Received: 29 June 2017 | Revised: 4 April 2018 | Accepted: 11 June 2018

DOI: 10.4218/etrij.2017-0095

WILEY ETRI Journal

ORIGINAL ARTICLE

# Low-power heterogeneous uncore architecture for future 3D chip-multiprocessors

**Aniseh Dorostkar**[1] | **Arghavan Asad**[1] | **Mahmood Fathy**[1] | **Mohammad Reza Jahed-Motlagh**[1] | **Farah Mohammadi**[2,3]

[1]Computer Engineering Department, Iran University of Science and Technology, Tehran, Iran.

[2]Electrical and Computer Engineering Department, Ryerson University, Toronto, Canada.

[3]BroadPack Corp. San Jose, CA, USA.

**Correspondence**
Mahmood Fathy, Computer Engineering Department, Iran University of Science and Technology, Tehran, Iran.
Email: mahfathy@iust.ac.ir

Uncore components such as on-chip memory systems and on-chip interconnects consume a large amount of energy in emerging embedded applications. Few studies have focused on next-generation analytical models for future chip-multiprocessors (CMPs) that simultaneously consider the impacts of the power consumption of core and uncore components. In this paper, we propose a convex-optimization approach to design heterogeneous uncore architectures for embedded CMPs. Our convex approach optimizes the number and placement of memory banks with different technologies on the memory layer. In parallel with hybrid memory architecting, optimizing the number and placement of through silicon vias as a viable solution in building three-dimensional (3D) CMPs is another important target of the proposed approach. Experimental results show that the proposed method outperforms 3D CMP designs with hybrid and traditional memory architectures in terms of both energy delay products (EDPs) and performance parameters. The proposed method improves the EDPs by an average of about 43% compared with SRAM design. In addition, it improves the throughput by about 7% compared with dynamic RAM (DRAM) design.

**KEYWORDS**
embedded chip-multiprocessor, heterogeneous memory system, nonvolatile memory, optimal placement, through silicon via, uncore

## 1 | INTRODUCTION

Chip-multiprocessor (CMP) architectures have been extensively adopted to meet the ever-increasing demands for performance in embedded systems. The increased number of cores in an embedded CMP (eCMP) comes with an increase in power consumption. In this context, power consumption is a primary concern in embedded systems because many of them are generally limited by the battery lifetime. In addition, high power consumption results in a temperature increase that negatively affects the chip reliability [1].

While technological advances are gradually moving toward the nanometer scale, complementary metal-oxide semiconductor (CMOS) very large-scale integration (VLSI) circuits have been faced with serious design challenges such as static power consumption and sensitivity. A study has shown that over 75% of the overall power dissipation in 32-nm generation is due to the static power [2], and this percentage is expected to increase in subsequent generations [2–4]. It should be noted that static power is unavoidable in modern nanoscale CMOS designs based on recently developed technologies such as Fin field-effect transistors (FinFETs) and fully depleted silicon on insulator (FDSOI) technologies [5,6]. At 22 nm and beyond, FinFETs encounter a tradeoff between delay and static power consumption [5]. In this context, [7] presents techniques to reduce static power consumption in FinFET and FDSOI structures under performance constraints.

One of the newest challenges in CMP design is the management of dark silicon [8–11]. The rise of utilization walls owing to thermal and power budgets limits the number of active components and results in a large region of dark silicon. Research shows that the increasing leakage power consumption is a major driver of the unusable portion or dark silicon in future many-core CMPs [8]. Uncore components such as on-chip memory systems and on-chip interconnection consume a significant proportion of the power. In this context, the power management of these components is important to maximizing the design performance in the dark silicon era [8].

While most of the previous studies on multicore processors have focused on the design of on-chip interconnection networks [12–19] and memory architectures [20–28] separately, in this study, to achieve power efficiency, we explore the role of uncore components on CMP performance and power behavior.

The use of traditional memory technologies such as static random-access memory (SRAM) or dynamic RAM (DRAM) cells as on-chip cache/memory systems results in several weaknesses. For instance, SRAM is a low-density technology that dissipates high leakage power [29,30]; DRAM requires refresh operations to preserve its data integrity. As the DRAM memory size increases, each refresh operation requires more energy, and more lines need to be refreshed in a given time; therefore, refresh operations become the main source of DRAM power dissipation [31]. To address these issues, the use of emerging nonvolatile memory (NVM) technologies for on-chip memories have attracted much attention [20–24,29]. Many desirable characteristics such as higher density, near zero leakage power, and high resilience against soft errors are some advantages that are offered [25,32]. However, NVM technologies have many advantages; for example, they suffer from longer write latency, limited write endurance, and higher write energy consumption when compared with the traditional SRAM and DRAM architectures. These challenges prevent NVMs from being directly used as a replacement for traditional memories. To address these issues, we propose a hybrid architecture for future eCMPs, where SRAM and DRAM technologies are integrated with NVMs to use advantages of both traditional and new technologies for the first time.

The use of two-dimensional (2D) interconnections in CMPs will result in long global wire lengths, causing a high delay and low performance. To continue the progress of Moore's law, three-dimensional (3D) integration is introduced by stacking multiple dies vertically. A number of researchers have proposed 3D CMP architectures with a 3D-stacked cache hierarchy/memory system [28,32,33] to improve performance and reduce power consumption. Stacking memory systems directly on top of a core layer is a natural way of addressing the memory wall problem. In order to fabricate hybrid cache architecture, a special process is needed. The fabrication of spin-transfer torque RAM (STT-RAM) involves a hybrid magnetization CMOS processor, and requires the growth of a magnetic stacked layer between metal layers. The fabrication of on-chip mixed-technology integration is more cost-effective with respect to the gains in power and performance. The use of through silicon vias (TSVs) is the most promising solution for building 3D CMPs by the vertical stacking of dies. The manufacturing process of the TSV is complex and expensive [34]. Moreover, TSVs suffer from crosstalk noise and temperature. Given that TSVs are bridges between layers, they are potentially more prone to thermal stress. The TSV overhead, such as area, manufacturing cost, routing congestion, and yield loss can increase significantly with the increase in the number of TSVs [35–37]. Therefore, a reduction in the number of TSVs such that it does not lead to performance degradation has the potential to improve the reliability.

In this paper, we propose a convex optimization approach for power-efficient uncore architecting in 3D CMPs with the aim of improving performance through the optimal placement of heterogeneous memory banks and an optimal number of TSVs. Figure 1 shows an overview of the proposed hybrid uncore architecture in a 3D CMP with two layers.

This study makes the following contributions:

1) We propose a stacked 3D memory architecture with the optimal placement of SRAM incorporated with DRAM and STT-RAM banks in the memory layer.
2) We optimized the number of TSVs and proposed their optimal placement in the target 3D CMP.

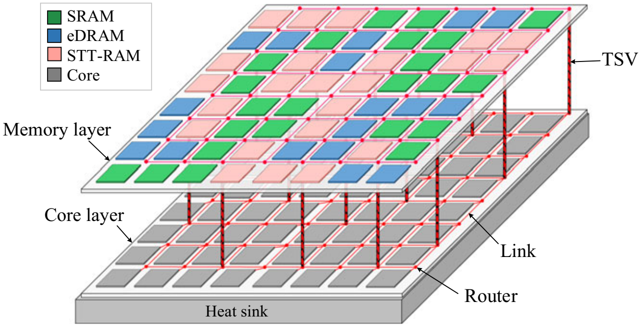

**FIGURE 1** An overview of a hybrid uncore architecture designed using the proposed convex method

3) We model and exploit architecture heterogeneity as an important feature to improve power efficiency, which is required in future many-core CMPs.
4) In the proposed model, we consider core and uncore leakage power consumption as an important contributor in the overall CMP power consumption in the nanoscale era.

## 2 | BACKGROUND

In this section, we first compare characteristics of different traditional and nonvolatile memory technologies with each other. Then, we review the STT-RAM technology as a well-known type of NVM technology.

The traditional and high-performance SRAM technology has been widely used in the on-chip caches owing to their standard logic compatibility, high endurance, and fast access-time features [30]. However, low-density SRAM technology dissipates a high leakage power because of the implementation of its six transistors [29], and this has become a bottleneck for energy-efficient design. The rising demand for increased memory in computing systems has made the use of conventional SRAM-based caches more expensive. DRAM technology has become a viable alternative for implementing on-chip caches because of its high density, high capacity, low-leakage, and good write-endurance features. It is possible to have more reliable large-last level cache with high memory bandwidth by stacking low leakage 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 the refresh energy to retain stored data, which have a negative impact on performance. The use of NVMs as a new emerging technology is an alternative option to addressing the weaknesses of traditional SRAM and DRAM memories, which are due to their ultra-low leakage power and higher density. Table 1 lists a brief comparison between SRAM, eDRAM, and STT-RAM technologies in 32-nm technology. As shown in Table 1, compared with eDRAM and SRAM technologies, STT-RAM commonly offers high cell density and zero leakage-power consumption. In addition, STT-RAM is around four times denser than SRAM in the same area. Therefore, STT-RAM is a promising candidate of NVMs that can be used to build larger on-chip memories and reduce the energy consumption of the memory design owing to its high density and near zero-leakage power consumption.

An STT-RAM cell consists of a magnetic tunnel junction (MTJ) to store bit information. The use of an MTJ as a fundamental building block in NVM technologies consists of two ferromagnetic layers separated by a dielectric layer, as shown in Figure 2C. While the direction of one ferromagnetic layer is fixed, the other layer can be controlled by passing a sufficiently large current through the MTJ. When the magnetization direction of the two layers will be paralleled, the MTJ will have a low resistance, which indicates a "0" logic (Figure 2B); otherwise, the magnetization directions of the two layers will be anti-paralleled, and the MTJ will have a high resistance, which indicates a "1" logic (Figure 2A). To design a memory cell, an MTJ is connected serially with an NMOS, as shown in Figure 2C [38].

