Zhao YL, Yang JL, Zhao WS *et al.* Power supply noise aware task scheduling on homogeneous 3D MPSoCs considering the thermal constraint. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 33(5): 966–983 Sept. 2018. DOI 10.1007/s11390-018-1868-6

Power Supply Noise Aware Task Scheduling on Homogeneous 3D MPSoCs Considering the Thermal Constraint

Ying-Lin Zhao[1,2,3], *Student Member*, *IEEE*, Jian-Lei Yang[1,4], *Member*, *IEEE*
Wei-Sheng Zhao[1,2,3], *Senior Member*, *IEEE*, Aida Todri-Sanial[5,6,*], *Member*, *IEEE*
and Yuan-Qing Cheng[2,3,*], *Member*, *CCF*, *ACM*, *IEEE*

[1] *Fert Beijing Research Institute, Beijing Advanced Innovation Center for Big Data and Brain Computing Beijing 100191, China*

[2] *School of Electronic and Information Engineering, Beihang University, Beijing 100191, China*

[3] *Qingdao Research Institute, Beihang University, Qingdao 266041, China*

[4] *School of Computer Science and Engineering, Beihang University, Beijing 100191, China*

[5] *Laboratory of Informatics, Robotics and Microelectronics, University of Montpellier, Montpellier 34095, France*

[6] *National Center for Scientific Research, Montpellier 34095, France*

E-mail: wssdzyl@sina.com; {jianlei, weisheng.zhao}@buaa.edu.cn; aida.todri@lirmm.fr; yuanqing@ieee.org

Received April 7, 2017; revised May 9, 2018.

Abstract Thanks to the emerging 3D integration technology, The multiprocessor system on chips (MPSoCs) can now integrate more IP cores on chip with improved energy efficiency. However, several severe challenges also rise up for 3D ICs due to the die-stacking architecture. Among them, power supply noise becomes a big concern. In the paper, we investigate power supply noise (PSN) interactions among different cores and tiers and show that PSN variations largely depend on task assignments. On the other hand, high integration density incurs a severe thermal issue on 3D ICs. In the paper, we propose a novel task scheduling framework considering both the PSN and the thermal issue. It mainly consists of three parts. First, we extract current stimuli of running tasks by analyzing their power traces derived from architecture level simulations. Second, we develop an efficient power delivery network (PDN) solver to evaluate PSN magnitudes efficiently. Third, we propose a heuristic algorithm to solve the formulated task scheduling problem. Compared with the state-of-the-art task assignment algorithm, the proposed method can reduce PSN by 12% on a $2 \times 2 \times 2$ 3D MPSoCs and by 14% on a $3 \times 3 \times 3$ 3D MPSoCs. The end-to-end task execution time also improves as much as 5.5% and 7.8% respectively due to the suppressed PSN.

Keywords power supply noise (PSN), power delivery network (PDN), task scheduling algorithm, temperature, 3D MPSoCs

1 Introduction

With the exponential increase of transistor count on-chip and the insatiable demand of high performance, the power density on-chip increases dramatically. "Power Wall" issue becomes a serious concern for modern VLSI (very large scale integrated circuits) designers[1]. The stringent power constraint makes it extremely difficult to squeeze extra performance by simply scaling the clock frequency. As a result, MPSoCs (multi-processor system on chips) emerge as an effective technique to continue Moore's law, especially in the embedded system domain[2] because of their better energy efficiency. Moreover, the design methodology of MPSoCs facilitates the integration of various IP cores from different vendors, which reduces the design complexity and accelerates time-to-market greatly.

Recently, the emerging 3D integration technology

Regular Paper

This work was supported by the National Natural Science Foundation of China under Grant Nos. 61401008 and 61602022, and the State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences, under Grant No. CARCH201602.

[*] Corresponding Author

has been adopted in the MPSoC design (i.e., 3D MPSoCs) to approach higher integration density and more functionalities than the 2D counterpart[3]. By stacking several thinned tiers vertically, data communication bandwidth and power consumption can be improved greatly with the aid of through-silicon-vias (TSVs). However, the 3D MPSoC architecture also presents several challenges due to the die-stacking structure. Among them, signal integrity is a critical issue for chip reliability. Since IP cores residing on different tiers commonly share a power delivery network[4], the activities on one core may propagate to cores in its vicinity by the shared power delivery network. The shrinking power supply voltage and increasing magnitudes of current transients make the power supply noise (PSN) more significant on 3D MPSoCs compared with their 2D counterparts. To constrain the PSN magnitude, many effective methods were proposed from different optimization perspectives. Todri *et al.* investigated the PSN interactions on a multi-processor platform, and derived several guidelines for workload assignments to suppress PSN[5]. Wang *et al.* optimized the task mapping and scheduling to minimize PSN induced by power gating[6]. Although their work provides insights of the PSN effect in traditional MPSoCs, the proposed method cannot be simply extended to 3D MPSoCs due to the following reasons. First, each tier manifests different characteristic impedance and may experience different voltage drop[7]. Second, PSN can propagate from one tier to other tiers through (temperature safety value) TSVs, which introduces non-negligible PSN couplings. As a result, the heterogeneity of die-stacking structure and complicated PSN interactions require a novel hardware-software codesign methodology for 3D MPSoCs to suppress PSN effectively.

On the other hand, the reliability of 3D MPSoCs is also threatened by the thermal issue[8]. Stacking several tiers together presents a severe challenge on thermal dissipation. High temperature can aggravate NBTI (negative bias temperature instability) and the hot carrier injection effect[9]. Since upper tiers are thinned before integration to facilitate die stacking, the vertical thermal correlation dominates the heat dissipation in 3D MPSoCs, which makes the thermal modeling of 3D MPSoCs different from those of 2D counterparts.

In the paper, we explore to deal with the above two reliability concerns from the task assignment perspective for hard real-time applications on homogeneous 3D MPSoCs. In the hard read-time application, the task graph extracted is predetermined and the deadline must be guaranteed[10]. With an illustrative example, we show that different task mapping and scheduling schemes may result in significantly different PSN distributions, which implies the large optimization potential by PSN-aware task scheduling. However, we observe that the conventional thermal aware task assignment scheme may cause severe PSN, which motivates our work. In order to optimize task scheduling with both thermal and PSN concerns, we propose a framework to evaluate PSN and thermal distributions effectively. First, we extract representative power traces from real applications, and convert them to current stimuli traces instead of the traditional triangular or trapezoidal current stimuli to make the PSN calculation more accurate and realistic. Second, we construct a power delivery network (PDN) model of 3D MPSoCs for PSN calculations, and develop a fast power grid solver to facilitate PSN analyses. Based on these modeling techniques, we formulate the task scheduling optimization as a linear programming problem, which minimizes the PSN with the peak temperature constraint. Then, a list-scheduling based heuristic algorithm is proposed to obtain the optimal task scheduling solution. Extensive simulation results show that the proposed algorithm can reduce PSN by 12% on a $2 \times 2 \times 2$ MPSoC and 14% on a $3 \times 3 \times 3$ MPSoC platform compared with the state-of-the-art thermal aware task scheduling scheme[11]. Additionally, task execution time can be improved by 5.5% and 7.8% respectively due to the reduced PSN.

Our main contributions are listed as follows.

- We propose an efficient architecture-level PSN simulation framework to fill the gap between circuit-level PSN simulation and architecture-level task scheduling for 3D MPSoCs.
- Our another contribution is considering both power supply noise and temperature to effectively avoid hotspot during the PSN optimization for 3D MPSoCs.
- We propose a novel list-scheduling based heuristic algorithm for the task scheduling problem on homogeneous 3D MPSoCs with both thermal and PSN considerations. Extensive simulation results show that the proposed algorithm can reduce PSN by 12% on a $2{\times}2{\times}2$ MPSoC and 14% on a $3{\times}3{\times}3$ MPSoC compared with the state-of-the-art thermal aware task scheduling scheme[11]. Additionally, task execution time can be improved by 5.5% and 7.8% respectively due to the reduced PSN.

The rest of the paper is organized as follows. Section 2 presents the preliminaries of 3D MPSoC PDN and thermal modeling techniques. Task graph is also introduced in this section. An illustrative example is presented in Section 3 to motivate our work. Section 4 formulates the PSN aware task scheduling problem with the thermal constraint and proposes a framework to solve it. The computing complexity analysis of the proposed algorithm is given in this section as well. Section 5 shows the experimental results by running our task scheduling algorithm extensively on different scales of 3D MPSoCs. Section 6 presents related work and Section 7 concludes the paper.

2 Preliminaries on Power Delivery Network and Thermal Modeling Techniques of 3D MPSoCs

For the ease of understanding, some necessary background knowledge is introduced as follows.

2.1 Power Delivery Network Modeling and PSN Evaluation for 3D MPSoCs

In general, a chip power delivery network consists of off-chip and on-chip power delivery interconnects as shown in Fig.1. A VRM (voltage regulator module) converts the high voltage (e.g., 2.5 V) to the chip operating voltage (e.g., 1 V). Then the supply voltage goes through the motherboard, P/G pins and on chip interconnects to drive transistors on chip. The PDN design needs to be determined at the early stage of chip design as any PDN change in the later phase of design flow may introduce expensive design iterations. Therefore, it is imperative to predict the supply voltage variations accurately to evaluate its impact on-chip performance and reliability as early as possible. As suggested in [12], it is necessary to consider both package and on-chip power delivery network for accurate PSN prediction. Traditionally, the package is modeled considering its inductive effect while on-chip interconnects are modeled considering their resistive and capacitive effects. However, as the working frequency increases, on-chip inductance can no longer be ignored[13]. In the paper, we model both the package and on-chip PDNs to capture PSN accurately.

We assume the on-chip PDN is in the mesh topology and shared by all cores on-chip as similar to [14]. The power and grid lines interleave with each other in one metal layer and are orthogonally aligned on different layers for the PSN reduction. P/G interconnects on different layers are connected by vias and deliver the power from global metal layers to local metal layers. A simplified model of a typical mesh PDN for homogeneous 3D MPSoC is shown in Fig.2. As shown in the figure, current is delivered from the package to the on-chip PDN through P/G C4 bumps, and the power supplies of upper tiers are provided from the bottom tier through P/G TSVs similar to the structure mentioned in [15].

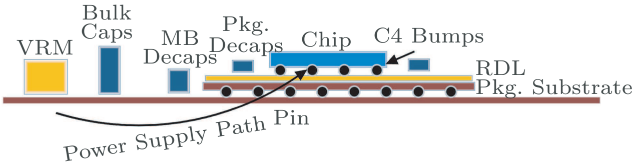


Fig.1. Power delivery system including both off-chip and on-chip components.

