EURASIP Journal on
Embedded Systems

RESEARCH Open Access

A novel power model for future heterogeneous 3D chip-multiprocessors in the dark silicon age

Arghavan Asad*, Aniseh Dorostkar and Farah Mohammadi

Abstract

Dark silicon has recently emerged as a new problem in VLSI technology. Maximizing performance of chip-multiprocessors (CMPs) under power and thermal constraints is very challenging in the dark silicon era. Providing next-generation analytical models for future CMPs which consider the impact of power consumption of core and uncore components such as cache hierarchy and on-chip interconnect that consume significant portion of the on-chip power consumption is largely unexplored. In this article, we propose a detailed power model which is useful for future CMP power modeling. In the proposed architecture for future CMPs, we exploit emerging technologies such as non-volatile memories (NVMs) and 3D techniques to combat dark silicon. Results extracted from the simulations are compared with those obtained from the analytical model. Comparisons show that the proposed model accurately estimates the power consumption of CMPs running both multi-threaded and multi-programed workloads.

Keywords: Chip-multiprocessor (CMP), Non-volatile memory (NVM), 3D integration, Dark silicon, Uncore components, Heterogeneous caches

1 Introduction

In today's chip-multiprocessor (CMP) architectures, power consumption is the primary constraint during system design. In the nanometer era, leakage power depletes the power budget and has substantial contribution in overall power consumption. A study by Kao and colleagues has shown that over 50% of the overall power dissipation in a 65-nm generation is due to the leakage power [1] and this percentage is expected to increase in the next generations [2, 3].

Due to the breakdown of Dennard scaling, the fraction of transistors that can be simultaneously powered on within the peak power and temperature budgets is dropping exponentially with each process generation. This phenomenon has been termed as the dark silicon era which is one of the newest challenges in multicore design [4]. Research shows that the increasing leakage power consumption is a major driver of unusable portion or dark silicon in future many-core CMPs [4]. Uncore components such as memory and on-chip interconnect play a significant role in consuming a large portion of power. Also, uncore components, especially those in the cache hierarchy, are the dominant leakage consumers in multi/many-core CMPs. Therefore, power management of these components can be critical to maximize design performance in the dark silicon era. Predictions in recent studies indicate that more than 50% of chips will be effectively dark, idle, dim, or under-clocked dark silicon [5], and this percentage will increase by scaling down in transistor dimension. Therefore, it is extremely important to provide next-generation architectural techniques, design tools, and analytical models for future many-core CMPs in the presence of dark silicon [6, 7]. Prior studies on dark silicon only focus on core designs to address the problem. In this work, we show that uncore components such as cache hierarchy and on-chip interconnect are significant contributors in the overall chip power budget in the nanoscale era and play important roles in the dark silicon age. Since the increase in the CMOS device's power density leads to the dark silicon phenomenon, the emerging power-saving materials manufactured with nanotechnology might be useful for illuminating the dark area of future CMPs.

* Correspondence: arghavan.asad@ryerson.ca
Electrical and Computer Engineering Department, Ryerson University, 350 Victoria Street, Toronto, Ontario M5B 2K3, Canada

The long switch delay and high switch energy of such emerging low-power materials are the main drawbacks which prevent manufactures from completely replacing the traditional CMOS in future processor manufacturing [8]. Therefore, architecting heterogeneous CMPs and integrating cores and cache hierarchy made up of different materials on the same die emerges as an attractive design option to alleviate the power constraint. In this work, we use emerging technologies, such as three-dimensional integrated circuits (3D ICs) [9, 10] and non-volatile memories (NVMs) [11–13] to exploit the device heterogeneity and design of dark silicon-aware multi/many-core systems. 3D die-stacking helps core and uncore components manufactured in different technologies to be integrated into a single package to reduce global wire lengths and improve performance. Among several benefits offered by 3D integrations compared to 2D technologies, mixed-technology stacking is especially attractive for stacking NVM on top of CMOS logics, and designers can take full advantage of the attractive benefits that NVM provides.

In this paper, we propose an accurate power model that formulates the power consumption of 3D CMPs with stacked cache layers. This model can be used for both of the homogenous and heterogeneous cache layers. Unlike the previous research on dark silicon which considers only the portion of power consumption related to on-chip cores [4, 14–16], the proposed model considers power impact of uncore components, such as cache hierarchy and on-chip interconnect, as important contributors in the total CMP power consumption.

In future many-core CMPs, at 22 nm and beyond, emerging leakage-aware technologies such as FinFETs, FDSOI structures, and non-volatile memories are materials for architecting heterogeneous components. The proposed power model in this work can be applied for different technologies with changing power and latency parameters of the new technology.

McPAT [17] (an integrated power, area, and timing modeling framework for multithreaded, multicore, and many-core architectures) cannot estimate the power consumption of 3D CMPs. The maximum number of cores which McPAT supports for power modeling in a many-core processor is 128 when attached to GEM5 [18] and the reason is limitations of existing 2D integration. NVmain [19] (a user-friendly memory simulator to model (non-) volatile memory systems) is a tool just for estimating the power consumption of memory components. It does not consider the power consumption of core and uncore components simultaneously. To the best of our knowledge, the proposed model is the first work in power modeling of network-on-chip (NoC)-based CMPs with stacked cache hierarchy as future CMPs.

In this paper, we make the following novel contributions:

1. We propose an accurate power model for future CMPs with stacked cache layers that support the impact of power consumption of core and uncore components in parallel.
2. The proposed power model for 3D CMPs supports power analysis for both multi-programed and multi-threaded workloads.
3. In the proposed power model, we target CMPs with a large number of cores (e.g., more than eight (many-core CMPs)) built based on scalable networks-on-chip (NoCs) and nonuniform cache architectures (NUCA) for the first time.
4. Our experimental results show that the value of the proposed model is truly close to the value derived by the simulation for each benchmark.

The rest of this paper is organized as follows. A brief background on traditional and NVM technology is explained in Section 2. Section 3 describes the related work. Section 4 analyzes the power consumption of core and uncore components in multicore processors. Section 5 explains the target heterogeneous 3D CMP architecture used in this work. Section 6 presents the power model for the target 3D CMP with the stacked cache hierarchy. In Section 7, evaluation results are presented. Finally, Section 8 concludes the paper.

2 Background

Since the proposed power model can be used for both of the homogenous and heterogeneous stacked cache layers, we first compare characteristics of different traditional and non-volatile memory technologies with each other. Then, we review the STT-RAM technology as a well-known type of NVM technologies.

The traditional and high-performance SRAM technology has been widely used in the on-chip caches due to its standard logic compatibility, high endurance, and fast access time features [20]. However, low-density SRAM technology dissipates high leakage power by its six-transistor implementation [21] and has become a bottleneck for energy-efficient designs. By increasing demand of larger memories in computing systems, using conventional SRAM-based caches becomes more expensive. DRAM technology has become a viable alternative for implementing on-chip caches due to its high density, high capacity, low leakage, and high write endurance features. It is possible to have large reliable last-level caches with high bandwidth by stacking low-leak and high-density DRAM as an on-die cache. However, conventional eDRAM technology tends to be slow compared with SRAM technology and consumes a significant amount of energy in the form of refresh energy to retain stored data which have negative impact on performance.

Compared with traditional memory technologies such as SRAM and eDRAM, NVM technologies such as

STT-RAM and PRAM commonly offer many desirable features like near-zero leakage power consumption, non-volatile property, high cell density, and high resilience against soft errors. Based on the mentioned characteristics for NVMs, the most important feature of the NVM technologies suitable to combat recent dark silicon challenge is near-zero leakage power consumption. As shown in Fig. 1, due to the magnetic characteristic of the MTJ blocks in NVM memory cells, there is not any leakage path between the source line and bit line; therefore, the static power consumption is near zero. However, it should be noted that NVMs suffer from shortcomings such as limited number of write operations and long write operation latency and energy. Compared with other technologies, PRAM is too slow and is not suitable for low-level caches but can be used as a large last-level cache. Table 1 provides a brief comparison between SRAM, STT-RAM, eDRAM, and PRAM technologies in 32-nm technology. The parameter values have been estimated by NVSim [22] and CACTI [23] in this table.

In this section, STT-RAM as a well-known type of NVM technologies, shown in Fig. 1, is briefly explained.

As shown in Fig. 1c, a STT-RAM cell consists of a magnetic tunnel junction (MTJ) to store bit information. A MTJ as a fundamental building block in NVM technologies consists of two ferromagnetic layers separated by a dielectric layer. While the direction of one ferromagnetic layer is fixed, the other layer can be controlled by passing a large enough current through the MTJ. If this current exceeds the critical value, the magnetization direction of the two layers will become antiparallel and MTJ will be in high resistance indicating a “1” logic (Fig. 1b); otherwise the magnetization directions of the two layers are parallel and MTJ is in low resistance indicating a “0” logic (Fig. 1a). It should be noted that the resistance of the MTJ relates not only to the current intensity but also to the current direction matters. If the electrons flow from the reference to the free layer, the magnetic momenta become parallel resulting in a low resistance and the bit 1. If the electrons flow in the reverse direction, we obtain antiparallel momenta and bit 0.

3 Related work

A majority of prior low-power techniques focus on power management at the processor level and the only knob that they use to control the power of a multicore processor is the voltage/frequency level of the cores [24, 25]. A number of researches have proposed some proactive techniques such as thread scheduling, thread mapping, shutting-down schemes, and migration policies to reduce the power consumption in multicore systems [26–30]. However, these approaches limit their scope only to cores.