## 3 | PROPOSED OPTIMIZATION PROBLEM

In this section, we propose a convex optimization technique that targets the optimization of a linear objective function subject to linear constraints. The outputs of our optimization problem are: 1) finding the optimal number of SRAM, eDRAM, and STT-RAM memory banks based on the memory access behavior of mapped applications with respect to performance and endurance constraints 2) the optimal placement of SRAM incorporated with eDRAM and STT-RAM banks in the memory layer 3) minimizing the number of TSVs, and 4) the optimal placement of TSVs to reduce cost and improve reliability without performance degradation. The proposed optimization model is designed for embedded systems on which special-purpose

**TABLE 1** Comparison of different 32-mm memory technologies

| Technology | 1 MB SRAM | 4 MB eDRAM | 4 MB STT-RAM |
|---|---|---|---|
| Area | 3.03 mm$^2$ | 3.31 mm$^2$ | 3.39 mm$^2$ |
| Read latency | 0.702 ns | 1.26 ns | 0.880 ns |
| Write latency | 0.702 ns | 1.26 ns | 10.67 ns |
| Leakage power at 80°C | 444.6 mW | 386.8 mW | 190.5 mW |
| Read energy | 0.168 nJ | 0.142 nJ | 0.278 nJ |
| Write energy | 0.168 nJ | 0.142 nJ | 0.765 nJ |
| Endurance | $10^{16}$ | $10^{16}$ | $4 \times 10^{12}$ |

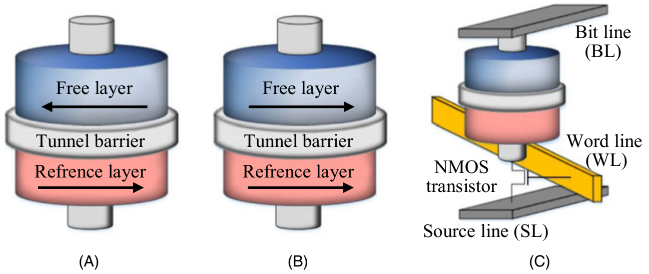

**FIGURE 2** Structure of a STT-RAM cell: (A) antiparallel state "1," (B) parallel state "0," and (C) a STT-RAM cell

applications are run. Hence, the behavior of these applications is known for us in design time. The proposed power model considers in detail the micro-architectural features and workload behavior, and presents an accurate power function that is based on architecture specifications and application parameters of the CMPs. Figure 3 shows the block diagram of the proposed optimization model.

SRC, DRC, STC, and TSV represent our optimization variables. SRC, DRC, and STC indicate that each memory bank in the proposed design is either an SRAM, an eDRAM, or an STT-RAM bank. In addition, TSV indicates that each tile on the core layer has a TSV to connect to the memory layer. Based on these variables, the optimal placement of SRAM, eDRAM, and STT-RAM banks in the second layer, as well as the optimal placement of TSVs in the first layer are performed for the 3D eCMP design, as shown in Figure 4.

After architecting the first layer and determining the placement of TSVs, and also constructing the second layer and determining the actual placement of SRAM, eDRAM, and STT-RAM banks on it, we can count the number of TSVs and memory banks, and hence, find the optimal number of TSVs and each memory technology in our design. This section describes the optimal placement of TSVs and memory banks with different technologies, which is done simultaneously using the proposed convex optimization model. Table 2 gives the constant terms used in our convex formulation. Based on the proposed optimization model, we propose a greedy algorithm for efficiently calculating the number and placement of SRAM, DRAM, and STT-RAM memory banks together with finding the optimal number of TSVs to connect layers. Our approach uses 0–1 variables to specify the coordinates of each memory bank and TSV.

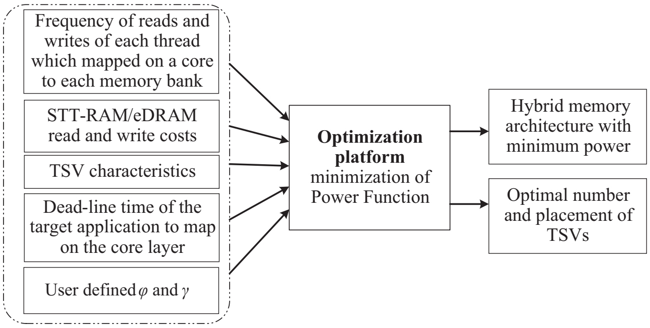
Frequency of reads and writes of each thread which mapped on a core to each memory bank
STT-RAM/eDRAM read and write costs
TSV characteristics
Dead-line time of the target application to map on the core layer
User defined φ and γ
Optimization platform minimization of Power Function
Hybrid memory architecture with minimum power
Optimal number and placement of TSVs

**FIGURE 3** Block diagram of our proposed optimization model

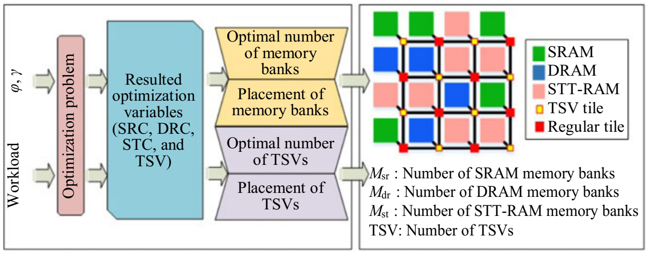
Workload
φ, γ
Optimization problem
Resulted optimization variables (SRC, DRC, STC, and TSV)
Optimal number of memory banks
Placement of memory banks
Optimal number of TSVs
Placement of TSVs
SRAM
DRAM
STT-RAM
TSV tile
Regular tile
$M_{sr}$ : Number of SRAM memory banks
$M_{dr}$ : Number of DRAM memory banks
$M_{st}$ : Number of STT-RAM memory banks
TSV: Number of TSVs

**FIGURE 4** Design steps of the 3D CMP

We used PC to identify the coordinates of a core in the core layer. More specifically,

- $\text{PC}_{p,x,y,l}$: indicates whether core $p$ is in $(x, y)$ in layer $l = 1$.

We used SRC, DRC, and STC in our formulation to identify the coordinates and technology of each memory bank. We have three types of memory banks, SRAM, eDRAM, and STT-RAM, so we have three memory variables.

- $\text{SRC}_{\text{sr},x,y,l}$: indicates whether SRAM bank sr is in $(x, y)$ in layer $l = 2$.
- $\text{DRC}_{\text{dr},x,y,l}$: indicates whether eDRAM bank dr is in $(x, y)$ in layer $l = 2$.
- $\text{STC}_{\text{st},x,y,x,l}$: indicates whether STT-RAM bank st is in $(x, y)$ in layer $l = 2$.

Similarly, we used TSV to identify the coordinates of a TSV on each tile on the core layer. More specifically,

- $\text{TSV}_{\text{tsv},x,y,l}$: indicates whether the located tile in $(x, y)$ in layer $l = 1$ has a TSV.
- $\text{REG}_{\text{reg},x,y,l}$: indicates whether the located tile in $(x, y)$ in layer $l = 1$ is a regular node without any TSV.
- $\text{Access}_{i,j,x,y,l}$: indicates whether the located tile in $(x, y)$ has access to the located tile in $(i, j)$ in layer $l = 1$.

The distances between a core and the nearest TSV on the dimensions $(x, y)$ are captured by $\text{Xdist}_{p,\text{tsv},x}$ and $\text{Ydist}_{p,\text{tsv},y}$. Specifically, we have:

- $\text{Xdist}_{p,\text{tsv},x,l}$: indicates whether the distance between core $p$ and the tile with the $\text{tsv}^{\text{th}}$ TSV is equal to $x$ on the $x$-axis in layer $l = 1$.
- $\text{Ydist}_{p,\text{tsv},y,l}$: indicates whether the distance between core $p$ and the tile with the $\text{tsv}^{\text{th}}$ TSV is equal to $y$ on the $y$-axis in layer $l = 1$.

The distances between a memory bank and the nearest TSV on the dimensions $(x, y)$ are captured by $\text{Xdist}_{\text{tsv},m,x}$ and $\text{Ydist}_{\text{tsv},m,y}$. Specifically, we have:

- $\text{Xdist}_{\text{tsv},m,x,l}$: indicates whether the distance between the tile with the $\text{tsv}^{\text{th}}$ TSV and memory bank $m$ is equal to $x$ on the $x$-axis in layer $l = 2$.
- $\text{Ydist}_{\text{tsv},m,y,l}$: indicates whether the distance between the tile with the $\text{tsv}^{\text{th}}$ TSV and memory bank $m$ is equal to $y$ on the $y$-axis in layer $l = 2$.

**TABLE 2** Constant terms used in our optimization problem

| Constant | Definition |
|---|---|
| $P$ | Number of cores in the core layer |
| $M$ | Number of memory banks in the memory layer |
| $M_{\text{sr}}$ | Number of SRAM memory banks |
| $M_{\text{dr}}$ | Number of eDRAM memory banks |
| $M_{\text{st}}$ | Number of STT-RAM memory banks |
| $C_X, C_Y$ | Dimensions of the chip |
| $N$ | Number of lines in STT-RAM memory bank |
| $l$ | Index of layers in the 3D CMP |
| $\text{FREQ}_{p,m,r}$ | Number of read access to memory bank $m$ by core $p$ |
| $\text{FREQ}_{p,m,w}$ | Number of write access to memory bank $m$ by core $p$ |
| $\varphi$ | Using STT-RAM vs SRAM and STT-RAM technologies ratio |
| $\gamma$ | Using eDRAM vs SRAM and eDRAM technologies ratio |
| $P_{\text{read}_{\text{SR}}}, P_{\text{write}_{\text{SR}}}$ | Average dynamic power consumption per read and write by SRAM memory bank |
| $P_{\text{read}_{\text{DR}}}, P_{\text{write}_{\text{DR}}}$ | Average dynamic power consumption per read and write by DRAM memory bank |
| $P_{\text{read}_{\text{ST}}}, P_{\text{write}_{\text{ST}}}$ | Average dynamic power consumption per read and write by STT-RAM memory bank |
| $P_{\text{static-sr}}$ | Static power consumed by each SRAM bank at maximum temperature limit |
| $P_{\text{static-dr}}$ | Static power consumed by each eDRAM bank at maximum temperature limit |
| $P_{\text{static-st}}$ | Static power consumed by each STT-RAM bank at maximum temperature limit |
| $\text{Endurance}_{\text{STT-line}}$ | Maximum number of writes for each line of a STT-RAM bank |
| $M_{\text{tsv}}$ | Maximum number of TSVs |
| $v$ | Number of virtual channels per link |
| $q$ | Size of a data block based on the packet size |
| $P_{\text{link}}^{\text{packet}}$ | Average power consumption required to transfer a packet from a link |
| $P_{\text{static}}^{\text{wire}}$ | Static power consumption of a link between two adjacent routers |
| $P_{\text{TSV}}^{\text{packet}}$ | Average power consumption required to transfer a packet from a TSV |
| $P_{\text{static}}^{\text{eRouter}}$ | Static power consumption of an empty router (without any packet) |
| $R_{\text{Cost-ST}}, W_{\text{Cost-ST}}$ | Cost of reading and writing to STT-RAM with respect to SRAM memory |
| $R_{\text{Cost-DR}}, W_{\text{Cost-DR}}$ | Cost of reading and writing to DRAM with respect to SRAM memory |
| $\tau_{\text{sr}}^{\text{r}}, \tau_{\text{sr}}^{\text{w}}$ | Average time to read/write a data packet from/to an SRAM bank |
| $\tau_{\text{dr}}^{\text{r}}, \tau_{\text{dr}}^{\text{w}}$ | Average time to read/write a data packet from/to an eDRAM bank |
| $\tau_{\text{st}}^{\text{r}}, \tau_{\text{st}}^{\text{w}}$ | Average time to read/write a data packet from/to a STT-RAM bank |
| $\tau_{\text{link}}^{\text{packet}}$ | Average time to transfer a packet from a link between two adjacent routers |
| $D$ | Deadline-time of the allocated program specified by the user in embedded applications |