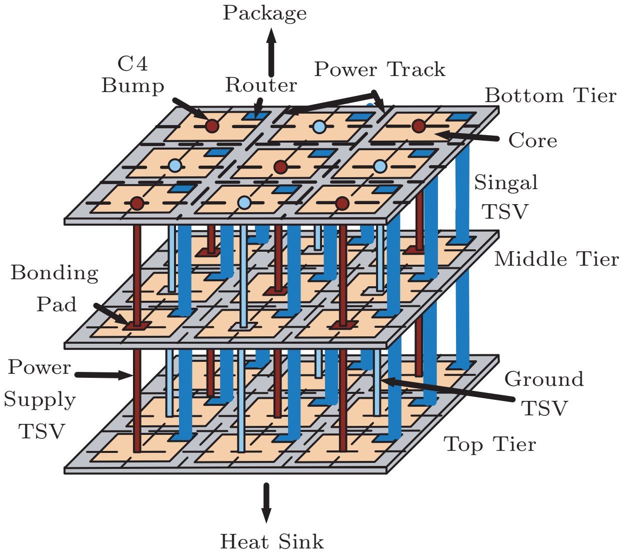


Fig.2. On chip mesh grid power delivery network of homogeneous 3D MPSoCs.

Then, PSN can be calculated as follows[16]:

$$V_{\rm pnoise} = \int_{t_{\rm s}}^{t_{\rm e}} \max\{V_{\rm dd} - V_{\rm p}, 0\}{\rm d}t/(t_{\rm e} - t_{\rm s}),$$

$$V_{\rm gnoise} = \int_{t_{\rm s}}^{t_{\rm e}} \max\{V_{\rm ss}, 0\}{\rm d}t/(t_{\rm e} - t_{\rm s}),$$

$$V_{\rm noise} = V_{\rm pnoise} + V_{\rm gnoise},$$

where $t_{\rm s}$ is the start of the timing window for the PSN calculation, and $t_{\rm e}$ is the end of the timing window. $V_{\rm dd}$ is the normal power supply voltage. $V_{\rm p}$ is the real voltage of the power grid node under investigation, and $V_{\rm ss}$ is the ground bounce of the corresponding ground grid node. Using the above equations, we can calculate the PSN magnitude of each P/G grid node considering both voltage drop ($V_{\rm pnoise}$) and ground bounce ($V_{\rm gnoise}$).

2.2 Thermal Modeling Techniques for 3D MPSoCs

Since the thermal issue is another important concern for 3D ICs, it is imperative to consider the thermal constraint during workload assignment optimization for 3D MPSoCs. In order to capture on-chip thermal distribution, an accurate thermal model is required. Thermal modeling techniques for 3D ICs can be classified into several categories.

- Solve the heat flow equation directly to get the temperature distribution on-chip by the finite element method or finite differential method[17-18].
- Utilize duality between the thermal and the electrical properties. The thermal conductivity can be modeled as the resistance in the equivalent electrical circuit. The temperature difference corresponds to the voltage difference. The power consumption can be modeled as the current source. Then, the temperature of each node can be solved by circuit analyses[19].
- Utilize on-chip power information such as the power gradient, to roughly estimate the temperature variations among functional blocks or cores[20].

The first method can obtain the accurate solution, but the computing complexity is too high to be adopted for the large scale of MPSoCs. The last one is the fastest method but it can only be used for qualitative analyses lacking of enough accuracy. The second method listed above makes a good trade-off between the accuracy and the efficiency. The widespread used thermal simulation tool Hotspot[21] is based on the second method and we will use it to evaluate the thermal distribution in this paper.

2.3 Introduction to Task Graph

In the paper, we assume that the running application can be split into a number of tasks with specified timing dependencies and constraints. Then, the application can be represented by a directed acyclic task graph, which is widely used for task scheduling research[11]. A task graph example is shown in Fig.3. In the figure, a node in the task graph denotes a task of the running application, and an edge denotes the task running priority and the running dependency. For example, task S_2 can only execute after S_1 finishes. The weight of the node (i.e., t_i) denotes the execution time of the task. For real-time applications, each task node has the deadline constraint and the task execution cannot violate it.

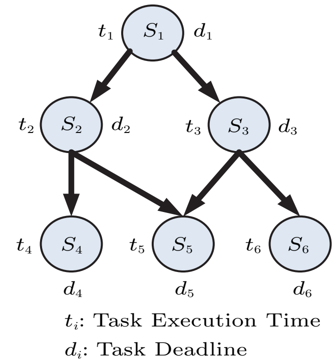


Fig.3. Illustrative task graph example.

3 Motivation

In this section, we investigate PSN distributions caused by different task allocation schemes. For the completeness of the paper, we briefly introduce the task allocation schemes used in our investigation. The first scheme is based on the conventional list scheduling algorithm[10] as shown in Fig.4(a). According to the task execution deadline, our scheduling algorithm calculates the earliest start time (EST) and the latest start time (LST) of each task. Then, all tasks are sorted with their mobilities (i.e., the difference of LST and EST). We define the schedule point as the time point when there is any task ending execution or beginning to run. At each schedule point, the scheduler assigns a ready task to an idle core randomly. When all ready tasks are allocated, the time will advance to the next scheduling point. If all cores are busy such that no ready task can be allocated at current scheduling point, the time will advance to the next one when at least one core becomes idle. The second scheme is the state-of-the-art thermal aware task scheduling algorithm proposed by [11] as shown in Fig.4(b). The scheduler evaluates the thermal distribution on-chip at each scheduling point for each possible task assignment, and the scheduler chooses the best candidate position approaching the lowest peak temperature. The third scheme is similar to the thermal aware scheduling but evaluates PSN instead. At each schedule point, the scheduler will assign the ready task to the available position approaching minimal PSN (the PSN evaluation procedure will be detailed in Section 4).

To illustrate scheduling results using the three task assignment schemes, we use TGFF[22] to generate a task graph and schedule it on a 2 × 2 × 2 3D MPSoC using these schemes. The task graph has 20 nodes and 30 edges. The task characterizations, detailed simulation setup, PDN and thermal model parameters are

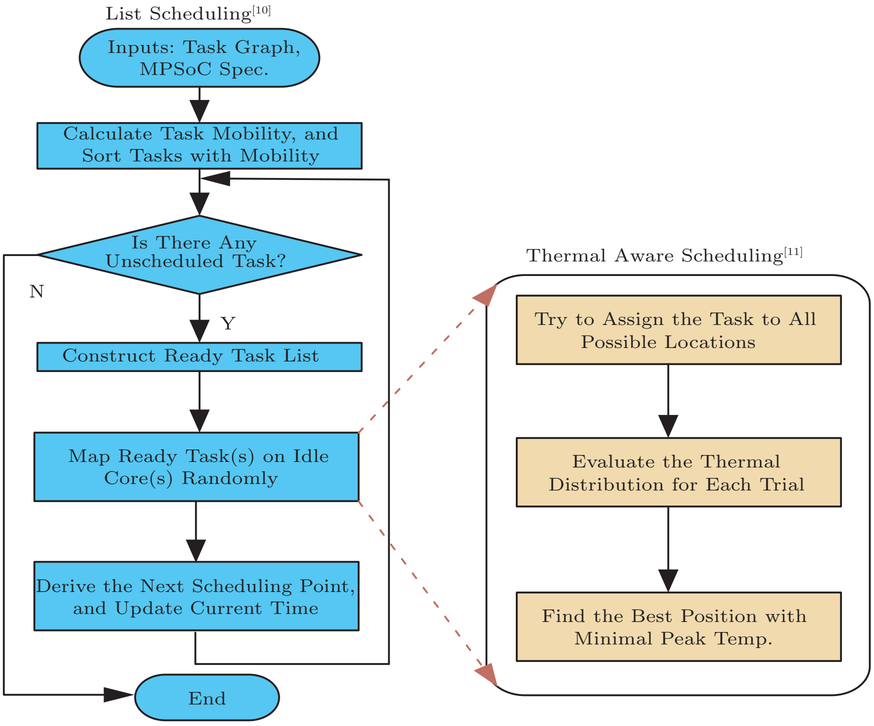

List Scheduling[10]
Inputs: Task Graph, MPSoC Spec.
Calculate Task Mobility, and Sort Tasks with Mobility
Is There Any Unscheduled Task?
N
Y
Construct Ready Task List
Map Ready Task(s) on Idle Core(s) Randomly
Derive the Next Scheduling Point, and Update Current Time
End
Thermal Aware Scheduling[11]
Try to Assign the Task to All Possible Locations
Evaluate the Thermal Distribution for Each Trial
Find the Best Position with Minimal Peak Temp.

Fig.4. (a) Flow chart of the list scheduling scheme[10]. (b) Thermal aware scheduling algorithm[11].

presented later. Table 1 indicates the comparisons among three algorithms (conventional list scheduling algorithm, i.e., the "random" algorithm in the table, thermal aware scheduling algorithm and PSN-aware scheduling algorithm without thermal consideration) in terms of temperature and PSN.

Table 1. PSN and Temperature Comparisons of Three Task Assignment Schemes

	Random	T-Aware	PSN-Aware
PSN (mV)	147.6	174.7	143.2
Temp. (K)	310.3	307.5	309.3

Note: T-aware denotes the thermal aware task scheduling scheme and PSN-aware denotes the PSN aware task scheduling scheme. Both PSN and temperature (temp.) values are the peak values among all cores.

As shown in Table 1, the random task assignment scheme can be neither thermal nor PSN friendly. Moreover, we can observe that the thermal aware scheduling scheme may generate severe peak PSN (increasing 18% compared with that of the PSN-aware case). In contrast, PSN aware task scheduling scheme may incur higher peak temperature than the thermal aware scheduling algorithm.

To examine the conflict between the thermal aware scheduling scheme and the PSN aware scheduling scheme, we investigate the thermal and PSN distributions of each tier when running a single task on different tiers as shown in Fig.5. In case 1, the task is assigned to the bottom tier. In case 2, it is assigned to the middle tier. In case 3, it is assigned to the top layer. In Fig.5, we assume that the bottom tier is attached to the PCB board and provides power supply for upper tiers while the top tier is attached to the heatsink for thermal dissipation. Both the peak temperature and the PSN magnitude of each tier are shown in Table 2.