Management of a problem recently known as dark silicon related to limited power budget is a new challenge in future multicore designs [4, 14–16, 31]. To address the challenges of the dark silicon, Esmailzadeh et al. [4] focused on using only general-purpose cores. They ignored the power impact of uncore components; however, they explained that this was a limitation of their work. The research in [14, 15] works on synthesis of heterogeneous CMPs to extract better energy efficiency and performance in the dark silicon era. Turakhia et al. [14] proposed a design time framework for synthesis of heterogeneous dark silicon CMPs. Raghunathan et al. [15] exploited process variation to evaluate the benefits of selecting a more suitable subset of cores for an application in a given fixed dark silicon power budget to maximize performance. Venkatesh et al. [16] introduced the concept of “conservation cores.” They are specialized processors that focus on reducing energy instead of increasing performance, used for computations that cannot take advantage of hardware acceleration. All of these prior works [4, 14–16] on the dark silicon phenomena over the past 6 years focus on core rather than uncore components. Dorostkar et al. in [31] proposed an optimization problem to minimize energy consumption of uncore components in heterogeneous cache hierarchy and 3D NoC under power budget constraint.

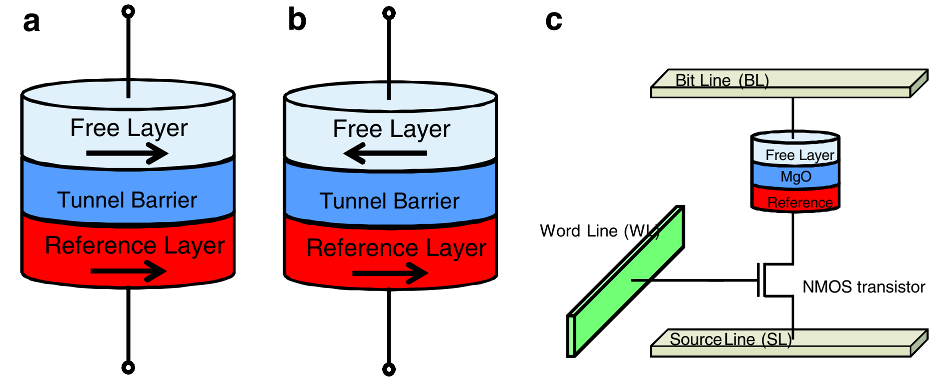

Fig. 1 **a** A parallel MTJ. **b** An anti-parallel MTJ. **c** A STT-RAM cell

Table 1 Different memory technology comparisons at 32 nm

Technology	Area (mm^2)	Read latency (ns)	Write latency (ns)	Leakage power at 80°C (mW)	Read energy (nJ)	Write energy (nJ)
1 MB SRAM	3.03	0.702	0.702	444.6	0.168	0.168
4 MB eDRAM	3.31	1.26	1.26	386.8	0.142	0.142
4 MB STT-RAM	3.39	0.880	10.67	190.5	0.278	0.765
16 MB PRAM	3.47	1.760	43.7	210.3	0.446	0.705

In these days, providing analytical models for future multi/many-core CMPs in the presence of dark silicon is essential [6]. None of the previous studies have presented analytical models for the future CMPs. To the best of our knowledge, this is the first work which proposes an accurate power model that formulates the power consumption of 3D CMPs with stacked cache layers. Unlike the previous researches on power management techniques and dark silicon which consider only the portion of power consumption related to on-chip cores [4, 14–16], the proposed model considers the power impact of uncore components as important contributors in the total CMP power consumption in parallel with cores. This accurate power model can help researchers to propose new power management techniques in future CMPs.

In addition, we note that all the power budgeting techniques and performance optimization under a given power budget proposed so far in the multicore systems [25–27, 32–34] only focus on multi-programed workloads where each thread is a separate application. These models are inappropriate for multi-threaded applications. With increasing parallelization of applications from emerging domains such as recognition, mining, synthesis, and, particularly, mobile applications, this issue has become important that in future many-core architectures, workloads are expected to be multi-threaded applications. To the best of our knowledge, this is the first study that presents an accurate power model for both multi-programed and multi-threaded workloads.

Therefore, an analytical power model is extremely essential in order to verify that power budgets are met by different parts of CMP including cores and uncores with different technology and different performance or low-power techniques and also model power consumption in heterogeneous and homogeneous CMP under running both multi-threaded and multi-programed applications in future CMP. To the best of our knowledge, this is the first work which proposes an accurate power model that formulates the power consumption of 3D CMPs with stacked cache layers for both multi-programed and multi-threaded workloads.

4 The contribution of core and uncore components in total future multicore processor power consumption

In this section, we analyze the power consumption of core and uncore components in multicore systems. To better understand the power distribution of a multicore processor, we use McPAT [17] and evaluate the power dissipation of core and uncore components including L2/L3 cache levels, the routers and links of NoC, integrated memory controllers, and integrated I/O controllers.

In recent years, more and more applications are shifting from compute bounding to data bounding; therefore, a hierarchy of cache levels and data storage components to efficiently store and manipulate large amounts of data is required. In this context, an increasing percentage of on-chip transistors is invested on the uncore components and architects have dramatically increased the size of cache levels in cache hierarchy, in an attempt to bridge the gap between fast cores and slow off-chip memory accesses in multi/many-core CMPs. We select *canneal* as a representative of future memory-intensive applications in Fig. 2.

Figure 2 illustrates the power breakdown of a multicore system with increasing number of cores under limited power budget. Cores in this multicore platform are based on Niagra2 processors [35] with an additional shared L3 as the last-level cache (LLC).

The size of LLC increases with increasing number of cores as shown in Fig. 2. We assume multicore systems in this experiment run *canneal* application from PARSEC [36]. We use technology 32 nm in this study. As shown in this figure, the power consumption of uncore components becomes more critical when the number of cores is increased in a multicore system and the power budget is a design limitation. In this work, we assume idle cores can be gated off (dark silicon) while other on-chip resources stay active or idle under limited power budget. Actually, the uncore components remain active and consume power as long as there is an active core on the chip. As illustrated in Fig. 2, more than half of the power consumption is due to the uncore components in the 16-core and 32-core systems.

In addition, Fig. 3 illustrates when technology scales from 32 to 22 nm, the ratio of leakage power increases and is expected to exceed the dynamic power in the future generations. We use 1 GHz frequency and 0.9 V supply voltage for an 8-core system in 32- and 22-nm technologies in Fig. 3. This figure shows that leakage power dominates the power budget in the nanoscale technologies and is a major driver for unusable portion or dark silicon in future many-core CMPs.

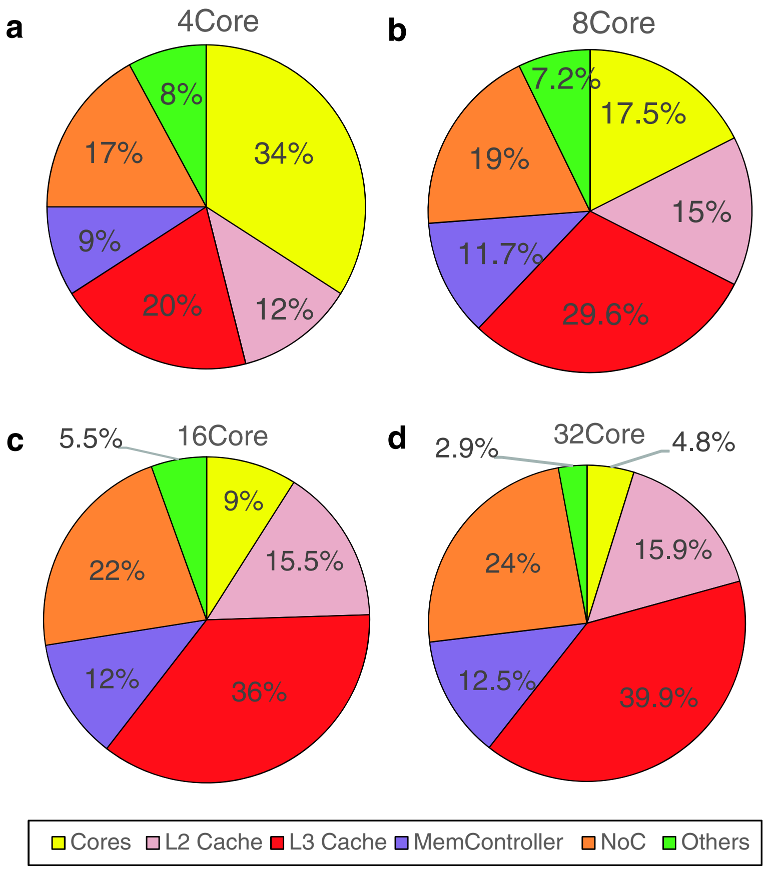

Fig. 2 Power breakdown of **a** a 4-core with 4 MB LLC, **b** an 8-core with 8 MB LLC, **c** a 16-core with 16 MB LLC, and **d** a 32-core with 32 MB LLC under limited power budget

In this context, for architecting new classes of low-power architectures, using emerging technologies such as NVMs with near-zero leakage power and three-dimensional integrated circuits (3D ICs) for stacking different technologies onto CMOS circuits brings new opportunities to the design of multi/many-core systems in the dark silicon era.

5 Target CMP architecture

With increasing parallelism levels of new applications (from emerging domains such as recognition, mining, synthesis, and especially mobile applications), which can efficiently use 100 to 1000 cores, shifting to multi/many-core designs has been aimed in recent years. Due to

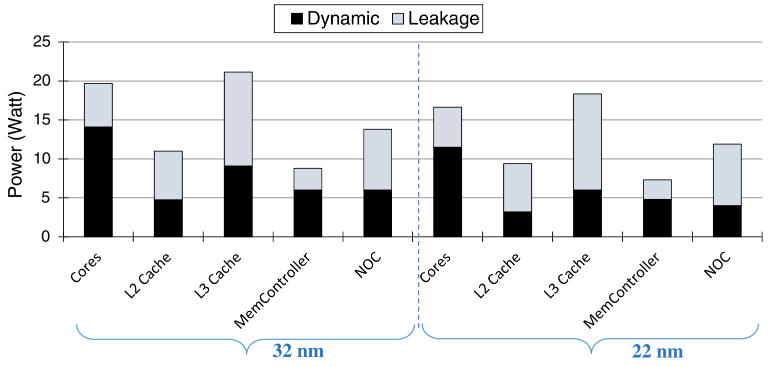

Fig. 3 Dynamic power vs. leakage power for an 8-core system in 32- and 22-nm technologies

the scalability limitations and performance degradation problems in 2D CMPs, especially, in future many-cores, in this work, we focus on 3D integration to reduce global wire lengths and improve performance of future CMPs. For instance, Apple's iPhone 4S is supposed to use the A5 processor, an SoC with two LPDDR2 SDRAM chips on top of the core layer, in the proposed system [37].