A core needs to be assigned to a single coordinate.

$$\sum_{i=0}^{C_X-1} \sum_{j=0}^{C_Y-1} \text{PC}_{p,i,j,l} = 1, \forall p, l = 1. \tag{1}$$

In (1), $i$ and $j$ correspond to the $x$ and $y$ coordinates, respectively. A memory bank also needs to be assigned to a unique coordinate.

$$\sum_{i=0}^{C_X-1} \sum_{j=0}^{C_Y-1} (\text{SRC}_{\text{sr},i,j,l} + \text{DRC}_{\text{dr},i,j,l} + \text{STC}_{\text{st},i,j,l}) = 1, \quad \forall\ \text{sr}, \forall\ \text{dr}, \forall\ \text{st}, l = 2. \tag{2}$$

The sum of the used SRAM, eDRAM, and STT-RAM banks in the second layer is equal to $M$ as follows:

$$\sum_{x=0}^{C_X-1} \sum_{y=0}^{C_Y-1} \left( \sum_{i=1}^{M_{\text{sr}}} \text{SRC}_{i,x,y,l} + \sum_{i=1}^{M_{\text{dr}}} \text{DRC}_{i,x,y,l} + \sum_{i=1}^{M_{\text{st}}} \text{STC}_{i,x,y,l} \right) = M, l = 2. \tag{3}$$

In this work, the size of memory banks and its associated router/controller in the upper layer is the same as the size of cores in the lower layer to prevent VLSI problems related to the layout and TSV design, as shown in Figure 1. The number of memory banks in the upper layer is equal to the number of cores in the lower layer owing to the regularity of the architecture, as shown in Figure 1. In this model, it means that $P = M$.

In (3), $M_{\text{sr}}$, $M_{\text{dr}}$, and $M_{\text{st}}$ are the maximum number of available SRAM, eDRAM, and STT-RAM banks that we can use in our design. Note that in this work, we assume $M_{\text{sr}}$, $M_{\text{dr}}$, and $M_{\text{st}}$ are equal to $P$ in this work. According to the specified constraints in this work, there is a possibility that all of the memory banks on top of the cores are selected from pure SRAM, eDRAM, or STTRAM technology, or they can be selected from a combination of SRAM, eDRAM and STT-RAM banks that leads to the design of a hybrid architecture.

In order to prevent multiple mappings of a coordinate in our grid, we force a coordinate in the first layer to belong to a single core, and a coordinate in the second layer to belong to a memory bank (SRAM or eDRAM, or STT-RAM).

For the optimal placement of TSVs, we assume that there are two types of tiles in the core layer, that is, a regular tile that contains a core without any TSV, and a tile that contains a core and a TSV, as shown in Figure 5. As given in (4), a TSV is assigned to a unique coordinate.

$$\sum_{x=0}^{C_X-1}\sum_{y=0}^{C_Y-1}(\mathrm{TSV}_{\mathrm{tsv},x,y,l}+\mathrm{REG}_{\mathrm{reg},x,y,l})=1,\quad \forall\ \mathrm{tsv},\ \mathrm{reg},\ \text{and}\ l=1. \tag{4}$$

The sum of the used regular tiles and TSV tiles in the first layer is equal to $P$ as follows:

$$\sum_{x=0}^{C_X-1}\sum_{y=0}^{C_Y-1}\Big(\sum_{i=1}^{M_{\mathrm{tsv}}}\mathrm{TSV}_{i,x,y,l}+\sum_{i=1}^{P}\mathrm{REG}_{i,x,y,l}\Big)=P,\ l=1. \tag{5}$$

The sum of TSVs is considered between a specified minimum and maximum number of TSVs based on (6).

$$\frac{M_{\mathrm{tsv}}}{16}\le\sum_{x=0}^{C_X-1}\sum_{y=0}^{C_Y-1}\sum_{\mathrm{tsv}=1}^{M_{\mathrm{tsv}}}\mathrm{TSV}_{\mathrm{tsv},x,y,l}\le M_{\mathrm{tsv}},\ l=1. \tag{6}$$

The minimum and maximum number of TSVs on a core layer with 64 cores is illustrated in Figure 5.

Equations (7) and (8) indicate that the TSVs are distributed well. These equations show that there are no adjacent TSVs on the core layer.

$$\mathrm{TSV}_{\mathrm{tsv},x,y,l}+\mathrm{TSV}_{\mathrm{tsv},x,y+1,l}\le 1,\ \forall x,y,\mathrm{tsv},\ \text{and}\ l=1, \tag{7}$$

$$\mathrm{TSV}_{\mathrm{tsv},x,y,l}+\mathrm{TSV}_{\mathrm{tsv},x+1,y,l}\le 1,\ \forall x,y,\mathrm{tsv},\ \text{and}\ l=1. \tag{8}$$

$\mathrm{Access}_{i,j,x,y,l}$ indicates that there is accessibility between tile $(i,j)$ and tile $(x,y)$ in layer $l$. This accessibility is specified based on the topology and used routing algorithm in advance as a problem input. Equation (9) shows that every regular node in layer $l=1$ should access at least $r_{\min}$ nodes with TSV. In addition, this equation shows that every regular node in layer $l=1$ should access $r_{\max}$ nodes with TSV at most.

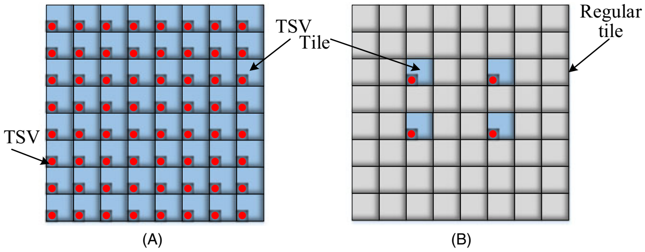

**FIGURE 5** Maximum and minimum number of TSVs on the core layer with 64 cores, $M_{\mathrm{tsv}}=64$: (A) maximum number of TSVs and (B) minimum number of TSVs

$$r_{\min}\le\sum_{i=0}^{C_X-1}\sum_{j=0}^{C_Y-1}\mathrm{REG}_{\mathrm{reg},i,j,l}\times\mathrm{Access}_{i,j,x,y,l}\times\mathrm{TSV}_{\mathrm{tsv},x,y,l}\le r_{\max},\ \forall x,y,\mathrm{reg},\ \mathrm{tsv},\ \text{and}\ l=1. \tag{9}$$

As we used index $l$ in this work, in the future, we can extend this proposed model to 3D CMPs with more than two layers.

$\mathrm{Xdist}_{p,\mathrm{tsv},x,l}$ and $\mathrm{Ydist}_{p,\mathrm{tsv},y,l}$ show the *Manhattan distance* between the core $p$ and its nearest TSV.

$$\mathrm{Xdist}_{p,\mathrm{tsv},x,l}\ge\mathrm{Access}_{x1,y1,x2,y2,l}\times\big(\mathrm{PC}_{p,x1,y1,l}+\mathrm{TSV}_{\mathrm{tsv},x2,y2,l}-1\big),\ \forall p,\mathrm{tsv},x1,x2,y1,y2,\ \text{and}\ l=1,x=|x1-x2|, \tag{10}$$

$$\mathrm{Ydist}_{p,\mathrm{tsv},y,l}\ge\mathrm{Access}_{x1,y1,x2,y2,l}\times\big(\mathrm{PC}_{p,x1,y1,l}+\mathrm{TSV}_{\mathrm{tsv},x2,y2,l}-1\big),\ \forall p,\mathrm{tsv},x1,x2,y1,y2,\ \text{and}\ l=1,y=|y1-y2|. \tag{11}$$

$\mathrm{Xdist}_{\mathrm{tsv},m,x}$ and $\mathrm{Ydist}_{\mathrm{tsv},m,y}$ show the *Manhattan distance* between the memory bank $m$ and its nearest TSV.

$$\mathrm{Xdist}_{\mathrm{tsv},m,x}\ge\mathrm{Access}_{x1,y1,x2,y2}\times\big(\mathrm{TSV}_{\mathrm{tsv},x1,y1}+\mathrm{DRC}_{dr,x2,y2,l}+\mathrm{SRC}_{\mathrm{sr},x2,y2,l}+\mathrm{STC}_{\mathrm{st},x2,y2,l}-1\big)\ \forall\ \mathrm{tsv},m,\mathrm{dr},\mathrm{sr},\mathrm{st},x1,x2,y1,y2,\ \text{and}\ l=2,x=|x1-x2|, \tag{12}$$

$$\mathrm{Ydist}_{\mathrm{tsv},m,y}\ge\mathrm{Access}_{x1,y1,x2,y2}\times\big(\mathrm{TSV}_{\mathrm{tsv},x1,y1}+\mathrm{DRC}_{\mathrm{dr},x2,y2,l}+\mathrm{SRC}_{\mathrm{sr},x2,y2,l}+\mathrm{STC}_{\mathrm{st},x2,y2,l}-1\big),\ \forall\ \mathrm{tsv},m,\mathrm{dr},\mathrm{sr},\mathrm{sr},x1,x2,y1,y2\ \text{and}\ l=2,y=|y1-y2|. \tag{13}$$

Figure 6 shows the *Manhattan distance* between the core $p$ and memory bank $m$. It should be noted that because our target CMP architecture has two layers in this work, the tile in the core layer, which is connected by a TSV to a memory bank in the upper layer, is specified as a TSV tile.

We denote the total power consumption of used SRAM banks on the memory layer as $P_{\mathrm{SR}}$. $P_{\mathrm{SR}}$ is comprised of dynamic and static power consumption.

$$P_{\mathrm{SR}}=P_{\mathrm{static_{SR}}}+P_{\mathrm{dynamic_{SR}}}. \tag{14}$$

The static power dissipation depends on temperature. Because this optimization approach is solved in the design phase, we consider the pessimistic worst-case temperature assumption and calculate $P_{\mathrm{static_{SR}}}$, $P_{\mathrm{static_{DR}}}$, and $P_{\mathrm{static_{ST}}}$ at the maximum temperature limit.