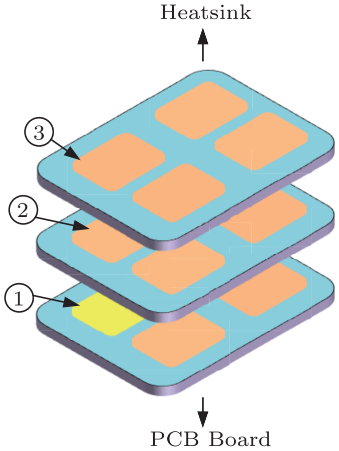


Fig.5. Running a single task on different tiers for thermal and PSN comparisons: ① running the task on the bottom tier; ② running the task on the middle tier; ③ running the task on the top tier.

Table 2. PSN and Temperature Comparisons of Three Task Running Scenarios

	Case 1			Case 2			Case 3		
	Bottom	Middle	Top	Bottom	Middle	Top	Bottom	Middle	Top
PSN (mV)	**54.8**	46.6	46.50	42.40	**59.80**	49.30	41.10	48.2	**63.40**
Temperature (K)	**304.8**	303.5	302.77	**303.67**	303.51	302.78	**303.26**	303.1	302.79

Note: Both PSN and temperature values are the peak values among all cores on the same tier.

As shown in Table 2, the peak PSN magnitude is the largest when the task is assigned to the top tier (63.4 mV). The peak PSN magnitude is the smallest when the task is assigned to the bottom tier (54.8 mV). From the power supply noise suppression perspective, it is preferable to schedule the task near the bottom tier as much as possible. Since the power supply has to go through the bottom and the middle tier to reach PDN on the top tier, the top tier suffers from more severe PSN than other tiers[23]. On the other hand, when the task is allocated on the bottom tier, the peak temperature is the highest (304.8 K) among all three cases. Therefore, from the heat dissipation perspective, it is better to assign the task in upper tiers which are closer to the heat sink. As a result, there is conflict between the PSN-aware task scheduling and the thermal-aware task scheduling, which coincides with [19]. The above analyses imply that it is necessary to consider both thermal and PSN issues during the task scheduling to ensure the reliability and availability of 3D MPSoCs.

4 Proposed Framework for Task Scheduling on 3D MPSoCs Considering Both Thermal and PSN Issues

In order to evaluate the temperature and PSN distributions, we need to build the thermal and electrical PDN models of 3D MPSoCs. Traditionally, considering the large scale of on-chip power grid, stimuli of PDN are usually simplified as current sources with some regular waveforms such as the triangular or trapezoidal waveforms. Although this assumption reduces the complexity of solving PDN equations, it sacrifices simulation accuracy and cannot reflect the real activities and different characteristics of various running tasks. To illustrate this point, we perform the power supply noise simulations with two different methods. In the first one, we extract the power traces by the architecture level simulation, and convert them to the stimuli to the PDN for PSN simulations. In the second method, the PDN stimuli waveforms are simplified as triangular waveforms, and the average power consumption is set as the same with that of the first method. The experimental results on a single core show that the PSN error of the first method can be as large as 51.35% (please refer to Section 5 for the experimental setup). Therefore, it is imperative to consider application running characteristics derived from the power traces for the accurate PSN evaluation.

In this paper, we propose a power trace based architecture level PSN evaluation framework. The advantages of this framework are as follows. First, the architecture level simulation can improve the PSN calculation efficiency and save significant computing overhead. Second, various characteristics of tasks can be captured by power traces extracted. Therefore, interactions of tasks running simultaneously can be captured by feeding power traces to the PDN model.

Our proposed framework can be divided into three steps as shown in Fig.6.

- First, we input the core architecture, technology parameters to set up the architecture-level simulator. Then, power traces of different tasks are extracted with simulations.
- Second, power traces are converted into current traces and fed into the 3D MPSoC PDN model for PSN calculations. We develop an efficient PDN solver to accelerate the PSN evaluations instead of the time-consuming HSpice simulations.
- At last, we formulate the task scheduling problem and propose a heuristic algorithm to solve it. The task graph and 3D MPSoC architecture specifications are inputs of the task scheduler. During the task scheduling, thermal simulations and PSN calculations are performed to find the optimal solution.

In the following, we will detail each step of the framework.

4.1 Power Trace Based Task Characterization

Tasks running simultaneously on the 3D MPSoC have different characteristics, such as peak current magnitudes, running frequencies, and power consumptions. Furthermore, their behaviors may affect one another

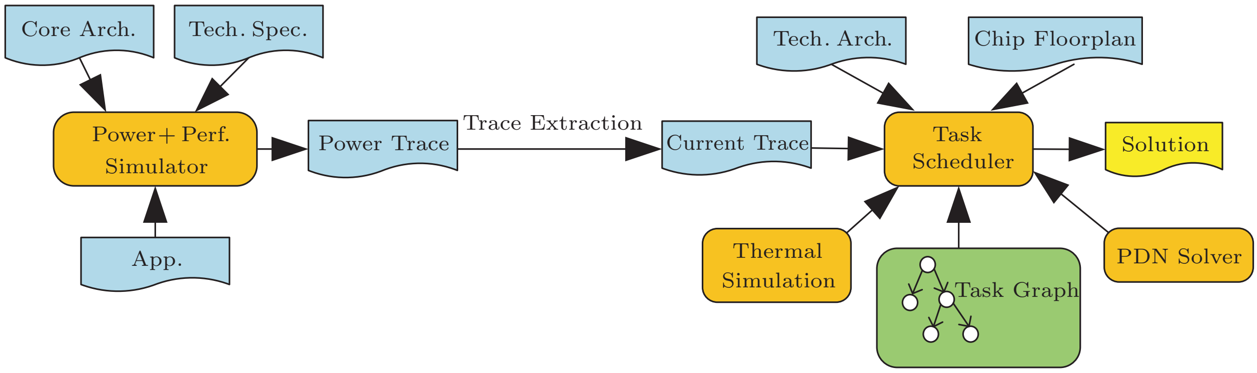


Fig.6. Proposed task scheduling framework considering both thermal and PSN issues.

when running at the same time. Individual workload[1] or workload combinations may cause different PDN responses due to PSN propagation and inter-coupling. In the paper, we use the architecture level simulation to get the power consumption information and then derive waveforms of current stimuli, which can be integrated with the PDN model conveniently for the PSN analysis. Although there is another simulation tool Voltspot available for power trace driven PSN simulation[24], it could not scale very well for large-scale 3D MPSoCs and incurs unacceptable running time overhead. Therefore, we propose the "single core power trace extraction + custom PDN solver" for the PSN evaluation. We choose Wattch[25] for power trace extraction since it can get a satisfied trade-off between the running speed and the accuracy for a single core simulation. Note that our proposed framework does not depend on the specific simulator.

Then, the generated power trace can be converted to current traces and integrated with the PDN model. Here, we will take a SPEC2000 application "ammp" as an example to illustrate the workload characterization process as shown in Fig.7.

Considering the large scale of power grid, it cannot afford to apply the entire power trace for PSN calculations, which will incur unacceptable running time overhead. Therefore, it is necessary to capture the representative segment of the extracted power trace. We use SimPoint[26] to identify several different running phases of "ammp". For each identified running phase, we fast forward 1 million instructions for the cache warm-up and perform detailed timing simulation for another 1 million instructions (refer to Section 5 for our architecture simulation setup). Then, we extract 10 000 cycle long segment from the generated power trace, which can capture the peak power consumption. Although the power consumption has close relationship with power supply noise, we observe that the maximum PSN may not exactly happen at the same time when the peak power appears due to the propagation delay of PDN. To capture the worst PSN exactly, we take the generated power trace segment as input and use Voltspot[24] to identify the interval including the worst PSN[2]. Then, we back-annotate the interval to the original power trace segment for the PDN stimuli extraction. Consequently, the extracted stimuli are inserted at current source insertion points of the PDN model for PSN calculations, which will be detailed in Subsection 4.2.

4.2 Development of the PDN Model and the PDN Solver

Since we focus on the PSN at the core-level and emphasize the power supply interactions between different cores, it is enough to only consider the global power grid, which takes each core as one block as in [5]. Considering the on-chip PDN, each interconnect segment is modeled with its distributed RLC (resistance, inductance, and capacitance) parasitics. Decoupling capacitance, including intrinsic and external decoupling capacitance, is assumed to be distributed evenly across the core, and connected in parallel with the current source derived from the power trace. The power trace is generated based on the switching activity of the task running on cores. Note that both power and ground grids are considered for PSN calculations. Power/ground grids on each layer are interconnected by

[1] In the paper, we use workload and task interchangeably.

[2] Since the Voltspot-based PSN simulation is only performed on a single task in this paper, the run-time overhead of Voltspot is acceptable.

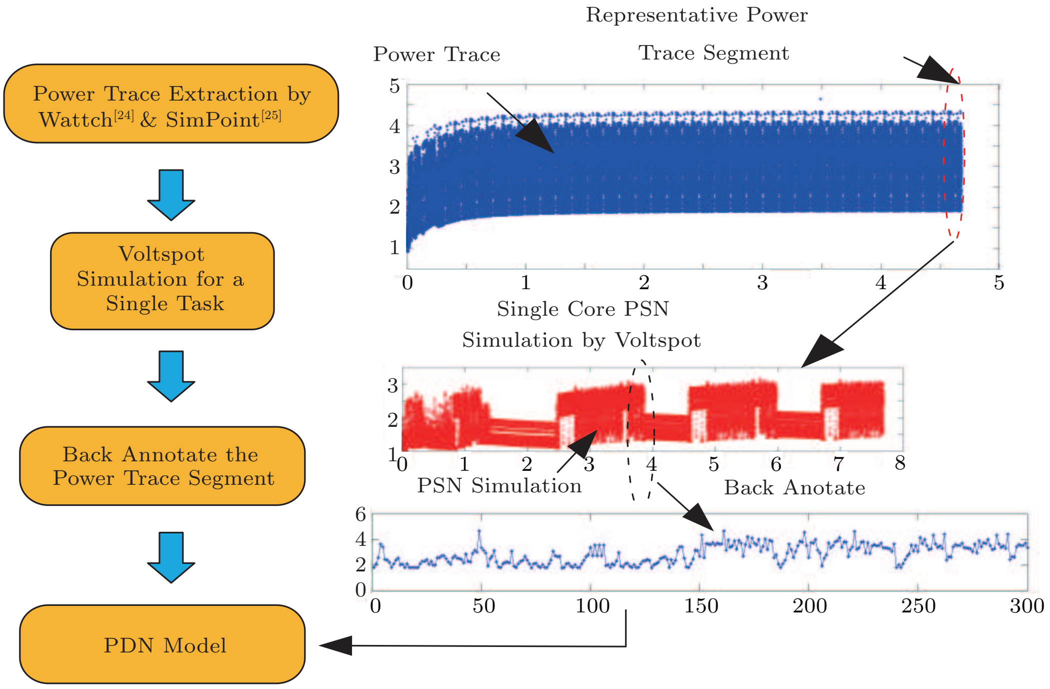

Power Trace Extraction by Watttch[24] & SimPoint[25]
Voltspot Simulation for a Single Task
Back Annotate the Power Trace Segment
PDN Model
Power Trace
Representative Power Trace Segment
5
4
3
2
1
0
1
2
3
4
5
Single Core PSN Simulation by Voltspot
3
2
1
0
1
2
3
4
5
6
7
8
PSN Simulation
Back Anotate
6
4
2
0
0
50
100
150
200
250
300

Fig.7. Illustration of the workload characterization process.

