Abstract—3-D technology that stacks silicon dies with through silicon vias (TSVs) is a promising solution to overcome the interconnect scaling problem in giga-scale integrated circuits (ICs). Thermal dissipation is a major challenge for 3-D integration and prior thermal-balanced task scheduling methods for 3-D multiprocessor system-on-chips (MPSoCs) typically balance power gradient across vertical stacks based on the assumption of strong thermal correlation among processing cores within a stack. On the other hand, 3-D MPSoCs typically employ network-on-chip (NoC) as the communication infrastructure which consumes a large portion of the energy budget. As TSVs consume much less energy than horizontal links in 3-D MPSoCs when transmitting the same amount data due to the reduced interconnect distance between vertical adjacent cores, it motivates to allocate heavily communicating tasks within the same vertical stack as much as possible, and thus traffic is restricted in the third dimension to reduce interconnect energy. However, aggregating active tasks within the same stack probably exacerbates the power density and result in hot spots. In this paper, we explore the tradeoff between thermal and interconnect energy when allocating tasks in 3-D Homogeneous MPSoCs, and propose an efficient heuristic. Experimental results show that the proposed technique can reduce interconnect energy by more than 25% on average with almost the same peak temperature when compared with prior thermal-balanced solutions.

Index Terms—3-D multiprocessor system-on-chips (MPSoCs), interconnect energy, network-on-chip (NoC), real-time systems, thermal-aware scheduling and assignment.

I. INTRODUCTION

WITH THE aggressive scaling of process technologies, on-chip interconnects have become a major bottleneck for the performance of giga-scale integrated circuits (ICs), because global interconnects do not scale accordingly with technology [1]. Significant research has been undertaken on 3-D technology, which is considered to be one of the most promising solutions to overcome the interconnect scaling problem [2]. In a 3-D IC, multiple planar device layers are stacked and bonded together using through silicon vias (TSVs), as shown in Fig. 1. Each layer except the bottom one is thinned to only tens of micrometers for the ease of stacking. By building the chip in the third dimension, circuit blocks connected using global wires in a planar IC can instead be placed vertically to reduce wire lengths drastically, thus leading to higher performance and lower power consumption when compared to traditional 2-D counterparts [3].

When dies from different layers are bonded together, the stacking structure will generate high power densities, and thus result in severe thermal issues, which is a major challenge for 3-D integration, as high temperature can impair a chip’s reliability and reduce its lifetime. Many methods from different perspectives have been proposed in the literature, including thermal via insertion [4], layout optimization [5], novel cooling systems [6], dynamic thermal management (DTM) [7], [8], thermal-aware task allocation and scheduling [9], [10], etc. Most of the above solutions are orthogonal to each other, and 3-D IC designs typically employ combinations of them to efficiently alleviate the on-chip thermal condition. In this paper, we mainly focus on the task allocation scheme for 3-D homogeneous multiprocessor system-on-chips (MPSoCs).

For 3-D MPSoCs, we can expect that more and more processing cores will be integrated on a chip. In terms of communication infrastructure, network-on-chip (NoC) is generally regarded as the most promising interconnect architecture due to its scalability and high throughput [11]–[16]. However, with the ever increasing demand for interconnect bandwidth, power becomes a major concern for the design of NoCs. On-chip network consumes a substantial portion of the entire chip power budget, even though this portion may vary among chip architectures and applications running on top [17]. For example, routers...
and links in the 80-core teraflops processor built by Intel take up 28% of the total power when they operate at 4 GHz, and this ratio will increase to 39% under the maximum frequency, i.e., 5 GHz [18]. The latest 48-core chip demonstrated by Intel uses much more complex heavy weight cores, and employs more advanced power management techniques, the interconnects still consume 10% of full chip power [19]. This problem is more obvious and serious for communication intensive applications. Significant prior work focused on task allocation and schedule techniques to reduce communication energy, for both 2-D and 3-D NoC-based MPSoCs [20]–[22].

Fig. 2 depicts a conceptual NoC-Bus hybrid architecture for 3-D MPSoC, in which planar 2-D cores are distributed across multiple layers. The vertical interconnect utilizes a wide, low-latency multi-drop bus to better explore the benefits of TSVs, such as shorter vertical connections, increased bandwidth, etc. Due to the reduced interconnect distance between vertical adjacent cores, which introduces smaller resistance and capacitance, TSVs consume much less energy than horizontal links when transmitting the same amount of data [23], [24]. Therefore, from energy consumption perspective, it is intuitive to allocate heavily communicating tasks within the same vertical stack as much as possible and thus restrict traffic vertically. However, aggregating active tasks within the same stack probably exacerbates the power density and results in hot spots due to strong vertical thermal correlations between adjacent dies. On the other hand, prior thermal-balanced allocation proposals for 3-D MPSoCs consider independent tasks and generally try to scatter hot tasks among different stacks to alleviate the heat generation problem [4], [5], [9], [25], [26]. However, this will achieve suboptimal solutions when communication energy and thermal issues are both taken into considerations. A later given example will present more detailed illustration on this problem.

