Runtime Performance Optimization of 3-D Microprocessors in Dark Silicon

Hai Wang, *Member, IEEE,* Wei Li, Wenjie Qi, Diya Tang,
Letian Huang, *Member, IEEE,* and He Tang, *Member, IEEE*

Abstract—Because the increasing power density is limited by the thermal constraint, multi-core integrated systems have stepped into the dark silicon era recently, meaning not all parts of the system can be powered on at the same time. Dark silicon effects are especially severe for 3-D microprocessors due to the even higher power density caused by the stacked structures, which greatly limit the system performances. In this work, we propose a greedy based core-cache co-optimization algorithm to optimize the performance of 3-D microprocessors in dark silicon at runtime. The new method determines many runtime settings of the 3-D system on the fly, including the active core and cache bank positions, active cache bank number, and the voltage/frequency (V/f) level of each active core, which optimizes the performance of the 3-D microprocessor under thermal constraint. Because the core-cache settings are co-optimized in the 3-D space and the power budgets are computed dynamically according to the running state of the 3-D microprocessor, the new method leads to a higher system performance compared with the existing methods. Experiments on two 3-D microprocessors show the greedy based core-cache co-optimization algorithm outperforms the state-of-the-art 3-D dark silicon microprocessor performance optimization method by achieving a higher processing throughput with guaranteed thermal safety.

Index Terms—Performance optimization, thermal management, heterogeneous system, 3-D IC, dark silicon.

1 Introduction

Three-dimensional (3-D) microprocessors have been proposed to explore the vertical integration potential of the integrated systems [1]. With stacking multiple dies layer by layer vertically, 3-D microprocessors achieve better performance than traditional 2-D microprocessors in many aspects including better heterogeneous integration ability, higher computing density, and shorter interconnection delay [2]. In order to realize the 3-D microprocessor design, several 3-D stacking technologies were introduced including the through-silicon via (TSV) based stacking, face-to-face (F2F) bonded stacking, and monolithic 3-D (M3D) technology. Electronic design automation programs for 3-D systems were also developed like the thermal modeling [3] and physical design [4] tools. Some commercial 3-D chips and prototypes were built recently, such as the AMD Radeon R9 Fury GPU [5] and the 3D-MAPS CPU [6].

The major issue that 3-D structure solves is the memory wall problem [7], [8], which describes the fact that the long memory access delay in traditional 2-D microprocessors is the bottleneck that limits the overall performance of the system. There are many 3-D architectures proposed including the stacking main memory architecture, the stacking cache architecture, and the stacking cache+core architecture [9]. Among these 3-D architectures, the stacking cache+core architecture is promising because it resolves the memory wall problem between cache and core [10], and a test chip has been designed and realized as the first fully-functioning general purpose many-core 3-D processor [6]. The stacking cache+core 3-D microprocessor structure has one or several cache memory layers stacked on the top of the core layer, as shown in Fig. 1. With the reduced interconnection length, 3-D microprocessors achieve faster cache access speed, which leads to higher computing throughput compared with the 2-D microprocessors [11].

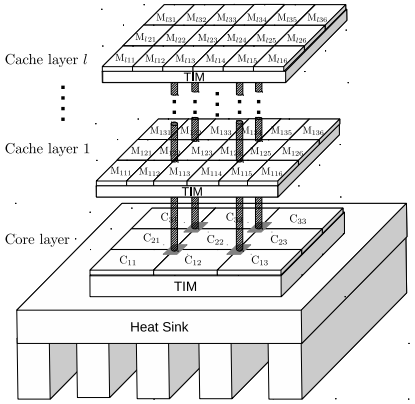


Fig. 1: The 3-D microprocessor structure with one core layer and l cache layers connected by TSVs (shown as the vertical bars). The cores and memory controllers (shown as the gray squares in the Core layer) are connected by crossbar switch interconnection.

Despite many advantages, 3-D microprocessors experience severe thermal induced dark silicon problem. Specifically, Dennard scaling, which states the power density remains constant with technology scaling and integration,

- *H. Wang, W. Li, W. Qi, D. Tang, L. Huang, and H. Tang are with State Key Laboratory of Electronic Thin Films and Integrated Devices, University of Electronic Science and Technology of China, Chengdu, 610054 China, and also with School of Electronic Science and Engineering, University of Electronic Science and Technology of China, Chengdu, 610054 China.*

(Corresponding author: He Tang).

becomes invalid in recent years [12]. The rising power density in the post Dennard scaling era leads to the *dark silicon* phenomenon, which means not all parts of the integrated system can be powered on at the same time in order to satisfy the thermal constraint. The part of the system which is deactivated is called dark silicon [12], [13], [14], [15]. Previous studies have revealed that 3-D microprocessors have higher power density due to the stacked structure [3], [16], [17], indicating they have more severe dark silicon phenomenon than the traditional 2-D microprocessors. It means an even larger part of the circuity in 3-D microprocessors cannot be powered on at the same time, which reduces the performance advantage brought by the 3-D integration [18].

In order to improve the performance of the microprocessors in dark silicon, special power budgeting methods were proposed, such as the greedy dynamic power (GDP) [13] and the thermal safe power (TSP) [14]. In addition, many dark silicon thermal/power management techniques were introduced based on the power budgets [19], [20], [21]. Unfortunately, the aforementioned methods cannot be applied to 3-D microprocessors, because they cannot consider the vertical thermal coupling effects of the active components across multiple die layers. For 3-D microprocessors, the active component distribution and the power budgets in one die layer will greatly affect the active component distribution and the power budgets in other layers, due to the high vertical heat conductivity in the 3-D package. In addition, the overall performance of the 3-D microprocessor in dark silicon is not determined solely by the active components in one layer, but by the heterogeneous active components in all layers as a full system. As a result, a complex running state coordination across all layers is required for a 3-D microprocessor, in order to optimize its performance.

In this work, we seek for a runtime performance optimization method for 3-D microprocessors in dark silicon which determines the optimal core and cache settings, including the distribution and V/f levels of the active cores as well as the distribution and number of the cache banks, such that the overall system throughput is maximized.

The major contributions of this work include:

- We have proposed a core-cache co-optimization framework for 3-D microprocessors in dark silicon which considers both thermal coupling and performance coupling effects between cache and core. Through the core-cache co-optimization, the optimal core settings (active core distribution and V/f levels) and cache settings (active cache bank number and distribution) can be found which lead to the optimal overall system performance.
- A greedy based iterative algorithm has been introduced to determine the optimal power budgets and distribution of the active cores. The thermal impact from the active cache banks is taken into account during this greedy based process to optimize the core settings. Since the greedy based iterative algorithm has a polynomial time complexity, the optimal core settings can be found at runtime.
- We have proposed a cache setting optimization algorithm which adjusts the active cache bank number and positions dynamically to improve the system performance with the active core settings in mind.
- We have experimentally compared the new method with the state-of-the-art method on two 3-D microprocessors with different architectures. The results show that the new method outperforms the existing method in the overall system throughput with guaranteed system thermal safety.

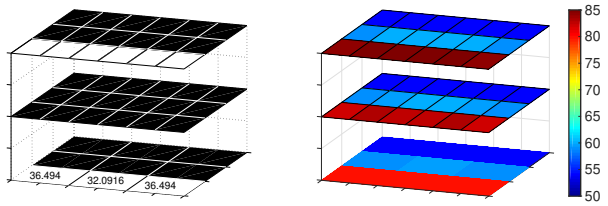
2 Related work

In this section, we review the related work on the performance optimization of 3-D microprocessors in dark silicon.

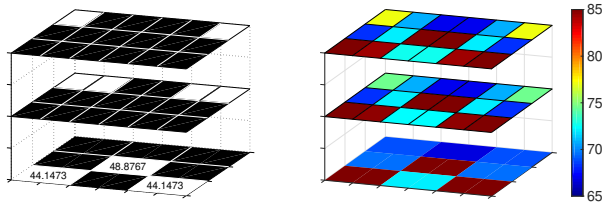
Many runtime performance optimization methods were proposed for 3-D microprocessors with temperature as the constraint. In [22] and [23], power and energy optimization methods were proposed with both power constraint and thermal constraint. Several task scheduling based methods were proposed to manage temperature dynamically for 3-D microprocessors [24], [25], [26], [27], [28]. In [11] and [29], thermal management and performance optimization methods were introduced for 3-D microprocessors with hybrid SRAM/MRAM L2 caches. Zou *et al.* proposed a thermal management method considering the thermal induced stress in 3-D systems [30]. With artificial neural network based runtime stress estimator [31], STREAM was introduced to optimize the 3-D microprocessor performance considering the thermal induced reliability problems [17]. However, the aforementioned methods were not designed for 3-D microprocessors in dark silicon, where only a fraction of the active components are powered on at the same time.

In recent years, many research studies were conducted to optimize the performance of the multi/many-core systems with the emerging dark silicon phenomenon. Thermal safe power (TSP) [14] was introduced to provide a less pessimistic static power budget than thermal design power (TDP) [32] for dark silicon multi-core systems. Greedy dynamic power (GDP) was proposed to provide a dynamic power budget according to the runtime state of the multi-core system in dark silicon [13]. In [19], researchers proposed a hierarchical power management scheme to improve the performance of the dark silicon system. A dark silicon aware scheduling method was proposed in [33] to boost the system performance considering process variation. In [20], researchers proposed a dynamic programming based algorithm to determine the optimal active core settings. Kanduri *et al.* proposed a dark silicon patterning method which increases power budget to enhance the system performance [21]. Recently, cache-aware task mapping [34] and task migration [35] algorithms were developed for static non-uniform cache access (S-NUCA) many-core systems based on TSP.

Optimizing the performance of the 3-D multi-core systems with dark silicon phenomenon is much more challenging than optimizing the 2-D systems. There are limited existing methods introduced to solve this problem. The state-of-the-art method was proposed in [36], which considers the impact of power consumptions of cores and un-core components simultaneously to improve the 3-D system performance in the dark silicon era. However, due to the limitation of the algorithm and the simplified two-dimensional model,



(a) The total power budget of the active cores is low when the active components cluster together in 3-D space.



(b) The total power budget of the active cores is high when the active components are uniformly distributed in 3-D space.

Fig. 2: The impact of the active cache bank and active core locations on the power budgets of the cores tested using the 9-core 3-D microprocessor shown in Fig. 1. In each subfigure, the left image shows the active component distribution (the active components are in white and the inactive components are in black) and the power budgets (shown as numbers in the active cores with unit W), the right image shows the temperature distribution with unit °C.

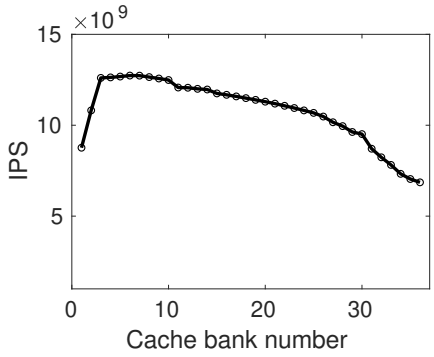


Fig. 3: The impact of the active cache number on the performance of the 9-core 3-D system. Throughputs (IPS) of the 3-D system with different number of active cache banks are plotted by running *Swaptions* benchmark on 3 active cores. When the active cache bank number increases, system throughput will first increase due to the decrease of cache access latency, but it will decrease later because the heat generated by the active cache banks forces the active cores to lower their operating frequencies.

the existing method requires the cache bank and the core in the same vertical column to be activated at the same time, which lowers the power budgets of the active cores. Moreover, this method is based on the static power budget, which over-constrains the system performance at runtime.