The architecture model assumed in this work is based on a 3D CMP with multi-level hybrid cache hierarchy stacked on the core layer similar to Fig. 4a. As shown in Fig. 4a, each cache level is assumed to be implemented using a different memory technology. For motivating about the proposed architecture for future CMPs in this paper, we design a scenario. In this scenario, we consider a 3D CMP with homogenous cache hierarchy as shown in Fig. 4b. In this scenario, we assume there is one layer per level in the homogenous cache hierarchy stacked on the core layer. Also, we assume there are four cores in the core layer, each of them running *art* application from SPEC 2000/2006 [38]. Figure 4b illustrates an example of the proposed architecture shown in Fig. 4a with four homogenous cache levels in the hierarchy and the core layer with four cores with more details about on-chip interconnection. Table 2 gives the properties of average memory access time (AMAT), as a suitable performance parameter for evaluation of the cache systems performance and system power consumption when the stacked cache levels in the homogenous hierarchy are made from SRAM, eDRAM, STT-RAM, or PRAM. Note that normalization reported in Table 2 is done based on the best case, that is, power consumption is normalized with respect to the SRAM, whereas AMAT is normalized with respect to the PRAM. Based on these views, SRAM is the fastest and a higher power-hungry option and it is better to be used in lower levels of the cache hierarchy to support faster accesses. According to the observations in Table 2, we decided to use SRAM in the L2 cache level, eDRAM in the L3 cache level, STT-RAM in the L4 cache level, and PRAM in the L5 cache level as shown in Fig. 4a. Details of all the experimental setup and power and performance estimation used in this motivation example will be shown in Section 7.

Because of strong thermal correlations between a core and cache banks directly stacked on the core, the core and the cache banks in the same stack is called a core set in our architecture (as shown in Fig. 5a).

6 Proposed power model for NoC-based CMPs with stacked cache hierarchy

In this section, we present an analytical power model for the 3D chip-multiprocessors (CMPs) with stacked cache hierarchy as future CMPs. Table 3 lists the parameters used in this model.

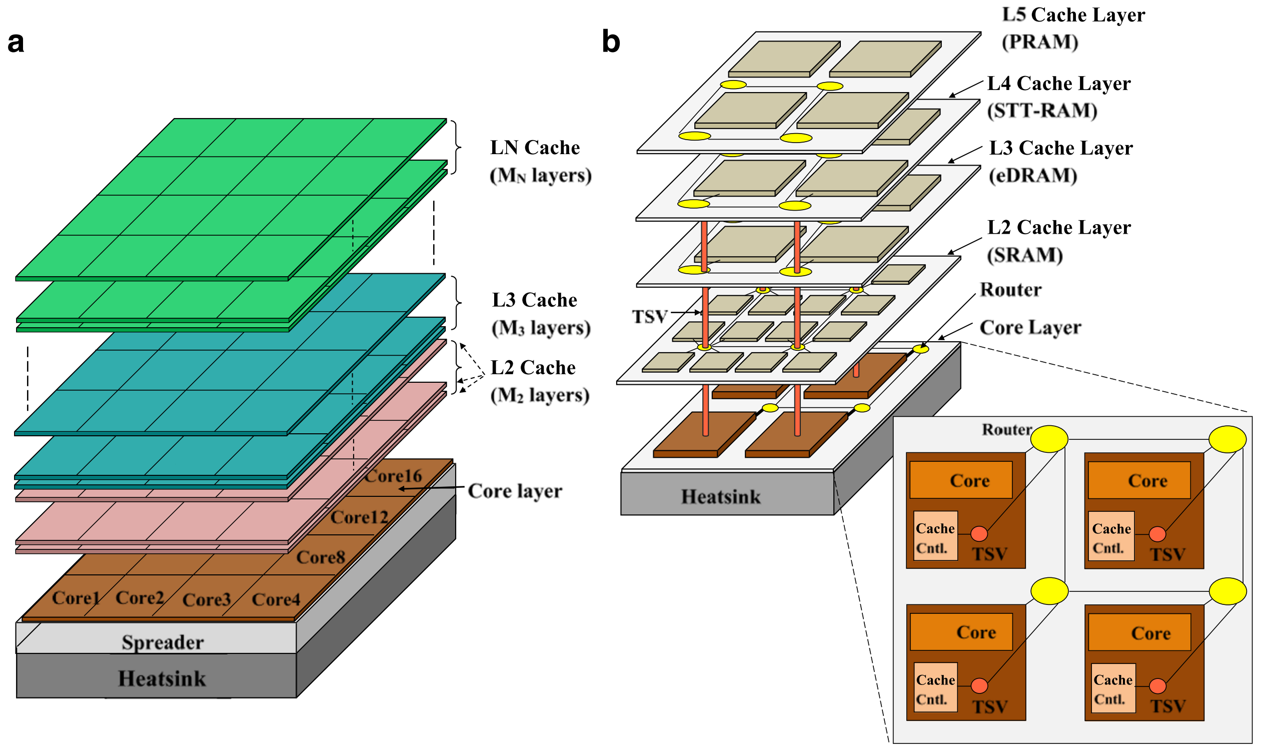

Fig. 4 a The target CMP architecture. **b** A 3D CMP architecture with homogenous cache hierarchy

Table 2 Comparison of AMAT and system power consumption

Technology	AMAT	Power consumption
SRAM	0.09	1
eDRAM	0.16	0.62
STT-RAM	0.3	0.37
PRAM	1	0.22

The total power consumption of a CMP mainly comes from three on-chip resources: cores, cache hierarchy, and interconnection network. CMPs with a large number of cores (more than eight) require building architectures through a scalable network-on-chip (NoC).

6.1 Components of the total power consumption of a 3D CMP

The total power consumption of a 3D CMP can be calculated as the sum of the power of individual on-chip resources (core and uncore components).

$$P_{\text{Total}} = P_{\text{cores}} + P_{\text{uncores}} \tag{1}$$

$$P_{\text{Total}} = P_{\text{cores}} + P_{\text{cache\_hierarchy}} + P_{\text{interconnection}} \tag{2}$$

6.1.1 Modeling of core power consumption

We denote the power consumption of core i as P_i^{core}.

$$P_{\text{cores}} = \sum_{i=1}^{n} P_i^{\text{core}} \tag{3}$$

The power consumption of core i is comprised of dynamic and leakage power components. The total power consumption of core i is written as:

$$P_i^{\text{core}} = P_{D,i} + P_{L,i}, \qquad \forall i \tag{4}$$

$$P_{D,i} = P_{\max} \frac{f_i^{\,2}}{f_{\max}^{\;\;2}}, \qquad \forall i \tag{5}$$

Since operating voltage of a core depends on the operating frequency, it is assumed that the square of the voltage scales linearly with the frequency of operation [39]. In Eq. 5, $P_{\max}$ is maximum power budget and $f_{\max}$ is maximum frequency of the core.

The leakage power dissipation depends on temperature. The leakage power of core i can be written as Eq. 6. T_t is ambient temperature at time t and h_i is empirical coefficient for temperature-dependent leakage power dissipation. h_i coefficients in cores with the same microarchitectures have the same value. h_i is based on the thermal behavior of a core and is calculated as presented in [40, 41].

$$P_{L,i} = h_i \times T_t, \qquad \forall i, t \tag{6}$$

In this work for core power modeling, we can consider the peak leakage power as other works [14, 15]. Therefore, in this model, we can use the maximum sustainable temperature for the chip.

$$P_{L,i} = h_i \times T_{\max}, \qquad \forall i \tag{7}$$

6.1.2 Modeling of cache hierarchy power consumption

6.1.2.1 Cache hierarchy power consumption modeling for multi-programed workloads As shown in Fig. 4a, the number of cache levels is N and each cache level is indexed as L_k, ($k = 1, 2, 3, \ldots, N$). There are M_k layers in the $k$thcache level, L_k. The lth cache layer ($l = 1, 2, 3, \ldots, M_k$) in the L_k is represented as $A_{k,\ l}$.

We assume that in multi-programed applications, each application mapped on each core effectively sees only its own slice of the dedicated cache banks in the cache hierarchy.