Motivated by this observation, in this paper we investigate and explore the tradeoff between thermal and interconnect energy when allocating tasks on 3-D homogeneous MPSoCs. We formulate the thermal-constrained task allocation problem to minimize interconnect energy and analyze its complexity. Relaxing the strict requirement of thermal balance, without exceeding the given constraint, gives larger space for interconnect energy optimization. We show this problem to be an instance of a well-known NP complete problem, i.e., the quadratic assignment problem (QAP). Our proposed novel heuristic can be divided into three steps. First, the thermal-balanced (TB) method is used to produce the starting solution point. Second, a proposed greedy procedure is applied on top of the starting solution to reduce interconnect energy as much as possible, under the thermal constraint. Finally, we adopt the state-of-the-art simulated annealing techniques [27], [28], which have yielded promising results for QAP, to further optimize the intermediate solution achieved by greedy. The above three steps together forms our greedy-based simulated-annealing algorithm (GSA). GSA achieves high quality task allocation results with polynomial time complexity. E3S benchmark suite is used to validate the effectiveness and efficiency of our algorithm. Extensive simulation results show that the proposed method can reduce interconnect energy by more than 25% without increasing the peak temperature when compared with thermal-balanced solutions, and this reduction can reach 48.5% if peak temperature constraint is relaxed by no more than 5.5°C.

The rest of this paper is organized as follows. Section II overviews preliminary knowledge and motivates this work. Section III formulates the problem investigated in this paper and analyzes its complexity. The proposed heuristic is described in detail in Section IV. Section V presents experimental results and analysis. Section VI briefly reviews prior related work and Section VII concludes this paper.

II. PRELIMINARIES AND MOTIVATION

A. Preliminaries

1) 3-D On-Chip Interconnection Architectures: Network-on-chip (NoC) is considered to be one promising option for future CMP and SoC designs to mitigate the interconnect scaling problem. In the following, we give a brief introduction on the architectural designs for the combination of both 3-D integration and NoC, i.e., 3-D NoC. Please note that, in this work we only consider the processing element, which could be a core or a cache bank itself, is a 2-D planar design. Generally, the 3-D NoC architecture falls into the following three categories [29].

1) Symmetric NoC Design: The simplest 3-D NoC design is to extend a 2-D router with two additional ports, i.e., up and down, within a 3-D mesh topology. Although this architecture introduces at least modifications to the traditional design paradigm, it does not take advantage of the benefits of 3-D integration. Vertical and horizontal data transmission are indistinguishable as both of them bear identical characteristics as hop-by-hop traversal. More importantly, the symmetric NoC router design incurs significant area and power overhead.

2) NoC-Bus Hybrid Design: To better explore the benefits of 3-D integration, NoC-Bus hybrid architecture is proposed, in which a low latency multi-drop shared bus is used to connect cores within the same stack. A centralized arbiter is used to resolve the contention among cores, and cores within the same stack can be accessed in a single hop [30].

3) True 3-D NoC Design: A true 3-D router architecture is proposed in [31], in which all major components of a router, including crossbar, are partitioned into different layers such that the vertical link can be used more efficiently.
2) 3-D NoC Energy Model: The communication energy is mainly consumed by two parts, i.e., routers and links. In this work, we mainly explore the beneficial attribute in 3-D integration, that is vertical links consume much less power than horizontal links because of differences in wire lengths between neighboring cores. However, for symmetric 3-D NoC design, energy reduction for data transmission along the vertical dimension is not remarkable due to the high hop router energy. Therefore, the above latter two architectures are adopted as the target platforms in this paper. Note that the choice between the last two designs has no essential impact on the results of our methods.

For 3-D NoC-Bus hybrid architecture, the energy consumed by transmitting one bit from core \(i\) to core \(j\) can be computed as

\[
E_{\text{bit}}^{P_i, P_j} = \left( n_{\text{hop}}^{\text{horizontal}} + 1 \right) \times E_{\text{bit}}^{\text{horizontal}} + n_{\text{hop}}^{\text{vertical}} \times E_{\text{bit}}^{\text{vertical}}
\]

(1)

where \(E_{\text{bit}}^{\text{outer}}\) represents the energy consumed on the router when transmitting one bit. \(E_{\text{bit}}^{\text{horizontal}}\) means the energy consumed by the link between horizontally adjacent cores to transmit one bit, while \(E_{\text{bit}}^{\text{vertical}}\) means the energy consumed by the TSVs between vertically adjacent cores. \(n_{\text{hop}}^{\text{horizontal}}\) is the number of hop distance between processor \(i\) and \(j\) in the planar 2-D mesh grid. \(E_{\text{bit}}^{\text{vertical}}\) is determined by the physical capacitance of horizontal link \(C_{\text{phy}}\), supply voltage \(V_{\text{dd}}\) and activity factor \(\alpha\) of the network communication

\[
E_{\text{bit}}^{\text{horizontal}} = 1/2 \times \alpha \times C_{\text{phy}}^{\text{horizontal}} \times V_{\text{dd}}^2
\]

In (2), \(C_{\text{phy}}^{\text{horizontal}}\) of horizontal link can be calculated according to its geometry parameters. \(E_{\text{bit}}^{\text{vertical}}\) can be calculated in a similar manner.