3 Motivation

In this section, we provide an example which motivates this research work.

In this example, we show two typical active component distributions of a 9-core 3-D microprocessor in Fig. 2. In Fig. 2a, the active cores are clustered and the active cache banks overlap the active cores vertically. Whereas in Fig. 2b, the active cores and the active cache banks stagger from each other vertically without forming any clusters. Obviously, the 3-D microprocessor with the latter active component distribution has a better heat dissipation condition than the former one, with two main reasons. First, inside each layer, the active components in Fig. 2b are more uniformly distributed without forming large active component clusters. With the uniform distribution, the active cores have better lateral heat conduction, which leads to higher power budgets and higher system performance potential. Second, in Fig. 2b, there is no active cache bank in each vertical stack with active core. This brings more power budget for the active core in each vertical stack, which further boosts the potential system performance. Experimentally, we have computed the power budgets of the 3-D microprocessor with two typical active component distributions in Fig. 2. Clearly, the 3-D microprocessor with a more uniform active component distribution in the three-dimensional space has a higher overall power budget, with the same temperature threshold as the constraint.

Despite the active component distribution, the active cache bank number also influences the performance of the 3-D microprocessor, in two opposite directions. For the influence in one direction, activating more cache banks enlarges the cache size, which increases the cache hit rate and improves the system performance. For the influence in the other direction, activating too many cache banks in the 3-D system will pose too much thermal pressure on the active cores, forcing them to lower their V/f levels to avoid thermal constraint violation. To see the impact of the active cache number on the system performance, we plot the throughputs (as measured by instructions per second (IPS)) of the 9-core 3-D microprocessor with different number of active cache banks in Fig. 3. In the beginning, the system throughput increases with the active cache bank number, because more active cache banks bring higher cache hit rate and reduce the cache access delay. When the active cache bank number grows beyond 6, activating more cache banks leads to a throughput decrease. This is because activating cache banks increases the power consumptions of the cache layers, the active cores have to reduce their power consumption by lowering their performances through dynamic voltage and frequency scaling (DVFS), in order to keep the 3-D microprocessor thermally safe.

As discussed above, the performance of the 3-D microprocessor in dark silicon is affected by many runtime parameters in a complex way. These parameters include the distribution of the active components in 3-D space, the active cache bank number, and the V/f levels of the active cores. In this work, we provide a systematic method named the greedy based core-cache co-optimization algorithm to optimize the performance of the 3-D microprocessor by finding the optimal runtime parameters at runtime.

4 Background

The basic structure of the 3-D microprocessor is shown in Fig. 1. We assume there are one core layer and l ($l \geq 1$) L2 cache layers in the 3-D system. The core layer contains n_c cores and each cache layer contains n_m cache banks (making a total of $n_m l$ cache banks in the full system). The power, performance, and thermal models of the 3-D microprocessor are presented in this section.

4.1 Power and performance models

Power of the 3-D microprocessor is composed of dynamic power and leakage power (which is also called static power).

Dynamic power, denoted as p_d, depends on the activity of the active component, which is expressed as

$$p_d = \alpha C_e V^2 f, \tag{1}$$

where α is the activity factor, C_e is the equivalent load capacitance, V and f are the operating voltage and frequency, respectively.

Leakage power, denoted as p_s, does not relate to the activity of the active component. Instead, it mainly depends on the temperature and is written as

$$p_s = V I_{leak}(T_p), \tag{2}$$

where T_p is the temperature in scalar form (we reserve T for the temperature in vector form), I_{leak} is the leakage current, which is a monotonic increasing nonlinear function of temperature T_p. For the details of leakage power modeling, simulation, and control, please refer to our previous work [37], [38], [39].

The performance of the system is measured as the total instructions per second (IPS) of all cores, abbreviated as TIPS in this paper. For each core, its IPS is expressed as

$$\text{IPS} = f/(\text{CPI}_b + \text{Memory stall cycles per instruction}) \tag{3}$$

where CPI_b stands for the base clock cycles per instruction with no memory access delay. The IPS estimation using performance counts and cache miss rate has been presented in [10], [11]. In order to estimate IPS with modified cache bank number, offline training or advanced methods like [40] can be used to predict the cache miss rate.

4.2 Thermal model

The thermal model of the 3-D microprocessor is built by exploiting the duality between the thermal system and the electrical circuit system. By using the finite difference method, we can discretize the 3-D microprocessor into q three-dimensional grids and connect them using thermal equivalent resistors, capacitors, current sources, and voltage sources.

Since there are n_c cores in the core layer and totally $n_m l$ cache banks in the l cache layers, the thermal model of the 3-D microprocessor is written as the following ordinary differential equations [17]:

$$\begin{aligned} GT(t) + C\frac{\mathrm{d}T(t)}{\mathrm{d}t} &= \underbrace{\begin{bmatrix} B_c & B_{m_1} & \cdots & B_{m_l} \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} P_c(t) \\ P_{m_1}(t) \\ \vdots \\ P_{m_l}(t) \end{bmatrix}}_{P(t)}, \\ \underbrace{\begin{bmatrix} Y_c(t) \\ Y_{m_1}(t) \\ \vdots \\ Y_{m_l}(t) \end{bmatrix}}_{Y(t)} &= \underbrace{\begin{bmatrix} L_c \\ L_{m_1} \\ \vdots \\ L_{m_l} \end{bmatrix}}_{L} T(t). \end{aligned} \tag{4}$$

In (4), $T(t) \in \mathbb{R}^q$ is the thermal vector representing temperatures of q grids of the 3-D microprocessor (including grids for cores, caches, and package parts), $G \in \mathbb{R}^{q \times q}$ matrix includes thermal resistance information; $C \in \mathbb{R}^{q \times q}$ matrix includes thermal capacitance information; $B \in \mathbb{R}^{q \times (n_c + n_m l)}$ matrix contains the power injection topology information; $P(t) \in \mathbb{R}^{(n_c + n_m l)}$ is the power vector with power dissipations of the cores and caches (as in power models (1) and (2)), and it is the input of the model. $Y(t) \in \mathbb{R}^{(n_c + n_m l)}$ is the thermal vector with temperature information of the n_c cores and $l \times n_m$ cache banks, and it is the output of the model; $L \in \mathbb{R}^{(n_c + n_m l) \times q}$ is the output selection matrix, which selects the $(n_c + n_m l)$ temperatures in the core and cache layers from $T(t)$.

We also divide the matrices B, $P(t)$, $Y(t)$, and L in (4) into block matrices according to the die layers. To be specific, B_c, $P_c(t)$, $Y_c(t)$, and L_c are the corresponding block matrices for the core layer; B_{m_i}, $P_{m_i}(t)$, $Y_{m_i}(t)$, and L_{m_i} are the corresponding block matrices for the cache layer i.

For the details of the internal structure of the thermal model matrices (G, C, B, and L), please refer to the thermal modeling work such as [23], [41].

5 The Greedy Based Core-Cache Co-Optimization Algorithm for 3-D Microprocessors in Dark Silicon

In this section, we present the greedy based core-cache co-optimization algorithm for runtime performance optimization of 3-D microprocessors in dark silicon.

5.1 The basic idea and flow

As discussed previously in Section 1, the major drawback of the existing method is the inability to perform co-optimization of cache and core in the three-dimensional space. The new method solves this problem with the basic flow for one optimization time step shown in Fig. 4.

The basic flow contains three main stages. In stage one, the new method will locate the active cores and compute their power budgets, by considering the cache effects using the cache settings (active cache bank number and distribution) from the previous optimization time step. This stage will be presented in detail in Section 5.2. In stage two, the power budgets of the active cores computed in stage one are updated according to the workloads, in preparation to

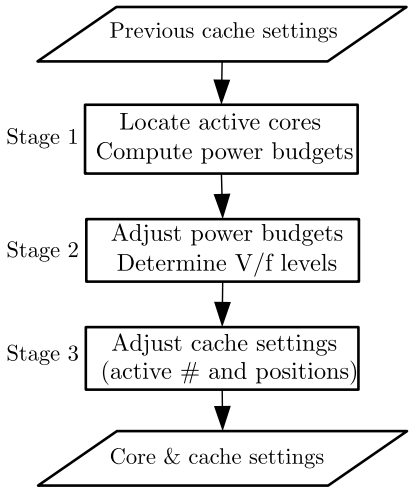


Fig. 4: The basic flow of the runtime performance optimization method for dark silicon 3-D microprocessors for one performance optimization time step.

determine the cache settings in stage three. The details of this stage will be presented in Section 5.3. In stage three, the cache settings will be adjusted using the core settings found in the previous two stages. The details of stage three will be presented in Section 5.4.

By following the basic flow presented above, the new method exploits the full performance potential of the 3-D structure by determining the optimal core and cache settings through the co-optimization in the three-dimensional space. Next, we will present the three main stages of the basic flow in detail. Please note that although the new method is fully compatible with transient temperature, we will demonstrate it in steady state for better presentation. The transient extension can be achieved by following our previous work (Section 4.3 of [13]), which will not be shown here due to page limitation.

5.2 Locate optimal positions and compute power budgets of the active cores

The active core positions have a significant impact on the performance of the dark silicon system as analyzed in [13], [14]: the total system power budget is generally higher when the active cores are more uniformly distributed, which further leads to a higher system performance. For 3-D microprocessors, the optimal active core distribution for high system performance is additionally influenced by the active cache bank distribution. Intuitively, the vertical overlapping between active core and active cache bank should be avoided, in order to gain more power budget to boost performance.

As a result, in each performance optimization time step, we will first determine the optimal positions and the power budgets of the active cores. To consider the impact of active cache banks on the core settings, we use the active cache bank number and distribution from the previous optimization time step.

The flow of locating the active cores and computing their power budgets is shown in Fig. 5, with the details presented in the following.

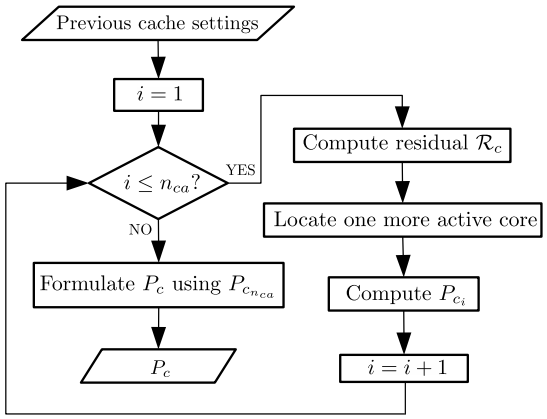


Fig. 5: The flow to locate the active cores and compute their power budgets.

5.2.1 Problem formulation

Our goal here is to determine the optimal distribution of the active cores which maximizes their total power budget to optimize the system performance, with the cache settings from the previous optimization step.