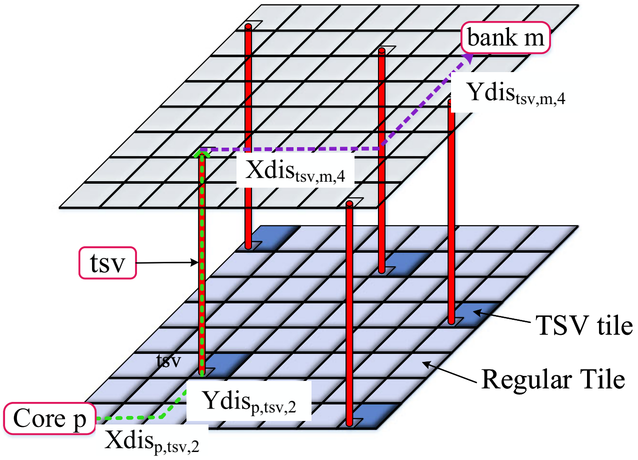

**FIGURE 6** Manhattan distance between core *p* and memory bank *m*

$$P_{\text{static}_{\text{SR}}} = \sum_{i=0}^{C_X-1} \sum_{j=0}^{C_Y-1} \left( \sum_{k=1}^{M_{\text{sr}}} \text{SRC}_{k,i,j,l} \times P_{\text{static-sr}} \right), l = 2. \quad (15)$$

In (16), $E_{\text{read}_{\text{sr}}}$ and $E_{\text{write}_{\text{sr}}}$indicate the average dynamic power consumed by a SRAM bank per read and write access, respectively. $E_{\text{dynamic}_{\text{SR}}}$ is the dynamic power consumption of the used SRAM memory banks on the core layer:

$$P_{\text{dynamic}_{\text{SR}}} = \sum_{i=0}^{C_X-1} \sum_{j=0}^{C_Y-1} \sum_{p=1}^{P} \left( \sum_{k=1}^{M_{\text{sr}}} \text{SRC}_{k,i,j,l} \times \left( \text{FREQ}_{p,k,r} \times P_{\text{read sr}} + \text{FREQ}_{p,k,w} \times P_{\text{write}_{\text{sr}}} \right) \right), l = 2. \quad (16)$$

In this context, the total power consumption of the used eDRAM banks on the memory layer is as shown in (17) to (19).

$$P_{\text{DR}} = P_{\text{static}_{\text{DR}}} + P_{\text{dynamic}_{\text{DR}}}, \quad (17)$$

$$P_{\text{static}_{\text{DR}}} = \sum_{i=0}^{C_X-1} \sum_{j=0}^{C_Y-1} \left( \sum_{k=1}^{M_{\text{dr}}} \text{DRC}_{k,i,j,l} \times P_{\text{static-dr}} \right), l = 2, \quad (18)$$

$$P_{\text{dynamic}_{\text{DR}}} = \sum_{i=0}^{C_X-1} \sum_{j=0}^{C_Y-1} \sum_{p=1}^{P} \left( \sum_{k=1}^{M_{\text{dr}}} \text{DRC}_{k,i,j,l} \times \left( \text{FREQ}_{p,k,r} \times P_{\text{read}_{\text{dr}}} + \text{FREQ}_{p,k,w} \times P_{\text{write}_{\text{dr}}} \right) \right), l = 2. \quad (19)$$

The total power consumption of the used STT-RAM banks on the memory layer is given as (20) to (22).

$$P_{\text{ST}} = P_{\text{static}_{\text{ST}}} + P_{\text{dynamic}_{\text{ST}}}, \quad (20)$$

$$P_{\text{static}_{ST}} = \sum_{i=0}^{C_X-1} \sum_{j=0}^{C_Y-1} \left( \sum_{k=1}^{M_{st}} \text{STC}_{k,i,j,l} \times P_{\text{static-st}} \right), l = 2, \quad (21)$$

$$P_{\text{dynamic}_{\text{ST}}} = \sum_{i=0}^{C_X-1} \sum_{j=0}^{C_Y-1} \sum_{p=1}^{P} \left( \sum_{k=1}^{M_{\text{st}}} \text{STC}_{k,i,j,l} \times \left( \text{FREQ}_{p,k,r} \times P_{\text{read}_{\text{st}}} + \text{FREQ}_{p,k,w} \times P_{\text{write}_{\text{st}}} \right) \right), l = 2. \quad (22)$$

The memory systems and on-chip interconnection network are the main contributors in the power consumption of uncore components. Equations (14) to (22) show the power consumption related to the memory system. In addition, (23) to (29) show the power consumption related to the 3D on-chip interconnection network in the target 3D CMPs in this work.

$$P_{\text{onchip-interconnection}} = P_{\text{TSVs}} + P_{\text{2Dlinks}}. \quad (23)$$

Based on (24) to (26), $P_{\text{TSVs}}$ is calculated as follows:

$$P_{\text{TSVs}} = P_{\text{static}_{\text{TSVs}}} + P_{\text{dynamic}_{\text{TSVs}}}, \quad (24)$$

$$P_{\text{static}_{\text{TSVs}}} = \sum_{x=0}^{C_X-1} \sum_{y=0}^{C_Y-1} \sum_{\text{tsv}=1}^{M_{\text{tsv}}} \text{TSV}_{\text{tsv},x,y,l} \times P_{\text{static}}^{\text{TSV}}, l = 1, \quad (25)$$

$$P_{\text{dynamic}_{\text{TSVs}}} = \sum_{p=1}^{P} \sum_{k=1}^{M} \sum_{\text{tsv}=1}^{M_{\text{tsv}}} \sum_{i=0}^{C_X-1} \sum_{j=0}^{C_Y-1} \sum_{x=0}^{C_X-1} \sum_{y=0}^{C_Y-1} P_{\text{TSV}}^{\text{packet}} \times q \times \left( \text{PC}_{p,i,j,l} \times \text{Access}_{i,j,x,y} \times \text{TSV}_{\text{tsv},x,y} \right) \times \left( \text{FREQ}_{p,k,r} + \text{FREQ}_{p,k,w} \right), l = 1, \quad (26)$$

Based on (27) to (29), $P_{\text{2Dlinks}}$is calculated as follows:

$$P_{\text{2Dlinks}} = P_{\text{static}_{\text{2Dlinks}}} + P_{\text{dynamic}_{\text{2Dlinks}}}, \quad (27)$$

$$P_{\text{dynamic}_{\text{2Dlinks}}} = \sum_{p=1}^{P} \sum_{k=1}^{M} \sum_{\text{tsv}=1}^{M_{\text{tsv}}} \left( \sum_{i=0}^{C_X-1} \sum_{j=0}^{C_Y-1} \left( i \times \text{Xdist}_{p,\text{tsv},i} \right) + \left( j \times \text{Ydist}_{p,\text{tsv},j} \right) + \sum_{i=0}^{C_X-1} \sum_{j=0}^{C_Y-1} \left( i \times \text{Xdist}_{\text{tsv},k,i} \right) + \left( j \times \text{Ydist}_{\text{tsv},k,j} \right) \right) \times \left( \text{FREQ}_{p,k,r} + \text{FREQ}_{p,k,w} \right) \times P_{\text{link}}^{\text{packet}} \times q, l1 = 1, l2 = 2, \quad (28)$$

$$P_{\text{static}_{\text{2Dlinks}}} = 2 \times (C_X - 1) \times (C_Y - 1) \times P_{\text{static}}^{\text{wire}} + C_X \times C_Y \times P_{\text{static}}^{\text{eRouter}} \times v. \quad (29)$$

Next, we model $T_{\text{EXE}}$, which is the time parameter for calculating the static power consumptions in (15), (18), (21), (25), and (29).

The cost of read and write requests to SRAM banks on the memory layer is shown in (30) to (35).

$$\begin{aligned} X_{\text{Cost-read-SR}} = {} & \sum_{p=1}^{P} \sum_{k=1}^{M_{\text{sr}}} \sum_{\text{tsv}=1}^{M_{\text{tsv}}} \left( \sum_{i=0}^{C_X-1} \left( i \times \text{Xdist}_{p,\text{tsv},i,l1} \right) \right. \\ & \left. + \sum_{x=0}^{C_X-1} \sum_{y=0}^{C_Y-1} \sum_{j=0}^{C_X-1} \left( j \times \text{Xdist}_{\text{tsv},k,j} \times \text{SRC}_{k,x,y,l2} \right) \right) \\ & \times \left( \text{FREQ}_{p,k,r} \right), l1 = 1, l2 = 2, \end{aligned} \tag{30}$$

$$\begin{aligned} Y_{\text{Cost-read-SR}} = {} & \sum_{p=1}^{P} \sum_{k=1}^{M_{\text{sr}}} \sum_{\text{tsv}=1}^{M_{\text{tsv}}} \left( \sum_{i=0}^{C_Y-1} \left( i \times \text{Ydist}_{p,\text{tsv},i,l1} \right) \right. \\ & \left. + \sum_{x=0}^{C_X-1} \sum_{y=0}^{C_Y-1} \sum_{j=0}^{C_Y-1} \left( j \times \text{Ydist}_{\text{tsv},k,j} \times \text{SRC}_{k,x,y,l2} \right) \right) \\ & \times \left( \text{FREQ}_{p,k,r} \right), l1 = 1, l2 = 2, \end{aligned} \tag{31}$$

$$XY_{\text{Cost-read-SR}} = X_{\text{Cost-read-SR}} + Y_{\text{Cost-read-SR}}, \tag{32}$$

$$\begin{aligned} X_{\text{Cost-write-SR}} = {} & \sum_{p=1}^{P} \sum_{k=1}^{M_{\text{sr}}} \sum_{\text{tsv}=1}^{M_{\text{tsv}}} \left( \sum_{i=0}^{C_X-1} \left( i \times \text{Xdist}_{p,\text{tsv},i,l1} \right) \right. \\ & \left. + \sum_{x=0}^{C_X-1} \sum_{y=0}^{C_Y-1} \sum_{j=0}^{C_X-1} \left( j \times \text{Xdist}_{\text{tsv},k,j} \times \text{SRC}_{k.x,y,l2} \right) \right) \\ & \times \left( \text{FREQ}_{p,k,w} \right), l1 = 1, l2 = 2, \end{aligned} \tag{33}$$

$$\begin{aligned} Y_{\text{Cost-write-SR}} = {} & \sum_{p=1}^{P} \sum_{k=1}^{M_{\text{sr}}} \sum_{\text{tsv}=1}^{M_{\text{tsv}}} \left( \sum_{i=0}^{C_Y-1} \left( i \times \text{Ydist}_{p,\text{tsv},i,l1} \right) \right. \\ & \left. + \sum_{x=0}^{C_X-1} \sum_{y=0}^{C_Y-1} \sum_{j=0}^{C_Y-1} \left( j \times \text{Ydist}_{\text{tsv},k,j} \times \text{SRC}_{k,x,y,l2} \right) \right) \\ & \times \left( \text{FREQ}_{p,k,w} \right), l1 = 1, l2 = 2, \end{aligned} \tag{34}$$

$$XY_{\text{Cost-write-SR}} = X_{\text{Cost-write-SR}} + Y_{\text{Cost-write-SR}}. \tag{35}$$

The cost of read and write requests to eDRAM banks on the memory layer is shown in (36) to (41). $R_{\text{Cost-DR}}$ and $W_{\text{Cost-DR}}$ are new constant parameters that are used in the eDRAM request cost function compared with the SRAM cost function.