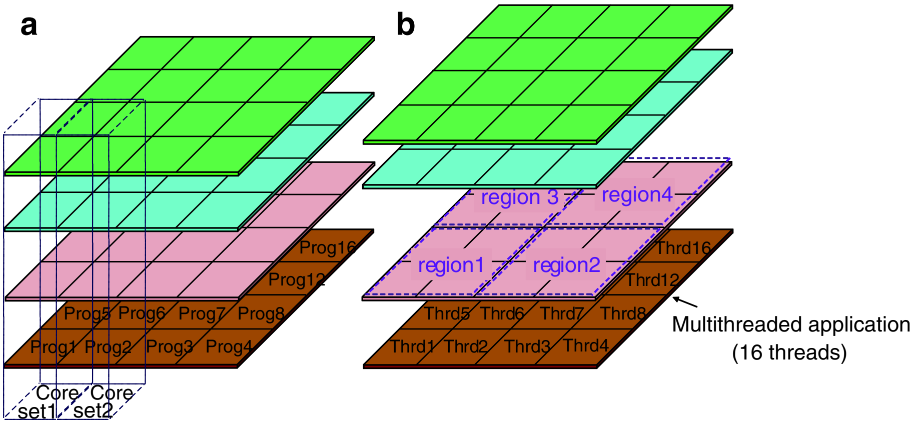

Fig. 5 The style of using cache hierarchy in **a** a multi-programed workload and **b** a multithreaded workload

Table 3 Parameters used in the power model

Parameter	Definition
n	Number of cores in the core layer
f_i	Operation frequency of core i
P_i^{core}	Power consumption of core i
$P_{D,\,i}$	Dynamic power consumption of core i
$P_{L,\,i}$	Leakage power consumption of core i
$P_i^{\text{cache\_hierarchy}}$	Sum of power consumption related to the dedicated cache banks in each level of the cache hierarchy stacked on core i from the 1st to the kth level
$P_{\text{static}_k}(T)$	Static power consumed by each layer of the kth cache level (L_k) at temperature T °C
N	Number of cache levels L_1, L_2, …, L_N
C_k	Capacity of the kth cache level (L_k)
$b_{i,\,k}$	Number of active cache layers in the region-set bank i stacked on core i at the kth cache level
$B_{i,\,k}$	Accumulated cache capacity in the region-set bank i stacked on core i at the kth cache level
a^r, a^w	Number of read and write accesses of an application
HiT_k	Hit time per hit access
$APPH_k$	Average power consumption per hit access
γ	Number of accesses per second
α	Sensitivity coefficient from the cache misses power law
E_n	Data sharing factor [53]
T_s	Total execution time of the mapped applications
$E^s_{\text{interconnection}}$	Energy consumption of the interconnection between nodes in T_s
$P_{\text{interconnection}}$	Power consumption of the interconnection network between nodes
$P^q_{n,n',n''}$	Static power consumption of an interconnection network based on mesh topology with n nodes in dimension 1, n' nodes in dimension 2, and n'' nodes in dimension 3
$P^{\text{static}}_{\text{Links}}$	Static power consumption of links
P^{static}_{TSVs}	Static power consumption of TSVs
E^s_{NP}	Average total energy dissipated in the on-chip interconnection network for transferring of NP packets in T_s
P^{qC}_R	Static power consumption of a router (without any packet)
P^c_R	Static power consumption of a router with one virtual channel (without any packet)
P^c_{Link}	Static power consumption of a link (without any packet)
P^c_{TSV}	Static power consumption of a TSV (without any packet)
TSV	Total number of TSVs
E^s_{NP}	Average total energy dissipated in the on-chip interconnection network for transferring of NP packets in T_s
E^s_1	Average total energy dissipated for transferring of one packet from the source to the destination in the on-chip interconnection network
E^P_R	Average constant energy dissipated in a router and the related link for a packet transfer
E^f_R	Average constant energy dissipated in a router and the related link for a flit transfer
D_{mesh}	The average distance of the mesh topology (the average number of links which a packet transits from the source to reach the destination)
v	Number of virtual channels per link
l	Size of a packet based on number of flits

$$P_{\text{cache\_hierarchy}} = \sum\nolimits_{i=1}^{n} P_i^{\text{cache\_hierarchy}} \tag{8}$$

$$P_i^{\text{cache\_hierarchy}} = P_{\text{dynamic}_i}^{\text{cache\_hierarchy}} + P_{\text{static}_i}^{\text{cache\_hierarchy}} \tag{9}$$

The first part of Eq. 9, $P_{\text{dynamic}_i}^{\text{cache\_hierarchy}}$, depends on dynamic energy. Dynamic energy consumed by cache depends on average memory access time (AMAT). Reducing AMAT leads to lower cache dynamic energy. Therefore, for formulating the first part of Eq. 9 based on accessible variables in the model, first we model the AMAT. The AMAT for a cache hierarchy with N levels is shown in Eq. 10. As shown in this equation, the AMAT is a function of miss rate and access time at each cache level.

$$\text{AMAT} = HiT_1 + \sum_{k=1}^{N-1} HiT_{k+1} \times R_k^{\text{miss}} \tag{10}$$

where HiT_k denotes hit time at the kth cache level and R_k^{miss} is the product of cache miss rates from the 1st to the kth cache level. The average HiT_k at the kth cache level is computed as Eq. 11 due to the different access time of reading and writing in non-volatile memories (i.e., STTRAM-based or PRAM-based cache):

$$HiT_k = \frac{a_k^r \times \tau_k^r + a_k^w \times \tau_k^w}{a_k^r + a_k^w} \tag{11}$$

where a_k^r and a_k^w are the number of read and write accesses of the running program at the kth cache level, respectively. τ_k^r and τ_k^w are latencies of read and write at the kth cache level.

In this trend, we can compute the average power per access (APPA) by:

$$\text{APPA} = APPH_1 + \sum_{k=1}^{N-1} APPH_{k+1} \times R_k^{\text{miss}} \tag{12}$$

$$APPH_k = \frac{a^r \times \tau_k^r \times p_k^r + a^w \times \tau_k^w \times p_k^w}{a^r + a^w} \tag{13}$$

where p_k^r and p_k^w are power consumption of read and write at the kth cache level, respectively. We can rewrite Eq. 13 as:

$$APPH_k = \frac{a_k^r \times E_{\text{read}k} + a_k^r \times E_{\text{write}k}}{a_k^r + a_k^r} \tag{14}$$

where $E_{\text{read}k}$ and $E_{\text{write}k}$ are read and write energy at the kth cache level, respectively.

$$R_k^{\text{miss}} = \mu \times \left(\frac{B_k}{\sigma}\right)^{-\alpha} \tag{15}$$

where σ is baseline cache size. μ is baseline cache miss rate. α is power law exponent and typically lies between 0.3 and 0.7 [42]. B_k is the sum of allocated cache capacity from the 1st to the kth cache level and is obtained by:

$$B_k = \sum_{m=1}^{k} c_m \times b_m \tag{16}$$

where c_m and b_m are the capacity of each cache layer and the number of active cache layers at the mth cache level, respectively.

We can rewrite the first part of Eq. 9, $P_{\text{dynamic}_i}^{\text{cache\_hierarchy}}$, based on the accessible variables as:

$$P_{\text{dynamic}_i}^{\text{cache\_hierarchy}} = \gamma \times \left(APPH_1 + \sum_{k=1}^{N-1} APPH_{k+1} \times \mu \times \left(\frac{B_{i,k}}{\sigma}\right)^{-\alpha} \right) \tag{17}$$

where γ is the number of accesses to cache layer per second. In Eq. 18, d_i is a constraint that shows the time-to-deadline of the program allocated to core i.

$$\gamma = \frac{a^r + a^w}{d_i} \tag{18}$$

As one of the worst cases, we can assume all of the accesses of the mapped application are to the Nth cache level of the hierarchy with biggest latency. Therefore, we can set d_i as:

$$d_i = a^r \times \tau_N^r + a^w \times \tau_N^w \tag{19}$$

The second part of Eq. 9, $P_{\text{static}_i}^{\text{cache\_hierarchy}}$, is the total leakage power consumption related to the dedicated cache banks to core i which is the main contributor to the total power consumption.

$$P_{\text{static}_i}^{\text{cache\_hierarchy}} = \sum_{k=1}^{N} b_{i,k} \times P_{\text{static}_k}(T_{\max}) \tag{20}$$

$$b_{i,k} = \frac{B_{i,k} - B_{i,k-1}}{c_k} \tag{21}$$

In Eq. 20, $P_{\text{static}_k}(T_{\max})$ is the static power consumed by each layer of the kth cache level, L_k, at temperature $T_{\max}$. Equation 21 shows the number of active cache layers in the region set bank i stacked on core i at the kth cache level, $b_{i,\,k}$, that is proportional to the difference between accumulated cache capacity at the kth cache level, $B_{i,\,k}$, and that at the $(k-1)$th level, $B_{i,\,k-1}$. c_k shows the capacity of the kth cache level, L_k.

6.1.2.2 Cache hierarchy power consumption modeling for multi-threaded workloads Equations 8–21 model cache power consumption in multi-programed workloads which each program only using the dedicated cache banks in its own core set privately as shown in Fig. 5a. Larger classes of multi-threaded applications are based on barrier synchronization and consist of two phases of execution (shown in Fig. 6): a sequential phase, which consists of a single thread of execution, and a parallel phase in which multiple threads process data in parallel. The parallel threads of execution in a parallel phase typically synchronize on a barrier. In parallel phase, all threads must finish execution before the application can proceed to the next phase. In multi-threaded workloads, cache levels are shared across the threads. In parallel phase, threads share regions at each layer of the cache levels in the hierarchy as shown in Fig. 5b. For example, for a performance-maximization problem with respect to power budget, first, we dedicate region 1 in each level to the threads, as an initialized value. Then based on power budget and other constraints in the optimization problem, we can increase the number of regions or keep it fixed in each level in order to obtain the maximum performance.

Since multi-threaded applications use cache hierarchy in shared style, we can rewrite Eq. 9 for them as follows:

$$P_{\text{cache\_hierarchy}} = \mathrm{P}_{\text{dynamic}}^{\text{cache\_hierarchy}} + P_{\text{static}}^{\text{cache\_hierarchy}} \tag{22}$$

Because of the impact of multi-threaded data sharing on the cache miss rate, we replace Eq. 15 with Eq. 23:

$$R_{j,k}^{\text{miss}} = \mu \times \left(\frac{B_{j,k}}{n \times \sigma}\right)^{-\alpha} \times E_n \tag{23}$$

where E_n is data sharing impact on miss rate. n is number of cores in the core layer. μ and σ are the same as these parameters in Eq. 15 [42].