However, directly maximizing the total power budget leads to an optimization problem which is difficult to solve [13]. Instead, we formulate an equivalent thermal optimization problem which maximizes the average temperature of the innermost layer (cache layer l), which has the highest temperature among all die layers,[1] under thermal constraint as

$$\begin{aligned} &\text{minimize } \|\mathcal{Y} - Y_{m_l}\|_2 \\ &\text{subject to } \begin{cases} \text{card}(P_c) = n_{ca}, \\ Y_{m_l} \preceq \mathcal{Y}, \end{cases} \end{aligned} \tag{5}$$

where $\mathcal{Y} = [T_{th}, T_{th}, \ldots, T_{th}]^T \in \mathbb{R}^{n_m}$ is the temperature threshold vector with scalar T_{th} as the user defined temperature threshold value, n_{ca} is the active core number, $\preceq$ denotes for vector inequality, $\text{card}(P_c)$ means the cardinality of the vector P_c defined as the number of nonzero elements in P_c. The equivalence of maximizing the average temperature of the innermost layer and maximizing the total power budget has been theoretically proved in [13] and also experimentally demonstrated in [13], [14].

In order to consider the thermal impact of the cache layers on the optimal distribution and power budgets of the active cores, we need to subtract the thermal effects of the active cache banks from the temperature threshold vector in (5). To be specific, from the 3-D microprocessor thermal model in (4), we can derive the steady state response as:

$$\begin{aligned} Y_{m_l} &= L_{m_l} T \\ &= L_{m_l} G^{-1} (B_c P_c + B_{m_1} P_{m_1} + \cdots + B_{m_l} P_{m_l}). \end{aligned} \tag{6}$$

Please note that the cache related terms $L_{m_l} G^{-1} B_{m_1} P_{m_1}$, $\ldots$, $L_{m_l} G^{-1} B_{m_l} P_{m_l}$ in (6) are constants representing the thermal effects of the active cache banks, which should be subtracted. Then, the optimization problem in (5) is

1. Please note that the innermost layer has the highest temperature because it is at the end of the heat dissipation path.

equivalently transformed into

$$\begin{aligned} &\text{minimize } \|\mathcal{Y}_c - A_c P_c\|_2 \\ &\text{subject to } \begin{cases} \text{card}(P_c) = n_{ca}, \\ A_c P_c \preceq \mathcal{Y}_c, \end{cases} \end{aligned} \tag{7}$$

where $\mathcal{Y}_c = \mathcal{Y} - (L_{m_l} G^{-1} B_{m_1} P_{m_1} + \cdots + L_{m_l} G^{-1} B_{m_l} P_{m_l}) \in \mathbb{R}^{n_m}$ is the new constant temperature threshold vector by subtracting the thermal effects of the active cache banks, $A_c = L_{m_l} G^{-1} B_c \in \mathbb{R}^{n_m \times n_c}$ is a known constant matrix. The only variable in the optimization problem (7) is P_c, which needs to be solved. The distribution of the nonzero elements in P_c denotes the optimal distribution of the active cores, and the values of these nonzero elements are the power budgets of the active cores.

5.2.2 The greedy algorithm to locate the active cores

The problem in (7) is a combinational optimization problem, which requires enumeration to find the global optimal solution, making its solving process infeasible at runtime. In order to solve the optimization problem with low time complexity, a greedy iterative method is used instead to find the sub-optimal solution. The basic idea is to locate one additional optimal active core in each iteration, with the previously located active cores fixed.

To illustrate the iterative algorithm, we present the procedure of its $(i+1)$th iteration. Because we have already located i active cores in the previous i iterations, in the $(i+1)$th iteration, our goal is to find one more core to activate (i.e., the $(i+1)$th active core), which optimizes the optimization goal.

First, we subtract the thermal impact of the i fixed active cores from $\mathcal{Y}_c$ to form the residual $\mathcal{R}_c$ as

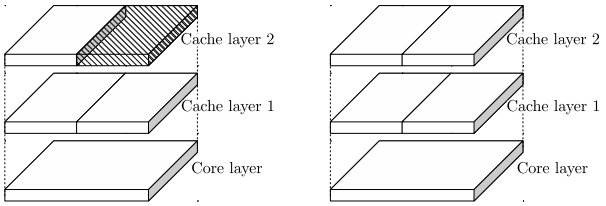
$$\mathcal{R}_c = \mathcal{Y}_c - A_{c_i} P_{c_i}, \tag{8}$$

where $A_{c_i} \in \mathbb{R}^{n_m \times i}$ is formed by the i columns of A_c corresponds to the i fixed active cores, $P_{c_i} \in \mathbb{R}^i$ contains the power budgets of the i fixed active cores computed in the previous iteration (the computation of power budgets will be presented in Section 5.2.3).

The physical meaning of the residual $\mathcal{R}_c$ is the temperature rise headroom (with temperature threshold as the ceiling) left when the previously located i cores are activated. As a result, in order to determine the optimal inactive core to be activated next, we check which remaining inactive core has the greatest ability to consume this temperature rise headroom. This is achieved by comparing the inner products of each remaining column of A_c (corresponds to an inactive core) and $\mathcal{R}_c$. The inactive core with the largest inner product is selected as the $(i+1)$th active core.

5.2.3 Compute power budgets in each greedy iteration

After the $(i+1)$th active core is picked, we need to update the power budgets of all $i+1$ active cores. Because of the thermal coupling between caches and cores, the active cache bank pattern in the vertical direction needs to be considered in this step. We can classify the active cache patterns above an active core in two cases:



(a) A non-uniform example: the right cache bank in cache layer 2 above an active core is inactive. The highest temperature will appear on the left cache bank of cache layer 2.

(b) A uniform example: all cache banks above an active core are active. The temperature distribution can be relatively uniform for cache layer 2.

Fig. 6: Examples of the two active cache patterns above an active core in a 3D system with 2 cache layers ($l = 2$). The white core/cache is active and the grey core/cache is inactive.

1) The active cache banks are non-uniformly distributed above the active core. In this case, the highest temperature of the innermost layer will appear on the side which has more active cache banks than the other side. An example of this non-uniform case is given in Fig. 6a.
2) The active cache banks are uniformly distributed above the active core. In this case, the temperature distribution of the innermost layer (cache layer l) can be relatively uniform. An example of this uniform case is shown in Fig. 6b.

For the non-uniform active cache pattern case, because the temperature of cache layer l is higher on the side with more active cache banks, we compute the power budget of the active core such that the temperature of the cache bank in cache layer l on the side with more vertical active cache banks (the temperature of the left cache bank of cache layer 2 (with $l = 2$) in Fig. 6a) reaches the threshold temperature. In other words, the power budget of the active core is constrained by the highest temperature in cache layer l, to avoid thermal violation.

For the uniform active cache pattern case, because the temperature distribution of cache layer l can be relatively uniform, we compute the power budget of the active core such that the average temperature of the two cache banks in cache layer l reaches the threshold temperature. In other words, the power budget of the active core is constrained by the average temperature of the two cache banks in cache layer l.

In order to realize the idea above, in the $(i+1)$th iteration, we introduce a selection matrix $L_s \in \mathbb{R}^{(i+1) \times n_m}$, whose values change according to the active core and active cache patterns. The job of L_s is to select the proper $i+1$ temperatures out of all the n_m temperatures of cache layer l, according to the active cache bank patterns above the $i+1$ active cores.

Then, the power budget of the $i+1$ active cores $P_{c_{i+1}}$ can be computed by simply solving

$$L_s \mathcal{Y}_c = L_s A_{c_{i+1}} P_{c_{i+1}}. \tag{9}$$

This completes the $(i+1)$th iteration of the greedy based algorithm.

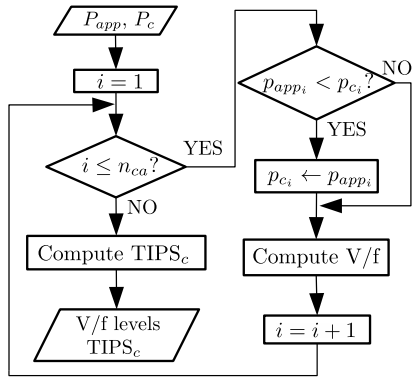


Fig. 7: The flow to adjust the power budgets and determine the V/f levels of the active cores.

When we finally finish all n_{ca} iterations, P_c can be readily formulated by filling its elements corresponding to the active cores with the elements in $P_{c_{n_{ca}}}$. The remaining elements of P_c, which correspond to the inactive cores, are left as zeros.

5.3 Adjust the power budgets and determine the V/f levels of the active cores

Although the positions and power budgets of the active cores are determined in Section 5.2, the power budgets need to be adjusted according to the workloads of the cores, before we can assign V/f levels to the active cores correctly. For example, some applications are memory bound, which spend the majority of time on memory and do not consume much CPU power even when it is allowed to. In this case, the power budgets of the corresponding cores should be lowered to match the real power consumption value, in order to make power budget headroom for the other active components.

The flow of this stage is shown in Fig. 7. To explain the steps in the flow, let us denote the power consumption of the ith active core with nominal V/f level as p_{app_i} and its power budget computed in Section 5.2 as p_{c_i}. If there is $p_{app_i} < p_{c_i}$, which means the ith core cannot fully consume the given power budget, its power budget should be adjusted by assigning p_{app_i} to p_{c_i} as $p_{c_i} \leftarrow p_{app_i}$.

Then, we can assign the V/f level to each active core according to the adjusted power budget: if there is $p_{c_i} < p_{app_i}$, DVFS is applied to the ith core, with the V/f level computed using the power models.

Finally, we can compute the throughput of the 3-D microprocessor under the current active component settings by computing the total instruction per second (TIPS) of all cores using the performance model. This system performance is recorded as TIPS$_c$.

5.4 Adjust the active cache bank number and positions

In the previous stages, the core settings have been computed using the previous cache settings. Now, we can update the cache settings, including the active cache bank number and distribution, using the newly computed core settings. The flow for this stage is given in Fig. 8.

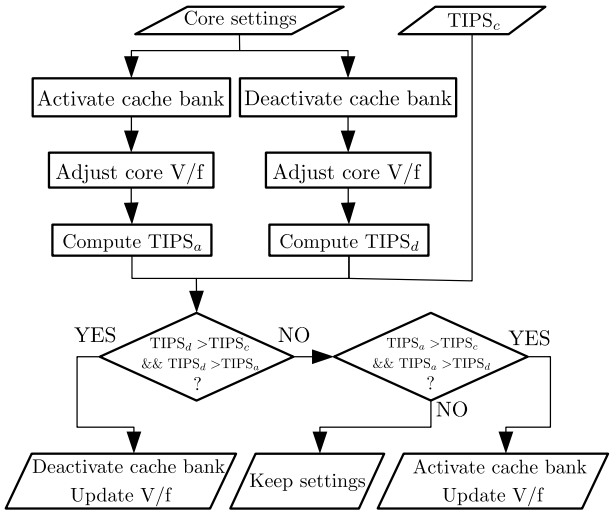


Fig. 8: The flow to adjust the active cache bank number and positions.

In determining the active cache bank number and distribution, we activate or deactivate only one cache bank in maximum within each performance optimization time step, to avoid large computing overhead. Then, there are three options in each step:

1) Activate one more cache bank from the inactive cache banks.
2) Deactivate one cache bank from the active cache banks.
3) Keep the current active cache banks.