The architecture-level power model for routers can be used to derive \(E_{\text{bit}}^{\text{outer}}\), in which a router can be divided into three major building blocks, i.e., first-input first-output (FIFO) buffer, crossbar, and arbiters [32]. The NoC-Bus hybrid router design needs a 6 × 6 crossbar with one additional I/O port to connect with the vertical bus. Given the communication data volume \(v\) between processor \(i\) and processor \(j\), the communication energy can be calculated via the following equation:

\[
E_{\text{bit}}^{P_i, P_j} = E_{\text{bit}}^{\text{horizontal}} \times v.
\]

(3)

3) 3-D Thermal Evaluation: Sun et al. [25] proposed a 3-D MPSoC thermal model to evaluate thermal impact of their task scheduling algorithm. Hung et al. [33] proposed a thermal-aware task allocation scheme using HotSpot to compute the temperature of the 3-D chip. Although their models are accurate, using them in the task allocation algorithm to predict thermal impacts will be very time-consuming, as the solution space is huge and a lot of trials need to be conducted.

In this work, we assume tasks running for a sufficient long time period, they the temperature impact can be approximated by power consumption of the core running that task. For 3-D MPSoCs, as cores in the same stack have strong thermal correlation, we assume all the cores within the same stack have similar temperatures as in [9]. Therefore, thermal evaluation is performed via power gradient computation in our algorithm.

B. Motivation

In 3-D integration, except the bottom layer, other layers are thinned to only tens of micrometers for integration in contrast to hundreds of micrometers thickness of silicon substrate in the 2-D chip. Since the length of TSV is much smaller than that of the horizontal link (tens versus thousands of micrometers), for the same amount of data, traverse between vertical adjacent cores consumes much less energy than horizontal adjacent cores. Prior work on thermal-aware scheduling in 3-D MPSoCs assumes tasks are independent from each other without any communications among them or generally ignores this beneficial property. The following example illustrates this potential in which both thermal and interconnect energy are taken into considerations.

The capacitance of the horizontal link can be calculated as

\[
C_{\text{phy}}^{\text{horizontal}} = \varepsilon_{\text{ox}} \left[ 2.42 + \frac{W}{x_{\text{int}}} - 0.44 \times \frac{x_{\text{int}}}{W} + \left( 1 - \frac{x_{\text{int}}}{W} \right)^6 \right] \times L
\]

(4)

As Fig. 3 shows, \(W\) and \(L\) denote the width and length of horizontal link respectively. \(x_{\text{int}}\) is the oxide thickness and \(\varepsilon_{\text{ox}}\) is the permittivity of \(SiO_2\). On the other hand, the capacitance of TSV can be expressed as [34]

\[
C_{\text{phy}}^{\text{TSV}} = \frac{6.34\varepsilon_0 l_c}{\ln \left( 1 + 5.26 \frac{l_c}{r_c} \right)} + \frac{\varepsilon_0 l_c}{\ln \left( \frac{r_e}{r_v} \right)}
\]

\[ \times \left[ k_1 \left( \frac{p_e}{r_v} \right)^{k_2} + k_3 \left( \frac{p_e}{l_v} \right)^{k_4} \right]
\]

(5)

where \(l_c, r_c,\) and \(p_e\) denote the length, radius, and pitch of TSV, respectively, and \(k_1, k_2,\) and \(k_3, k_4\) are constants obtained through simulations. We derive the bit transmission energy of horizontal link under SMIC 90 nm 6 metal layer process. Then, we calculate the TSV bit energy using the parameters from ITRS [35]. The former is 0.127 pJ and the latter is \(9.56 \times 10^{-3}\) pJ, i.e., bit transmission energy for TSV is only 7.5% of that of horizontal link, respectively 0.127 pJ and the latter is 9.56 × 10⁻³ pJ, i.e., bit transmission energy for TSV is only 7.5% of that of horizontal link, which provides substantial space for energy optimization. Detailed simulations will be described at the experimental section.

Based on the above, communication energy can be reduced by restricting tasks within the same stack to take advantages of TSVs. Unfortunately, though this scheme is beneficial from the
energy perspective, aggregating tasks into one stack aggravates the thermal problem. If tasks cluster in the same stack, the power density of this stack will increase sharply with a high possibility of generating hot spot. Thus, a tradeoff should be made between energy consumption and thermal dissipation which motivates the research of this paper. An illustrative example is shown in Fig. 4.

The right-hand side of Fig. 4 shows a homogeneous $2 \times 2 \times 3$ NoC-Bus hybrid 3-D MPSoC. The $x$-$y$-$z$ deterministic routing algorithm is adopted to avoid the livelock and deadlock. The task graph of the application is illustrated in the left part of Fig. 4. Task graph is an acyclic directed graph derived from application profiling in advance. Every node in the graph represents a task in the application to be assigned to the core. Every edge represents the communication requirement between the two corresponding nodes (tasks). The weight on the edge denotes the communication data volume between them. The power consumptions of cores running corresponding tasks, and communications between them in this hypothetical example are listed in the task graph of Fig. 4.

First, a thermal-balanced task allocation method which is similar with that in [9] is used. The method minimizes the stack power gradient across the 3-D MPSoC. The obtained solution, i.e., solution A is shown in Fig. 5(a). Tasks assigned in the same stack are grouped in a rectangular box. Communications across the stack are denoted using solid lines and those within the same stack are represented as dashed lines. By adjusting locations of tasks, solution B in Fig. 5(b) reduces inter-stack communications while the power gradient remains the same as in solution A. We relax the constraint of power gradient by 50%, and achieve the solution C as in Fig. 5(c), which turns more inter-stack communications into intra-stack. To quantitatively evaluate these solutions, their stack power distributions and peak temperatures are shown in Fig. 6(b) and the corresponding interconnect energy is illustrated in Fig. 6(a).