TSVs, which are modeled as resistors and inductors. C4 bumps distribute on the bottom layer to connect the on-chip PDN to the off-chip PDN. Fig.8 illustrates the on-chip PDN model[27] while the off-chip model is derived from [28].

In contemporary VLSI design, the power grid mesh can easily contain millions of nodes for the PSN calculation. It imposes a great challenge on power grid analysis. As the traditional simulation using HSpice can no longer be applied for such a very large-scale problem, many efficient numerical solving methods have been proposed considering the regularity of the power grid structure, such as multi-grid method[29-30], preconditioned conjugated gradient method[31].

In order to reduce the simulation complexity, we construct a power grid hierarchy, i.e., the power grid in one core consists of several base grids while each base grid is composed of several base cells as in [5]. Although the PDN model is largely simplified by the hierarchical structure, it would be still very time-consuming for PSN computations in task allocation and scheduling as the scheduling algorithm usually involves many trial-and-fail iterations to obtain the final solution. Therefore, we develop an efficient PDN solver to accelerate PSN analysis based on the modified nodal analysis (MNA).

To verify the effectiveness of the PDN solver, we generate the netlist of a $2 \times 2 \times 2$ 3D MPSoC and feed it into the PDN solver to calculate its pulse response. The simulation results on some selected power grid nodes are shown in Fig.9. The simulation error of the PDN solver is negligible compared with the HSpice simulation.

4.3 Problem Formulation

As stated in Section 3, to ensure reliable operations of 3D MPSoCs, we need to consider both thermal and PSN issues during the task scheduling. To clearly formulate the problem, we firstly give 3D MPSoC architecture and task graph definitions as follows.

3D MPSoC Architecture. The 3D MPSoC architecture can be described as $AR(m, n, l)$, where l is the layer[3] count, and each layer has $m \times n$ cores. All processing elements (PEs for short) within a layer are interconnected by a 2D mesh network-on-chip (NoC) structure. PEs on different layers are connected through TSVs, as proposed in [32]. All PEs within the 3D MPSoC share a global power grid, and the power supply is delivered from the bottom layer to the upper layers through power supply TSVs, as shown in Fig.2.

Task Graph Definition. The application running on the MPSoC can be split into a set of tasks executing concurrently or in a specified order. As shown in Fig.3,

[3] In the paper, we use tier and layer interchangeably.

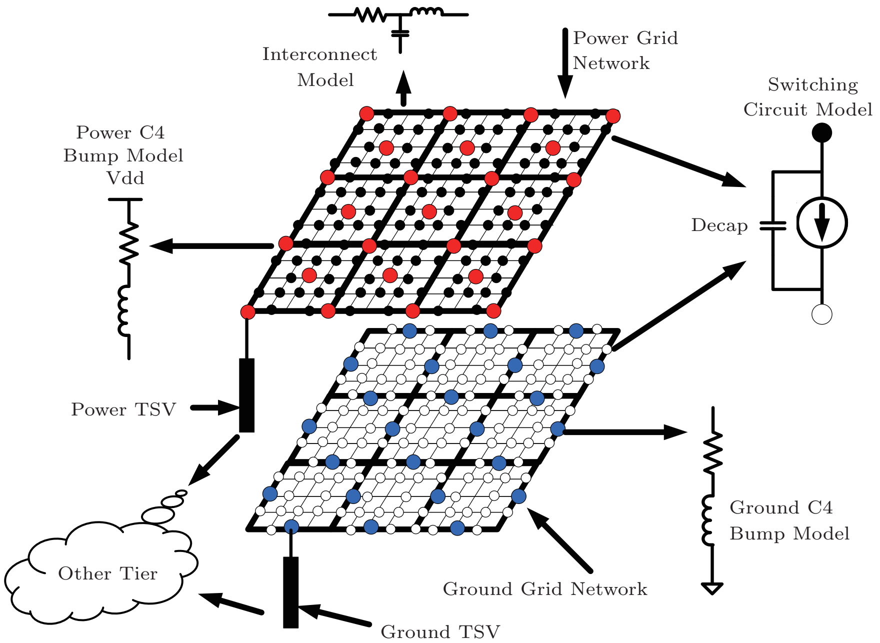
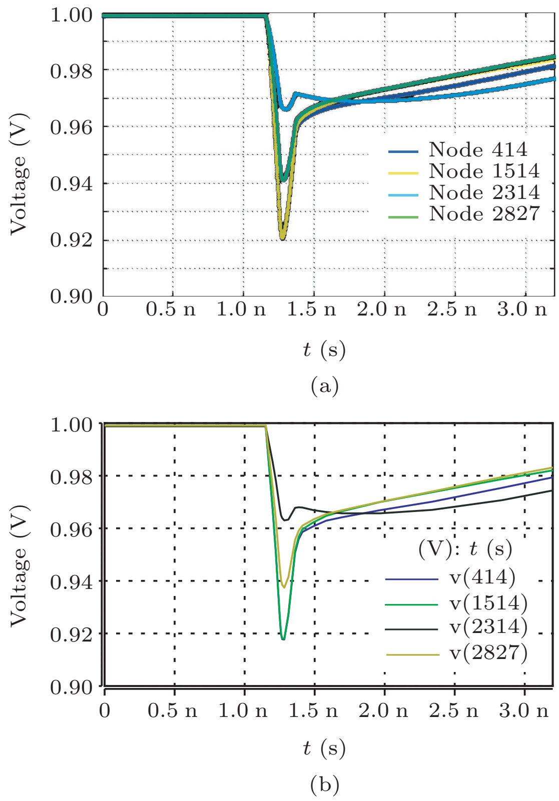


Fig.8. Illustration of the on-chip PDN model[27].



(a)

(b)

Fig.9. Simulation result comparisons between our PDN solver and HSpice simulations on four sample power grid nodes. (a) Simulation results of the PDN solver. (b) HSpice simulation results. t means the simulation time.

S is the vertex set of task graph $G(S, E)$. S_i denotes task i. Its weight t_i denotes the execution time. e_{ji} represents the edge from node i to node j in the task graph. If task i is a predecessor of task j, then they should satisfy the following constraints, $t_{\rm s}(j) \geqslant t_{\rm e}(i)$, where $t_{\rm s}(j)$ is task j's start time and $t_{\rm e}(i)$ is task i's finish time. Additionally, all tasks should meet the deadline constraints, i.e., $t_{\rm e}(i) \leqslant d(i)$ where $d(i)$ is the deadline of task i.

Then, the task scheduling problem can be formulated as follows:

$$
\begin{aligned}
\text{min.} \quad & \sum_{S_i \in S, p \in P} x_{ip} PSN(p, S_i) \\
\text{s. t.} \quad & \forall S_i \sum_{p} x_{ip} = 1, && (1) \\
& t_{\rm e}(S_i) = t_{\rm s}(S_i) + t_i, && (2) \\
& t_{\rm s}(S_i) \geqslant \max\nolimits_{e_{ji} \in E} \{t_{\rm e}(S_j)\}, && (3) \\
& t_{\rm e}(S_i) \leqslant d_{S_i}, && (4) \\
& t_{\rm s}(S_i) \geqslant t_{\rm e}(S_j), && (5) \\
& p_{ij} + p_{ji} = 1, && (6) \\
& \max\nolimits_{p \in P} T_p(t) \leqslant T_0, && (7) \\
& t \leqslant SL, \\
& \boldsymbol{GV}(x) + \boldsymbol{C}\tfrac{{\rm d}V(x)}{{\rm d}t} = \boldsymbol{MAI}, && (8) \\
& x \in \text{power grid nodes}, \\
& PSN(p, S_i) = \max |V(x_k, S_i) - V_{\rm dd}|, && (9) \\
& x_k \in \text{core } p,
\end{aligned}
$$

where,

$$x_{ip} = \begin{cases} 1, \text{ if task } S_i \text{ is allocated to core } p, \\ 0, \text{ otherwise,} \end{cases}$$

$$p_{ij} = \begin{cases} 1, \text{ if task } S_i \text{ is scheduled before } S_j \\ \quad \text{and both of them are allocated} \\ \quad \text{to the same core,} \\ 0, \text{ otherwise.} \end{cases}$$

As shown above, our optimization target is to minimize the PSN magnitudes of running cores by task scheduling. (1) means that every task S_i can only be allocated to a single core. S is the task set and P is the core set. (2) derives the finish time of task S_i, where t_i denotes the task i's execution time. (3) is used to guarantee the task precedence during the task scheduling, i.e., a task can only be scheduled after all its predecessors finish execution. (4) guarantees that the deadline constraint cannot be violated. (5) and (6) maintain the execution order when two tasks are allocated on the same core. (7) makes sure that the peak temperature of the 3D MPSoC must be lower than the temperature constraint T_0. SL is the task schedule time length. At each schedule point, power supply noise can be calculated by (8) and (9).

$\boldsymbol{I}$ is a $K \times 1$ task current vector derived by task characterization, and K is the number of tasks in the task graph. $\boldsymbol{A}$ is a $Pg\_x\ K$ assignment matrix. P is the number of cores in 3D MPSoC. g is the number of power grid points belonging to the core. $\boldsymbol{A}$ indicates which task is assigned to some core. If task i is assigned to core p, $A[x][i] = 1$, where node x belongs to core p. The grid node x is covered by processor p. $\boldsymbol{G}$ is the conductance matrix of power grid, including resistance and inductance of both off-chip package and on-chip PDN interconnects. $\boldsymbol{C}$ is the capacitance matrix contains both decoupling and intrinsic capacitances of off-chip package and on-chip PDN.

In the next subsection, we will propose a heuristic algorithm to solve the problem effectively.