Since the goal is to find the cache bank settings which lead to the best system performance, for the first two options, we need to activate/deactivate the *optimal* cache bank and estimate the corresponding system performances. For the third option, no operation is needed and its system performance has already been estimated as TIPS$_c$. Then, we pick the option which leads to the best system performance. The detailed steps are described in the following.

5.4.1 Activate one more cache bank

For the first option, we need to choose the optimal cache bank to activate, which leads to the best system performance. This problem is equivalent to finding one more cache bank to activate, which has the greatest potential in complementing the thermal impact of the fixed active cores (or in other words, which has the least thermal impact to the fixed active cores). For better presentation, we define two new boolean vectors $X_{cur} = [x_{cur}(1), x_{cur}(2), \ldots, x_{cur}(n_m l)] \in \mathbb{B}^{n_m l}$ and $X_{pre} = [x_{pre}(1), x_{pre}(2), \ldots, x_{pre}(n_m l)] \in \mathbb{B}^{n_m l}$ to represent the active cache bank distribution for the current time step and the previous time step, respectively, where $x_{cur}(i) \in \{0, 1\}$ and $x_{pre}(i) \in \{0, 1\}$ for $i = 1, 2, \ldots, n_m l$. For each element in the boolean vectors, $x_{cur}(j) = 0$ ($x_{pre}(j) = 0$) means the jth cache bank is inactive at the current time step (previous time step), and $x_{cur}(j) = 1$ ($x_{pre}(j) = 1$) represents the jth cache bank is active at the current time step (previous time step). Now,

we can formulate the optimization problem with the core settings fixed as

$$\begin{aligned} &\text{minimize } \|\mathcal{Y} - Y_{m_l}\|_2 \\ &\text{subject to } \begin{cases} \text{card}(P_m) = n_{ma} + 1, \\ \text{card}(X_{cur} - X_{pre}) = 1, \\ Y_{m_l} \preceq \mathcal{Y}, \end{cases} \end{aligned} \tag{10}$$

where $P_m = \left[P_{m_1}^T, \cdots, P_{m_l}^T\right]^T$, n_{ma} is the active cache bank number. Please note that the term $L_{m_l} G^{-1} B_c P_c$ in (6) is constant now because we fix the core settings to determine the cache settings. Then, we can rewrite the optimization problem (10) as the following by plugging (6):

$$\begin{aligned} &\text{minimize } \|\mathcal{Y}_m - A_m P_m\|_2 \\ &\text{subject to } \begin{cases} \text{card}(P_m) = n_{ma} + 1, \\ \text{card}(X_{cur} - X_{pre}) = 1, \\ A_m P_m \preceq \mathcal{Y}_m, \end{cases} \end{aligned} \tag{11}$$

where $\mathcal{Y}_m = \mathcal{Y} - L_{m_l} G^{-1} B_c P_c \in \mathbb{R}^{n_m}$ is a known constant vector, $A_m = L_{m_l} G^{-1} \left[B_{m_1}, \cdots, B_{m_l}\right] \in \mathbb{R}^{n_m \times n_m l}$ is a known constant matrix.

To solve this optimization problem, we simply test all the previously inactive cache banks (there are $n_m l - n_{ma}$ inactive cache banks), and pick the optimal one.

For example, to test the jth cache bank (assume it was inactive previously), we make it active and compute the cost function explicitly as

$$\|\mathcal{Y}_m - A_{m_{pre}} P_{m_{pre}} - a_{m_j} p_{m_j}\|_2, \tag{12}$$

where $A_{m_{pre}} \in \mathbb{R}^{n_m \times n_{ma}}$ and $P_{m_{pre}} \in \mathbb{R}^{n_{ma}}$ are composed of columns of A_m and elements of P_m corresponding to the active cache banks in the previous time step, respectively. $a_{m_j} \in \mathbb{R}^{n_m}$ and $p_{m_j} \in \mathbb{R}$ are the jth column of A_m and jth element of P_m, respectively, which correspond to the jth cache bank under test.

Then, the previous inactive cache bank which leads to the smallest cost function (12) value is chosen as the one to activate.

Finally, we update the system performance with the chosen cache bank activated. This is first achieved by computing the power budget of the cores using equation (9) of the (n_{ca})th iteration (i.e., set $i = n_{ca} - 1$ in (9)) with L_s updated according to the new active cache bank distribution. Then, the system performance is updated by following the steps presented in Section 5.3. This system performance is recorded as TIPS_a.

5.4.2 Deactivate one cache bank

To formulate the optimization problem of deactivating a previously active cache bank, we slightly modify the optimization problem in (11) as

$$\begin{aligned} &\text{minimize } \|\mathcal{Y}_m - A_m P_m\|_2 \\ &\text{subject to } \begin{cases} \text{card}(P_m) = n_{ma} - 1, \\ \text{card}(X_{cur} - X_{pre}) = 1, \\ A_m P_m \preceq \mathcal{Y}_m. \end{cases} \end{aligned} \tag{13}$$

To solve this problem, we test all the previously active cache banks (there are n_{ma} active cache banks), and pick the optimal one.

For example, to test the jth cache bank (assume it was active previously), we make it inactive and compute the cost function explicitly as

$$\|\mathcal{Y}_m - A_{m_{pre}} P_{m_{pre}} + a_{m_j} p_{m_j}\|_2. \tag{14}$$

Then, the previous active cache bank which leads to the smallest cost function (14) is chosen as the one to deactivate.

Finally, we update the system performance with the chosen cache bank deactivated, using the same corresponding step presented previously in Section 5.4.1. This system performance is recorded as TIPS_d.

5.4.3 Pick the best option for cache bank settings

Now, we have the system performances for the three options: TIPS_a (activate one cache bank), TIPS_d (deactivate one cache bank), and TIPS_c (keep the previous cache bank settings). The decision can be readily made by picking the option with the largest system performance. The corresponding V/f levels will also be assigned to the active cores. This concludes the full performance optimization flow for one time step.

5.5 Full algorithm flow and time complexity analysis

To summarize the new performance optimization method for 3-D microprocessors, we provide the pseudo code of the new method for one performance optimization time step in algorithm 1. The algorithm can be executed in an active core of the 3-D system (or in a co-processor of the 3-D system if available) in a centralized way. Then, the computed results (core settings and cache settings) are sent to cores through the interconnection network on chip. The algorithm computing and the main workload can be handled by classical single core scheduling algorithms like priority scheduling and round-robin scheduling, so the interference can be well controlled.

Time complexity is important for the runtime performance optimization algorithm. For stage 1 (as shown in Fig. 4), there are totally n_{ca} iterations. In each iteration, testing all the remaining inactive cores requires approximately $n_{ca} \cdot n_m$ operations (including the computation of the residual $\mathcal{R}_c$) and computing the power budget needs around n_{ca}^3 operations. As a result, the time complexity for stage 1 is $O(n_{ca}^2 \cdot n_m + n_{ca}^4)$. Stage 2 has a low time complexity of $O(n_{ca})$, which can be ignored. For stage 3, trying to activate and deactivate one cache bank requires around $l \cdot n_m^2 + n_m \cdot n_{ma}$ operations in total.[2] Finally, we can summarize that the time complexity of the full algorithm is $O(n_{ca}^2 \cdot n_m + n_{ca}^4 + l \cdot n_m^2 + n_m \cdot n_{ma})$.

For a typical multi-core 3-D system ($n_{ca} \leq 16$, $n_m \leq 18$, $l \leq 2$, $n_{ma} \leq n_m$), the operation number is below 10^5, leading to a computing overhead which is smaller than $1\,\text{ms}$ as shown later in Table 1 and Table 2 in the experiments.

5.6 The application fairness problem and a simple solution

The target of this work is to maximize the overall throughput of the 3-D system, so the total instruction per

2. Please note that we only need to compute $A_{m_{pre}} P_{m_{pre}}$ once.

Algorithm 1 The greedy based core-cache co-optimization algorithm for 3-D microprocessors in dark silicon

1: Locate active cores with previous cache settings.
2: Compute the power budgets of the active cores.
3: **for** $i \leftarrow 1, n_{ca}$ **do** ▷ *Adjust the power budget of each core according to its workload.*
4: **if** $p_{app_i} < p_{c_i}$ **then**
5: $p_{c_i} \leftarrow p_{app_i}$
6: **end if**
7: Set V/f level according to p_{c_i}.
8: **end for**
9: Compute the total IPS of all cores as TIPS_c.
▷ *Compute performance if activate one more cache bank*
10: Find the optimal cache bank to activate.
11: Re-compute V/f levels (line 2 to 8).
12: Re-compute the total IPS as TIPS_a.
▷ *Compute performance if deactivate one more cache bank*
13: Find the optimal cache bank to deactivate.
14: Re-compute V/f levels (line 2 to 8).
15: Re-compute the total IPS as TIPS_d.
▷ *Choose the cache bank settings for the best performance*
16: **if** $\mathrm{TIPS}_d > \mathrm{TIPS}_c$ && $\mathrm{TIPS}_d > \mathrm{TIPS}_a$ **then**
17: Deactivate the optimal cache bank, $n_{ma} \leftarrow n_{ma} - 1$.
18: Set V/f levels as the ones in line 14.
19: **return**
20: **else if** $\mathrm{TIPS}_a > \mathrm{TIPS}_c$ && $\mathrm{TIPS}_a > \mathrm{TIPS}_d$ **then**
21: Activate the optimal cache bank, $n_{ma} \leftarrow n_{ma} + 1$.
22: Set V/f levels as the ones in line 11.
23: **return**
24: **else**
25: **return** ▷ *Keep n_{ma} and use original V/f levels in line 7*
26: **end if**

second (TIPS) is used as the optimization target. However, with this target, the cache settings may prefer the computing intensive applications to the memory intensive ones, because the IPS of the memory intensive applications is lower. This is called the application fairness problem, which has been researched for the 2-D systems in [42].

Discussing the application fairness problem thoroughly is beyond the scope of this work. But we still provide a simple solution here to ease this problem: just define the total IPS improvement ratio (TIIR) ($\mathrm{TIIR}_c = \frac{\mathrm{TIPS}_c - \mathrm{TIPS}_p}{\mathrm{TIPS}_p}$, $\mathrm{TIIR}_a = \frac{\mathrm{TIPS}_a - \mathrm{TIPS}_p}{\mathrm{TIPS}_p}$, $\mathrm{TIIR}_d = \frac{\mathrm{TIPS}_d - \mathrm{TIPS}_p}{\mathrm{TIPS}_p}$) to replace the TIPS ($\mathrm{TIPS}_c$, TIPS_a, TIPS_d) target in the new method, where TIPS_p is the TIPS of the previous step. Now, the computing intensive applications will not be overly preferred because they lost the advantage in absolute IPS.

6 Experimental Results

In this section, we present the experimental results to analyze the performance of the proposed runtime optimization method for 3-D microprocessors.