From Fig. 6(a), we can see that interconnect energy consumption varies significantly among different solutions. With the same stack power gradient, 26.8% energy can be reduced by solution B compared with A. Meanwhile, with 1.6 W power gradient in C, we can gain another 16.1% energy saving. The peak temperatures of three solutions are almost the same. From the above example, we can conclude that thermal-balanced task scheduling is suboptimal when interconnect energy is taken into consideration. Relaxing the thermal constraint while not exceeding a specified threshold can achieve impressive interconnect energy reduction. The rest of this paper investigates this problem and proposes a novel heuristic to attack it.

### III. Problem Formulation and Analysis

The task allocation problem investigated in this paper is to offline assign tasks in a task graph to cores in a MPSoC. In addition, we assume that tasks run periodically for sufficient long time and two connecting tasks have to transfer the specified data volume to continue their executions. The task graph can be defined as follows.

**Definition 1 (Task Graph):** Given an undirected graph $G(V, E)$, where $V$ denotes the task set of $G$, $E$ represents communication relationships. The weight $W$ of each edge denotes communication data volume between the two connecting tasks.

We use NoC-Bus hybrid architecture in this paper, and it can be described as follows.

**Definition 2 (3-D NoC-Bus Hybrid MPSoC):** The 3-D MPSoC has $L$ layers and each layer contains $M \times N$ cores which are connected by 2-D mesh in every layer. These layers communicate with each other through TSV busses vertically. The routing algorithm is $R$. The data width of NoC channel is $B$ bit. Thus, the architecture can be represented by $A(L, M, N, R, B)$.

Based on the above definitions, the allocation problem investigated in this paper can be expressed as follows.

**Definition 3 (Thermal-Constrained Allocation Problem for Interconnect Energy Minimization):** Given a task graph $G(V, E)$ and the 3-D MPSoC architecture $A(L, M, N, R, B)$. Find an optimal task allocation scheme to minimize interconnect energy consumption

$$\sum_i \sum_j E(i,j) \times \delta(i,j)$$

where core $i$, core $j$ $\in A(i \neq j)$, and $\delta(i,j)$ is

$$\delta(i,j) = \begin{cases} 1, & \text{if core } i \text{ communicates with core } j \\ 0, & \text{otherwise} \end{cases}$$

subject to

$$\max \Delta P(\text{stack}_i, \text{stack}_j) \leq P$$

where

$$\Delta P(\text{stack}_i, \text{stack}_j) = |P(\text{stack}_i) - P(\text{stack}_j)|.$$

In (6), $E(i,j)$ denotes the interconnect energy between core $i$ and core $j$. In (8), we use the power gradient constraint to represent thermal constraint and $P(\text{stack}_i)$ denotes the stack, where core $i$ resides. In the following, we show that the above problem is essentially an instance of a well known NP-complete problem.

Recall that the communication energy is dependent on the relative locations and data volume of the communicating task pair. We assume the Manhattan distance of the two communicating task is $d$, then the communication energy can be expressed as

$$E(i,j) = E_{\text{horizontal}} \times d_{\text{horizontal}} + E_{\text{vertical}} \times d_{\text{vertical}} + E_{\text{outside}} \times (d_{\text{horizontal}} + 1)$$

where $E_{\text{horizontal}}$, $E_{\text{vertical}}$, and $E_{\text{outside}}$ represent horizontal, vertical, and outside communication energy, respectively.
Fig. 5. Different task allocation results. (a) Solution A (thermal-balanced design), (b) solution B, and (c) solution C.

Fig. 6. (a) Interconnect energy comparisons and (b) stack power/peak temperature distributions of the example.

The famous “backboard wiring problem” is a typical application of QAP, which concerns how to place computer components to minimize the total amount of wiring required to connect them. The above conclusion is easy to reach if we let

$$a_{ij} = v$$

(13)

$$b_{ij} = E_{\text{hit}} \times d_{\text{horizontal}} + E_{\text{hit}} \times d_{\text{vertical}} + E_{\text{hit}} \times (d_{\text{horizontal}} + 1).$$

(14)

The above brief analysis gives us two important inspirations. First, we cannot expect an algorithm with polynomial time complexity for this problem. Second, and more importantly, we can borrow and adapt prior efficient heuristics for QAP to tackle this problem, as in the following section.

IV. PROPOSED HEURISTIC ALGORITHM

Since we have shown the above problem to be an instance of QAP, we can adopt previous heuristic approaches for QAP to tackle it. However, simple and direct applying of prior work is not a good idea as they do not consider any characteristics of our problem. Moreover, most of the prior solutions are quite time-consuming to explore many random solutions before achieving a satisfactory result. In this work, we propose an efficient and effective heuristic algorithm, which is called GSA, short for greedy-based simulated annealing algorithm. GSA achieves high quality task allocation results with polynomial time complexity. It can be divided into three steps as follows.