4.4 Our Proposed Task Scheduling Algorithm

As shown in Algorithm 1, the inputs to the algorithm include the 3D MPSoC specification, current stimuli derived from the power trace extraction, and the task graph. The output is the optimal task schedule. The algorithm is constructed based on the widely-used list scheduling algorithm[33]. First of all, the earliest start time of each task is calculated by ASAP (as soon as possible) algorithm. The latest start time is calculated by ALAP (as late as possible) algorithm. Then, all tasks are sorted by their mobilities (i.e., the difference of the latest start time and the earliest start time of the task) in the ascending order such that the task with the least mobility can be scheduled firstly to avoid the violation of the deadline requirement.

At the beginning of the schedule, the first task is put into the ready list. During the schedule, all tasks whose parents finish executions are put into the ready list for scheduling. At each schedule point, ready tasks are assigned to idle cores in an iterative manner. For each assignment trial, the PSNs of all running cores are calculated by the PDN solver mentioned above. At the same time, the peak temperature of each trial is evaluated using Hotspot[21]. As a result, the core incurring the minimum PSN value while meeting the peak temperature constraint is chosen for the task assignment. Since the thermal constant is in the range of milliseconds[34], we use the average power consumption of each task instead of the transient power for the thermal simulation to accelerate the simulation speed. Additionally, as mentioned in [35], significant thermal/PSN fluctuation occurs when there is a new task beginning to run or an old one finishing execution. Therefore, we only perform PSN and thermal evaluations at each scheduling point, which reduces the computing overhead further. When the ready list becomes empty, it means all tasks have been scheduled and the optimal schedule can be obtained.

The timing complexity of Algorithm 1 is analyzed as follows. Assume the task graph has S tasks and the 3D MPSoC has $C = l \times m \times n$ cores, where l is the number of tiers, and each tier has $m \times n$ cores. The worst case is that every task has only one parent and only one child. In this case, at each schedule point, the core candidate for assignment is C. For each candidate, we need to perform one thermal simulation and one PSN calculation, whose running time mainly depends on the thermal and power grid sizes. Assume that the base grid size is a constant. Then, the number of thermal and PDN grids depends on the number of tiers and the number of cores on each tier. Therefore, the time complexity of the thermal simulation or the PSN calculation is $O(Cp_0q_0)$. p_0 is the time of one thermal simulation for a single core, and q_0 is the time of one PSN calculation for a single core. In the worst case, we need totally $S \times C$ thermal simulations and PSN calculations to obtain the optimal schedule. As a result,

Algorithm 1 . PSN and Thermal Aware Task Scheduling Algorithm

```
procedure pt_schedule(G, spec, PDN_param, thermal_param, PSN_profiles_table)
    // G: task graph, spec: 3D MPSoC specification, PDN_param: PDN parameters
    // thermal_param: thermal parameters, PSN_profiles_table: PSN profiles of each task
    construct_ready_list(&ready_list, G, &task_map); cur_time ← 0
    while ready_list! = NULL do
        ready_list_sort(&ready_list, G); task ← task_select(ready_list, cur_time)
        for i ← 0, spec.core_count do
            if spec.core[i] = free then
                update_task_map(task.id, i, &task_map)
                if task.finish_time > deadline[task.id] then
                    restore(task.id, i, &task_map); continue
                else
                    thermal_sim_time ← finish_time(task_map)
                    peak_temp ← thermal_sim(task.id, i, spec.power_table, thermal_param, sim_time)
                    if peak_temp > max_temp then
                        restore(task.id, i, &task_map); continue
                    else
                        PSN_value ← PSN_eval(task.id, i, PSN_profiles_table, PDN_param)
                        if PSN_value < optimal_value then
                            optimal_value ← PSN_value; spec.core[i] ← busy;
                            find_solution ← true
                            update_solution(&PSN_schedule_solution, task.id, i, cur_time)
                        else
                            restore(task.id, i, &task_map); continue
        if find_solution = false && slack = −1 then
            return NULL
        else
            update_ready_list(task.id, i, &G, &task_map, &scheduling_point)
            update_core_status(spec.core, scheduling_point); cur_time ← scheduling_point
    return PSN_schedule_solution
```

the time complexity of the algorithm is $O(SC^2p_0q_0)$ or $O(SC^2)$ since p_0 and q_0 are constants.

Note that we assume that the task graph to be run on the 3D MPSoC is determined in advance similar to [11, 20]. Therefore, our proposed algorithm can run off-line to derive the optimal task scheduling. Although the iterative optimization procedure involves power supply noise and temperature evaluations, the running time overhead is acceptable due to the linear increase of the time complexity with the number of tasks. Moreover, since the algorithm is off-line, it does not need to make the task assignment decision during the run time, and can guarantee quick scheduling response during the running time.

5 Experimental Results

In this section, we describe the experimental setup firstly and then present the experimental results of the proposed task scheduling algorithm.

5.1 Experimental Setup

Our proposed task scheduling algorithm is implemented by C++. E3S (Embedded System Synthesis Benchmarks Suite) benchmark suite is adopted in our experiments. E3S benchmarks are extracted from EEMBC (Embedded Microprocessor Benchmarks Consortium) benchmark suite and widely used in the task scheduling research[11,36]. The E3S benchmarks can be classified into five application categories, i.e., auto-indust, consumer, networking, office automation, and telecommunications (telecom).

In addition to E3S benchmarks, we also use TGFF (Task Graphs for Free)[22] to generate eight different hypothetical task graphs suitable for evaluations on the large-scale 3D MPSoC. The specifications of E3S and synthesized benchmarks are shown in Table 3. The first column is the name of the benchmark. The second and the third column denote the task number and the edge count between tasks respectively. Since task power traces used for PSN evaluations are not available from task graph specifications, we take power trace extracted from SPEC2000 benchmarks instead, and different SPEC2000 benchmarks correspond to different tasks. In practice, task power traces could be obtained with the technique described in Section 4.

Architecture parameters of the core used for the power trace extraction are listed in Table 4. The extracted power traces are converted to current stimuli corresponding to the running tasks and fed into the PDN solver for PSN calculations. The PDN and TSV parameters used in our PDN model are listed in Table 5. Core size used in our experiments is derived from a 45 nm 48-core IA-32 processor[37]. The interconnect parasitics are derived based on 45 nm PTM interconnect model[4]. TSV parasitics are obtained from [38]. To evaluate the temperature using HotSpot, it requires to obtain the power trace of the running task. As mentioned in Subsection 4.1, we firstly use the architecture level simulation to get the power trace when a specific task runs on a core. Because the thermal constant lies within the range of several milliseconds which is much larger than the clock cycle time (within nanosecond range), we pick the average power for the thermal simulation. The leakage power of idle cores is also taken into account during the thermal simulation. With these power information and the Hotspot configuration listed in Table 6, we can get the temperature distributions on-chip in each scheduling step.

Table 3. Task Graphs Used in the Experiments

Benchmark	Task Count	Edge Count
Auto-indust	24	21
Consumer	13	14
Networking	15	11
Office-automation	5	5
Telecom	34	28
Tgff1	23	35
Tgff2	16	21
Tgff3	18	25
Tgff4	21	28
Tgff5	17	24
Tgff6	26	30
Tgff7	22	33
Tgff8	18	27

Table 4. Architecture Parameters of a Single Core of the 3D MPSoCs

Configuration	Parameter
CPU	Alpha 21264 2 GHz
Predictor	Bimodal predictor, BTB with 2-bit counter
IFQ size/LSQ size	4/8
L1 D$/I$	32 KB/32 KB 64 B block size 4-way/1-way associative LRU replacement
Unified L2$	1 MB, 64 B block size 4-way associative LRU replacement

Table 5. PDN Model Parameters in Our Experiments

PDN Model Parameter	Value
Interconnect segment length	200 μm
Interconnect segment resistance	48 Ω
Interconnect segment capacitance	6.8 pF
Interconnect segment inductance	196 pH
TSV diameter	10 μm
TSV aspect ratio	1:8
TSV resistance	20 mΩ
TSV inductance	25 pH
C4 Bump resistance	9.52 mΩ
C4 Bump inductance	12.65 pH
Core size (mm)	3.2 × 3.2
Core mesh grid	16 × 16

Table 6. Thermal Model Parameters Used for Temperature Evaluations

Model Parameter	Value
Bulk Si thickness of bottom die (next to heat sink)	150 μm
Bulk Si thickness of other dies	50 μm
Cu metal layer thickness	0.42 μm
Si thermal conductivity	100.0 W/(mK)
Heat sink thermal conductivity	400.0 W/(mK)
HotSpot grid resolution	64 × 64
Ambient temperature	27°C

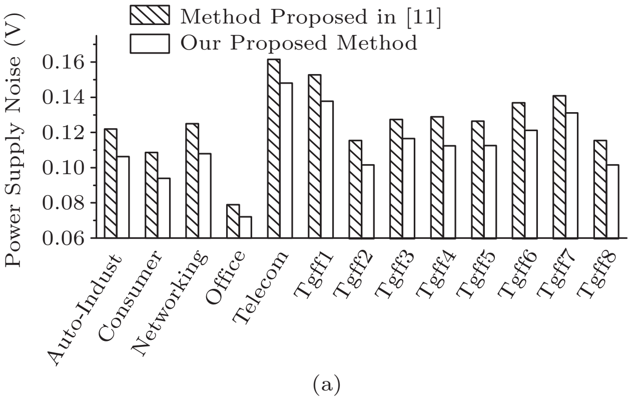
Two kinds of MPSoCs are used in our evaluation. The first one has two tiers and each tier has 2×2 homogeneous cores. The power consumption of the core is assumed to be 4.6 W according to measurements from a prototype of many-core processor[37]. Another one has 3 tiers and each tier has 3×3 homogeneous cores. To verify the effectiveness of our proposed algorithm, we implement a thermal-aware task scheduling algorithm proposed by [11] for comparisons.

5.2 PSN Reductions of Our Proposed Scheme Compared with the State-of-the-Art 3D MPSoC Task Scheduling Algorithm

First, the $2 \times 2 \times 2$ homogeneous MPSoC is considered. During the PSN calculations, the worst case voltage drop during the whole task scheduling procedure among all power grid nodes is chosen for comparison. The temperature constraint is set to 320 K. The comparison results are shown in Fig.10(a).

[4] PTM interconnect model. http://ptm.asu.edu, July 2018.

As shown in Fig.10(a), different benchmarks have significantly different PSN distributions due to their various switching activities. Our proposed task scheduling algorithm can reduce PSN by 12% on average compared with the task scheduling scheme proposed in [11].