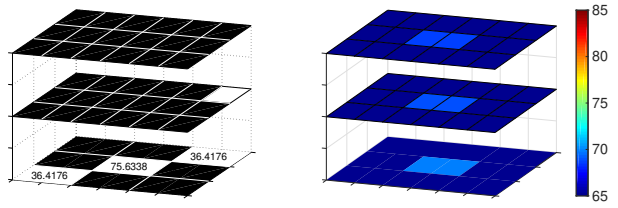
6.1 Experimental settings

The experiments are performed using a PC with Intel i7-6700HQ CPU and $8\,\mathrm{GB}$ memory.

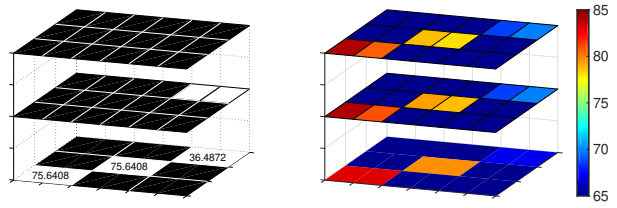
We built two 3-D microprocessors to test the performance of the new method. The first microprocessor (the 9-core system) has a structure as shown in Fig. 1. It has one core layer and two cache layers. The core layer has 9 Alpha cores and each cache layer has 18 cache banks, making a total of 36 cache banks in the system. The second microprocessor (the 16-core system) has the same basic structure as the 9-core system but with one core layer and one cache layer. The core layer has 16 Alpha cores and the cache layer has 16 cache banks, with one cache bank located on the top of each core. The cores are connected by crossbar switch interconnection and different die layers are connected by TSVs. The V/f level of each core ranges from ($800\,\mathrm{MHz}$, $0.4\,\mathrm{V}$) to ($2.0\,\mathrm{GHz}$, $1\,\mathrm{V}$). The size of the L1 cache is $32\,\mathrm{kB}$. The L2 cache association is 8 and the L2 cache bank size is $64\,\mathrm{kB}$ for the 9-core system and is $128\,\mathrm{kB}$ for the 16-core system. The capacity of memory is $16\,\mathrm{GB}$. The average delays of each memory access and each cache access are set as $100\,\mathrm{ns}$ and $8\,\mathrm{ns}$, respectively.

We use gem5 [43] as the performance estimator and McPAT [44] as the power estimator. The benchmarks used in experiments are from PARSEC v2.1 [45]. Since activating and deactivating the cache banks for the multithreading simulation in gem5 is difficult, we explore the multitasking (multiprogramming) parallelism in this experiment. To be specific, we run each application in single-thread and assign one application/thread to one active core. At the end of each performance optimization cycle, gem5 is stopped to re-map the cache and then resumed for the next optimization cycle. In this way, we can activate and deactivate cache banks without experiencing any cache coherence and shared-memory problems. Offline training is performed to obtain the cache miss rate with different cache size for the performance estimation in cache setting adjustment. The average error for cache miss rate estimation is 2.4% and the average error for performance estimation is 5%. The power of the interconnection is ignored in the experiment, because it is very small (less than $2\,\mathrm{mW}$ for each router including both dynamic power and leakage power) compared with the power of the active components as analyzed in [46], [47]. We assume the greedy based core-cache co-optimization algorithm is executed in an active core of the 3-D microprocessor. The performance, power, and thermal impact of the algorithm computing are counted in the measurements. We also take into account the delay of information gathering and dispatching through the interconnection network on chip, which is estimated as 100 cycles [46], [47].

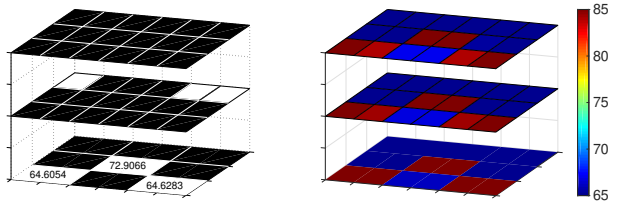
Deactivating a cache bank requires its dirty blocks to be written to the memory and activating a new cache bank requires the bank to wake up. We count the energy overhead of deactivating a cache bank as $50\,\mu\mathrm{J}$, according to the studies in [10], [11]. For activating a cache bank, the wake-up time is only several clock cycles, which is ignored, according to [11], [48]. There will be start up misses in a short period of time when the cache bank is newly activated. It is not a problem of the new method when the workload mode is stable (cache bank activation does not often happen immediately after a deactivation) since they are not additional misses. In case that the workload mode is unstable, drowsy cache can be used instead of power gating for deactivating a cache bank, as advised by many



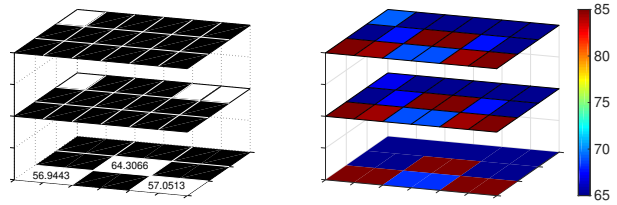
(a) Time step 1. The optimal active core positions are automatically determined by the new method. The new method also activates one active cache bank to increase system performance. Because the 3-D system has not reached threshold temperature yet, placing the active cache bank on top of an active core does not harm performance.



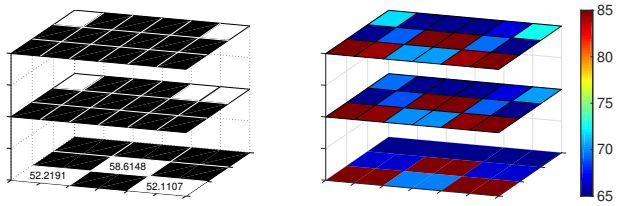
(b) Time step 2. The new method activates one more cache bank to increase system performance.



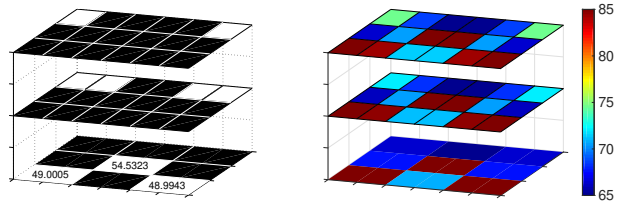
(c) Time step 3. The new method activates one more cache bank to increase system performance. The active core positions are automatically updated by the new method to minimize the thermal impact from the existing activated cache banks. The new cache bank is activated by avoiding any active core in the vertical direction, such that the total system performance is maximized.



(d) Time step 4. The new method activates one more cache bank to increase system performance. The cache bank with the smallest thermal impact to the active cores is activated, such that the total system performance is maximized.



(e) Time step 5. The new method activates one more cache bank at the optimal location to increase system performance.



(f) Time step 6. The new method activates one more cache bank at the optimal location to increase system performance.

Fig. 9: The active core and active cache bank locations of the 9-core system determined by the new method for the first 6 performance optimization time steps. Benchmark used for this test is *Swaptions* and the active core number is 3. In each subfigure, the left image shows the active component distribution (the active components are in white and the inactive components are in black) and the power budgets (shown as numbers in the active cores with unit W), the right image shows the temperature distribution with unit °C.

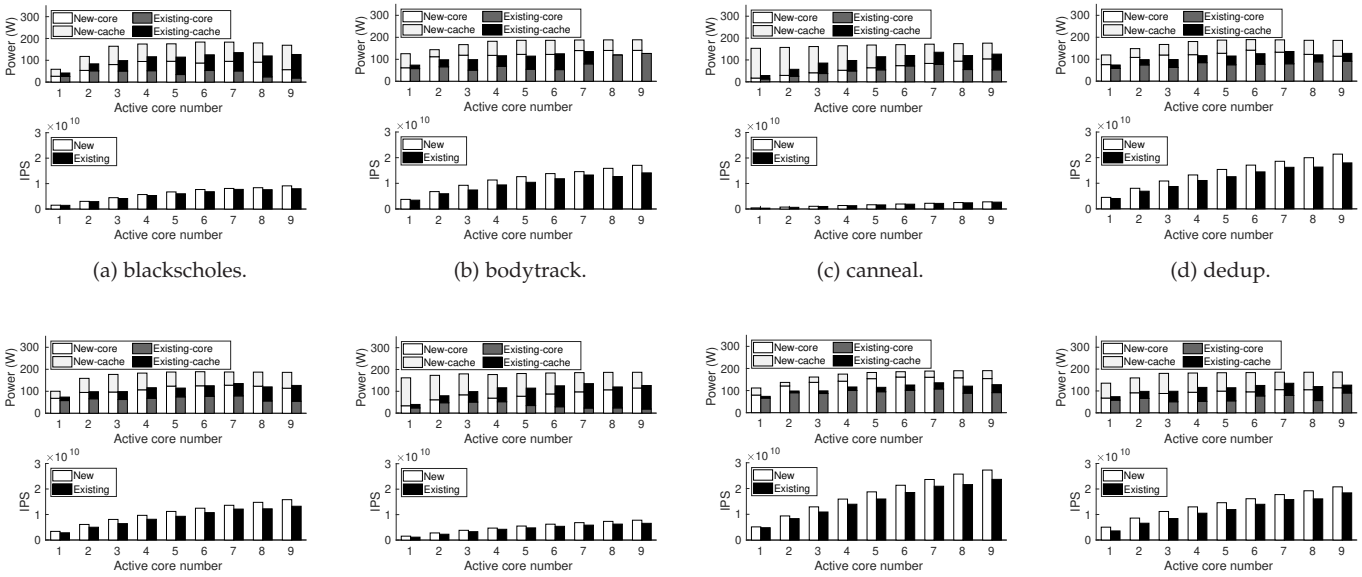
dynamic cache tuning studies [49]. It avoids the start up miss overhead by introducing small energy overhead [48].

The thermal models of the 3-D microprocessors are extracted from HotSpot [50], where each core/cache layer has the dimension of $12\,\mathrm{mm} \times 12\,\mathrm{mm} \times 0.15\,\mathrm{mm}$ for the die and $12\,\mathrm{mm} \times 12\,\mathrm{mm} \times 0.02\,\mathrm{mm}$ for the TIM. The convection resistance of heatsink is set as $0.2\,\mathrm{K\,W^{-1}}$ and the convection capacitance of heatsink is $14.4\,\mathrm{J\,K^{-1}}$. For all test cases, we set the ambient temperature as $25\,^{\circ}\mathrm{C}$ and the temperature constraint as $85\,^{\circ}\mathrm{C}$. The performance optimization time step is set as $1\,\mathrm{s}$.

In order to show the advantage of the new method, we compare it with the state-of-the-art performance optimization method for 3-D dark silicon system proposed in [36] (we call it the existing method), which uses a heuristic method to find the power of each core and the corresponding cache bank number. In the original work of the existing method [36], thermal design power (TDP) is used as the power budget constraint for the performance optimization. However, TDP has been proved to be overly conservative for the dark silicon system [14]. In order to fully release the potential of the existing method for a better comparison, we use thermal safe power (TSP) [14], which is the power budget specially designed for the dark silicon system, as the power budget of the existing method.

6.2 Runtime behavior analysis of the 3-D microprocessor with the new performance optimization method

First, we analyze the runtime behavior of the 3-D microprocessor with the new performance optimization method. We use the 9-core system with 3 active cores running the *Swaptions* benchmark for this analysis. For initiation, the temperature of the whole packaged system is set as the ambient temperature, and the active cache bank number is zero.