Step 1: A Thermal-Balanced Method Generates Initial Solution. A thermal-balanced (TB) method which is similar with the one proposed in [9] is used to generate the initial solution. Note that our algorithm is an offline allocation scheme which is different from the online scheduling method in [9]. Their method minimizes the stack power gradient across the 3-D MPSoC without considering interconnect energy. Our method firstly sorts tasks in descending order according to their power consumption running on the core. Hereafter, tasks are allocated one by one to the stack with the minimum stack power currently until all the tasks are allocated. Within one stack, tasks with higher power consumption will be allocated nearer to the heat sink to facilitate its heat dissipation. Fig. 7(a) shows an initial solution on a $2 \times 2 \times 3$ 3-D MPSoC. Communications between stacks are drawn in solid lines in the figure while those lies within the same stack are drawn with dashed lines.

We propose a fast deterministic greedy procedure to optimize the above initial solution and achieve a high quality intermediate solution as follows.

We use the same example in Step I for illustration. Based on the initial solution derived in the first step, we examine communications lie across different stacks and select the largest one [labeled 1000 in Fig. 7(a)] to move it from inter-stack to intra-stack communication by exchanging task 1 and 7 as depicted in Fig. 7(b). Note that, 1000 is normalized communication energy considering both distance and data volume. However, this change leads to another intra-stack communication [labeled 50 in Fig. 7(a)] out of the stack, and thus increases its energy from 50 to 500 accordingly. If the task exchange does not result in the total interconnect energy reduction, it will be dropped. After that, the power gradient constraint is verified not to be violated. Then, this exchange can be accepted. This procedure iterates until the energy consumption cannot be improved any further, as depicted in Fig. 7(c).

Step III: Simulated Annealing Produces Final Solution.

Through the above two steps, we can achieve a much better result. Since we have shown that the problem is an instance of QAP, we finally adopt the state-of-the-art simulated annealing techniques, which have yielded promising results for QAP, to further optimize the intermediate solution.

In the annealing procedure, any pair of cores are exchanged to derive a new solution and the interconnect energy is computed accordingly. For an MPSoC with $n$ cores, the number of 2-exchanges will be $n \times (n - 1)/2$. After all exchange operations complete, the best result (the one with the least energy consumption) is selected as the current optimal solution. During the search process, power gradient is evaluated in every task allocation trial to check the violation of the constraint. If the constraint cannot be satisfied by this adjustment, the energy consumption of this allocation scheme is marked as infinite to prevent it from being chosen later. After that, the annealing temperature decreases. The annealing procedure continues until no improvement can be obtained within the specified iteration number. When the optimal result in the above search is produced, it is optimized by the TABOO method proposed in [27] further. In addition, to avoid being trapped in the sub-optimal region during a single annealing procedure, we adoptannealing scheme which follows the ramping wave with several repeated fluctuations as in [28]. For the detailed implementation of the revised annealing algorithm, please see Algorithm 1.

Algorithm 1 The greedy-based simulated annealing algorithm

1: procedure GSA ($n, S^*, \text{IterCnt}, \mathcal{P}^*$) 
2: // $n$: core count, IterCnt: iteration count 
3: // $S^*$: initial task allocation scheme 
4: // $\mathcal{P}^*$: stack power constraint 
5: $\mathcal{P} \leftarrow \mathcal{P}^*$ // set power gradient 
6: $S \leftarrow S^*$ // set initial assignment achieved by greedy 
7: minenergy $\leftarrow \text{ComputeEnergy}(S, \mathcal{P})$ 
8: $Q \leftarrow \text{IterCnt}$, $K \leftarrow n(n - 1)/2$ 
9: $L_0 \leftarrow QK$, oscillation $\leftarrow$ false 
10: set initial temperature $t \leftarrow t_0$ // according to $S$
11: for $k \leftarrow 0$ to $L_0$ do 
12: 
13: make exchange operation based on $S$ to derive $S'$ 
14: energy $\leftarrow \text{ComputeEnergy}(S', \mathcal{P})$ 
15: // if constraint $\mathcal{P}$ is violated, energy $\leftarrow \infty$ 
16: if energy $< \text{minenergy}$ then 
17: accept $\leftarrow$ true 
18: else if $RAN\text{D}[\cdot] < e^{-\Delta/t}$ then 
19: accept $\leftarrow$ true 
20: else 
21: accept $\leftarrow$ false 
22: energy $\leftarrow \infty$ 
23: end if 
24: if accept $\leftarrow$ true then 
25: $S \leftarrow S'$ 
26: else 
27: reject $++$ 
28: end if 
29: if oscillation $\leftarrow$ false then 
30: if reject $\geq$ threshold then 
31: set $t$ to $t_0$, oscillation $\leftarrow$ true, $k' \leftarrow 0$ 
32: // start another annealing procedure 
33: end if 
34: else 
35: $k'++$ 
36: end if 
37: if $k' > 0$ then 
38: decrease temperature $t$ // temperature annealing 
39: end if 
40: $S_{opt} \leftarrow \text{TABOO Search}(S)$ 
41: end procedure

Please note that, in this paper we consider the scenario that the number of tasks in an application is less than the number of cores available on a 3-D MPSoC, and once a task is allocated to a core, it cannot be preempted by other tasks. GSA algorithm is an offline solution to allocate tasks to processing cores on a 3-D MPSoC before the application starts execution. The three steps are performed one by one. Suppose an application with $M$ tasks and $C$ interconnection between them to be allocated to a 3-D MPSoC with $L$ layers and $N$ cores within each layer. Step I involves sorting tasks’ power consumptions in descending order, with time complexity $O(M \log M)$. In Step II, all interconnections between tasks are sorted and exchanged to check whether there are energy reduction. Thus, the complexity is $O(C^2 \log C)$. The innermost procedure ComputeEnergy evaluated all $NL$ cores and the complexity of simulated annealing procedure is $O((NL)^3)$. Therefore, the complexity of Step III is $O((NL)^4)$.

