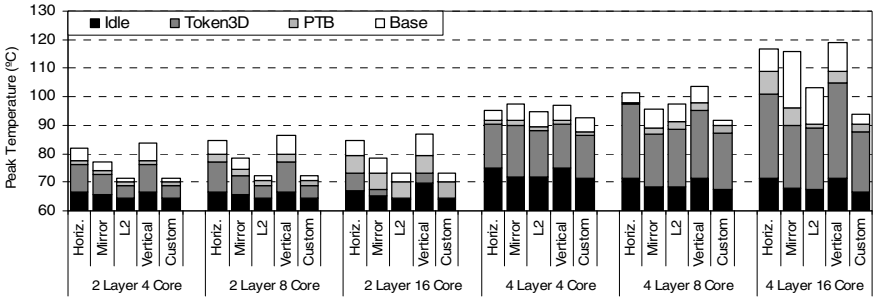

Fig. 3. Peak temperature for PTB, Token3D and the base case for different floorplans and core configurations

Another important parameter is the cooling system. The regular thermal resistance of a cooling system ranges from 0.25 K/W for the all-copper fan model at the highest speed setting (very good), to 0.33 K/W for the copper/aluminum variety at the lowest setting. In this work we model a real-world Zalman CNPS7700-Cu heatsink with 0.25 K/W thermal resistance and an area of 3.268 cm^2 (136mm side).

4.2 Effects of *Token3D* on Peak Temperature

Figure 3 shows the peak temperature for different floorplan configurations and a varying number of cores (from 4 to 16) using stacked bars. The reported *idle*

temperature corresponds to the average idle temperature of the cores[1]. The studied floorplans are: Horizontal (Figure 1.a), Mirror (Figure 1.b), L2 (Figure 1.c), Vertical (Figure 1.d) and Custom. As cited before, this last floorplan corresponds to a new configuration that places hotspots into upper layers of the 3D stack, giving more chances for them to cool down, and will be further discussed later in the next subsection. In Figure 3 we can clearly see that both L2 and Custom are the best designs to reduce peak temperature of the processor. This is due to the fact that both designs place the L2 in lower layers, and, as it can be seen in Figure 4, the L2 is the coolest structure within a core, even though we are accounting for leakage to calculate temperature. This placement leaves hotspots close to the surface and hot structures can cool down easily. We can also see that even a simple change in the floorplan such as mirroring between layers gives substantial temperature reduction (5-6ºC) compared to the horizontal design.

When considering the vertical design we can observe a higher peak temperature than the horizontal one. This vertical design was introduced in [19] by Puttaswamy *et al.* along with a dynamic power saving mechanism, *Thermal Herding*, that disables layers at runtime, depending on the number of bits used by the different instructions. This vertical design assumes each structure is vertically implemented across all layers. In our evaluation of this vertical design, the area occupied by each structure and its power consumption is divided by the number of available layers, but we do not disable any layer, to isolate our proposed power control mechanisms from the benefits obtained by *Thermal Herding*. For instance, in a 4-layer vertical design the implemented thermal model calculates the temperature of a structure in layer *i* by considering one fourth of its original power and area, however, the fraction of that structure is stacked on top of another equal portion of the same structure, with all portions simultaneously accessed, and therefore, increasing temperature. Note, however, that the use of *thermal herding* and its ability to disable unused layers for the vertical design is orthogonal to the use of our proposed PTB and *Token3D* approaches.

When it comes to the studied power control mechanisms both the original PTB and *Token3D* are able to reduce peak temperature by 2-26ºC depending on the floorplan configuration. *Token3D* is always 1-3% better than the original PTB balancing mechanism. We must also note that, as we get closer to the idle temperature, any temperature reduction comes at a higher performance degradation.