(a) blackscholes. (b) bodytrack. (c) canneal. (d) dedup.

(e) ferret. (f) freqmine. (g) swaptions. (h) x264.

Fig. 10: The performance comparison results between the new method and the existing method on the 9-core system with different benchmarks.

The runtime behavior of the 3-D microprocessor for the first 6 optimization time steps with the new method is demonstrated in Fig. 9. From each subfigure in Fig. 9, we can see the active core and cache bank positions, the power budgets of the active cores, and the temperature distributions of the three die layers, for each time step.

Fig. 9a shows the status of the 3-D microprocessor at the first time step. We see that the three active cores are located optimally by the new method, without forming active core clusters. One cache bank is activated by the new method, because it will increase the total performance of the 3-D microprocessor. We notice that the cache bank is activated directly on the top of an active core (C9 in Fig. 1). It looks like a mistake by the first instinct but it is actually correct as explained in the following. If we pay attention to the temperature distribution in Fig. 9a, we see the highest temperature of the die is much lower than the temperature threshold because the whole package was at ambient temperature just one time step ago for initiation. In this condition, activating the cache bank on the top of the active core does not degrade the total system performance because the power budget of the active core has not been limited by the temperature constraint yet.

In time step 2, as shown in Fig. 9b, the active cores are kept the same as the previous time step. The new method activates one more cache bank because this will increase the total system performance. The newly activated cache bank is on the top of an active core (C9 in Fig. 1), due to the same reason presented previously in time step 1.

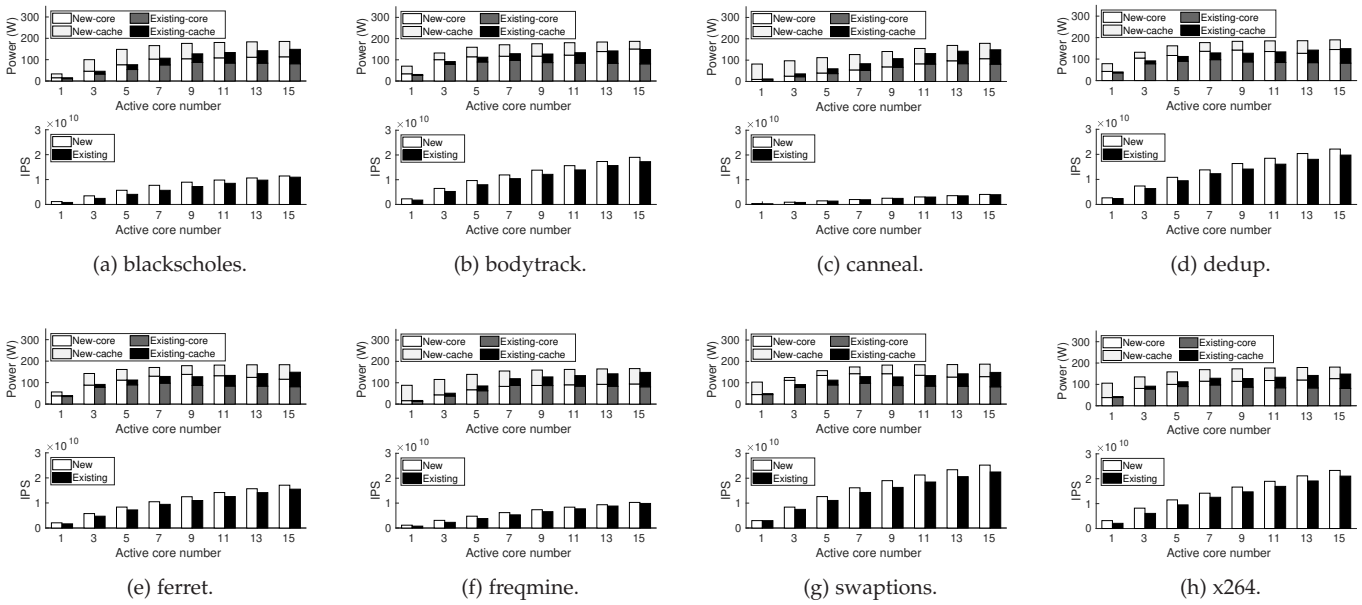
In time step 3, as shown in Fig. 9c, the active core distribution is re-mapped by the new method to avoid overlapping with the active caches. This is because the whole packaged system is heated up now, the power budgets of the active cores start to be constrained by the temperature threshold. By avoiding the active caches vertically, the active cores are able to gain more power budgets with the same temperature threshold. The new method activates one more cache bank in this step to further improve the system performance. Cache bank M13 is correctly chosen by the new method as the new active cache bank because it has minimum thermal impact on the active cores. With the power budgets computed by the new method, the highest temperatures (which appear at the cache banks in cache layer 2 above the active cores) just reach the temperature threshold, meaning the full performance potential of the 3-D microprocessor is released without violating the thermal constraint.

From time step 4 to time step 6, as shown from Fig. 9d to Fig. 9f, more cache banks are activated by the new method, because system performance can still be improved with larger cache size. These new active cache banks are all activated at the positions with minimum thermal impact on the active cores, leading to high power budgets of the active cores and high system performance.

Although activating more cache banks can increase the hit rate to benefit the system performance, it will also decrease the frequencies of the active cores which harms the performance, because the increasing power of the active cache banks will limit the power budgets of the active cores due to the thermal constraint. After time step 6, the balance is reached, and the new method judges that activating more cache banks will not improve the system performance further.

6.3 Comparison against the existing method

Now we compare the new method with the state-of-the-art performance optimization method for 3-D microprocessor in dark silicon proposed recently in [36] (called the existing



(a) blackscholes. (b) bodytrack. (c) canneal. (d) dedup.

(e) ferret. (f) freqmine. (g) swaptions. (h) x264.

Fig. 11: The performance comparison results between the new method and the existing method on the 16-core system with different benchmarks.

method). Both methods are applied to the same 3-D microprocessors with the same experimental settings as shown previously in Section 6.1.

We will first compare the performances of the 3-D microprocessors optimized by the new method and the existing method. Then, we will look into the transient temperatures of the 3-D microprocessors managed by the two methods, which provide explanations and experimental supports to the performance comparison results. Lastly, the computing overheads of the two methods are compared and analyzed.

6.3.1 System performance comparison and analysis

In this comparison, we run multiple tests with each PARSEC benchmark by changing the active core number from 1 to n_c, in order to consider all dark silicon conditions. For each test, the PARSEC benchmark is applied to all active cores, then we collect the average instruction per second (IPS) of the 3-D microprocessor as the performance measurement. In order to analyze the performance results, we also record the power consumptions of the cores and caches, for each test.

The system performance comparison results on the 9-core microprocessor and the 16-core microprocessor are shown in Fig. 10 and Fig. 11, respectively, for different benchmarks and different active core numbers. For all test cases, the 3-D microprocessor managed by the new method has a higher system performance than the one optimized by the existing method.

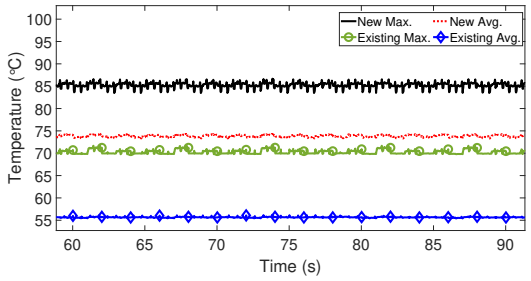
The new method leads to a better system performance mainly because it brings a higher power budget than the existing method, as revealed by the power consumption comparison in Fig. 10 and Fig. 11, thanks to two good properties of the new method. First, the new method is able to perform the joint optimization of the active core and active cache bank distributions. Whereas the existing method only optimizes the active core distribution in a heuristic way. With the joint optimization, the true power budget potential of the 3-D system is released with the same thermal constraint. Second, the new method is able to compute the power budget at runtime according to the current active component distribution. In contrast, the existing method relies on the static power budget computed offline. Because such static power budget cannot be updated at runtime, it is computed in a pessimistic way so that the absolute thermal safety is guaranteed for any situation [13], [14]. However, such power budget is too low for most of the running conditions, which greatly limits the performance of the 3-D microprocessor.

6.3.2 Transient temperature comparison and analysis

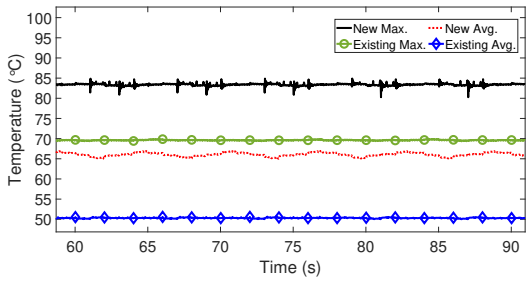
The temperature behavior of the 3-D microprocessor is also important in order to justify both the performance optimization ability and the system reliability consideration of a performance optimization method. To be specific, with a good method, the temperature of the 3-D microprocessor shall satisfy the following two conditions at the same time. First, the average temperature of the 3-D microprocessor should be high. This means the power budget provided by the optimization method is sufficient to achieve high system performance. Second, the highest temperature of the 3-D microprocessor should always be lower than the temperature threshold. This indicates the reliability of the 3-D microprocessor is ensured when the system performance is being optimized.

Due to the reasons above, we have performed a transient temperature comparison to analyze the differences between the new method and the existing method. In this comparison, the same set of benchmarks is used for both methods.

We plot the transient temperature comparison results with the 9-core microprocessor and the 16-core microprocessor in Fig. 12. We see that the average temperature of the



(a) Transient temperature comparison on the 9-core system.



(b) Transient temperature comparison on the 16-core system.

Fig. 12: Transient temperature comparison between the new method and the existing method [36]. The temperature threshold is set to be 85 °C for both methods.

processor with the new method is higher than that with the existing method. It means the power budget dynamically optimized by the new method is much higher than the static power budget used by the existing method, which further leads to a higher system performance as revealed previously in Section 6.3.1.

The highest temperature traces of the 3-D microprocessor managed by the new method and the existing method are also plotted in Fig. 12. The highest temperature appears at cache layer 2, because it is the innermost layer in the heat dissipation path. We can see that the highest temperatures of the 3-D microprocessor with both methods are always below the temperature threshold. It means both methods keep the 3-D microprocessor in a safe state during the performance optimization process. We also notice that the highest temperature with the new method just reaches the temperature threshold without violating it. It indicates the power budgets computed by the new method are not only conservative to guarantee the system safety, but also accurate to release the system performance potential.

We also plot the power and temperature snapshots of the 9-core system and the 16-core system at 70 s in Fig. 13 and Fig. 14, respectively. From the figures, we see that the active component distribution of the 3-D microprocessor optimized by the new method is more uniform in the 3-D space. The power budget provided by the new method is also higher than the existing method, thanks to the optimized active component distribution and the dynamic power budget computing. The temperatures of all 3-D microprocessors are below the temperature threshold (85 °C), but the temperature distributions with the new method are more uniform and the average temperatures with the new method are also higher. This confirms again that the new method releases the system performance potential with the system thermal safety ensured.