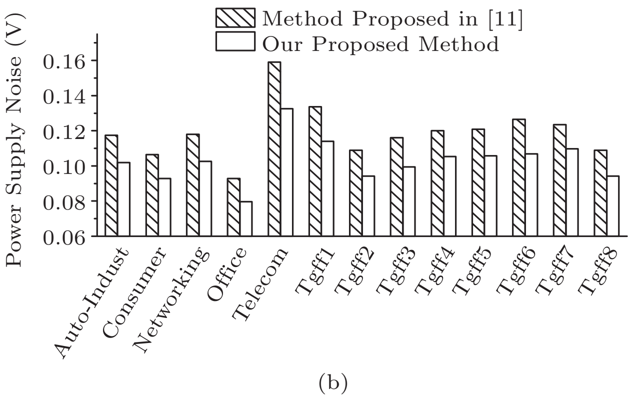
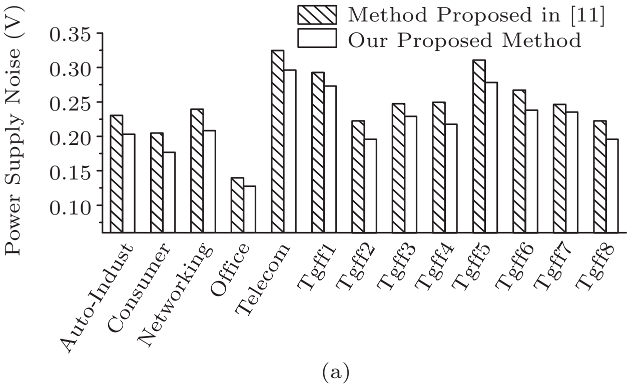


Fig.10. PSN comparisons of our proposed task scheduling algorithm and the thermal aware task scheduling algorithm[11] given the same thermal constraint. (a) Running tasks on the $2 \times 2 \times 2$ MPSoC. (b) Running tasks on the $3 \times 3 \times 3$ MPSoC.

Then, we schedule tasks on the $3 \times 3 \times 3$ MPSoC and the comparison results are shown in Fig.10(b). The power supply noise magnitudes on the $3 \times 3 \times 3$ MPSoC are generally smaller than those of the $2 \times 2 \times 2$ MPSoC case. It is because more cores are idle during the scheduling procedure and can provide more capacitance to suppress the transient PSN. Our proposed algorithm can reduce as much as 17% power supply noise compared with the work in [11] with the same thermal constraint. It indicates the good scalability of our algorithm.

In the above experiments, we observe that there are at most four tasks running simultaneously during the task schedule. Considering the peak power of each core is 4.6 W in our assumption, the total power consumption is at most 18.4 W. To explore the PSN reducing potential of the proposed task scheduling algorithm, we scale the core power by 2x to evaluate the scalability of our proposed technique in terms of the power consumption. Fig.11(a) and Fig.11(b) plot PSN comparisons for $2 \times 2 \times 2$ and $3 \times 3 \times 3$ MPSoC cases respectively. The thermal constraint is still set to 320 K. As shown in Fig.11, increasing core power induces more severe PSN. Compared with [11], our proposed method can reduce PSN by 10.2% for the $2 \times 2 \times 2$ MPSoC case and by 14.8% for the $3 \times 3 \times 3$ MPSoC case.



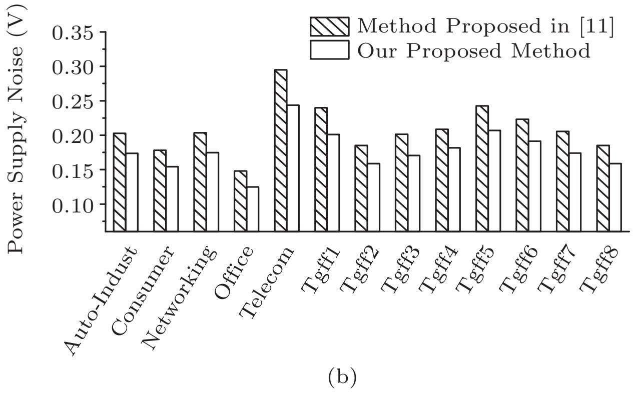


Fig.11. PSN comparisons of our proposed task scheduling algorithm and the thermal aware task scheduling algorithm[11] given the same thermal constraint when the core power is doubled. (a) Running tasks on the $2 \times 2 \times 2$ MPSoC. (b) Running tasks on the $3 \times 3 \times 3$ MPSoC.

5.3 Running Performance & Temperature Comparisons

PSN may not only be detrimental to the system reliability but also increase critical path delay variations, which may severely degrade the task running performance. For example, Saint-Laurent and Swaminathan analyzed the relationship between the power supply noise and the clock frequency, and claimed that 63 mV PSN variation can slow down clock frequency by 6.7% at 130 nm technology node[39]. Therefore, the PSN-aware task scheduling algorithm is beneficial for the task running performance as well. We use the formula proposed in [39] to model the relationship between the average PSN and the critical path delay, i.e.,

$$\frac{D}{D_0} = 1 - k_1\frac{\Delta V}{V_{\rm dd} - V_{\rm t}} + k_2(\frac{\Delta V}{V_{\rm dd} - V_{\rm t}})^2, \qquad (1)$$

where $V_{\rm dd}$ is the nominal supply voltage. $V_{\rm t}$ is the transistor threshold voltage. k_1 and k_2 are process dependent constants. D_0 is the ideal critical path delay.

The critical path delay variations may degrade the processor running frequency and the task execution time. For example, when the critical path delay increases by 20% due to PSN, the clock frequency would have to be decreased accordingly to prevent logic errors. Therefore, the task execution time will increase. In the experiment, according to the PSN distributions during the task scheduling procedure, the clock frequency when running different tasks can be obtained by (10). Then, the updated clock frequency is used to derive the task execution time. Fig.12 illustrates the final completion time when running different task graphs with the thermal-constrained algorithm proposed in [11] and our proposed algorithm assuming the core power is 9.2 W.

As shown in Fig.12, due to the reduced PSN, the task completion time also improves by the proposed task scheduling algorithm. For the 2 × 2 × 2 MPSoC case, the task completion time can be improved by 3.2% on average and 5.5% in maximum. For the 3×3×3 MPSoC case, the task completion time can be improved by 4.4% on average and 7.8% in maximum. The task completion time of task graphs with 4.6 W core power also shows the similar trend and is not plotted due to the lack of space. The above experimental results validate the effectiveness of our proposed algorithm in terms of the PSN reduction and the running task performance improvement.

The peak temperature comparisons caused by different task scheduling techniques are shown in Table 7. As shown in the table, both task scheduling methods can meet the 320 K peak temperature constraint. The thermal-aware task scheduling can obtain lower peak

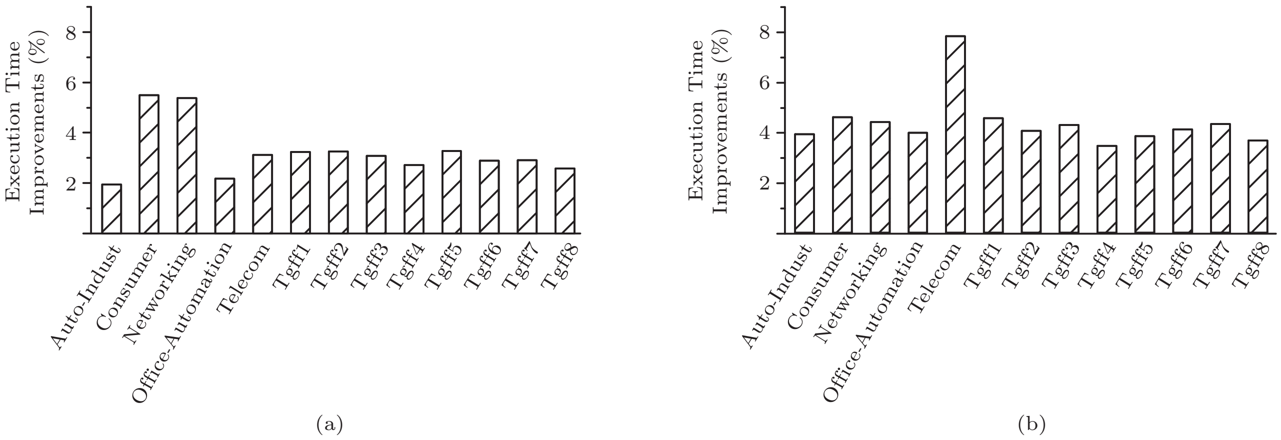


Fig.12. Task graph completion time improvements of the proposed task scheduling algorithm compared with that of the algorithm proposed in [11] under the same thermal constraint (assume the core power is 9.2 W). (a) 2 × 2 × 2 MPSoC. (b) 3 × 3 × 3 MPSoC.

Table 7. Peak Temperature Comparisons of the Thermal-Aware Task Scheduling Algorithm[11] and Our Proposed Algorithm (Unit: Kelvin)

Benchmark	2 × 2 × 2		3 × 3 × 3	
	Thermal-Aware[11]	Our Method	Thermal-Aware[11]	Our Method
Auto-Indust	309.88	311.36	312.03	316.99
Consumer	308.33	310.79	311.20	315.82
Networking	309.34	311.58	312.19	316.96
Office-Automation	306.59	308.71	309.87	314.04
Telecom	313.79	315.08	317.15	320.44
Tgff1	313.43	313.87	313.44	318.17
Tgff2	310.30	313.87	311.59	315.78
Tgff3	311.39	312.68	312.15	316.60
Tgff4	309.76	311.75	312.36	317.33
Tgff5	312.88	314.31	313.60	318.54
Tgff6	311.73	312.39	312.90	317.90
Tgff7	311.93	312.13	312.11	316.37
Tgff8	310.30	310.56	311.59	315.63

temperature while the task scheduling solution derived by our proposed method can achieve an optimal tradeoff between temperature and power supply noise.

5.4 Discussion of the Extension to the Heterogeneous 3D MPSoCs

Although we focus on the homogeneous 3D MPSoCs in this work, our proposed method can be easily extended to the heterogeneous case. For the homogeneous case, the task execution time is the same even when it runs on different cores if we do not take the PSN effect into account. Therefore, we set the PSN magnitude as the optimization target for homogeneous 3D MPSoCs. Whereas, for the heterogeneous case, the same task can have different power consumptions and execution time on different cores, and the optimization target should be set as the final execution time considering the PSN effect. In each task scheduling step, we need to evaluate the new execution time with the power supply noise consideration. And we will take this extension as our future work.