Figure 4-left shows a more detailed analysis on the effects of both PTB and *Token3D* in the peak temperature of the different core structures. We selected the most probable configuration for 3D die-stacked cores (Mirror, Figure 1.b) and a 4-layer 16-core CMP for this *per structure* temperature analysis. As cited before, PTB and *Token3D* are evaluated with a preset power budget of 50% of the original average power consumption. For comparison purposes we also evaluate DVFS trying to match the same target power budget of 50%. Figure 4-left helps us to locate our design hotspots (I-cache, TLB, Branch predictor, Load store queue) and see how both cycle-level power control mechanisms are able to reduce peak temperature by 20-36%. For example, the I-cache goes from 150ºC down to 110ºC, 30ºC less than DVFS. We can

[1] We define "idle" temperature as the temperature of the whole CMP in idle state (i.e., only the operating system is running).

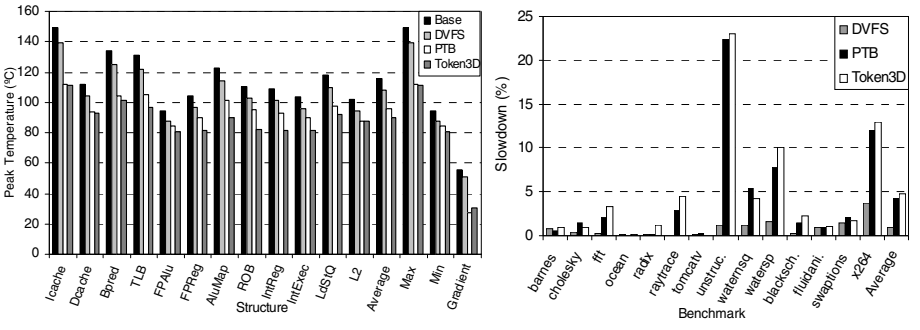

Fig. 4. Peak temperature (left) and performance (right) of a 4-layer 16-core CMP using the mirror floorplan

also see that, on average for this selected design, *Token3D* is 5-6ºC better than regular PTB. It is also important to note that our cycle-level mechanisms are able to reduce all hotspots peak temperature and put them close to the average core temperature. This last result is specially interesting on 3D architectures, as they exacerbate thermal problems and a much tighter power control is necessary. This is the benefit we expected from the highly accurate power budget matching our mechanisms provide, that ensures minimal deviation from the target power budget and, therefore, temperature. In Figure 4-left we also show the *spatial gradient* (temperature difference between the hottest and coolest structure of the core). Reducing spatial gradients is important because they can cause clock skew and impact circuit delay [1]. In particular, both PTB and *Token3D* are able to reduce this gradient by more than 50%, from 50ºC to 22ºC.

In terms of performance degradation (Figure 4-right), regular PTB behaves slightly better than *Token3D*, as power is equally divided between all cores and they can get to the next synchronization point more evenly, while *Token3D* will unbalance cores and make them wait at the synchronization point more time.

4.3 Further Temperature Optimizations

In addition to the PTB temperature analysis and the introduction of *Token3D* we also wanted to perform some optimizations for the vertical 3D die-stacked layout. More specifically, we will analyze the effects on peak temperature of MLP-based instruction window (ROB) resizing [18] and ALU selection based on instruction criticality (from ALUs placed on different layers) while varying the number of cores.

Figure 5 shows the effects on peak temperature of different instruction window (IW) sizes for a 4-layer vertical core design (Figure 1.d). Each core has a 128-entry IW that is equally distributed across the different layers in the vertical design (as we are working with 4 layers, each layer has 32 entries). Entries are disabled by layer, so we disable entries in groups of 32. In order to decide the current IW size we use a dynamic MLP-based IW resizing mechanism as proposed in [18]. In Figure 5-left, we also show the distribution of the average IW size for different benchmark suites (represented with lines). This average window size highly varies between