$$\begin{aligned} X_{\text{Cost-read-DR}} = {} & \sum_{p=1}^{P} \sum_{k=1}^{M_{\text{dr}}} \sum_{\text{tsv}=1}^{M_{\text{tsv}}} \left( \sum_{i=0}^{C_X-1} \left( i \times \text{Xdist}_{p,\text{tsv},i,l1} \right) \right. \\ & \left. + \sum_{x=0}^{C_X-1} \sum_{y=0}^{C_Y-1} \sum_{j=0}^{C_X-1} \left( j \times \text{Xdist}_{\text{tsv},k,j} \times \text{DRC}_{k,x,y,l2} \right) \right) \\ & \times \left( \text{FREQ}_{p,k,r} \times R_{\text{Cost-DR}} \right), l1 = 1, l2 = 2, \end{aligned} \tag{36}$$

$$\begin{aligned} Y_{\text{Cost-read-DR}} = {} & \sum_{p=1}^{P} \sum_{k=1}^{M_{\text{dr}}} \sum_{\text{tsv}=1}^{M_{\text{tsv}}} \left( \sum_{i=0}^{C_Y-1} \left( i \times \text{Ydist}_{p,\text{tsv},i,l1} \right) \right. \\ & \left. + \sum_{x=0}^{C_X-1} \sum_{y=0}^{C_Y-1} \sum_{j=0}^{C_Y-1} \left( j \times \text{Ydist}_{\text{tsv},k,j} \times \text{DRC}_{k,x,y,l2} \right) \right) \\ & \times \left( \text{FREQ}_{p,k,r} \times R_{\text{Cost-DR}} \right), l1 = 1, l2 = 2, \end{aligned} \tag{37}$$

$$XY_{\text{Cost-read-DR}} = X_{\text{Cost-read-DR}} + Y_{\text{Cost-read-DR}}, \tag{38}$$

$$\begin{aligned} X_{\text{Cost-write-DR}} = {} & \sum_{p=1}^{P} \sum_{k=1}^{M_{\text{dr}}} \sum_{\text{tsv}=1}^{M_{\text{tsv}}} \left( \sum_{i=0}^{C_X-1} \left( i \times \text{Xdist}_{p,\text{tsv},i,l1} \right) \right. \\ & \left. + \sum_{x=0}^{C_X-1} \sum_{y=0}^{C_Y-1} \sum_{j=0}^{C_X-1} \left( j \times \text{Xdist}_{\text{tsv},k,j} \times \text{DRC}_{k,x,y,l2} \right) \right) \\ & \times \left( \text{FREQ}_{p,k,w} \times W_{\text{Cost-DR}} \right), l1 = 1, l2 = 2, \end{aligned} \tag{39}$$

$$\begin{aligned} Y_{\text{Cost-write-DR}} = {} & \sum_{p=1}^{P} \sum_{k=1}^{M_{\text{dr}}} \sum_{\text{tsv}=1}^{M_{\text{tsv}}} \left( \sum_{i=0}^{C_Y-1} \left( i \times \text{Ydist}_{p,\text{tsv},i,l1} \right) \right. \\ & \left. + \sum_{x=0}^{C_X-1} \sum_{y=0}^{C_Y-1} \sum_{j=0}^{C_Y-1} \left( j \times \text{Ydist}_{\text{tsv},k,j} \times \text{DRC}_{k,x,y,l2} \right) \right) \\ & \times \left( \text{FREQ}_{p,k,w} \times W_{\text{Cost-DR}} \right), l1 = 1, l2 = 2, \end{aligned} \tag{40}$$

$$XY_{\text{Cost-write-DR}} = X_{\text{Cost-write-DR}} + Y_{\text{Cost-write-DR}}. \tag{41}$$

$XY_{\text{Cost-read-ST}}$ and $XY_{\text{Cost-write-ST}}$ are read and write cost functions of STT-RAM banks, and are calculated in the same way as DRAM banks with the difference being that we used $R_{\text{Cost-ST}}$ and $W_{\text{Cost-ST}}$ instead of $R_{\text{Cost-DR}}$ and $W_{\text{Cost-DR}}$. We do not show these equations because of space limitations.

The execution times on SRAM, eDRAM, and STT-RAM banks are calculated as shown in (42) to (44).

$$\begin{aligned} T_{\text{EXE-SR}} = {} & \left( XY_{\text{Cost-read-SR}} \right) \times q \times \left( \tau_{\text{sr}}^{r} + \tau_{\text{link}}^{\text{packet}} \right) \\ & + \left( XY_{\text{Cost-write-SR}} \right) \times q \times \left( \tau_{\text{sr}}^{w} + \tau_{\text{link}}^{\text{packet}} \right), \end{aligned} \tag{42}$$

$$\begin{aligned} T_{\text{EXE-DR}} = {} & \left( XY_{\text{Cost-read-DR}} \right) \times q \times \left( \tau_{\text{dr}}^{r} + \tau_{\text{link}}^{\text{packet}} \right) \\ & + \left( XY_{\text{Cost-write-DR}} \right) \times q \times \left( \tau_{\text{dr}}^{w} + \tau_{\text{link}}^{\text{packet}} \right), \end{aligned} \tag{43}$$

$$\begin{aligned} T_{\text{EXE-ST}} = {} & \left( XY_{\text{Cost-read-ST}} \right) \times q \times \left( \tau_{st}^{r} + \tau_{link}^{packet} \right) \\ & + \left( XY_{\text{Cost-write-ST}} \right) \times q \times \left( \tau_{\text{st}}^{w} + \tau_{\text{link}}^{\text{packet}} \right). \end{aligned} \tag{44}$$

The total execution time of the mapped embedded application, $T_{\text{EXE}}$, is as follows:

$$T_{\text{EXE}} = T_{\text{EXE-SR}} + T_{\text{EXE-DR}} + T_{\text{EXE-ST}}. \tag{45}$$

Equation (46) shows that the total time of execution of the mapped embedded application should be less than the accepted time-to-deadline by the user.

$$T_{\text{EXE}} \le D. \tag{46}$$

We consider the endurance problem of STT-RAM in our convex optimization model. We used an endurance model to decide between two types of memory banks based on the limited write endurance of STT-RAM for optimal placement. Note that, this endurance model can be used for other NVM memory types. This endurance constraint can be expressed as follows:

$$\frac{\sum_{i=1}^{P} \mathrm{FREQ}_{i,\mathrm{st},w}}{\mathrm{Endurance}_{\mathrm{STT\text{-}line}}} \times \mathrm{STC}_{\mathrm{st},x,y,2} < \frac{N}{2}, \forall x, y, \mathrm{st}. \tag{47}$$

Given that STT-RAM has an endurable write threshold, we can only write a limited number of times in each line of STT-RAM. If the number of write accesses into one line exceeds the threshold, that line will be destroyed. We assume a worst-case scenario where all of the write operations are written in one line until that line is destroyed, after which a new line is selected for the rest of the write operations. In this equation, we assume that when 50% of the lines of an STT-RAM bank are destroyed, that bank will be corrupted. Hence, the maximum tolerable number of destroyed lines required for us to use a special STT-RAM bank is $N/2$. Thus, in our endurance constraint model, if placing an STT-RAM memory bank in the special position leads to the destruction of more than half of the lines of that memory owing to the writing frequency of cores, the STT-RAM bank is not chosen for that position.

We denote the total power consumption of the proposed uncore architecture as $P_{\mathrm{Uncore}}$. Consequently, our objective function can be expressed as:

$$\text{minimize}\{P_{\mathrm{Uncore}} = P_{\mathrm{SR}} + \varphi . P_{\mathrm{ST}} + \gamma . P_{\mathrm{DR}} + P_{\mathrm{onchip\text{-}interconnection}}\}. \tag{48}$$

In (48), the target is to minimize the objective function subject to the constraints (1) to (47). A weighted objective function is considered in order to determine the potential effects on the overall performance. This is achieved by the $\gamma$ and $\varphi$ constants. The $\varphi$ constant is as a knob for choosing STT-RAM vs SRAM and eDRAM banks in each $x$ and $y$. In addition, the $\gamma$ constant is like a knob that is used to choose eDRAM vs SRAM and STT-RAM banks in each of the $x$ and $y$ dimensions, respectively, of the memory layer

### 3.1 | Solving the proposed optimization problems

In the optimization problem presented in (1) to (48), the constraints and objective function are linear. As discussed in [39], the linear functions are convex. In addition, based on the convexity proof fact [39], all of the constraints in the optimization problem should be convex functions. Therefore, the proposed optimization problem is a convex problem.

To solve the proposed convex optimization model, we can use CVX [40], which is an efficient optimization solver. Another approach for solving the optimization problems is to propose greedy algorithms that are less time consuming and less expensive. For the proposed optimization model, we propose a greedy algorithm to optimally pick number and placement of STT-RAM and SRAM banks on the memory layer, as well as to find the optimal number and placement of TSVs required to vertically connect the layers to each other. This is shown by details in Algorithm 1. In this algorithm, $\mathrm{Access}_{d_{\max}}(n)$ denotes the number of tiles that are accessible by tile $n$ at the maximum Manhattan distance of $d_{\max}$. The order of this algorithm is $O(m, t)$ in a CMP with $m$ memory banks and $t$ TSVs. Note that the proposed algorithm is performed only once for a system in the design phase, and the timing overhead for this is negligible.

**Algorithm 1:** (Greedy algorithm used to find the optimal number and placement of SRAM, eDRAM, and STT-RAM bank and TSVs)

```
SET ALL memory bank_type to SRAM
SET TSV to all tiles
Calculate $\mathrm{Power_{total}}$ *
for $i \in [0, C_x - 1]$
    for $j \in [0, C_y - 1]$:
        for $m \in [1, M]$:
            if $m$ is STT-RAM:
                Calculate STT-limit for bank ($m$)
                if (STT-limit > $N/2$ or Write_FREQ($m$) > $\mathrm{Write_{threshold}}$):
                    change bank_type($m$) to eDRAM
            else:
                change bank_type($m$) to STT-RAM
                if $T_{\mathrm{EXE}} > D$:
                    change bank_type($m$) to SRAM
                Calculate $\mathrm{Power_{total}}^{\mathrm{new}}$
                if($\mathrm{Power_{total}}^{\mathrm{new}} < \mathrm{Power_{total}}$ *):
                    $\mathrm{Power_{total}}$ * = $\mathrm{Power_{total}}^{\mathrm{new}}$
                    bank_type$^*$ = bank_type
            for $x \in [0, Cx - 1]$:
                for $y \in [0, Cy - 1]$:
                    while num_TSV > $M_{\mathrm{tsv}}/16$:
                        TSV_coordinate (x, y) = False
                        for all n of tiles:
                            if ($\mathrm{Access}_{d_{\max}}(n) > r_{\max}$ or $\mathrm{Access}_{d_{\max}}(n) < r_{\min}$):
                                TSV_coordinate ($x,y$) = True
                                Break
                        calculate $\mathrm{Power_{total}}^{\mathrm{new}}$
                        if ($T_{\mathrm{exe}} > D$ and $\mathrm{Power_{total}}^{\mathrm{new}} > \mathrm{Power_{total}}$ *):
                            TSV_coordinate ($x,y$) = True
                        else:
                            $\mathrm{Power_{total}}$ * = $\mathrm{Power_{total}}^{\mathrm{new}}$
                            TSV_coordinate$^*$ = TSV_coordinate
return bank_type$^*$, TSV_coordinate$^*$, $\mathrm{Power_{total}}$*
```

## 4 | EXPERIMENTAL EVALUATION

### 4.1 | Platform setup

In order to validate the efficacy of 3D CMP architectures in this work, we employed a detailed simulation framework that is driven by traces extracted from real applications running on a full-system simulator. The traces have been extracted from the GEM5 full-system simulator [41]. To simulate a 3D CMP architecture, the extracted traces from GEM5 are interfaced with 3D Noxim, as a 3D NoC simulator [42]. Figure 1 shows the eCMP at 32-nm technology for use in our experiment, which contains 64 cores on a core layer and 64 memory banks on the memory layer. In the core layer, each processing core is connected to a router. The area of the core tile (consisting of a processing core, private 32KB L1 instruction and data caches, and the cache controller with cache tags) is 3.5 mm$^2$, as estimated by McPAT [43] and CACTI 6.0 [44]. The detailed system configurations are given in Table 3.