6 Related Work

With the scaling down of power supply voltage, the noise margin reduces remarkably. The signal integrity issue is of a great concern for modern VLSI circuit design. Arabi *et al.* investigated the power supply noise (PSN) impact on performance and reliability of SoCs[40]. The authors observed that PSN can affect the timing of critical path and the chip reliability. Chen and Ling proposed a systematic technique to predict the PSN distributions across the chip in the early stage of chip design, and used it to guide the decoupling capacitor insertion[12]. Firouzi *et al.* formulated the PSN estimation as a linear programming problem, and proposed an efficient method to solve it[41]. All the work above explored the PSN calculation from the circuit level. Gupta *et al.* constructed the power delivery network from the architecture level, and obtained voltage variations within CMP when running different applications, and several hazardous activity sequences are identified based on the PSN simulation[28]. Joseph *et al.* proposed a voltage simulation method which used the power trace as stimuli to the PDN model for voltage simulation[42]. Grochowski *et al.* proposed a hardware implementation to accelerate the simulating[43]. Due to TSV parasitics, 3D ICs have some different properties, such as PSN distribution heterogeneities among layers, vertical PSN coupling, compared with 2D ICs. Huang *et al.* proposed a compact model for 3D power delivery networks analysis[44]. The model has only less than 4% errors compared with HSpice simulations but with much higher simulation speed. Healy and Lim explored the TSV topology impact on the power supply noise for 3D ICs[45]. Zhang *et al.* investigated efficient IR drop calculation for 3D ICs[46]. Their proposed method can speed up simulation by 10× to 20× compared with the preconditioned conjugated gradients method. Todri-Sanial and Cheng studied the PDN modeling of 3D ICs with multiple clock domains[47].

As power density increases, chip temperature becomes another important issue affecting the chip reliability. High temperature not only degrades the chip performance, but also makes the material fragile under repeated thermal cyclings[48]. Thermal evaluation and optimization has got much attention recently, especially for 3D IC whose heat removal is more difficult than the 2D counterpart. Huang *et al.* explored thermal modeling for micro-architecture blocks, and developed Hotspot tool for architecture level thermal simulation[21]. Shang *et al.* claimed that on-chip networks heat dissipation cannot be ignored either. They proposed a NoC thermal model and the ThermalHerd technique for online thermal management[49]. Coskun *et al.* investigated the thermal scheduling policy for MPSoCs[48]. By combining current temperature profile with past thermal history, the authors of [48] proposed a novel OS-level dynamic thermal management heuristic algorithm to reduce performance overhead induced by traditional power/thermal management techniques. Jung *et al.* held the view point that due to thermal measurement errors and varying application characteristics, stochastic-based method can model thermal dissipation more efficiently and effectively compared with previous work[50]. They proposed the dynamic thermal management for multi-core system based on Markov decision process to maximize performance under the given temperature constraint. With the emerging of 3D integration technology, heat dissipation becomes a great concern for chip designers and requires more efficient and effective thermal modeling methods. Qian and Zhu proposed an analytical three-dimensional thermal model for 3D ICs considering the TSV effect[51]. Hameed *et al.* adjusted the core activities at different layers according to their distances from the heatsink such that the hotspot will not occur[52]. This method is combined with DVFS to improve the chip performance by run-time adaption with the thermal consideration.

For MPSoCs, task mapping and/or scheduling is another important research topic. Some work focuses on improving the performance under some constraints such as [48, 53-54] (thermal constraint), [55] (power aware task scheduling). On the other hand, increasing power density inspires the research for energy minimization, such as [56-57]. Since task mapping/scheduling gets rid of changing the low-level hardware design, it can improve the system adaptability for different types of applications with relative low overheads, and will become more and more important with the prevalence of 3D MPSoCs.

7 Conclusions

As transistor integration density increases continuously, more cores can be fabricated on a single chip to implement more functionalities and approach higher energy efficiency. Moreover, with the emergence of 3D integration technology, many core MPSoCs can be packaged in smaller footprints and achieve higher performance. Although 3D MPSoCs bring huge opportunities for high performance system design, they also have some challenging problems. Among them, power supply noise interactions among different cores and tiers become an imminent issue for consideration during the software-hardware co-design. In the paper, we investigated the PSN optimization of 3D MPSoCs from the task scheduling perspective. To capture PSN accurately and efficiently, we proposed a framework including PDN stimuli extraction based on the architecture level simulation, an efficient PDN solver for PSN calculations and a heuristic algorithm for task scheduling taking both PSN and thermal issues into account. Compared with the state-of-the-art task scheduling algorithm for 3D MPSoCs, our proposed algorithm could reduce PSN by 12% on the $2 \times 2 \times 2$ 3D MPSoC and by 14% on the $3 \times 3 \times 3$ 3D MPSoC. The execution time can also be improved by 5.5% and 7.8% respectively due to the reduced PSN magnitudes. Moreover, the experimental results also showed the good scalability of our proposed algorithm as power consumptions of 3D MPSoCs increase.

References

[1] Borkar S, Chien A A. The future of microprocessors. *Communications of the ACM*, 2011, 54(5): 67-77

[2] Martin G. Overview of the MPSoC design challenge. In *Proc. the 43rd ACM/IEEE Design Automation Conference*, July 2006, pp.274-279.

[3] Todri-Sanial A, Tan C S. Physical Design for 3D Integrated Circuits (1st edition). CRC Press, 2015.

[4] Tendler J M, Dodson J S, Fields J S, Le H, Sinharoy B. POWER4 system microarchitecture. *IBM Journal of Research and Development*, 2002, 46(1): 5-25.

[5] Todri A, Marek-Sadowska M, Kozhaya J. Power supply noise aware workload assignment for multi-core systems In *Proc. the 2008 IEEE/ACM International Conference on Computer-Aided Design*, November 2008, pp.330-337.

[6] Wang Y, Xu J, Xu Y *et al.* Power gating aware task scheduling in MPSoC. *IEEE Transactions on Very Large Scale Integration Systems*, 2011, 19(10): 1801-1812.

[7] Huang G, Bakir M, Naeemi A, Chen H, Meindl J D. Power delivery for 3D chip stacks: Physical modeling and design implication. In *Proc. the 2007 IEEE Electrical Performance of Electronic Packaging*, October 2007, pp.205-208.

[8] Sabry M M, Sridhar A, Atienza D, Temiz Y, Leblebici Y, Szczukiewicz S, Borhani N, Thome J R, Brunschwiler T, Michel B. Towards thermally-aware design of 3D MPSoCs with inter-tier cooling. In *Proc. the 2011 Design, Automation and Test in Europe Conference and Exhibition*, March 2011, pp.1466-1471.

[9] Kuuoglu H, Alam M A. A unified modeling of NBTI and hot carrier injection for MOSFET reliability. In *Proc. the 10th International Workshop on Computational Electronics*, October 2004, pp.28-29.

[10] Micheli G D. Synthesis and Optimization of Digital Circuits (1st edition). McGraw-Hill Science/Engineering/Math, 1994.

[11] Chantem T, Hu X S, Dick R P. Temperature-aware scheduling and assignment for hard real-time applications on MPSoCs. *IEEE Transactions on Very Large Scale Integration Systems*, 2011, 19(10): 1884-1897.

[12] Chen H H, Ling D D. Power supply noise analysis methodology for deep-submicron VLSI chip design. In *Proc. the 34th Design Automation Conference* June 1997, pp.638-643.

[13] Zhuo C, Wilke G, Chakraborty R *et al.* A silicon-validated methodology for power delivery modeling and simulation. In *Proc. the 2012 IEEE/ACM International Conference on Computer-Aided Design*, November 2012, pp.255-262.

[14] Khan N H, Alam S M, Hassoun S. Power delivery design for 3-D ICs using different through-silicon via (TSV) technologies. *IEEE Transactions on Very Large Scale Integration Systems*, 2011, 19(4): 647-658.

[15] Healy M B, Lim S K. Power delivery system architecture for many-tier 3D systems. In *Proc. the 60th Electronic Components and Technology Conference*, June 2010, pp.1682-1688.

[16] Conn A R, Haring R A, Visweswariah C. Noise considerations in circuit optimization. In *Proc. the 1998 Computer-Aided Design of Integrated Circuits and Systems*, November 1998, pp.220-227

[17] Sun C, Shang L, Dick R P. Three-dimensional multiprocessor system-on-chip thermal optimization. In *Proc. the 5th IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*, September 2007, pp.117-122.

[18] Huang W, Stan M R, Skadron K. Parameterized physical compact thermal modeling. *IEEE Transactions on Components & Packaging Technologies*, 2005, 28(4): 615-622.

[19] Todri A, Kundu S, Girard P *et al.* A study of tapered 3-D TSVs for power and thermal integrity. *IEEE Transactions on Very Large Scale Integration Systems*, 2013, 21(2): 306-319.

[20] Zhou X, Yang J, Xu Y, Zhang Y, Zhao J. Thermal-aware task scheduling for 3D multicore processors. *IEEE Transactions on Parallel & Distributed Systems*, 2010, 21(1): 60-71.

[21] Huang W, Ghosh S, Velusamy S, Sankaranarayanan K, Skadron K, Stan M R. HotSpot: A compact thermal modeling methodology for early-stage VLSI design. *IEEE Transactions on Very Large Scale Integration Systems*, 2006, 14(5): 501-513.

[22] Dick R P, Rhodes D L, Wolf W. TGFF: Task graphs for free. In *Proc. the 6th International Workshop on Hardware/Software Codesign*, March 1998, pp.97-101.

[23] Xu Z, Gu X, Scheuermann M, Rose K, Webb B C, Knickerbocker J U, Lu J Q. Modeling of power delivery into 3D chips on silicon interposer. In *Proc. the 62nd IEEE Electronic Components and Technology Conference*, June 2012, pp.683-689.

[24] Zhang R, Wang K, Meyer B H, Stan M R, Skadron K. Architecture implications of pads as a scarce resource. In *Proc. the 41st International Symposium on Computer Architecture*, June 2014, pp.373-384.

[25] Brooks D, Tiwari V, Martonosi M. Wattch: A framework for architectural-level power analysis and optimizations. In *Proc. the 27th International Symposium on Computer Architecture*, June 2000, pp.83-94.

[26] Sherwood T, Perelman E, Hamerly G, Calder B. Automatically characterizing large scale program behavior. In *Proc. the 10th International Conference on Architectural Support for Programming Languages and Operating Systems*, October 2002, pp.45-57.