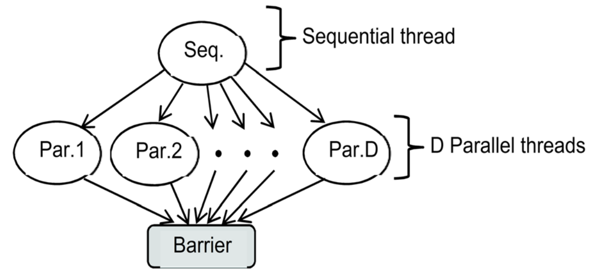

Fig. 6 An example of a multi-threaded application with D parallel threads

$$B_k = \sum_{m=1}^{k} \sum_{j=1}^{regn_m} j \times x_{j,k} \times \frac{c_m}{regn_m} \tag{24}$$

$$\sum_{j=1}^{regn_k} x_{j,k} = 1, \quad \forall k \tag{25}$$

Let $x_{j,\ k}$, $x_{j,\ k} \in \{0, 1\}$, $j \in [1, regn_k]$, $k \in [1, N]$, be a binary variable. $x_{j,\ k}$ is set to 1, when the multi-threaded application uses j regions (region 1, region 2, ..., region $j - 1$ and region j) at the kth cache level. Its optimal value is founded by performance-maximization problem. Note that $regn_k$ represents the total number of regions in the kth cache level of the hierarchy. Equation 25 guarantees that the value of $x_{j,\ k}$ is unique and just j regions is used at the kth cache level.

The first part of Eq. 22, $P_{\text{dynamic}}^{\text{cache\_hierarchy}}$, based on accessible variables, is as follows:

$$P_{\text{dynamic}}^{\text{cache\_hierarchy}} = \gamma \times \left(APPH_1 + \sum_{k=1}^{N-1} \sum_{j=1}^{regn_k} \left(APPH_{k+1} \times \mu \left(\frac{B_k}{n \times \sigma} \right)^{-\alpha} \times E_n \right) \right) \tag{26}$$

where γ is the number of accesses per second in Eq. 26 and is computed by using Eqs. 18 and 19.

The second part of Eq. 22, $P_{\text{static}}^{\text{cache\_hierarchy}}$, can be modeled as follows:

$$P_{\text{static}}^{\text{cache\_hierarchy}} = \sum_{k=1}^{N} \sum_{j=1}^{regn_k} j \times x_{j,k} \times P_{\text{static}_{regk}}(T_{\max}) \tag{27}$$

Equations 12, 13, and 14 in the multi-programed cache power modeling are repeated again for a multi-threaded workload.

6.1.3 Modeling of on-chip interconnection power consumption

Energy consumption of the on-chip interconnection network in the total execution time of mapped workload, T_s, is calculated by Eq. 28 [43], which contains static energy of an interconnection network and dynamic energy of transferring packets to the network.

$$\begin{aligned} E_{\text{interconnection}}^{s} &= E_{\text{static}} + E_{\text{dynamic}} \\ &= P_{n,n',n''}^{q} \times T_s + E_{NP}^{s} \end{aligned} \tag{28}$$

$$\begin{aligned} E_{NP}^{s} &= NP \times E_1^{s} = NP \times (D_{\text{mesh}} + 1) \times E_R^{P} \\ &= NP \times (D_{\text{mesh}} + 1) \times l \times E_R^{f} \end{aligned} \tag{29}$$

Total dynamic energy dissipation contains energy dissipated for transferring NP packets, where each packet dissipates E_1^s on average for transferring from the source to the destination in the on-chip interconnection network. When one packet is forwarded from the source to the destination, on average, it goes across $D_{\text{mesh}} + 1$ routers and links (E_R^P is average constant energy dissipated in a router and the related link for a packet transferring). It should be noted that a packet contains l flits and in this context, E_R^f is the average of energy dissipated in a router and the related link for a flit transferring. Therefore, to transfer NP packets in T_s in the on-chip interconnection network, dynamic energy dissipation (E_{NP}^s) of an on-chip interconnection will be formulated as Eq. 29.

In a mesh topology with d dimensions, where there are k_i nodes in the ith dimension, the average distance that a packet must traverse to reach the destination can be calculated by Eq. 30 [44]:

$$D_{\text{mesh}} = \frac{1}{3} \times \sum\nolimits_{i=1}^{d} \left(k_i - \frac{1}{k_i} \right) \tag{30}$$

In a 2D mesh with n nodes in each dimension ($d = 2$ and $k_{1,\ 2} = n$), the average distance between two nodes can be calculated as follows:

$$D_{\text{mesh}} = \frac{2n}{3} - \frac{2}{3n} \tag{31}$$

In a many-core platform based on 2D mesh topology ($n \geq 32$), the value of the second part in Eq. 31 will be negligible and can be ignored. Therefore, the average distance is:

$$D_{\text{mesh}} \cong \frac{2n}{3} \tag{32}$$

$P_{n,n',n''}^{q}$ is the static power consumption of an interconnection network based on mesh topology with n nodes in dimension 1, $n^{'}$ nodes in dimension 2, and $n^{''}$ nodes in dimension 3 and contains power consumption of

Table 4 Specification of CMP configurations evaluated in this work

Component	Description
Number of cores	Experiment 1, 16, 4 × 4 mesh Experiment 2, 64, 8 × 8 mesh
Core configuration	Alpha21164, 3 GHz, area 3.5 mm^2, 32 nm
L1 cache	SRAM, 4 way, 32B line, size 32 KB private per each core
L2/L3/L4 caches	L2: SRAM, L3: SRAM, L4: SRAM (baseline) L2: SRAM, L3: eDRAM, L4: STT-RAM (hybrid)
Network router	2-stage wormhole switched, XYZ routing, virtual channel flow control, 2 VCs per port, a buffer with depth of 5 flits per each VC, 8 flits per data packet, 1 flit per address packet, each flit is set to be 16-byte long
Network topology	3D network, each layer is a 4 × 4 mesh, each node in layer 1 has a router, 16 TSV links which are 128b bi-directional in each layer
$P_{\max}$, $T_{\max}$	110 W, 80°C

Table 5 Multi-programed workloads used in the experiment

Test program suite	Benchmarks
Memory Bounded set1 (MB1)	zeusmp(3), libquantum(3), lbm(3), GemsFDTD(3), art(2), swim(2)
Memory Bounded set2 (MB2)	zeusmp(2), libquantum(2), lbm(2), GemsFDTD(2), art(4), swim(4)
Medium set1 (MD1)	mcf(3), sphinx3(3), leslie3d(2), gcc(2), cactusADM(2), milc(2), omnetpp(2)
Medium set2 (MD2)	mcf(2), sphinx3(2), bzip2(2), calculix(2), leslie3d(2), gcc(2), cactusADM, milc, omnetpp, wupwise
Computation Bounded set1 (CB1)	parser(2), applu(2), face_rec(2), equake(2), astar(2), hmmer(2), bzip2(2), calculix(2)
Computation Bounded set2 (CB2)	parser(2), applu(2), face_rec(2), equake(2), astar(2), hmmer(2), bzip2, calculix, mpeg_dec(2)
Mixed set1 (Mix1)	sphinx3(2), mcf, astar(2), hmmer, gamess(2), perlbench(2), soplex, gromacs, gcc(2), leslie3d(2)
Mixed set2 (Mix2)	sphinx3, mcf, astar(2), hmmer(2), gamess(2),perlbench(2), gromacs(2), tonto(2), gcc, leslie3d

TSVs, links, and routers without packets. There is $n \times n^{'} \times n^{''}$ routers with v virtual channels, $n \times n^{'}$ links on the core layer, and TSV TSVs in the 3D network on chip.

Finally, power consumption of on-chip interconnection between nodes can be calculated as:

$$\begin{aligned} P_{\text{interconnection}} &= \frac{E^s_{\text{interconnection}}}{T_s} = P^q_{n,n^{'},n^{''}} + \frac{E^s_{NP}}{T_s} \\ &= \left(n \times n^{'} \times n^{''} \times P^{qC}_R + P^{\text{static}}_{\text{Links}} + P^{\text{static}}_{TSVs}\right) + \frac{E^s_{NP}}{T_s} \\ &= n \times n^{'} \times n^{''} \times v \times P^c_R + n \times n^{'} \times P^c_{\text{link}} + TSV \times P^c_{TSV} + \frac{E^s_{NP}}{T_s} \end{aligned} \tag{33}$$

Since Eq. 33 is the function of total execution time of the mapped applications, T_s, and T_s has a big value compare to E_{NP}, the last term of Eq. 31 can be ignored; therefore,

$$P_{\text{interconnection}} = n \times n' \times n'' \times v \times P^c_R + n \times n' \times P^c_{link} + TSV \times P^c_{TSV} \tag{34}$$

As described in [45–47], also based on observation from Fig. 3, particularly problematic for NoC structures is leakage power, which is dissipated regardless of communication activity. At high network utilization, static power may comprise more than 75% of the total NoC power at the 22-nm technology and this percentage is expected to increase in future technology generations. This fact is captured by Eq. 34.

6.2 Dark silicon constraint

Equations 35 and 36 represent the dark silicon constraints for CMPs with multi-programed and multi-threaded workloads when, for example, the goal is maximizing performance of the system. Maximizing performance under power constraint is an important target in digital system design in these days. The peak power dissipation during the entire execution must be less than the maximum power budget.

$$\sum_{i=1}^{n} P_i^{\text{core}} + \sum_{i=1}^{n} P_i^{\text{cache\_hierarchy}} + P_{\text{interconnection}} \le P_{\text{budget}} \tag{35}$$

$$\sum_{i=1}^{n} P_i^{\text{core}} + \sum_{k=1}^{N} \sum_{j=1}^{regn_k} j.x_{j,k}.P_j^{\text{cache\_hierarchy}} + P_{\text{interconnection}} \le P_{\text{budget}} \tag{36}$$

Equations 35 and 36 can be used in design time and run time optimization techniques and other power management methods to combat dark silicon.

The proposed model is linear polynomial since all formulas are linear and degree of variables is one. To solve the models, we use CVX [48], an efficient convex optimization solver. Solving this model can be exhaustively done (in polynomial time) to determine the best solution that maximizes performance within the dark silicon peak power budget. The overall runtime overhead for this polynomial computation is negligible in our experiment.