V. EXPERIMENTAL RESULTS AND ANALYSIS

A. Experimental Setup

1) Benchmark Suites: To evaluate the effectiveness of the proposed heuristic, we use the well-known E3S benchmarks [32], [36], [37] in experiments. E3S contains automation/industrial, office automation, networking, telecommunication and
consumer-electronic benchmark suites, which are all derived from EEMBC benchmarks. Each benchmark program is comprised of a task graph and pre-defined communication patterns. Table I lists the detailed information of E3S benchmarks. The inter-task communication data volumes for E3S benchmarks range from kilobits to megabits as shown in Table I.

2) **Simulation Platform:** The simulation platform is comprised of PowerPC 750CX cores that are connected by a 3-D NoC-Bus hybrid structure. The 3-D integration uses Face-to-Back technology in this work for the ease of integration of multiple layers. Each layer is assumed to be thinned to 50 μm except the bottom layer which is hundreds of micrometers thick. The feature size of TSV is assumed to be 5 μm as in [35].

The size of PowerPC 750CX is set to 4 mm × 4 mm, which is derived by extrapolating the size of the core in 90-nm technology node. The task power consumptions on this core are obtained from E3S benchmark specifications.

The routers in NoC-Bus interconnection network employ classic virtual channel design with four pipeline stages. The deterministic x-y-z routing and credit-based flow control are adopted in our platform. The channel width of the network is set to 64 bits with 500 MHz operating frequency. The router and link energy are computed based on the energy model in 90-nm technology discussed in Section II.

3) **Thermal Simulation:** HotSpot-5.0 is used for thermal simulation in our experiments. HotSpot-5.0 supports thermal simulation of 3-D chips with grid model [38], [39]. With power trace of each core and layout of each layer as inputs, it produces transient and steady temperatures of each core. In this work, we assume tasks to run on cores periodically for sufficient long time, we only take steady temperature into consideration. As the thermal constant of the silicon chip is generally about several hundreds of milliseconds [40], we simulate the thermal state of the chip for one second to obtain the steady-state temperature.

At this point, we collect the interconnect energy consumption by using various methods. The sample rate is 10 K cycles/sample and the trace file contains 500 M/10 K = 50 000 sample points. The power trace file containing all running tasks’ power consumptions can then be generated. An idle core is set to consume 1/10 of the normal power. The detailed configuration of thermal simulation is shown in Table II.

### B. Experimental Results and Discussion

In order to show the potential interconnect energy reduction that the proposed GSA algorithm can achieve under the same power gradient, thermal-balanced (TB) method is employed as the baseline algorithm. We also evaluate other two algorithms in experiments, i.e., GSA-50% and GSA-100% to demonstrate more interconnect energy savings when power gradient is relaxed by 50% and 100% compared with GSA, respectively.

1) **Experiment I:** In this experiment, the target platform is a 12-core 3-D MPSoC with 2 × 2 × 3 network topology (3 layers and 2 × 2 2-D Mesh within each layer). We apply various E3S benchmarks on top to evaluate the previously mentioned four algorithms on interconnect energy saving and thermal impact. The interconnect energy consumed for E3S benchmark suite is depicted in Fig. 8. We divide the benchmarks into two groups. The consumer, networking and office-automation are all communication-intensive applications with large data volume exchanged among tasks, as can be seen in Table I. Auto-indust and telecom belong to the other group. The peak temperature achieved by each method is marked on top of each bar correspondingly.

First, it can be seen from the figure that GSA can achieve remarkable interconnect energy reductions for all benchmarks, except telecom 1 and 4, for which TB happened to produce good results. Take the benchmark “consumer1” for example, after the allocation of tasks by TB, the maximum power gradient is 5.4 W with peak temperature 317.13 K. The same power gradient is

---

**TABLE I**

<table>
<thead>
<tr>
<th>Category</th>
<th>Task graphs</th>
<th>Overall tasks</th>
<th>Task → Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>auto-indust</td>
<td>4</td>
<td>24</td>
<td>1Kb ~ 15Kb</td>
</tr>
<tr>
<td>consumer</td>
<td>2</td>
<td>12</td>
<td>1Mb ~ 6Mb</td>
</tr>
<tr>
<td>networking</td>
<td>4</td>
<td>13</td>
<td>1.7Mb ~ 8.4Mb</td>
</tr>
<tr>
<td>office-automation</td>
<td>1</td>
<td>5</td>
<td>1Kb ~ 787Kb</td>
</tr>
<tr>
<td>telecom</td>
<td>9</td>
<td>30</td>
<td>1Kb ~ 10Kb</td>
</tr>
</tbody>
</table>