TABLE 1: The computing overhead comparison on the 9-core system.

Active core #		1	2	3	4	5	6	7	8	9
Time (ms)	New	0.28	0.37	0.45	0.51	0.57	0.67	0.73	0.82	0.90
	Existing	0.27	0.26	0.19	0.28	0.28	0.21	0.27	0.25	0.16

TABLE 2: The computing overhead comparison on the 16-core system.

Active core #		1	3	5	7	9	11	13	15
Time (ms)	New	0.21	0.31	0.42	0.56	0.68	0.81	0.95	1.1
	Existing	0.66	0.86	0.77	0.85	0.88	0.78	0.65	0.57

6.3.3 *Computing overhead comparison and analysis*

Finally, we compare and analyze the computing overhead of the new method and the existing method.

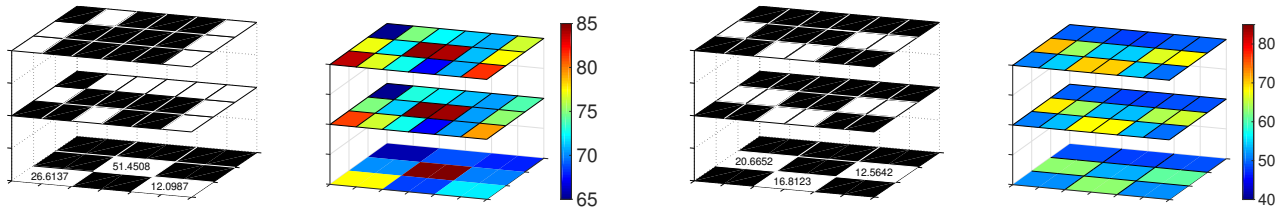
The computing overheads of the two methods for the 9-core system and the 16-core system are collected in Table 1 and Table 2, respectively. The overhead is measured as the computing time for each performance optimization time step (1 s in this experiment).

From the tables, we can see that both methods have small computing overhead, with around 1 ms. Thanks to the greedy based algorithm, the new method manages to compute the power budget dynamically at a fast speed. The computing overhead of the new method is slightly higher, mainly because it computes power budget dynamically at runtime, whereas the existing method uses the static power budget computed off-line. However, with the slightly higher computing overhead, the new method provides a higher power budget than the static power budget used in the existing method. Such higher power budget leads to a significantly higher overall system throughput, which overweighs the slight disadvantage in overhead. It is also noted that the overhead of the new method increases with the active core number, because more iterations are needed to locate more active cores.

6.4 Analysis of using TIIR for application fairness

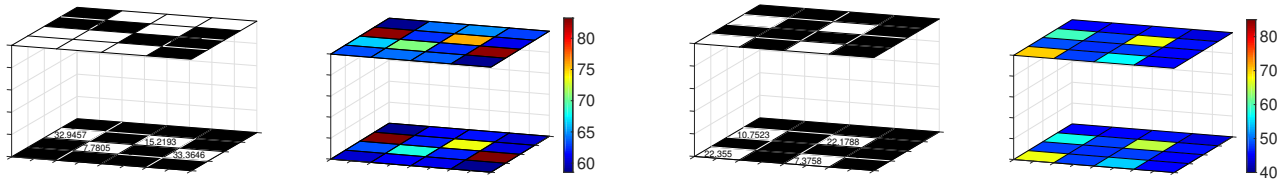
As presented in Section 5.6, the new method using TIPS as the optimization target may prefer the computing intensive applications to the memory intensive applications, so using TIIR as the target is provided as a simple solution.

In order to see the effectiveness of using TIIR, we perform the new method on the 9-core system with 6 active cores running 4 computing intensive applications (all are "swaptions") and 2 memory intensive applications (both are "canneal"). The throughput comparison results of using TIPS and TIIR as the optimization targets are shown in Fig. 15. By using the TIIR target, although the overall throughput is lower, the memory intensive application ("canneal") gains a higher performance since the IPS prejudice is alleviated, compared with using the TIPS target.



(a) The 3-D microprocessor with the new method. (b) The 3-D microprocessor with the existing method.

Fig. 13: The power and temperature snapshots of the 9-core system at 70 s of Fig. 12a. In each subfigure, the left image shows the active component distribution (the active components are in white and the inactive components are in black) and the power budgets (shown as numbers in the active cores with unit W), the right image shows the temperature distribution with unit °C.



(a) The 3-D microprocessor with the new method. (b) The 3-D microprocessor with the existing method.

Fig. 14: The power and temperature snapshots of the 16-core system at 70 s of Fig. 12b. In each subfigure, the left image shows the active component distribution (the active components are in white and the inactive components are in black) and the power budgets (shown as numbers in the active cores with unit W), the right image shows the temperature distribution with unit °C.

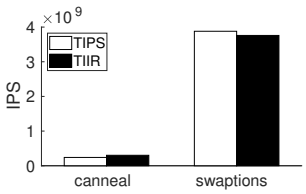


Fig. 15: The throughput comparison of using TIPS and TIIR as the optimization targets on the 9-core system with 6 active cores running 4 "swaptions" and 2 "canneal" applications. The IPS is measured as the average IPS of one core running the specific application.

7 Limitations and Future Work

Although the new method can be applied to the widely adopted 3-D system composed of one core layer and multiple cache layers, it is not compatible with the 3-D system with cores and caches mixed in the same layer. We plan to develop the performance optimization algorithm for the latter 3-D system as our future work.

Moreover, the new method in current form cannot be applied to the many-core 3-D microprocessors, mainly because of the following two problems. First, the computing overhead grows with the core and cache numbers as shown in the time complexity analysis, making the overhead unacceptable for the many-core 3-D systems. Second, the new method assumes all cache banks can be accessed with the same latency for each active core and gem5 is directly used for architectural modeling, because the memory controllers are connected with the cores using crossbar switch in the core layer. However, the many-core systems are usually Non-Uniform Cache Access (NUCA) based, meaning the cache access latency depends on the locations of the core and cache bank [34], [35] which are connected by a 3-D NoC. As a result, a future research direction is to develop a distributed runtime performance optimization algorithm with NUCA consideration and new architectural model for the many-core 3-D systems.

8 Conclusion

In this article, we have presented a greedy based core-cache co-optimization algorithm to optimize the performance of 3-D microprocessors in dark silicon at runtime. With a greedy based joint optimization scheme, the new method determines a sub-optimal distribution of active cores and active cache banks in the three-dimensional space across different die layers. The active cache bank number is adjusted dynamically by the new method to improve the overall system performance. The V/f levels of the active cores are also tuned by the new method according to the power budgets computed at runtime under the optimized active component distributions. The experiments show that the new greedy based core-cache co-optimization algorithm outperforms the state-of-the-art performance optimization method for 3-D microprocessors in dark silicon with a higher system throughput and guaranteed system thermal safety.

Acknowledgments

This research is supported in part by National Natural Science Foundation of China under grant No. 61974018, in

part by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry, in part by the Department of Science and Technology of Sichuan Province under grant No. 2018GZDZX0002.

REFERENCES

[1] K. Banerjee, S. J. Souri, P. Kapur, and K. C. Saraswat, "3-D ICs: a novel chip design for improving deep-submicrometer interconnect performance and systems-on-chip integration," *Proc. of the IEEE*, vol. 89, no. 5, pp. 602–633, May 2001.

[2] R. Weerasekera, L.-R. Zheng, D. Pamunuwa, and H. Tenhunen, "Extending systems-on-chip to the third dimension: performance, cost and technological tradeoffs," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, November 2007, pp. 212–219.

[3] A. Sridhar, A. Vincenzi, D. Atienza, and T. Brunschwiler, "3D-ICE: A compact thermal model for early-stage design of liquid-cooled ICs," *IEEE Trans. on Computers*, vol. 63, no. 10, pp. 2576–2589, October 2014.

[4] B. W. Ku, K. Chang, and S. K. Lim, "Compact-2D: A physical design methodology to build two-tier gate-level 3D ICs," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 6, pp. 1151–1164, June 2020.

[5] M. Alfano, B. Black, J. Rearick, J. Siegel, M. Su, and J. Din, "Unleashing fury: A new paradigm for 3-D design and test," *IEEE Design & Test*, vol. 34, no. 1, pp. 8–15, January/February 2017.

[6] D. H. Kim, K. Athikulwongse, M. B. Healy, M. M. Hossain, M. Jung, I. Khorosh, G. Kumar, Y.-J. Lee, D. L. Lewis, T.-W. Lin, C. Liu, S. Panth, M. Pathak, M. Ren, G. Shen, T. Song, D. H. Woo, X. Zhao, J. Kim, H. Choi, G. H. Loh, H.-H. S. Lee, and S. K. Lim, "Design and analysis of 3D-MAPS (3D massively parallel processor with stacked memory)," *IEEE Trans. on Computers*, vol. 64, no. 1, pp. 112–125, January 2015.

[7] C. C. Liu, I. Ganusov, M. Burtscher, and S. Tiwari, "Bridging the processor-memory performance gap with 3D IC technology," *IEEE Design & Test of Computers*, vol. 22, no. 6, pp. 556–564, 2005.

[8] G. H. Loh, "3D-stacked memory architectures for multi-core processors," in *Proc. Int. Symp. on Computer Architecture (ISCA)*, July 2008, pp. 453–464.

[9] K. Cao, J. Zhou, T. Wei, M. Chen, S. Hu, and K. Li, "A survey of optimization techniques for thermal-aware 3D processors," *Journal of System Architecture*, vol. 97, pp. 397–415, January 2019.

[10] S. Lee, K. Kang, J. Jung, and C.-M. Kyung, "Runtime 3-D stacked cache data management for energy minimization of 3-D chip-multiprocessors," in *Proc. Int. Symp. on Quality Electronic Design (ISQED)*, March 2014.

[11] S. Lee, K. Kang, and C.-M. Kyung, "Runtime thermal management for 3-D chip-multiprocessors with hybrid SRAM/MRAM L2 cache," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 3, pp. 520–533, March 2015.

[12] M. Taylor, "A landscape of the new dark silicon design regime," *IEEE MICRO*, vol. 33, no. 5, pp. 8–19, October 2013.

[13] H. Wang, D. Tang, M. Zhang, S. X.-D. Tan, C. Zhang, H. Tang, and Y. Yuan, "GDP: A greedy based dynamic power budgeting method for multi/many-core systems in dark silicon," *IEEE Trans. on Computers*, vol. 68, no. 4, pp. 526–541, April 2019.

[14] S. Pagani, H. Khdr, J.-J. Chen, M. Shafique, M. Li, and J. Henkel, "Thermal safe power (TSP): Efficient power budgeting for heterogeneous manycore systems in dark silicon," *IEEE Trans. on Computers*, vol. 66, no. 1, pp. 147–162, January 2017.

[15] H. Esmaeilzadeh, E. Blem, R. S. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," *IEEE MICRO*, vol. 32, no. 3, pp. 122–134, May 2012.

[16] A. K. Coskun, J. L. Ayala, D. Atienza, T. S. Rosing, and Y. Leblebici, "Dynamic thermal management in 3D multicore architectures," in *Proc. Design, Automation and Test in Europe Conf. (DATE)*, April 2009, pp. 1410–1415.