[27] Cheng Y, Todri-Sanial A, Bosio A, Dilillo L, Girard P, Virazel A. Power supply noise-aware workload assignments for homogeneous 3D MPSoCs with thermal consideration. In *Proc. the 19th Asia and South Pacific Design Automation Conference*, January 2014, pp.544-549.

[28] Gupta M S, Oatley J L, Joseph R, Wei G Y, Brooks D M. Understanding voltage variations in chip multiprocessors using a distributed power-delivery network. In *Proc. the 2007 Design, Automation and Test in Europe Conference and Exhibition*, April 2007.

[29] Nassif S R, Kozhaya J N. Fast power grid simulation. In *Proc. the 37th Design Automation Conference*, June 2000, pp.156-161.

[30] Su H, Liu F, Devgan A, Acar E, Nassif S. Full chip leakage estimation considering power supply and temperature variations. In *Proc. the 2003 International Symposium on Low Power Electronics and Design*, August 2003, pp.78-83.

[31] Chen T H, Chen C C P. Efficient large-scale power grid analysis based on preconditioned Krylov-subspace iterative methods. In *Proc. the 38th Design Automation Conference*, June 2001, pp.559-562.

[32] Li F, Nicopoulos C, Richardson T, Xie Y, Narayanan V, Kandemir M. Design and management of 3D chip multiprocessors using network-in-memory. In *Proc. the 33rd International Symposium on Computer Architecture*, June 2006, pp.130-141.

[33] Kwok Y, Ahmad I. Static task scheduling and allocation algorithms for scalable parallel and distributed systems: Classification and performance comparison. In *Annual Review of Scalable Computing*, Kwong Y C (eds.), World Scientific Publishing Company, 2003, pp.107-227.

[34] Choi J, Cher C Y, Franke H, Hamann H, Weger A, Bose P. Thermal-aware task scheduling at the system software level. In *Proc. the 2007 International Symposium on Low Power Electronics and Design*, August 2007, pp.213-218.

[35] Huang L, Yuan F, Xu Q. Lifetime reliability-aware task allocation and scheduling for MPSoC platforms. In *Proc. the 2009 Design, Automation & Test in Europe Conference and Exhibition*, April 2009, pp.51-56.

[36] Cheng Y, Zhang L, Han Y, Li X. Thermal constrained task allocation for interconnect energy reduction in 3-D homogeneous MPSoCs. *IEEE Transactions on Very Large Scale Integration Systems*, 2013, 21(2): 239-249.

[37] Howard J, Dighe S, Hoskote Y *et al.* A 48-Core IA-32 message-passing processor with DVFS in 45nm CMOS. In *Proc. the 2010 International Solid-State Circuits Conference*, February 2010, pp.108-109.

[38] Cadix L, Rousseau M, Fuchs C *et al.* Integration and frequency dependent electrical modeling of Through Silicon Vias (TSV) for high density 3DICs. In *Proc. the 2010 International Interconnect Technology Conference*, January 2010, pp.1-3.

[39] Saint-Laurent M, Swaminathan M. Impact of power-supply noise on timing in high-frequency microprocessors. *IEEE Transactions on Advanced Packaging*, 2004, 27(1): 135-144.

[40] Arabi K, Saleh R, Meng X. Power supply noise in SoCs: Metrics, management, and measurement. *IEEE Design & Test of Computers*, 2007, 24(3): 236-244.

[41] Firouzi F, Kiamehr S, Tahoori M B. Modeling and estimation of power supply noise using linear programming. In *Proc. the 2011 IEEE/ACM International Conference on Computer-Aided Design*, October 2011, pp.537-542.

[42] Joseph R, Brooks D, Martonosi M. Control techniques to eliminate voltage emergencies in high performance processors. In *Proc. the 9th International Symposium on High-Performance Computer Architecture*, February 2003, pp.79-90.

[43] Grochowski E, Ayers D, Tiwari V. Microarchitectural simulation and control of di/dt-induced power supply voltage variation. In *Proc. the 8th High-Performance Computer Architecture*, February 2002, pp.7-16.

[44] Huang G, Sekar D C, Naeemi A *et al.* Compact physical models for power supply noise and chip/package co-design of gigascale integration. In *Proc. the 57th Electronic Components and Technology Conference*, May 2007, pp.1659-1666.

[45] Healy M B, Lim S K. Distributed TSV topology for 3-D power-supply networks. *IEEE Transactions on Very Large Scale Integration Systems*, 2012, 20(11): 2066-2079.

[46] Zhang C, Pavlidis V F, Micheli G D. Voltage propagation method for 3-D power grid analysis. In *Proc. the 2012 Design, Automation & Test in Europe Conference & Exhibition*, April 2012, pp.844-847.

[47] Todri-Sanial A, Cheng Y. A study of 3-D power delivery networks with multiple clock domains. *IEEE Transactions on Very Large Scale Integration Systems*, 2016, 24(11): 3218-3231.

[48] Coskun A K, Rosing T S, Whisnant K. Temperature aware task scheduling in MPSoCs. In *Proc. the 2007 Design, Automation and Test in Europe Conference and Exhibition*, March 2007, pp.1659-1664.

[49] Shang L, Peh L S, Kumar A *et al.* Thermal modeling, characterization and management of on-chip networks. In *Proc. the 37th IEEE/ACM International Symposium on Microarchitecture*, December 2004, pp.67-78.

[50] Jung H, Rong P, Pedram M. Stochastic modeling of a thermally-managed multicore system. In *Proc. the 45th ACM/IEEE Design Automation Conference*, June 2008, pp.728-733.

[51] Qian L, Zhu Z. Analytical heat transfer model for three-dimensional integrated circuits incorporating through silicon via effect — RETRACTED. *LET Micro & Nano Letters*, 2012, 7(9): 994-996.

[52] Hameed F, Faruque M A A, Henkel J. Dynamic thermal management in 3D multicore architecture through run-time adaptation. In *Proc. the 2011 Design, Automation & Test in Europe Conference & Exhibition*, March 2011.

[53] Jayaseelan R, Mitra T. Temperature aware task sequencing and voltage scaling. In *Proc. the 2008 IEEE/ACM International Conference on Computer-Aided Design*, November 2008, pp.618-623.

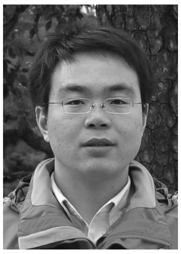
[54] Liao C H, Wen C H P, Chakrabarty K. An online thermal-constrained task scheduler for 3D multi-core processors. In *Proc. the 2015 IEEE/ACM Design, Automation & Test in Europe Conference & Exhibition*, March 2015, pp.351-356.

[55] Momtazpour M, Sanaei E, Goudarzi M. Power-yield optimization in MPSoC task scheduling under process variation. In *Proc. the 11th International Symposium on Quality Electronic Design*, March 2010, pp.747-754.

[56] Hu J, Marculescu R. Energy- and performance-aware mapping for regular NoC architectures. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2005, 24(4): 551-562.

[57] Ghasemazar M, Pakbaznia E, Pedram M. Minimizing the power consumption of a chip multiprocessor under an average throughput constraint. In *Proc. the 11th International Symposium on Quality Electronic Design*, March 2010, pp.362-371.

Ying-Lin Zhao received his B.S. degree in computer science from Xidian University, Xi'an, in 2014, and now is pursuing his Master's degree in electrical engineering at School of Electrical and Information Engineering, Beihang University, Beijing. His research interests include architecture design and optimization of 3D ICs and non-volatile memory systems.



Jian-Lei Yang received his B.S. degree in microelectronics from Xidian University, Xi'an, in 2009, and his Ph.D. degree in computer science and technology from Tsinghua University, Beijing, in 2014. From 2013 to 2014, he was a research intern at Intel Labs China, Intel Corporation. He worked as a post-doctoral researcher with the Department of Electrical and Computer Engineering, University of Pittsburgh, Pittsburgh, from 2014 to 2016. Since 2016, he has joined in School of Computing Science and Engineering, Beihang University, Beijing, as an associate professor. His current research interests include numerical algorithms for VLSI power grid analysis and verification, spintronics and neuromorphic computing. He was the recipient of the first place on TAU Power Grid Simulation Contest in 2011, and the second place on TAU Power Grid Transient Simulation Contest in 2012. He was a recipient of IEEE ICCD Best Paper Award in 2013, and ACM GLSVLSI Best Paper Nomination in 2015.

Wei-Sheng Zhao received his Ph.D. degree in physics from the University of Paris-Sud, Orsay, France, in 2007. From 2004 to 2008, he investigated Spintronic devices based logic circuits and designed a prototype for hybrid Spintronic/CMOS (90 nm) chip in cooperation with STMicroelectronics. Since 2009, he has joined CNRS (French National Center for Scientific Research) as a tenured research scientist and his interest includes the hybrid integration of nano-devices with CMOS circuit and new non-volatile memory (40 nm technology node and below) like MRAM circuit and architecture design. He has authored or co-authored more than 150 scientific papers (e.g., Advanced Material, Nature Communications, IEEE Transactions); he is also the principal inventor of four international patents. From 2014, he becomes a Youth 1 000 Plan Distinguished Professor in Beihang University, Beijing, and the associated editor for IEEE Transactions on Nanotechnology.

Aida Todri-Sanial received her B.S. degree in electrical engineering from Bradley University, Peoria, in 2001, her M.S. degree in electrical engineering from Long Beach State University, CA, in 2003 and Ph.D. degree in electrical and computer engineering from the University of California, Santa Barbara, in 2009. She is currently a research scientist for French National Center of Scientific Research (CNRS) attached to Laboratory of Informatics, Robotics and Microelectronics (LIRMM), University of Montpellier, Montpellier. Previously she was an RD Engineer for Fermi National Accelerator Laboratory, IL where she was the recipient of John Bardeen Fellow in Engineering in 2009. She has also held visiting positions at Mentor Graphics, Cadence Design Systems, STMicroelectronics and IBM TJ Watson Research Center.

Yuan-Qing Cheng received his Ph.D. degree from the Key Laboratory of Computer System and Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing. After spending one year post-doc study at Laboratory of Informatics, Robotics and Microelectronics (LIRMM) and French National Center of Scientific Research (CNRS), University of Montpellier, Montpellier, he joined Beihang University, Beijing, as an assistant professor. His research interests include VLSI design for 3D integrated circuits considering thermal and defect issues, as well as spintronics computing system architecture design. He is currently a senior member of CCF, and a member of ACM and IEEE.