**TABLE II**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bulk Si thickness of bottom die (next to heat sink)</td>
<td>150μm</td>
</tr>
<tr>
<td>Bulk Si thickness of other dies</td>
<td>50μm</td>
</tr>
<tr>
<td>Cu metal layer thickness</td>
<td>0.42μm</td>
</tr>
<tr>
<td>Si thermal conductivity</td>
<td>100.0W/(cm-K)</td>
</tr>
<tr>
<td>Heat sink thermal conductivity</td>
<td>400.0W/(cm-K)</td>
</tr>
<tr>
<td>HotSpot grid resolution</td>
<td>64 × 64</td>
</tr>
<tr>
<td>Ambient temperature</td>
<td>27°C</td>
</tr>
</tbody>
</table>
set as the constraint for GSA, which achieves 28.8% interconnect energy reduction while the peak temperature is 317.33 K. As the power gradient for GSA and TB are the same, thermal simulation result shows that the peak temperature achieved by GSA is almost the same as that of TB method. On average, GSA achieves 26.3% and 25.4% energy savings for the two groups of benchmarks, respectively, when compared with TB method. The above results prove the superioritiy of GSA over TB: significant interconnect energy reductions without aggravating the thermal problems of 3-D ICs.

Second, when the power gradient is relaxed by 50% and 100%, respectively, we can further gain small amount of interconnect energy reduction as shown in the figure. This is because we can allocate more tasks within the same stack and make better use of vertical communication links. However, as mentioned before, this will aggravate thermal issues for 3-D ICs, as the stack power density will increase. This is clear in the figure that the steady-state peak temperatures increase by more or less 1K and 5K on average from GSA to GSA-50% and GSA-100%, respectively. Depending on the thermal constraints set by users, GSA can achieve minimal interconnect energy consumption.

2) Experiment II: In this experiment, we show the scalability of GSA when there are more layers in 3-D MPSoCs. We implement 36-core MPSoC simulation platforms with different topologies, i.e., 3 × 4 × 3, 3 × 3 × 4, 2 × 3 × 6 with 3, 4, and 6 layers, respectively. More layers can provide more opportunities to save interconnect energy. As the number of tasks within an E3S benchmark is fixed, and is much smaller than the number of cores (36), we mix several benchmarks from the same category and apply them on the three simulation platforms. The interconnect energy consumed for synthesized E3S benchmark suite is depicted in Fig. 9.

First, it is clear that when the number of layers are increased, GSA can achieve more interconnect energy reduction, while for TB, the interconnect energy fluctuates. This is because TB method is unaware of the difference between horizontal and vertical links in NoC-Bus architecture, while GSA can fully explore the benefits of TSVs to allocate more data transfers along vertical links.

It is also very obvious that when there are more layers in 3-D ICs, the peak temperatures are increased accordingly, which is due to the increased power density within a stack. In addition, it can be seen that relaxing the power gradient will increase the peak temperature more drastically when there are more layers. This is because with more layers and more relaxed power gradient, GSA heuristic is free to allocate communicating tasks into the same stack as much as possible, thus aggravates the thermal problem.

3) Experiment III: To further validate the efficiency of our method, in this experiment, we evaluate the execution time of each benchmark when applying the above allocation solutions. As stated above, the proposed GSA algorithm try to cluster communication intensive tasks within the same stack, which cannot only achieve power benefit but also performance benefit. This is because vertical communication not only consume less power but also take less time, when compared with horizontal communication if all else being equal. The saved communication time will convert to less execution time of the entire application.

We have shown in Table III the execution time results of E3S benchmarks for different algorithms on a 2 × 2 × 3 3-D MPSoC simulation platform. The results are the execution time for one single period of an application. It is clear that the execution time are less for all applications by using GSA than TB method. For consumer and networking benchmarks which are all communication intensive applications, the execution time can be reduced by at most 47%.

VI. RELATED WORK

Since power densities increase dramatically with the shrinking technology node, it introduces severe thermal emer-
gery which threatens the performance and reliability of modern MPSoCs. Thus, thermal optimization has attracted much attention by academia and industry. Coskun et al. [41] proposed several thermal-aware heuristics to obtain uniform temperature distribution across the whole MPSoC chip. Rosing et al. [42] claimed that reliability and power management should be considered simultaneously to meet reliability constraint while not aggravating power consumptions. Marcon et al. [43] formulated the task scheduling problem considering both thermal hot spot and temperature gradient across the chip and proposed a task scheduling algorithm using integer linear programming (ILP) to solve it. Jayaseelan et al. [44] took thermal optimization into consideration in the system configured with DTM (dynamic thermal management mechanism) and proposed a heuristic algorithm to construct the task sequence such that the peak temperature can be minimized. Chantem et al. [36] formalized the temperature-aware real-time task allocation and schedule problem for MPSoCs and proposed mixed linear programming-based heuristics to solve it. Due to these methods are 2-D-oriented, they may be unsuitable for 3-D MPSoC which has significant different thermal property from 2-D counterparts.