[17] H. Wang, D. Huang, R. Liu, C. Zhang, H. Tang, and Y. Yuan, "STREAM: Stress and thermal aware reliability management for 3-D ICs," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 11, pp. 2058–2071, November 2019.

[18] N. Hardavellas, "The rise and fall of dark silicon," *USENIX*, vol. 37, no. 2, pp. 7–17, April 2012.

[19] T. S. Muthukaruppan, M. Pricopi, V. Venkataramani, T. Mitra, and S. Vishin, "Hierarchical power management for asymmetric multi-core in dark silicon era," in *Proc. Design Automation Conf. (DAC)*, May 2013.

[20] H. Khdr, S. Pagani, M. Shafique, and J. Henkel, "Thermal constrained resource management for mixed ILP-TLP workloads in dark silicon chips," in *Proc. Design Automation Conf. (DAC)*, 2015.

[21] A. Kanduri, M.-H. Haghbayan, A. M. Rahmani, M. Shafique, A. Jantsch, and P. Liljeberg, "adBoost: Thermal aware performance boosting through dark silicon patterning," *IEEE Trans. on Computers*, vol. 67, no. 8, pp. 1062–1077, August 2018.

[22] K. Kang, J. Kim, S. Yoo, and C.-M. Kyung, "Runtime power management of 3-D multi-core architectures under peak power and temperature constraints," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 6, pp. 905–918, 2011.

[23] J. Meng, K. Kawakami, and A. K. Coskun, "Optimizing energy efficiency of 3-D multicore systems with stacked DRAM under power and thermal constraints," in *Proc. Design Automation Conf. (DAC)*, June 2012, pp. 648–655.

[24] F. Hameed, M. A. A. Faruque, and J. Henkel, "Dynamic thermal management in 3D multi-core architecture through run-time adaptation," in *Proc. Design, Automation and Test in Europe Conf. (DATE)*, March 2011.

[25] T.-H. Tsai and Y.-S. Chen, "Thermal-aware real-time task scheduling for three-dimensional multicore chip," in *Proc. ACM Symp. on Applied Computing (SAC)*, March 2012, pp. 1618–1624.

[26] W.-K. Cheng and T.-W. Hsu, "Thermal-aware task allocation, memory mapping, and task scheduling for 3D stacked memory and processor architecture," in *IEEE 2013 Tencon-Spring*. IEEE, 2013, pp. 95–98.

[27] D. Zhao, H. Homayoun, and A. V. Veidenbaum, "Temperature aware thread migration in 3D architecture with stacked DRAM," in *Proc. Int. Symp. on Quality Electronic Design (ISQED)*, March 2013.

[28] S. Aljeddani and F. Mohammadi, "A novel migration technique to balance thermal distribution for future heterogeneous 3D chip multiprocessors," in *Proc. Int. Conf. on Information Science and Technology (ICIST)*, June 2018.

[29] S. Lee, K. Kang, J. Jung, and C.-M. Kyung, "Hybrid L2 NUCA design and management considering data access latency, energy efficiency, and storage lifetime," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 10, pp. 3118–3131, March 2016.

[30] Q. Zou, E. Kursun, and Y. Xie, "Thermomechanical stress-aware management for 3-D IC designs," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 9, pp. 2678–2682, 2017.

[31] H. Wang, T. Xiao, D. Huang, L. Zhang, C. Zhang, H. Tang, and Y. Yuan, "Runtime stress estimation for three-dimensional IC reliability management using artificial neural network," *ACM Trans. on Design Automation of Electronic Systems*, vol. 24, no. 6, pp. 69:1–69:29, November 2019.

[32] E. Rotem, A. Naveh, D. Rajwan, A. Ananthakrishnan, and E. Weissmann, "Power-management architecture of the Intel microarchitecture code-named Sandy Bridge," *IEEE MICRO*, vol. 32, no. 2, pp. 20–27, March–April 2012.

[33] M. Shafique, D. Gnad, S. Garg, and J. Henkel, "Variability-aware dark silicon management in on-chip many-core systems," in *Proc. Design, Automation and Test in Europe Conf. (DATE)*, March 2015, pp. 387–392.

[34] M. Rapp, M. Sagi, A. Pathania, A. Herkersdorf, and J. Henkel, "Power- and cache-aware task mapping with dynamic power budgeting for many-cores," *IEEE Trans. on Computers*, vol. 69, no. 1, pp. 1–13, January 2020.

[35] M. Rapp, A. Pathania, T. Mitra, and J. Henkel, "Prediction-based task migration on S-NUCA many-cores," in *Proc. Design, Automation and Test in Europe Conf. (DATE)*, March 2019.

[36] A. Asad, O. Ozturk, M. Fathy, and M. R. Jahed-Motlagh, "Optimization-based power and thermal management for dark silicon aware 3D chip multiprocessors using heterogeneous cache hierarchy," *Microprocessors and Microsystems*, vol. 51, pp. 76–98, June 2017.

[37] H. Wang, J. Wan, S. X.-D. Tan, C. Zhang, H. Tang, Y. Yuan, K. Huang, and Z. Zhang, "A fast leakage-aware full-chip transient thermal estimation method," *IEEE Trans. on Computers*, vol. 67, no. 5, pp. 617–630, May 2018.

[38] H. Wang, X. Guo, S. X.-D. Tan, C. Zhang, H. Tang, and Y. Yuan, "Leakage-aware predictive thermal management for multi-core systems using echo state network," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 7, pp. 1400–1413, July 2020.

[39] H. Wang, L. Hu, X. Guo, Y. Nie, and H. Tang, "Compact piecewise linear model based temperature control of multi-core systems considering leakage power," *IEEE Trans. on Industrial Informatics*, 2020.
[40] Y. Zhong, S. G. Dropsho, X. Shen, A. Studer, and C. Ding, "Miss rate prediction across program inputs and cache configurations," *IEEE Trans. on Computers*, vol. 56, no. 3, pp. 328–343, March 2007.
[41] H. Wang, S. X.-D. Tan, D. Li, A. Gupta, and Y. Yuan, "Composable thermal modeling and simulation for architecture-level thermal designs of multi-core microprocessors," *ACM Trans. on Design Automation of Electronic Systems*, vol. 18, no. 2, pp. 28:1–28:27, March 2013.
[42] X. Wang, K. Ma, and Y. Wang, "Cache latency control for application fairness or differentiation in power-constrained chip multiprocessors," *IEEE Trans. on Computers*, vol. 61, no. 10, pp. 1371–1385, October 2012.
[43] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, and D. A. Wood, "The gem5 simulator," *ACM SIGARCH Comput. Archit. News*, vol. 39, no. 2, pp. 1–7, May 2011.
[44] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi, "The McPAT framework for multicore and manycore architectures: Simultaneously modeling power, area, and timing," *ACM Trans. on Architecture and Code Optimization*, vol. 10, no. 1, pp. 5:1–5:29, April 2013.
[45] C. Bienia, "Benchmarking modern multiprocessors," Ph.D. dissertation, Princeton University, January 2011.
[46] L. Huang, J. Wang, M. Ebrahimi, M. Daneshtalab, X. Zhang, G. Li, and A. Jantsch, "Non-blocking testing for network-on-chip," *IEEE Trans. on Computers*, vol. 65, no. 3, pp. 679–692, March 2016.
[47] L. Wang, S. Jiang, S. Chen, J. Wang, and L. Huang, "Optimized mapping algorithm to extend lifetime of both NoC and cores in many-core system," *Integration, the VLSI Journal*, vol. 67, pp. 82–94, July 2019.
[48] H. Homayoun, M. Makhzan, and A. Veidenbaum, "Multiple sleep mode leakage control for cache peripheral circuits in embedded processors," in *Proc. Intl. Conf. on Compilers, architectures and synthesis for embedded systems (CASES)*, October 2008, pp. 197–206.
[49] W. Zang and A. Gordon-Ross, "A survey on cache tuning from a power/energy perspective," *ACM Computing Surveys*, vol. 45, no. 3, pp. 32:1–32:49, June 2013.
[50] W. Huang, K. Sankaranarayanan, K. Skadron, R. J. Ribando, and M. R. Stan, "Accurate, pre-RTL temperature-aware processor design using a parameterized, geometric thermal model," *IEEE Trans. on Computers*, vol. 57, no. 9, pp. 1277–1288, 2008.

Hai Wang received the B.S. degree from the Huazhong University of Science and Technology, Wuhan, China, in 2007, and the M.S. and Ph.D. degrees from the University of California at Riverside, Riverside, CA, USA, in 2008 and 2012, respectively.

He is currently an associate professor with the University of Electronic Science and Technology of China, Chengdu, China. His research interests include modeling, optimization, and artificial intelligence assisted design automation of VLSI circuits and systems.

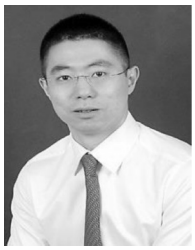
Dr. Wang was a recipient of the Best Paper Award nomination from ASP-DAC in 2019. He has served as a Technical Program Committee Member of several international conferences, including DATE, ASP-DAC and ISQED, and also served as a Reviewer of many journals including the IEEE Transactions on Computers, the IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, the IEEE Transactions on Parallel and Distributed Systems, and ACM Transactions on Design Automation of Electronic Systems.

Wei Li received the B.S. and M.S. degrees from the University of Electronic Science and Technology of China, Chengdu, China, in 2017 and 2020, respectively. His main research interests are computer architectures, especially the ARM and RISC-V architectures.

Wenjie Qi received the bachelor's degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2018. Currently, he is working toward the master's degree at UESTC. His current research interests include thermal analysis, power analysis, and thermal management of integrated circuit.

Diya Tang received the B.S. and M.S. degrees from the University of Electronic Science and Technology of China, Chengdu, China, in 2017 and 2020, respectively. Her main research directions are mobile operating system and machine learning algorithms.

Letian Huang received the M.S. and Ph.D. degrees in communication and information system from the University of Electronic Science and Technology of China (UESTC), Chengdu, China in 2009 and 2016, respectively. He is an associate professor with UESTC and IEEE CEDA Chengdu Chapter Chair. His scientific work contains more than 40 publications including book chapters, journal articles and conference papers. His research interests include heterogeneous multi-core system-on-chips, network-on-chips, and mixed signal IC design.



He Tang received the BSEE degree from the University of Electronic Science and Technology of China, Chengdu, China, the MS degree in electrical and computer engineering from the Illinois Institute of Technology, Chicago, and the PhD degree in electrical engineering from University of California, Riverside, in 2005, 2007, and 2010. From 2010 to 2012, he was with OmniVision Technologies Inc., in Santa Clara, California, as an Analog IC Designer, where he worked on high-speed I/O interface. Since 2012, he has been an associate professor and subsequently a professor with the University of Electronic Science and Technology of China, Chengdu, China. He has authored or coauthored more than 40 papers. His research interests focus on data converters and analog/mixed-signal IC designs. His past work includes high-speed high-resolution pipelined ADCs with digital calibration and high-performance ultra-low-power SAR ADCs. He has served on IEEE CAS Analog Signal Processing Technical Committee (ASPTC) since 2013. He is a member of the IEEE.