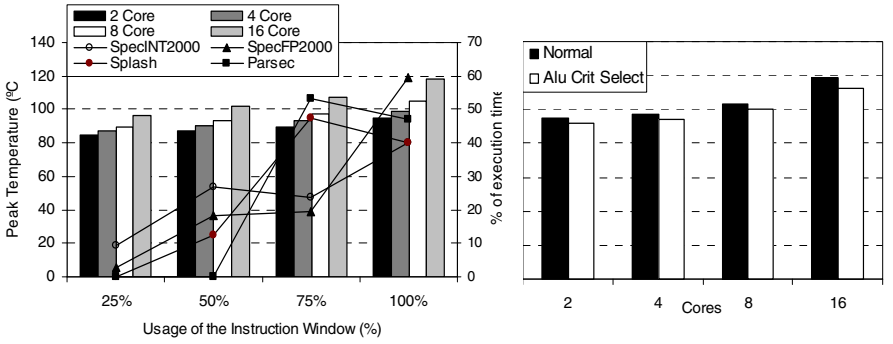

Fig. 5. Peak temperature of the instruction window (left) and ALUs (right) for a varying number of cores

benchmarks, as memory-bound benchmarks require many IW entries to bring more data simultaneously from memory, while CPU-bound applications do not need that many entries. Therefore, instead of just showing the peak temperature reduction of the average benchmarks (bars in Figure 5-left) we decided to do a design exploration of the peak temperature based on the IW size. For example, Parsec benchmarks use 0% of the time 25% and 50% of the IW, 55% of the time use a 75% of the IW (12ºC reduction) and 45% of the time use the whole IW.

When working with vertical designs we can think of having different types of ALUs placed into different layers: fast (and hot) ALUs placed on upper layers for critical instructions plus slower power-saving ALUs placed in lower layers. As our core design includes an instruction criticality predictor we can use this information to decide where we want to send a specific instruction. Figure 5-right shows the effect on peak core temperature having half of the ALUs placed in layers 2-3 (upper layers) and half of the ALUs placed in layers 0-1 (lower layers). The ALUs in the lower layers consume 25% of the original power consumption but are also 25% slower than the original ALUs. Results show a peak temperature reduction of 3-5ºC. This small temperature reduction is due to the fact that in our core design ALUs are not a hotspot (as it can be seen in Figure 4-left: IntExec and FPAlu structures) for the studied benchmarks, and thus, their temperature contribution has almost no impact on the average peak temperature of the processor. However, we can expect better results with other CPU-bound applications where ALUs become a hotspot.

Finally, we want to introduce a custom floorplan design that merges both vertical and horizontal designs. This design is an extension of the L2 design (Figure 1.c) for a 4-layer core. Based on the information provided by Figure 4-left we can separate cool from hot structures and place them in different layers. Hot structures are placed in the top layer (Bpred, Icache, Alumap, TLB, LdStQ, IntReg and ROB), which is the closest to the heatsink. The second layer consists of the rest of structures except the L2, and the last two layers hold the L2 cache and memory controllers. This custom design has the additional advantage of reducing inter-layer communication when bringing data from memory, as memory controllers and the L2 are placed close to the socket. As we can see in Figure 3 (last bar on each group), this design is able to reduce peak temperature by almost 12ºC for a 4-layer 16-core processor.

5 Conclusions

3D die-stacked integration offers a great promise to increase scalability of CMPs by reducing both bandwidth and communication latency problems. However, the increase on core density leads to an increase in temperature and hotspots in these designs. Moreover, as building process scales down below 32nm, leakage becomes an important source of power consumption and, as it increases exponentially with temperature, causes a power/temperature loop that negatively affects to 3D die-stacked processors. To control temperature, regular DTM mechanisms detect overheating in any of the temperature sensors and trigger a power control mechanism to limit power consumption and cool the processor down. However, neither DVFS nor task migration (the most frequently used mechanisms) offer accurate ways to match this target power budget.

Power tokens and *Power Token Balancing* (PTB) were introduced by Cebrian *et al.* as an accurate way to account for power and match a power constraint with minimal performance degradation by balancing power among the different cores of a 2D CMP. In this paper we evaluate these mechanisms in a new design scenario, 3D die-stacked processors. In this scenario PTB is able to reduce average peak temperature by 2-20ºC depending on the selected floorplan. For specific hotspot structures (i.e., instruction cache) PTB can reduce peak temperature by almost 40% in a 4-layer 16-core CMP. In addition, we have proposed *Token3D*, a novel policy that takes into account temperature and layout information when balancing power, giving priority to cool cores over hot ones. This new policy enhances PTB by providing an additional 3% temperature reduction over the original PTB approach. Also note that task migration is orthogonal to PTB and can be applied simultaneously for further temperature reductions.