In this study, GEM5 is augmented with McPAT and 3D Noxim with ORION [45] to calculate the power consumption. The cache capacities and power consumption of SRAM, DRAM, and NVMs are estimated from CACTI and NVSIM [46], respectively. We employed Hotspot5.0 [47] as a grid-based thermal modeling tool to estimate the 3D temperature. In this work, the simulation platform for the evaluation of our proposed and other 3D CMP architectures is illustrated in Figure 7.

To perform our experiments, we used multithreaded applications *Simlarge* input set consisting of 64 threads from PARSEC benchmarks [48]. The percentage of read and write accesses for each PARSEC benchmark is listed in Table 4.

**TABLE 3** Specification of CMP configurations evaluated in this work

| Component | Description |
|---|---|
| Number of cores | 64, 8 × 8 mesh |
| Core configuration | Alpha21164, 3 GHz, area 3.5 mm$^2$, 32 nm |
| Private L1 cache | SRAM, 4 way, 32B line, size 32-KB per core |
| Shared L2 cache | 1. SRAM-Baseline: 64 MB (1 MB each SRAM bank)<br>2. eDRAM-Baseline: 256 MB (4 MB each eDRAM bank)<br>3. STTRAM-Baseline: 256 MB (4 MB each STT-RAM bank)<br>4. Hybrid-Fix-Symmetric: 128 MB STT-RAM (32 banks, 4 MB each) and 128 MB eDRAM (32 banks, 4 MB each)<br>5. Hybrid-Fix-Centric: 192 MB STT-RAM and 64-MB eDRAM (48 STT-RAM and 16 eDRAM banks, 4 MB each)<br>6. Proposed: the proposed hybrid memory based on the convex optimization model |
| Main memory | 4 GB, 320 cycle access, 4 on-chip memory controllers at each corner node |
| Network router | 2-stage wormhole switched, virtual channel flow control, 2 VCs per port, 5 flits buffer depth, 8 flits per a data packet, 1 flit per address packet, 16-byte in each flit |

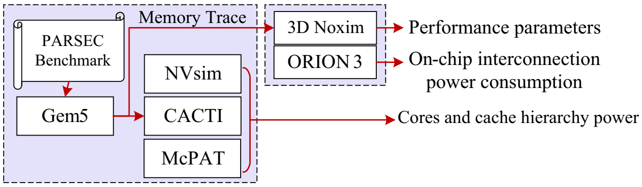

**FIGURE 7** Simulation platform of work

## 4.2 | Experimental result

In this subsection, we evaluate the target 3D eCMP with stacked memory for six different cases: 1) the CMP with SRAM-only stacked memory (SRAM-baseline), 2) the CMP with eDRAM-only stacked memory (DRAM-baseline), 3) the CMP with STT-RAM-only stacked memory (STT-RAM-baseline), 4) the CMP with hybrid stacked memory that has 16 eDRAM banks in the central part and 48 STT-RAMs around the eDRAM banks (hybrid-fix-centric), 5) the CMP with hybrid stacked memory that has 32 STT-RAM banks on the downside and 32 eDRAM banks on the upper side (hybrid-fix-symmetric), and 6) the CMP with the proposed hybrid stacked memory based on our proposed convex optimization model (proposed). In the proposed method, we consider a total of 64 SRAM banks (1 MB each), 64 STT-RAM banks (4 MB each), and 64 eDRAM banks (4 MB each) as the maximum available memory banks that can be used to design the hybrid memory architecture. We compared the results of the baseline designs with the proposed architecture to evaluate our work. In addition, we compared the proposed design with the new reconfigurable hybrid cache architecture

**TABLE 4** Percentage of read and write accesses for each benchmark

| Benchmark | Read access | Write access |
|---|---|---|
| Blackscholes | 90.23% | 9.77% |
| Bodytrack | 93.64% | 6.36% |
| Canneal | 97.57% | 2.43% |
| Dedup | 95.96% | 4.04% |
| Facesim | 64.10% | 35.90% |
| Ferret | 92.46% | 7.54% |
| Fluidanimate | 90.46% | 9.54% |
| Freqmine | 91.77% | 8.23% |
| Rtview | 85.20% | 14.80% |
| Streamcluster | 93.85% | 6.15% |
| Swaption | 95.92% | 4.08% |
| Vips | 92.04% | 7.96% |
| X264 | 94.59% | 5.41% |

(RHC) [29] design. Cheng et al. [29] proposed a novel reconfigurable hybrid cache design (RHC) for the last level cache, which supports STT-RAM memory beside SRAM to reduce the leakage energy. They power on/off SRAM and NVM arrays in a way-based manner in order to change the cache size and improve the memory requirements for different workloads. In their work, hardware-based mechanisms are proposed to detect the memory requirements for the RHC method. The specification of the RHC architecture in our work is the same as in Table 3.

To show the results of our work, the generated hybrid uncore architectures based on the proposed optimization model for the *canneal* and *facesim* benchmarks are shown in Figure 8. As illustrated in Table 4, *facesim* is a write-intensive workload in which about 35.9% of accesses are write operations. Meanwhile, *canneal* is a read-intensive application, and about 97% of accesses to L2 cache are read operations. As shown in Figure 9, in a read-intensive workload such as *canneal*, the contribution of STT-RAM banks is greater than that of eDRAM and SRAM banks. However, in a write-intensive workload, the contribution of eDRAM banks is increased. As shown in Figure 9, in the *facesim* workload, about 43% of banks are STT-RAM, while in the *canneal* workload, this percentage is 67%.

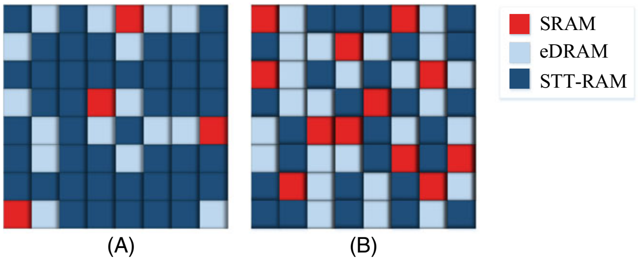

**FIGURE 8** Proposed hybrid uncore architectures for the *canneal* and *facesim* benchmarks: (A) canneal and (B) facesim

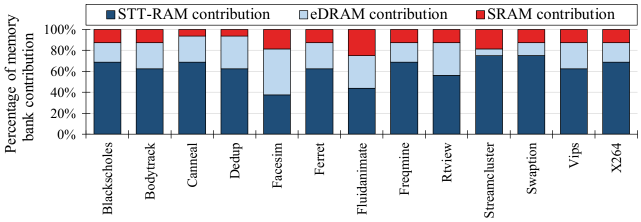

**FIGURE 9** Contribution of SRAM, eDRAM, and STT-RAM banks

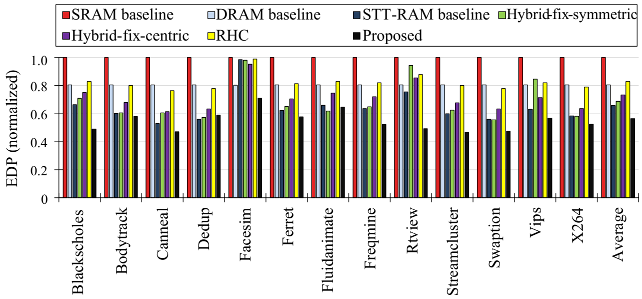

**FIGURE 10** Comparison of energy delay product (EDP) normalized to SRAM-baseline

Figure 10 compares the energy delay product (EDP) of the mentioned architectures normalized with the SRAM-baseline. As shown in this figure, owing to the higher leakage power consumption, the SRAM-baseline and DRAM-baseline demonstrate a higher EDP compared with STT-RAM and hybrid architectures. Given that hybrid-fix-symmetric and hybrid-fix-centric have static architectures and blind placement without considering the access behavior of workloads, there is the probability for a higher number of write accesses to STT-RAM banks compared with DRAM banks. Therefore, most of the workloads are not as efficient as the proposed approach. According to the lower static power consumption and larger density of STT-RAM and DRAM compared with SRAM, the hybrid fix methods and proposed methods work better than RHC. As shown in this figure, the proposed architecture improves the EDP by an average of 43%, 29%, and 31% compared with SRAM-baseline, DRAM-baseline, and RHC design, respectively.

Figure 11 compares the number of instructions per cycle (IPC) normalized with the SRAM-baseline. As shown in Table 1, the read latency of the eDRAM bank is more than the read latency of SRAM and STT-RAM. However, the write latency of STT-RAM is more than the write latency of the eDRAM and SRAM. Therefore, in read-intensive workloads such as *canneal*, the STT-RAM-baseline outperforms the DRAM-baseline. However, in write-intensive workloads such as *facesim*, the number of IPC of STT-RAM is lower than in other cases owing to the high write latency of STT-RAM. In the proposed architecture, there is an improvement of about 7% compared with DRAM-baseline and an average degradation of about 6% and 22% compared with STT-RAM-baseline and RHC architecture, respectively.

Figure 12 shows the optimal number of TSVs in the proposed architecture for each workload. As shown in this

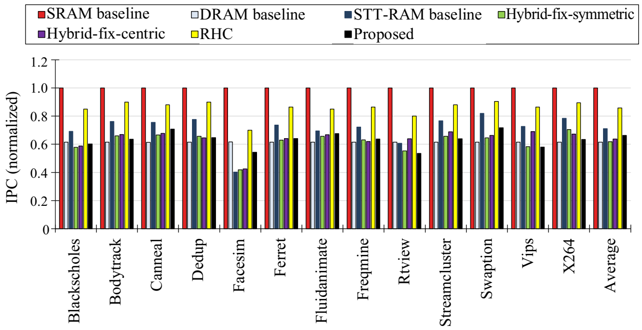

**FIGURE 11** Comparison of IPC normalized to SRAM-baseline

figure, the proposed reduced number of TSVs is about 60% on average.