7 Experimental evaluation

7.1 Platform setup

In order to validate the efficiency of 3D CMP architectures in this work, we employed a detailed simulation framework driven by traces extracted from real application workloads running on a full-system simulator. The traces have been extracted from the GEM5 full-system simulator [17]. For simulating a 3D CMP architecture, the extracted traces from GEM5 were interfaced with 3D Noxim, as a 3D NoC simulator [49]. GEM5 was augmented with McPAT and 3D Noxim with ORION [50] to calculate the power consumption in this paper. The cache capacities and energy consumption of SRAM and NVMs were estimated from CACTI and NVSIM [22], respectively. A full-system simulation of a 16-core CMP

Table 6 Multi-threaded workloads used in the experiment

Multi-threaded workload	blackscholes, bodytrack, canneal, dedup, facesim, swaption, ferret, fluidanimate, vips, freqmine, × 264

Table 7 Workload characteristics (cache hierarchy)

WL	LLC_Util (%)	Miss rate (%)	Hit rate (%)	WL	LLC_Util (%)	Miss rate (%)	Hit rate (%)
blackscholes	0.14	5	95	MB1	97	79	21
bodytrack	0.15	8	92	MB2	92	73	27
canneal	74	73	27	MD1	62.4	61	39
dedup	25.3	61	39	MD2	58.6	57	43
facesim	10.7	57	43	CB1	3.6	4	96
ferret	9.1	55	45	CB2	2	3	97
swaptions	0.16	7	93	Mix1	7.8	9	91
fluidanimate	27.6	59	41	Mix2	14.3	17	83
freqmine	0.72	41	59				
vips	5	33	77				
× 264	5.3	30	70				

architecture with three cache levels in the hierarchy at the 32-nm technology is performed for evaluation in this work. In each cache level of the stacked hierarchy, there are three layers. In the hybrid architecture, the capacity of each layer of L2 cache is 16 × 1 MB SRAM bank, the capacity of each layer of L3 cache is 16 × 4 MB eDRAM bank, and the capacity of each layer of L4 cache is determined 16 × 4 MB STT-RAM bank. In the baseline architecture, the capacity of each layer of L2, L3, and L4 caches is 16 × 1 MB SRAM bank. The detailed properties of cache banks in different technologies are listed in Table 1. The system configuration used for evaluation in this work is listed in Table 4.

We use multi-programed workloads consisting of 16 applications and multi-threaded workloads with 16 threads for performing our experiments. The applications in multi-programed workloads are selected from the SPEC2000/2006 benchmark suites [38]. Based on memory demand intensity of benchmark applications, we classified them into three groups: memory-bounded, medium, and computation-bounded benchmarks. From this classification, we generated a range of workloads (combinations of 16 benchmarks), as summarized in Table 5. Note that the number in parentheses is the number of instances. In our setup, programs in a given workload are randomly mapped to cores to avoid a specific OS policy. Table 6 summarizes the multi-threaded workloads based on PARSEC [51] used in this work.

7.2 Experimental results

In this subsection, we evaluate the target 3D CMP with stacked cache hierarchy in two different cases: the CMP with SRAM-only stacked cache levels on the core layer (baseline) and the proposed CMP with hybrid stacked cache levels on the core layer (hybrid). Results extracted from the simulations are compared with those obtained from the analytical model. We define a new parameter named LLC_Util which shows the utilization of the last-level cache in the hierarchy and is the best parameter to show the workload characteristics among the other parameters. Workloads with LLC_Util less than 5% are computation intensive. Tables 7 and 8 show used workload characteristics based on the related LLC_Util parameter, number of cache hits, number of cache

Table 8 Workload characteristics (NoC)

WL	Packet latency (ns)	No. of packet transfer	WL	Packet latency (ns)	No. of packet transfer
blackscholes	0.01	1,000,000	MB1	0.98	11,000,000
bodytrack	0.016	1,150,000	MB2	0.92	10,000,000
canneal	0.61	9,700,000	MD1	0.67	7,400,000
dedup	0.52	6,000,000	MD2	0.65	7,100,000
facesim	0.49	5,300,000	CB1	0.12	2,000,100
ferret	0.39	4,880,005	CB2	0.11	1,900,000
swaptions	0.013	1,000,010	Mix1	0.31	2,700,000
fluidanimate	0.59	8,000,400	Mix2	0.34	3,000,050
freqmine	0.29	3,000,000			
vips	0.21	2,700,000			
×264	0.20	2,870,000			

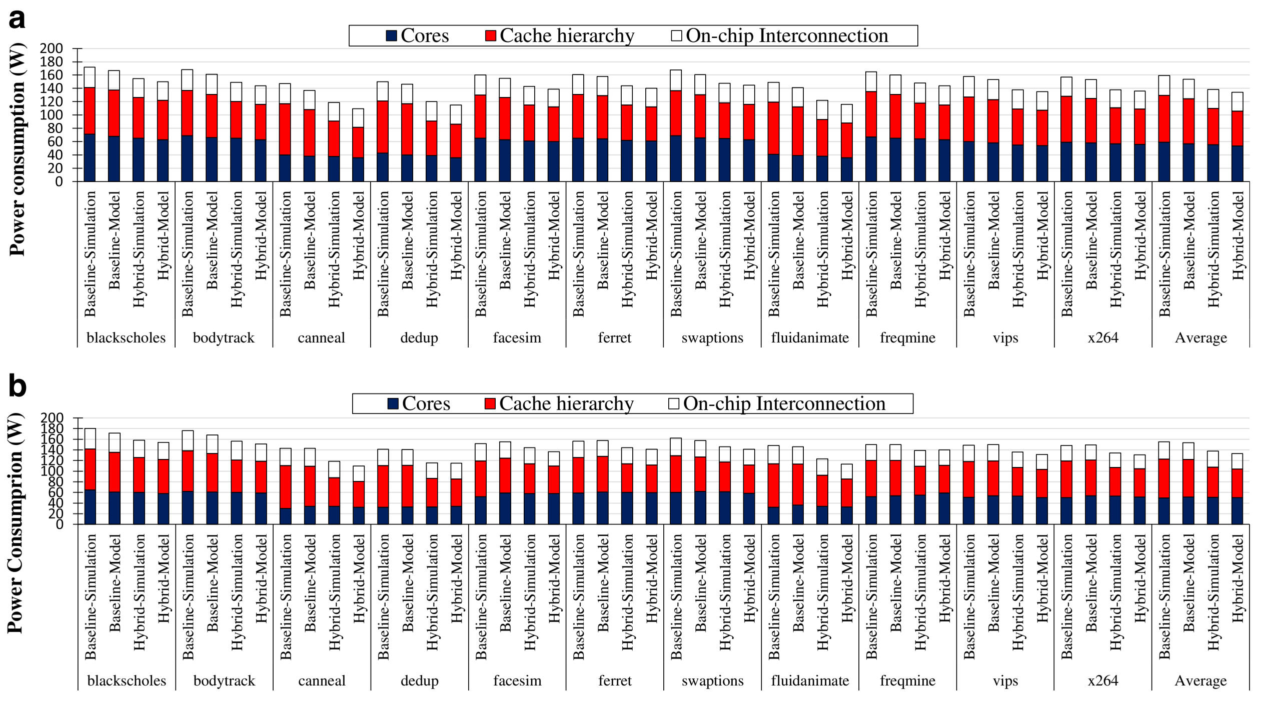

Fig. 7 Validation of the power model under multi-threaded workloads for **a** 16-core CMP and **b** 64-core CMP

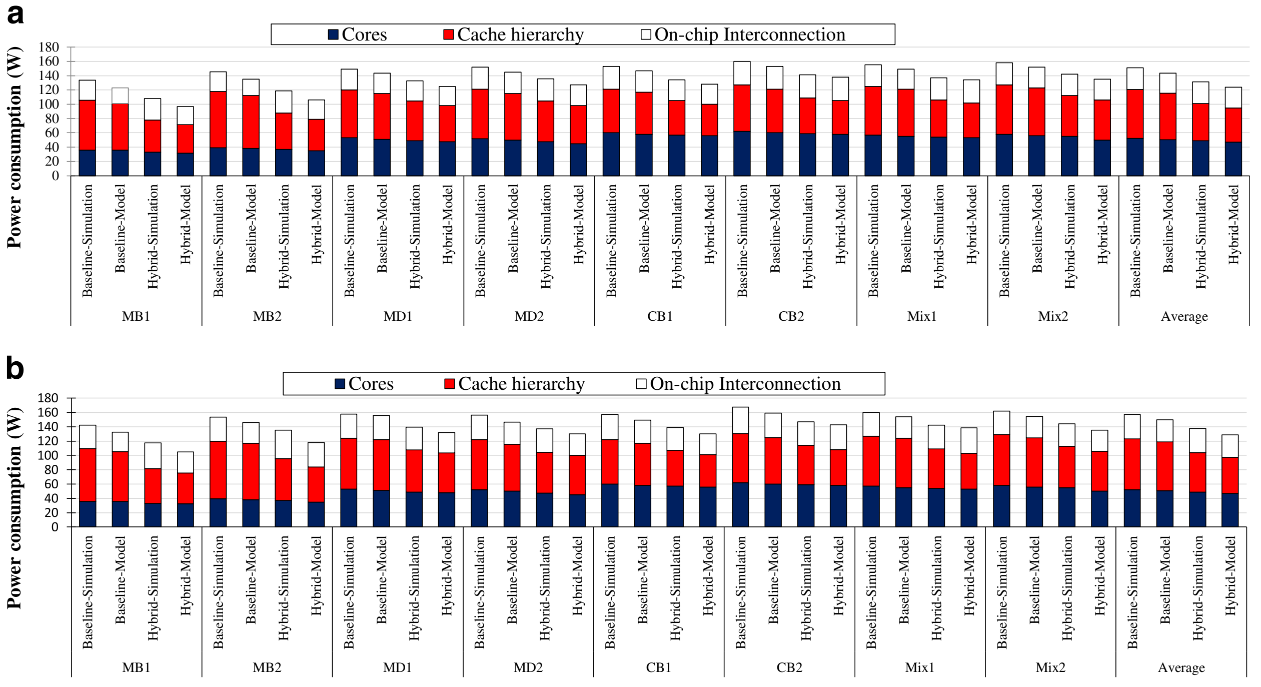

Fig. 8 Validation of the power model under multi-programed workloads for **a** 16-core CMP and **b** 64-core CMP

misses, packet latency, and number of packets transferred. When the utilization of the last-level cache of the hierarchy is high (LLC_Util > 5%), the number of cache miss rate increases and the workload needs a larger cache capacity to better fit the active memory footprint.