To conclude this work we have also extended 3D die-stacked vertical designs with additional power control mechanisms. First, we enabled instruction window resizing based on MLP. CPU-intensive applications are highly dependent on cache, but do not show performance degradation if the instruction window is reduced. On the other hand, memory-intensive applications require big instruction windows to locate loads and stores and take advantage of MLP. Based on these properties we extended previous vertical designs with adaptive instruction window resizing. Second, we split ALUs in different groups, low latency and high latency ALUs. Low latency ALUs consume more power and should be placed in upper layers of the 3D design, on the other hand, high latency ALUs are more energy-friendly and can be placed in lower layers of the 3D stack, lowering the chances of becoming a potential hotspot. An instruction criticality predictor was used to decide where an instruction should be placed, either in a fast but expensive or in a slow but efficient unit.

Finally, we explored a custom 3D design that merges both vertical and horizontal designs trying to minimize hotspots. In this design hot processor structures are placed in upper layers while cool structures are placed in lower layers. The design is able to reduce peak temperature by an additional 10% / 85% over the best horizontal / vertical designs.

Acknowledgements. This work was supported by the Spanish MEC, MICINN and EU Commission FEDER funds under Grants CSD2006-00046 and TIN2009-14475-C04. Also by the EU-FP7 ICT Project "Embedded Reconfigurable Architecture (ERA)", contract No. 249059. Finally, the EU-FP7 HiPEAC funded an internship of J.M. Cebrián at U. Uppsala.

References

[1] Ajami, A.H., Banerjee, K., Pedram, M.: Modeling and analysis of nonuniform substrate temperature effects on global ULSI interconnects. IEEE Trans. on CAD 24(6), 849–861 (2005)

[2] Black, B., Annavaram, M., Brekelbaum, N., DeVale, J., Jiang, L., Loh, G.H., McCaule, D., Morrow, P., Nelson, D.W., Pantuso, D., Reed, P., Rupley, J., Shankar, S., Shen, J., Webb, C.: Die Stacking (3D) Microarchitecture. In: Proc. of the 39th Annual IEEE/ACM International Symposium on Microarchitecture, pp. 469–479 (December 2006)

[3] Brooks, D., Martonosi, M.: Dynamic thermal management for high-performance microprocessors. In: Proc. of the 7th Int. Symposium on High Performance Computer Architecture, HPCA (2001)

[4] Cebrián, J.M., Aragón, J.L., García, J.M., Petoumenos, P., Kaxiras, S.: Efficient Microarchitecture Policies for Accurately Adapting to Power Constraints. In: Proc. of the 23rd Int. Parallel and Distributed Processing Symposium, IPDPS (2009)

[5] Cebrián, J.M., Aragón, J.L., Kaxiras, S.: Power Token Balancing: Adapting CMPs to Power Constraints for Parallel Multithreaded Workloads. To appear in Proc. of the 25rd Int. Parallel and Distributed Processing Symposium (May 2011)

[6] Coskun, A.K., Rosing, T.S., Whisnant, K.A., Gross, K.C.: Static and dynamic temperature-aware scheduling for multiprocessor SOCS. IEEE Trans. on VLSI 16(9), 1127–1140 (2008)

[7] Coskun, A., Ayala, J., Atienza, D., Rosing, T., Leblebici, Y.: Dynamic Thermal Management in 3D Multicore Architectures. In: Proc. of the Int. Conf. on Design, Automation and Test in Europe (2009)

[8] Flynn, M.J., Hung, P.: Microprocessor Design Issues: Thoughts on the Road Ahead. IEEE Micro 25(3) (2005)

[9] Gomaa, M., Powell, M.D., Vijaykumar, T.N.: Heat-and-Run: leveraging SMT and CMP to manage power density through the operating system. In: Proc. of the 10th Int. Conf. on Architectural Support for Programming Languages and Operating Systems ASPLOS (2004)