We assumed the endurable maximum number of writes for SRAM, DRAM, STT-RAM, and PRAM based on Table 5 [30]. To evaluate the lifetime, we assumed that programs in a test program suite continuously run until one of the cache blocks exceeds the maximum number of endurable writes in each cache level. Given that the endurance of SRAM technology is the same as DRAM, as shown in Table 5, we reported only the results for SRAM in Figure 13. As shown in this figure, the life time of the SRAM-baseline architecture is an average of 1.8X compared with that of the proposed architecture because of the low endurance problem of NVM technologies.

Figure 14 shows the percentage of time that the memory layer of the 3D CMP spent at different temperature points while executing *canneal* and *facesim* benchmarks in each case. As shown in this figure, the proposed method always ensures that the memory layers of the 3D CMP are below the maximum temperature of 80°C. For the *canneal* benchmark as one of the computation intensive workloads, the SRAM-baseline, hybrid-fix-symmetric and hybrid-fix-centric baseline designs spend up to 58%, 32%, and 20% of the time above the maximum temperature, respectively. Figure 15 shows the temperature distribution of the memory layer while executing the *canneal* and *facesim* benchmarks. As shown in Figure 15A, the temperature of *canneal* is higher than that of *facesim* because it is a computation intensive workload; however, in *facesim*, the memory access distribution is practically uniform over banks.

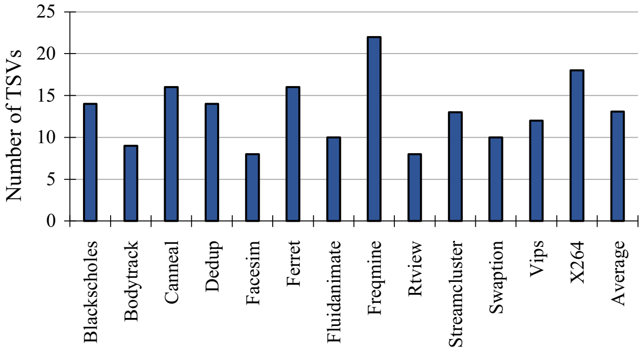

FIGURE 12 Number of TSVs in the proposed 3D-NoC for each workload

TABLE 5 Endurance maximum number of writes for various memory technologies

| Technology | SRAM | eDRAM | STT-RAM | PRAM |
|---|---|---|---|---|
| Endurance | $10^{16}$ | $10^{16}$ | $4 \times 10^{12}$ | $10^{9}$ |

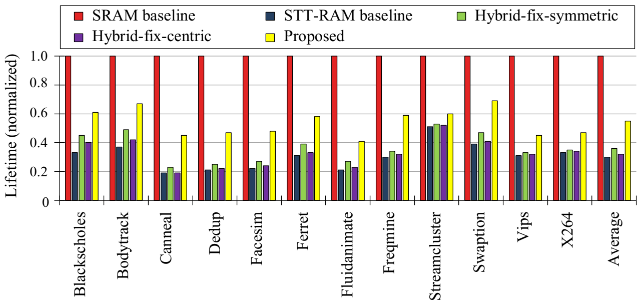

FIGURE 13 Comparison of life time normalized to SRAM-baseline

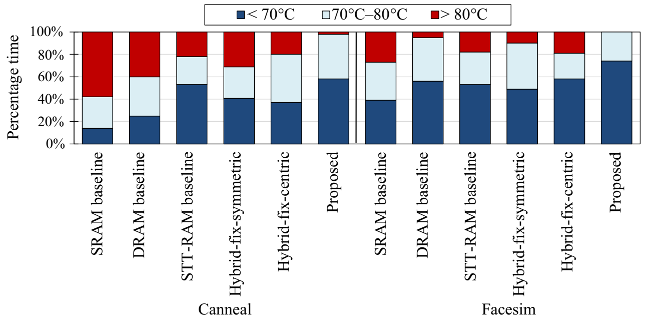

FIGURE 14 Comparison of percentage time spent on average by the memory layer at different temperature points for canneal and facesim

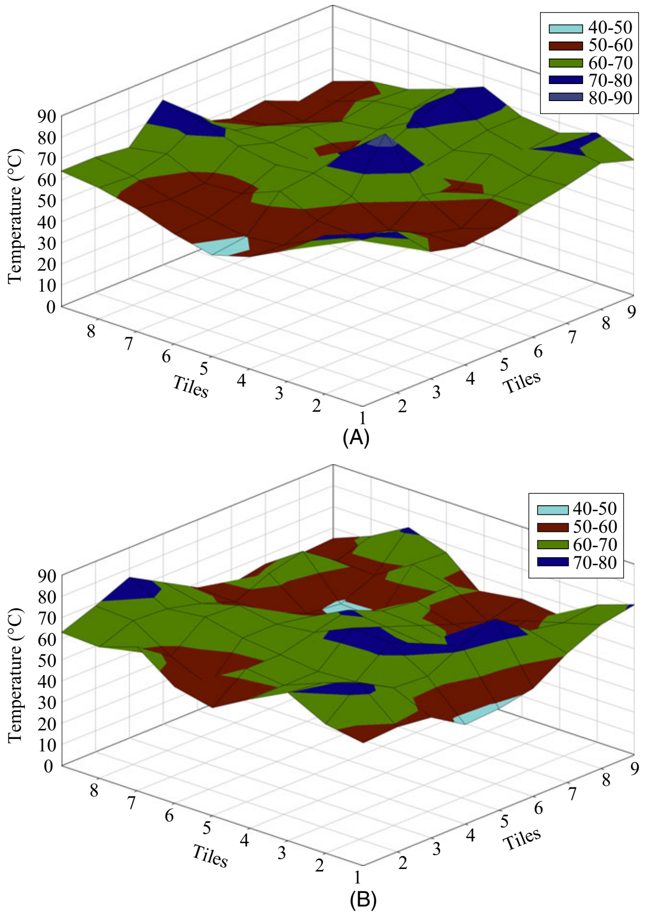

FIGURE 15 Thermal maps of the memory layer in a 64-core CMP under executing (A) canneal and (B) facesim benchmarks

In the next set of experiments, the effect of the $\varphi$ and $\gamma$ parameters in the optimization platform is tested. As can be expected, the energy savings increase with lower $\varphi$ values. The main reason for this behavior is the reduction in the use of SRAM banks in the proposed heterogeneous architecture and increasing the mapping of STT-RAM banks in this architecture. However, from a performance point of view, it is preferable to minimize the use of STT-RAM banks in architecting the memory layer. Figure 15 shows the improvement in the reliability and performance and the reduction in the energy effects of the $\varphi$ and $\gamma$ parameters. Because of space limitations, we illustrate three parts related to $\gamma = 0.1$, $\gamma = 0.5$, and $\gamma = 0.9$, and a large range for $\varphi$, as shown in Figure 16A, B, and C. As can be seen in this figure, when $\varphi$ is increased, the overall performance improvement increases owing to the increased usage of SRAM and DRAM banks.

The energy consumption, performance, and reliability are three important parameters in embedded system design. In the proposed platform, which can be a methodology for future embedded architecting, based on the assignment of the suitable values to $\varphi$ and $\lambda$ by the designer, one of the target parameters (energy, performance, or reliability) or the trade-off between them can be obtained. As the aim of this paper is to propose an energy-efficient architecture, in our experiment evaluation, we consider the value for $\varphi$ and $\gamma$, (0.5, 0.1) to enable a better performance compared with a lower value for $\varphi$. Hence, all energy, performance, and reliability values are normalized with respect to $\varphi = 0.5$ and $\gamma = 0.1$. Because of space limitations, we only present Figure 16 for the *canneal* multithreaded program. All of the test programs used in this paper were experimented, and in all of them, the trend in Figure 16 was observed.

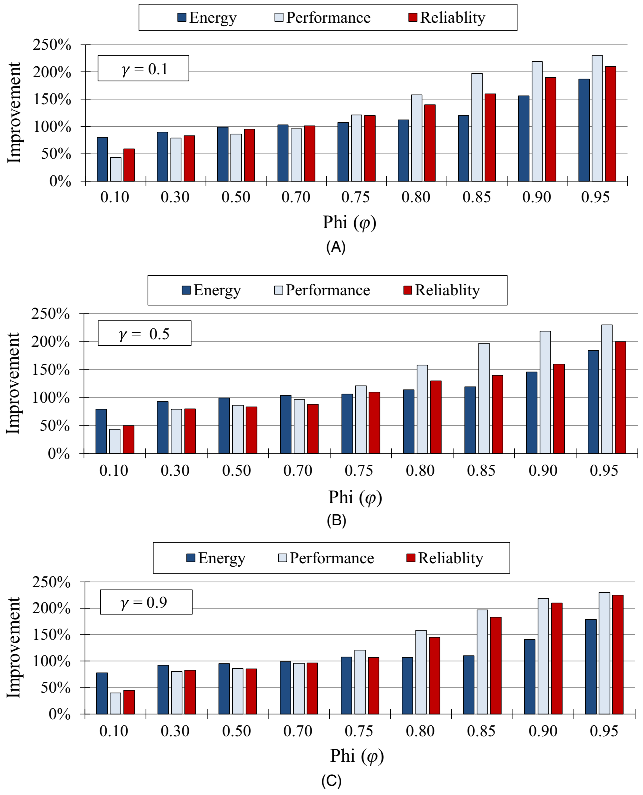

**FIGURE 16** Normalized performance-energy-reliability costs under the different $\varphi$ and $\gamma$ values for *canneal* program

## 5 | CONCLUSION

In this work, we proposed a convex optimization model to design an optimal heterogeneous uncore architecture for future many-core CMPs. Our proposed convex optimization-based model finds the optimal number and placement of different memory banks in the memory layer. Then, this model finds the optimal number and placement of TSVs in 3D CMPs to reduce performance degradation and minimize power consumption. Experimental results show that the proposed method improves the EDP by about 43% compared with the SRAM-baseline. Further, it improves the IPC of the 3D CMP by about 7% compared with the DRAM-baseline. Moreover, it has an IPC degradation of about 6% compared with the STT-RAM design.

## ORCID

*Aniseh Dorostkar* http://orcid.org/0000-0003-2596-0721

## REFERENCES

1. H. Tajik, H. Homayoun, and N. Dutt, VAWOM: Temperature and process variation aware wearout management in 3D multicore architecture, *Design Autom. Conf. (DAC)*, Austin, Texas, USA, 2013, pp. 1–8.
2. Z. Abbas and M. Olivieri, *Impact of technology scaling on leakage power in nano-scale bulk CMOS digital standard cells*, Microelectron. J. **45** (2014), no. 2, 179–195.
3. W. Wang and P. Mishra, *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** (2012), no. 5, 902–910.
4. H. Jeon, Y.-B. Kim, and M. Choi, *Standby leakage power reduction technique for nanoscale CMOS VLSI systems*, IEEE Trans. Instrum. Meas. **59** (2010), no. 5, 1127–1133.
5. P. Mishra, A. Muttreja, and N. K. Jha, FinFET circuit design, *Nanoelectronic Circuit Design*, N. K. Jha and D. Chen, Eds. Springer New York, New York, NY, USA, 2011, pp. 23–54.
6. O. Weber et al., Static and dynamic power management in 14 nm FDSOI technology, *Int. Conf. IC Design Tech.*, Leuven, Belgium, 2015, pp. 1–4.
7. B. Sriram, *Nanoscale thin-body MOSFET design and applications*, University of California, Berkeley, 2006.