For the SPEC benchmarks, we fast-forward 500M instructions and run in detailed mode for the next 1 billion instructions. For PARSEC benchmarks, we run 1 billion instructions starting from the region of interest (ROI), using the *simlarge* input set. We used Ruby in the Gem5 that considers stalls in cores and blocking time effect in generating traces for the workloads. The proposed model considers the stalls and packet blocking time as well according to the use of the concept of stall time and blocking effect modelling in recent studies [43, 52].

Figures 7 and 8 compare the result of power consumption for the simulation and analytical model of baseline and proposed architecture under running both multi-threaded and multi-program workload, respectively.

According to Table 7, *canneal* and *MB1* applications, with the largest LLC_Util, are memory-intensive workloads which utilize the last-level cache heavily. In these applications, as shown in Figs. 7a and 8a, cache hierarchy consumes more power consumption compared with cores because cores are mostly in stall stage. *Swaption* and *CB2* applications, with the smallest LLC_Util, are computation-intensive workloads and, as shown in Figs. 7a and 8a, have higher core power consumption compared with other workloads. Compared with baseline CMP, the proposed hybrid CMP improves the power consumption of cores, cache hierarchy, and on-chip interconnection by about 6.3, 22.5, and 5.0% on average under execution of multi-threaded workload. The hybrid CMP improves cores and cache hierarchy power consumption by about 6.14 and 24.14%, respectively, and worsens the on-chip interconnection power consumption by about 0.14% on average that is negligible.

Table 9 Difference of simulation and proposed model under multi-threaded workloads for a 16-core CMP

Workload	CMP	Cores (%)	Cache hierarchy (%)	NoC (%)
blackscholes	Baseline	4.225	0.714	6.452
	Hybrid	3.077	3.279	1.754
bodytrack	Baseline	4.348	4.412	3.226
	Hybrid	3.077	3.636	3.448
canneal	Baseline	5.000	9.091	3.333
	Hybrid	4.762	14.151	0.000
dedup	Baseline	6.977	1.282	0.000
	Hybrid	7.692	3.846	0.000
facesim	Baseline	3.077	3.077	3.333
	Hybrid	1.639	3.704	3.571
ferret	Baseline	1.991	0.915	3.333
	Hybrid	1.613	3.774	3.448
swaptions	Baseline	4.360	4.425	3.226
	Hybrid	2.326	1.852	0.000
fluidanimate	Baseline	4.878	6.410	3.333
	Hybrid	5.263	5.455	3.448
freqmine	Baseline	2.985	2.941	3.333
	Hybrid	1.563	3.704	3.333
vips	Baseline	5.172	16.923	3.333
	Hybrid	1.818	1.852	3.448
x264	Baseline	1.695	2.899	3.448
	Hybrid	1.754	1.852	0.000
Average	Baseline	3.726	4.424	3.029
	Hybrid	3.144	4.282	2.041

Table 10 Difference of simulation and proposed model under multi-programed workloads for a 16-core CMP

Workload	CMP	Cores (%)	Cache hierarchy (%)	NoC (%)
MB1	Baseline	0.084	6.871	20.929
	Hybrid	4.545	11.111	16.667
MB2	Baseline	3.357	5.852	16.847
	Hybrid	5.149	13.725	12.903
MD1	Baseline	3.774	4.755	2.069
	Hybrid	2.449	9.222	4.643
MD2	Baseline	3.846	5.797	3.226
	Hybrid	5.263	7.018	6.452
CB1	Baseline	3.333	3.279	6.250
	Hybrid	1.754	8.333	3.448
CB2	Baseline	3.226	6.154	3.030
	Hybrid	1.695	6.000	− 3.125
Mix1	Baseline	3.509	2.941	6.667
	Hybrid	1.852	5.769	− 3.226
Mix2	Baseline	3.448	2.899	6.452
	Hybrid	9.091	1.754	3.333
Average	Baseline	3.072	4.819	8.184
	Hybrid	3.975	7.867	5.137

Table 11 Standard deviation of the simulation and proposed model

Workload	CMP	Cores	Cache hierarchy	NoC
Multi-threaded (16 cores)	Baseline	0.0153	0.0472	0.0144
	Hybrid	0.0198	0.0345	0.0169
Multi-threaded (64 cores)	Baseline	0.0653	0.0471	0.0336
	Hybrid	0.0393	0.0311	0.0435
Multi-program (16 cores)	Baseline	0.0123	0.0159	0.0692
	Hybrid	0.0258	0.0364	0.0696
Multi-program (64 cores)	Baseline	0.0123	0.0108	0.0250
	Hybrid	0.0258	0.0272	0.0435

In this trend, we evaluate the scalability of our proposed model for a 64-core CMP as shown in Figs. 7b and 8b. By increasing the number of cores in memory-intensive workloads, the increasing of power consumption is much higher due to higher uncore power consumption in comparison with computation-intensive applications in both multi-threaded workloads and multi-programed workloads. It should be noted that the power consumption of the target architecture is limited under power budget and temperature limit that are given by a designer-specified value.

As shown in Figs. 7 and 8, the proposed power model estimates the power consumption of heterogeneous (hybrid) and homogenous (baseline) 3D CMPs, with a good degree of accuracy, under running both multi-programed and multi-threaded workloads. Tables 9 and 10 show the difference of values between the simulation and proposed model for both multi-threaded and multi-program workloads in a 16-core CMP. To evaluate the degree of accuracy, we calculate standard deviation (STDEV) of the simulation and proposed model under different benchmarks and architectures. As reported in Table 11, the value of the proposed model is truly close to the value of the simulation. The STDEV of the baseline and hybrid CMP is negligible and about 0.0198 and 0.0153 for estimation of core power under running multi-threaded workloads for a 16-core CMP. In addition, we estimate the cache hierarchy power consumption with STDEV of 0.0472 and 0.159 for the baseline CMP and about 0.0345 and 0.0364 for hybrid CMP under the multi-threaded and multi-program workloads.

8 Conclusions

In this work, we proposed an accurate power model that formulates the power consumption of 3D CMPs with stacked cache layers. The proposed model that considers power impact of uncore beside the cores for the first time is appropriate for heterogeneous and non-heterogeneous CMPs under multi-threaded and multi-programed workloads. In the future, we can use this model in the optimization problems to minimize power consumption or maximize performance of CMPs under latency and temperature constraints. Moreover, we can use this power model in the prediction functions of machine learning-based power/thermal management techniques for future power-aware CMPs.

Authors' contributions

AA, the first author of this paper, proposed the mentioned power model for future embedded chip-multiprocessors. She implemented a platform setup for this work for the first time under the supervision of Professor FM, the third author of this paper. Professor FM participated in the evaluation of the proposed model and helped to draft the manuscript. All authors read and approved the final manuscript. AD as the second author of this paper helped to prepare the simulation results of this work.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 23 August 2017 Accepted: 2 July 2018
Published online: 27 July 2018