[10] Healy, M., et al.: Multiobjective microarchitectural floorplanning for 2-d and 3-d ICs. IEEE Transactions on CAD 26(1) (2007)

[11] Isci, C., Buyuktosunoglu, A., Cher, C., Bose, P., Martonosi, M.: An Analysis of Efficient Multi-Core Global Power Management Policies: Maximizing Performance for a Given Power Budget. In: Proc. of the 39th Int. Symposium on Microarchitecture, MICRO (2006)

[12] Kgil, T., D'Souza, S., Saidi, A., Binkert, N., Dreslinski, R., Mudge, T., Reinhardt, S., Flautner, K.: PicoServer: using 3D stacking technology to enable a compact energy efficient chip multiprocessor. In: Proc. of the 12th Int. Conf. on Arch. Support for Programming Languages and Operating Systems (2006)

[13] Li, S., Ahn, J.H., Strong, R.D., Brockman, J.B., Tullsen, D.M., Jouppi, N.P.: McPAT: an integrated power, area, and timing modeling framework for multicore and manycore architectures. In: Proc. of the 42nd Int. Symposium on Microarchitecture, MICRO (2009)
[14] Loi, G.L., Agrawal, B., Srivastava, N., Lin, S.C., Sherwood, T., Banerjee, K.: A thermally-aware performance analysis of vertically integrated (3-D) processor-memory hierarchy. In: Proc. of the 43rd Int. Conference on Design Automation (July 2006)
[15] Macken, P., Degrauwe, M., Paemel, V., Oguey, H.: A voltage reduction technique for digital systems. In: Proc. of the IEEE Int. Solid-State Circuits Conf. (February 1990)
[16] Madan, N., Balasubramonian, R.: Leveraging 3D Technology for Improved Reliability. In: Proc. of the 40th Annual IEEE/ACM Int. Symposium on Microarchitecture (December 2007)
[17] Martin, M.M.K., Sorin, D.J., Beckmann, B.M., Marty, M.R., Xu, M., Alameldeen, A.R., Moore, K.E., Hill, M.D., Wood, D.A.: Multifacet's general execution-driven multiprocessor simulator (gems) toolset. SIGARCH Comput. Archit. News 33(4), 92–99 (2005)
[18] Petoumenos, P., Psychou, G., Kaxiras, S., Cebrian, J.M., Aragon, J.L.: MLP-aware Instruction Queue Resizing: The Key to Power-Efficient Performance. In: Proc. of the 23rd Int. Conf. on Architecture of Computing Systems (ARCS) (February 2010)
[19] Puttaswamy, K., Loh, G.H.: Thermal Herding: Microarchitecture Techniques for Controlling Hotspots in High-Performance 3D-Integrated Processors. In: Proc. of the 13th Int. Symposium on High Performance Computer Architecture (HPCA), pp. 193–204 (2007)
[20] Sartori, J., Kumar, R.: Distributed Peak Power Management for Many-core Architectures. In: Proc. of the Int. Conference on Design, Automation and Test in Europe, DATE (2009)
[21] Skadron, K., Stan, M., Huang, W., Velusamy, S., Sankaranarayanan, K., Tarjan, D.: Temperature-aware microarchitecture. In: Proc. of the Int. Symposium on Computer Architecture, ISCA (2003)
[22] Souri, S.J., Banerjee, K., Mehrotra, A., Saraswat, K.C.: Multiple Si layer ICs: motivation, performance analysis, and design implications. In: Proc. of the Int. Conf. on Design Automation (2000)
[23] Xie, Y., Loh, G.H., Black, B., Bernstein, K.: Design space exploration for 3D architectures. J. Emerg. Technol. Comput. Syst. 2(2), 65–103 (2006)
[24] Zhu, C., Gu, Z., Shang, L., Dick, R.P., Joseph, R.: Three-dimensional chip-multiprocessor run-time thermal management. IEEE Transactions on CAD 27(8), 1479–1492 (2008)