8. H. Esmaeilzadeh et al., Dark silicon and the end of multicore scaling, *Int. Symp. Comput. Arch.*, San Jose, California, USA, June 4–8, 2011, pp. 365–376.
9. B. Raghunathan et al., Cherry-picking: Exploiting process variations in dark-silicon homogeneous chip multi-processors, *Design, Autom. Test Eur. Conf. Exh. (DATE)*, Grenoble, France, 2013, pp. 39–44.
10. J. Henkel et al., New trends in dark silicon, *Proc. Design Autom. Conf. (DAC)*, San Francisco, CA, USA, 2015, pp. 1–6.
11. A. Asad et al., *Optimization-based power and thermal management for dark silicon aware 3D chip multiprocessors using heterogeneous cache hierarchy*, Microprocess. Microsyst. **51** (2017) 76–98.
12. H. Bokhari et al., darkNoC: Designing energy-efficient network-on-chip with multi-Vt cells for dark silicon, *Design Autom. Conf. (DAC)*, San Francisco, CA, USA, 2014, pp. 1–6.
13. H. Bokhari et al., Malleable NoC: Dark silicon inspired adaptable network-on-chip, *Design, Autom. Test Eur. Conf. Exhi. (DATE)*, Dresden, Germany, 2015, pp. 1245–1248.
14. H. Jang et al., A hybrid buffer design with STT-MRAM for on-chip interconnects, *Int. Symp. Net. Chip (NoCS)*, Lyngby, Denmark, 2012, pp. 193–200.
15. J. Zhan et al., DimNoC: A dim silicon approach towards power-efficient on-chip network, *Design Autom. Conf. (DAC)*, San Francisco, CA, USA, 2015, pp. 1–6.
16. H. Lu et al., ShuttleNoC: Boosting on-chip communication efficiency by enabling localized power adaptation, *Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Chiba/Tokyo, Japan, Jan. 2015, pp. 142–147.
17. J. Zhan, Y. Xie, and G. Sun, NoC-sprinting: Interconnect for fine-grained sprinting in the dark silicon era, *Design Autom. Conf. (DAC)*, San Francisco, CA, USA, 2014, pp. 1–6.
18. L. Chen et al., Power punch: Towards non-blocking power-gating of NoC routers, *Int. Symp High Perfor. Comput. Arch. (HPCA)*, Burlingame, CA, USA, 2015, pp. 378–389.
19. J. Zhan et al., NoΔ: Leveraging delta compression for end-to-end memory access in NoC based multicores, *Asia South Pacific Design Autom. Conf. ASP-DAC*, Singapore, 2014, pp. 586–591.
20. J. Ahn, S. Yoo, and K. Choi, *Prediction hybrid cache: An energy-efficient STT-RAM cache architecture*, IEEE Trans. Comput. **65** (2016), 940–951.
21. C. Fu et al., Sleep-aware variable partitioning for energy-efficient hybrid PRAM and DRAM main memory, *Int. Symp. Low Power Elec. Design*, La Jolla, CA, USA, 2014, pp. 75–80.
22. S. Lee, K. Kang, and C.-M. Kyung, *Runtime thermal management for 3-D chip-multiprocessors with hybrid SRAM/MRAM L2 Cache*, IEEE Trans, Very Large Scale Integr. VLSI Syst. **23** (2015), 520–533.
23. I.-C. Lin and J.-N. Chiou, *High-endurance hybrid cache design in CMP architecture with cache partitioning and access-aware policies, IEEE Trans*, Very Large Scale Integr. VLSI Syst. **23** (2015), 2149–2161.
24. Z. Wang et al., Adaptive placement and migration policy for an STT-RAM-based hybrid cache, *Int. Symp. High Perfor. Comput. Arch. (HPCA)*, Orlando, FL, USA, 2014, pp. 13–24.
25. J. Ahn, S. Yoo, and K. Choi, DASCA: Dead write prediction assisted STT-RAM cache architecture, *Int. Symp. High Perfor. Comput. Arch. (HPCA)*, Orlando, FL, USA, 2014, pp. 25–36.
26. A. Valero et al., *Design of hybrid second-level caches*, IEEE Trans. Comput. **64** (2015), 1884–1897.
27. M. S. Haque et al., Accelerating non-volatile/hybrid processor cache design space exploration for application specific embedded systems, *Design Autom. Conf. (ASP-DAC)*, Chiba/Tokyo, Japan, 2015, pp. 435–440.
28. J. Meng and A. K. Coskun, Analysis and runtime management of 3D systems with stacked DRAM for boosting energy efficiency, *Design, Autom. Test Eur. Conf. Exhi. (DATE)*, 2012, Dresden, Germany, 2012, pp. 611–616.
29. Y.-T. Chen et al., Dynamically reconfigurable hybrid cache: An energy-efficient last-level cache design, *Design, Autom. Test Eur. Conf. Exhi. (DATE)*, Dresden, Germany, 2012, pp. 45–50.
30. M.-T. Chang et al., Technology comparison for large last-level caches (L3Cs): Low-leakage SRAM, low write-energy STT-RAM, and refresh-optimized eDRAM, *Int. Symp. High Perfor. Comput. Arch. (HPCA)*, Shenzhen, China, 2013, pp. 143–154.
31. C. Wilkerson et al., Reducing cache power with low-cost, multi-bit error-correcting codes, *Int. Symp. Comput. Arch. (ISCA)*, Saint-Malo, France, 2010, pp. 83–93.
32. A. K. Mishra et al., Architecting on-chip interconnects for stacked 3D STT-RAM caches in CMPs, *Int. Symp. Comput. Arch. (ISCA)*, San Jose, CA, USA, 2011, p. 69–80.
33. X. Dong and Y. Xie, System-level cost analysis and design exploration for three-dimensional integrated circuits (3D ICs), *Proc. Asia South Pacific Design Autom. Conf.*, Yokohama, Japan, 2009, pp. 234–241.
34. D. H. Woo et al., An optimized 3D-stacked memory architecture by exploiting excessive, high-density TSV bandwidth, *Int. Symp. High Perfor. Comput. Arch. (HPCA)*, Bangalore, India, 2010, pp. 1–12.
35. K. Manna et al., TSV placement and core mapping for 3D mesh based network-on-chip design using extended Kernighan-Lin partitioning, *IEEE Comput. Soc. Symp. VLSI (ISVLSI)*, Montpellier, France, 2015, pp. 392–397.
36. C.-H. Cheng, C.-H. Kuo, and S.-H. Huang, TSV number minimization using alternative paths, *Int. Conf. IC Design Tech.*, Ho Chi Minh, Vietnam, 2011, pp. 1–4.
37. B. Lee and T. Kim, Algorithms for TSV resource sharing and optimization in designing 3D stacked ICs, *Integr. VLSI J.* **47** (2014) 184–194.
38. Z. Diao et al., *Spin-transfer torque switching in magnetic tunnel junctions and spin-transfer torque random access memory*, J. Phys. Condens. Matter **19** (2007) 165–209.
39. S. P. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge, UK; New York: Cambridge University Press, 2004.
40. M. Grant, S. Boyd, and Y. Ye, Matlab software for disciplined convex programming [Online]. Available: www.stanford.edu/boyd/cvx/.
41. N. Binkert et al., The gem5 simulator, ACM SIGARCH Comput. Archit. News **39** (2011) 1–7.
42. V. Catania et al., Noxim: An open, extensible and cycle-accurate network on chip simulator, *Int. Conf. Applicat. -specific Syst., Architectures Processors (ASAP)*, Toronto, Canada, 2015, pp. 162–163.
43. S. Li et al., McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures, *Proc. Int. Symp. Microarchitecture*, New York, USA, 2009, pp. 469–480.

44. N. Muralimanohar, R. Balasubramonian, and N. P. Jouppi, CACTI 6.0: A tool to model large caches, *HP Lab.*, 2009, pp. 22–31.
45. A. B. Kahng, B. Lin, and S. Nath, *ORION3.0: A comprehensive NoC router estimation tool*, IEEE Embed. Syst. Lett. **7** (2015) 41–45.
46. X. Dong et al., NVSim: A circuit-level performance, energy, and area model for emerging non-volatile memory, *Emerging Memory Technologies*, Y. Xie, Ed. Springer New York, New York, NY, USA, 2014, pp. 15–50.
47. W. Huang et al., *HotSpot: A compact thermal modeling methodology for early-stage VLSI design*, IEEE Trans. Very Large Scale Integr. VLSI Syst. **14** (2006) 501–513.
48. M. Gebhart et al., Running PARSEC 2.1 on M5, *Univ. Tex. Austin Dep. Comput. Sci. Tech Rep*, 2009.

## AUTHOR BIOGRAPHIES

**Aniseh Dorostkar** received her BSc in computer engineering from Shahid Bahonar University of Kerman, Kerman, Iran, in 2012. She is currently an MSc student at Iran University of Science & Technology, Tehran, Iran. She is currently working on optimizing the placement of uncore components that contain hybrid memory hierarchy using emerging nonvolatile memory and network-on-chip architecture for future chip-multiprocessors.

**Arghavan Asad** received her MSc degree in computer architecture from Iran University of Science & Technology (IUST), Tehran, Iran, in 2010. She has been a PhD candidate at IUST since 2011. She is currently working on hybrid memory hierarchy design using emerging nonvolatile memory technologies for future chip-multiprocessors. Her research topics include the impact of uncore components' power consumption in dark silicon-aware multi/many-core systems.

**Mahmood Fathy** received his BSc degree in electronics from Iran University of Science & Technology (IUST), Tehran, Iran, in 1984, the MSc degree in computer architecture from Bradford University, West Yorkshire, UK, in 1987, and the PhD degree in image processing computer architecture from the University of Manchester, UK, in 1991. Since 1991, he has been an academic member with the Department of Computer Engineering, IUST.

**Mohammad Reza Jahed-Motlagh** received his BS degree in electrical and electronic engineering from Sharif University of Technology, Tehran, Iran, in 1978, the MS degree in control engineering in 1986, and the PhD degree in control engineering in 1990, both from Bradford University, West Yorkshire, UK. Since 1991, he has been an associate professor with the Department of Computer Engineering, Iran University of Science & Technology.

**Farah Mohammadi** received the BSc and the MSc degrees in telecommunication engineering from Iran University of Science & Technology in 1988 and 1991, respectively, and the PhD degree in electronic engineering from the Institute d'Electronique et Microelectronique du Nord at University of Science and Technology of Lille, France, in 1998. Since 2003, she has been an associate professor with the Electrical and Computer Engineering Department at Ryerson University.