References

1. Kao, J, Narendra, S, Chandrakasan, A (2002). Subthreshold leakage modeling and reduction techniques. In IEEE/ACM Int. Conf. Comput.-aided design (ICCAD) (pp. 141–148).
2. Kim, NS, et al. (2003). Leakage current: Moore's law meets static power. *Computer*, ***36***(12), 68–75.
3. Wang, W, & Mishra, P. (2012). System-wide leakage-aware energy minimization using dynamic voltage scaling and cache reconfiguration in multitasking systems. *IEEE Trans. Very Large Scale Integr. VLSI Syst.*, ***20***(5), 902–910.
4. Esmaeilzadeh, H, Blem, E, St. Amant, R, Sankaralingam, K, Burger, D (2011). Dark silicon and the end of multicore scaling. In *Proc. Int. Symp. Comput. Archit*. (pp. 365–376).
5. Henkel, J, Khdr, H, Pagani, S, & Shafique, M (2015). New trends in dark silicon. In *Proc. Design Automation Conf. (DAC)*, (pp. 1–6).
6. Bose, P. (2013). Is dark silicon real?: technical perspective. *Commun. ACM*, ***56***(2), 92.
7. Taylor, MB (2012). Is dark silicon useful?: harnessing the four horsemen of the coming dark silicon apocalypse. In *Proc. Design Autom. Conf. (DAC)*, (pp. 1131–1136).
8. Jammy, R (2009). Materials, process and integration options for emerging technologies. In *SEMATECH/ISMI symposium*.
9. Woo, DH, Seong, NH, Lewis, DL, Lee, H-HS (2010). An optimized 3D-stacked memory architecture by exploiting excessive, high-density TSV bandwidth. In *Int. Symp. High Perf. Comput. Arch. (HPCA)*, (pp. 1–12).
10. Loh, GH (2008). 3D-Stacked Memory Architectures for Multi-core Processors. In *Int. Symp. .Comput. Arch. (ISCA)*, (pp. 453–464).
11. Kultursay, E, Kandemir, M, Sivasubramaniam, A, Mutlu, O (2013). Evaluating STT-RAM as an energy-efficient main memory alternative. In *Int. Symp. Performance Analysis of Systems and Software (ISPASS)*, (pp. 256–267).
12. Lee, BC, et al. (2010). Phase-change technology and the future of main memory. *IEEE Micro*, ***30***(1), 143–143.
13. Diao, Z, et al. (2007). Spin-transfer torque switching in magnetic tunnel junctions and spin-transfer torque random access memory. *J. Phys. Condens. Matter*, ***19***(16), 165–209.
14. Turakhia, Y, Raghunathan, B, Garg, S, Marculescu, D (2013). HaDeS: architectural synthesis for heterogeneous dark silicon chip multi-processors. In *Design Autom. Conf. (DAC)*, (p. 1).
15. Raghunathan, B, Turakhia, Y, Garg, S, Marculescu, D (2013). Cherry-picking: exploiting process variations in dark-silicon homogeneous chip multi-processors. In *Design, Autom. Test in Europe Conf. & Exhibition (DATE)*, (pp. 39–44).
16. Venkatesh, G, Sampson, J, Goulding-Hotta, N, Venkata, SK, Taylor, MB, Swanson, S (2011). QsCores: trading dark silicon for scalable energy efficiency with quasi-specific cores. In *Proc. Int. Symp. Microarchitecture*, (p. 163).
17. Li, S, Ahn, JH, Strong, RD, Brockman, JB, Tullsen, DM, Jouppi, NP (2009). McPAT: an integrated power, area, and timing modeling framework for multicore and manycore architectures. In *Proc. Int. Sym. Microarchitecture*, (pp. 469–480).
18. Binkert, N, et al. (2011). The gem5 simulator. *ACM SIGARCH Comput. Archit. News*, ***39***(2), 1.
19. Poremba, M, Zhang, T, Xie, Y. (2015). NVMain 2.0: a user-friendly memory simulator to model (non-)volatile memory systems. *IEEE Comput. Archit. Lett.*, ***14***(2), 140–143.
20. Chang M-T, Rosenfeld, P, Lu S-L, Jacob, B (2013). Technology comparison for large last-level caches (L3Cs): Low-leakage SRAM, low write-energy STT-RAM, and refresh-optimized eDRAM. In *Int. Symp. High Perfor. Comput. Archit. (HPCA)*, (pp. 143–154).
21. Chen, Y-T, et al. (2012). Dynamically reconfigurable hybrid cache: An energy-efficient last-level cache design. In *Design, Autom.n Test in Europe Conf. Exhib. (DATE)*, (pp. 45–50).
22. Dong, X, Xu, C, Jouppi, N, Xie, Y (2014). NVSim: A Circuit-Level Performance, Energy, and Area Model for Emerging Non-volatile Memory. In Y. Xie (Ed.), *Emerging Memory Technologies*, (pp. 15–50).
23. Muralimanohar, N, Balasubramonian, R, Jouppi, NP. (2009). CACTI 6.0: a tool to model large caches. *HP Lab.*, 22–31.

24. Wonyoung Kim, Gupta, M-S, Wei, G-Y, Brooks, D (2008). System level analysis of fast, per-core DVFS using on-chip switching regulators. In *Int. Symp. High Perfor. Comput. Archit. (HPCA)*, (pp. 123–134).
25. Wang, Y, Ma, K, Wang, X. (2009). Temperature-constrained power control for chip multiprocessors with online model estimation. *ACM SIGARCH Comput. Archit. News*, ***37***(3), 314.
26. Heo, S, Barr, K, Asanović, K (2003). Reducing power density through activity migration. In *Int. Symp. Low Power Electronics Design*, (p. 217).
27. Chantem, T, Hu, XS, Dick, RP. (2011). Temperature-aware scheduling and assignment for hard real-time applications on MPSoCs. *IEEE Trans. Very Large Scale Integr. VLSI Syst.*, ***19***(10), 1884–1897.
28. Ebi, T, Al Faruque, MA, Henkel, J (2009). TAPE: thermal-aware agent-based power economy for multi/many-core architectures. In IEEE/ACM *Int. Conf. Comput.-Aided Design-Digest of Technical Papers*, (p. 302).
29. Ebi, T, Kramer, D, Karl, W, Henkel, J (2011). Economic learning for thermal-aware power budgeting in many-core architectures. In *Proc. Int. Conf. Hardware/Software Codesign and Syst. Synthesis (CODES+ISSS)*, (p. 189).
30. Al Faruque, MA, Jahn, J, Ebi, T, Henkel, J. (2010). Runtime thermal management using software agents for multi- and many-core architectures. *IEEE Des. Test Comput.*, ***27***(6), 58–68.
31. Dorostkar, A, Asad, A, Fathy, M, Mohammadi, F (2017). Optimal Placement of Heterogeneous Uncore Component in 3D Chip-Multiprocessors. In *Euromicro Conf. Digital System Design (DSD)*, (pp. 547–551).
32. Shelepov, D, et al. (2009). HASS: a scheduler for heterogeneous multicore systems. *ACM SIGOPS Oper. Syst. Rev*, ***43***(2), 66.
33. Asad, A, Ozturk, O, Fathy, M, & Jahed-Motlagh, MR. (2015). Exploiting Heterogeneity in Cache Hierarchy in Dark-Silicon 3D Chip Multi-processors. In Euromicro Conf. Digital Syst. Design (DSD) (pp. 314–321).
34. Sharifi, A, Mishra, AK, Srikantaiah, S, Kandemir, M, Das, CR (2012). PEPON: performance-aware hierarchical power budgeting for NoC based multicores. In *Proc. Int. Conf. Parallel archit. compilation techn.* (p. 65).
35. Nawathe, UG, Hassan, M, Yen, KC, Kumar, A, Ramachandran, A, Greenhill, D. (2008). Implementation of an 8-Core, 64-thread, power-efficient SPARC server on a chip. *IEEE J. Solid State Circuits*, ***43***(1), 6–20.
36. Gebhart, M, Hestness, J, Fatehi, E, Gratz, P, & Kwcker, SW. (2009). "Running PARSEC 2.1 on M5". Univ. Te. Austin, Dep. Comput. Sci.e, Tech. Rep.
37. Shimpi, AL, Klug, B. (2011, Oct.) Apple iphone 4s: Thoroughly reviewed. [Online]. Available: http://www.anandtech.com/show/4971/apple-iphone-4s-review-att-verizon/5.
38. "Standard performance evaluation corporation" [Online], Available: http://www.specbench.org.
39. Murali, S, Coenen, M, Radulescu, A, Goossens, K, De Micheli, G (2006). Mapping and configuration methods for multi-use-case networks on chips. In *Proc. Asia South Pacific Design Autom. Conf. (ASP-DAC)*, (pp. 146–151).
40. Skadron, K, Stan, MR, Sankaranarayanan, K, Huang, W, Velusamy, S, Tarjan, D. (2004). Temperature-aware microarchitecture: modeling and implementation. *ACM Trans. Archit. Code Optim.*, ***1***(1), 94–125.
41. Su, H, Liu, F, Devgan, A, Acar, E, Nassif, S (2003). Full chip leakage estimation considering power supply and temperature variations. In Proc. *Int. Symp. Low power electronics design (ISLPED)*, (pp. 78–83).
42. Hartstein, A, Srinivasan, V, Puzak, TR, Emma, PG (2006). Cache miss behavior: is it √2? In *Proc. Conf. Computing frontiers (CF)*, (pp. 313–320).
43. Asad, A, Zonouz, AE, Seyrafi, M, Soryani, M, Fathy, M (2009). Modeling and Analyzing of Blocking Time Effects on Power Consumption in Network-on-Chips. In *Int. Con. Reconfig. Computing FPGAs (ReConFig)*, (pp. 290–295).
44. Shen, Z. (2002). The calculation of average distance in mesh structures. *Comput. Math. Appl.*, ***44***(10–11), 1379–1402.
45. Zhan, J, Xie, Y, Sun, G (2014). NoC-Sprinting: Interconnect for Fine-Grained Sprinting in the Dark Silicon Era. In ACM/EDAC/IEEE *Design Autom. Conf. (DAC)*, (pp. 1–6).
46. Zhan, J, Ouyang, J, Ge, F, Zhao, J, Xie, Y (2015). DimNoC: a dim silicon approach towards power-efficient on-chip network. In ACM/EDAC/IEEE *Design Autom. Conf. (DAC)*, (pp. 1–6).
47. Sun, C, et al. (2012). DSENT - A Tool Connecting Emerging Photonics with Electronics for Opto-Electronic Networks-on-Chip Modeling. In IEEE/ACM Int. Symp. Net. on Chip (NoCS), (pp. 201–210).
48. Grant, M, Boyd, S, Ye, Y. CVX: Matlab software for disciplined convex programming. [online] Available: http://cvxr.com/cvx/.
49. Catania, V, Mineo, A, Monteleone, S, Palesi, M, Patti, D (2015). Noxim: An open, extensible and cycle-accurate network on chip simulator. In Int. Conf. *Application-specific Syst., Archit. Processors (ASAP)*, (pp. 162–163).
50. Kahng, AB, Lin, B, Nath, S. (2015). ORION3.0: a comprehensive NoC router estimation tool. *IEEE Embed. Syst. Lett.*, ***7***(2), 41–45.
51. Bienia, C, Kumar, S, Singh, JP, Li, K (2008). The PARSEC benchmark suite: characterization and architectural implications. In *Int. Conf. Parallel Archit. Compilation Techn. (PACT)*, (p. 72).
52. Ogras, UY, Marculescu, R (2007). Analytical Router Modeling for Networks-on-Chip Performance Analysis. In *Design, Autom. Test Euro. Conf. Exhib. (DATE)*, (pp. 1–6).
53. Krishna, A, Samih, A, Solihin, Y. (2012). Data sharing in multi-threaded applications and its impact on chip design. In *Int Symp Perfor. Analysis Sys. Software (ISPASS)*, (pp. 125–134).