Since 3-D integration introduces severe thermal problem, temperature-aware design is of vital importance. Goplen et al. [4] used thermal TSVs to provide sufficient thermal dissipation paths. Wong et al. [5] exploited the potential of layout optimization benefits to reduce temperature. Bakir et al. [6] proposed novel cooling hydraulic system to alleviate the thermal issue in 3-D ICs. While the above solutions attempted to alleviate thermal issue through novel cooling mechanisms, some other research focus on attacking it by task scheduling techniques. Zhou et al. [9] scheduled tasks on homogeneous 3-D multiprocessor platform to achieve uniform temperature distributions. However, all these 3-D thermal-related work only took thermal optimization into account without considering communication energy consumption.

In addition to thermal issue, energy consumption is another challenge, especially for battery-driven systems. Task mapping/scheduling plays a key role in enhancing the energy efficiency of MPSoCs, and thus has already been an active research field for several years [20], [21], [45]–[52]. Hu et al. [47] first formulated the energy-aware task allocation problem of 2-D NoC-based MPSoC and presented an branch-and-bound algorithm to attack it. Marcon et al. [43] took both the dynamic behavior of the application and communication contention into consideration to guide the task schedule. The experimental results showed that execution time and energy consumption can be reduced by their proposed methods. Although these task allocation and schedule research effectively reduce energy, thermal related issue were not considered.

Different from the above research work, in this paper we explore the potential tradeoffs between thermal and interconnect energy during the task mapping on homogeneous 3-D MPSoCs, and propose a novel heuristic algorithm to reduce interconnect energy significantly while maintaining the peak temperature increase in an acceptable range.
VII. CONCLUSION

Both 3-D integration and NoCs are proposed as alternatives for the future interconnect scaling problem. In this paper, we study jointly the thermal and interconnect energy concerns in 3-D integration and NoCs, respectively. The attractive property of 3-D integration is the low transmission energy consumed by TSVs when compared with horizontal links. Thus significant interconnect energy can be saved through aggregating communication intensive tasks within a single stack and restrict the traffic vertically. However, this probably will exacerbate the already severe thermal issue in 3-D ICs because of the increased stack power density. In this paper, we investigate the thermal-constrained task allocation problem for interconnect energy minimization in 3-D homogeneous MPSoCs, and explore the tradeoff between these two factors. We formulate the unique task allocation problem and show it is an instance of a well-known NP-complete problem, i.e., QAP. We then propose an efficient heuristic algorithm QSA to tackle it. Extensive simulation experiments using E3S benchmarks are performed and show that the proposed solution achieve significant amount of interconnect energy savings with negligible impact on peak temperature, when compared with the thermal-balanced method.

ACKNOWLEDGMENT

The authors would like to thank C. Liu for providing many constructive suggestions during the development of the motivation. They would also like to thank the anonymous reviewers for their insightful comments to help make the work more solid and meaningful.

REFERENCES


Yuancheng Cheng (S’11) received the B.Eng. degree from Xidian University, Xi’an, China, in 2003, and the M.S. degree from Harbin Institute of Technology, Harbin, China, in 2005. He is currently pursuing the Ph.D. degree from the Key Laboratory of Computer System and Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China.

His research interests include VLSI design for 3-D integrated circuits considering thermal and defect issues, as well as DFT techniques for 3-D SoC design.

Lei Zhang (M’09) received the B.Eng. degree in Computer Science from University of Electronic Science and Technology of China (UESTC), Sichuan, China, in 2003, and the Ph.D. degree in computer architecture from Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2008.

He is currently an Associate Professor with the State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. His research interests include network-on-chip, 3-D integration, fault-tolerant computing, and multi-core/multi-processor systems.

Yinhe Han (M’06) received the B.Eng. degree from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2001, and the M.Eng. and Ph.D. degrees in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2003 and 2006, respectively.

He is currently an Associate Professor with the Key Laboratory of Computer System and Architecture, Institute of Computing Technology, Chinese Academy of Sciences. His research interests include VLSI/SOC on-chip interconnection, testing, and fault-tolerance.

Dr. Han was the recipient of a Best Paper Award at Asian Test Symposium 2003. He is a member of the ACM, CCF, and IEEE. He was Program Co-Chair of Workshop of RTL and High Level Testing (WRHLT) in 2009, and serves on the Technical Program Committees of several IEEE and ACM conferences, including ATS, GVLSI, etc.

Xiaowei Li (SM’04) received the B.Eng. and M.Eng. degrees in computer science from Hefei University of Technology, Hefei, China, in 1985 and 1988, respectively, and the Ph.D. degree in computer science from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, China, in 1991.

From 1991 to 2000, he was an Assistant Professor and an Associate Professor (since 1993) with the Department of Computer Science, Peking University, Beijing, China. He joined the ICT, CAS in 2000. He is now the deputy (executive) director of the State Key Laboratory of Computer Architecture (ICT, CAS). His research interests include VLSI testing, design verification, dependable computing. He has co-published over 200 papers in academic journals and international conferences, holds 34 patents and 35 software copyrights.

Dr. Li served as Chair of China Computer Federation (CCF) Technical Committee on Fault Tolerant Computing since 2008. He served as IEEE Asian Pacific Regional Test Technology Technical Council (TTC) Vice Chair since 2004. He served as the Steering Committee Chair of IEEE Asian Test Symposium (ATS) since 2011. In addition, he serves on the Technical Program Committee of several IEEE and ACM conferences, including VTS, DATE, ASP-DAC, PRDC, etc. He also serves as member of editorial board of ICST, JETTA, JOLPE, etc.