Wang X, Liu P, Yang M *et al.* Energy efficient run-time incremental mapping for 3-D networks-on-chip. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 28(1): 54–71 Jan. 2013. DOI 10.1007/s11390-013-1312-x

Energy Efficient Run-Time Incremental Mapping for 3-D Networks-on-Chip

Xiao-Hang Wang[1,2] (王小航), Peng Liu[2,*] (刘　鹏), Mei Yang[3] (杨　梅), Maurizio Palesi[4]
Ying-Tao Jiang[3] (蒋颖涛), and Michael C Huang[5] (黄　巍)

[1] *Intelligent Chips and Systems Research Centre, Guangzhou Institute of Advanced Technology, Chinese Academy of Sciences Guangzhou 511458, China*

[2] *Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China*

[3] *Department of Electrical and Computer Engineering, University of Nevada, Las Vegas, Nevada, U.S.A.*

[4] *Faculty of Engineering, Kore University, Catania, Italy*

[5] *Department of Electrical and Computer Engineering, University of Rochester, Rochester, New York, U.S.A.*

E-mail: xh.wang@giat.ac.cn; liupeng@zju.edu.cn; mei.yang@unlv.edu; maurizio.palesi@unikore.it; yingtao@egr.unlv.edu michael.huang@rochester.edu

Received March 5, 2012; revised November 1, 2012.

Abstract 3-D Networks-on-Chip (NoC) emerge as a potent solution to address both the interconnection and design complexity problems facing future Multiprocessor System-on-Chips (MPSoCs). Effective run-time mapping on such 3-D NoC-based MPSoCs can be quite challenging, as the arrival order and task graphs of the target applications are typically not known a priori, which can be further complicated by stringent energy requirements for NoC systems. This paper thus presents an energy-aware run-time incremental mapping algorithm (ERIM) for 3-D NoC which can minimize the energy consumption due to the data communications among processor cores, while reducing the fragmentation effect on the incoming applications to be mapped, and simultaneously satisfying the thermal constraints imposed on each incoming application. Specifically, incoming applications are mapped to cuboid tile regions for lower energy consumption of communication and the minimal routing. Fragment tiles due to system fragmentation can be gleaned for better resource utilization. Extensive experiments have been conducted to evaluate the performance of the proposed algorithm ERIM, and the results are compared against the optimal mapping algorithm (branch-and-bound) and two heuristic algorithms (TB and TL). The experiments show that ERIM outperforms TB and TL methods with significant energy saving (more than 10%), much reduced average response time, and improved system utilization.

Keywords energy efficiency, Networks-on-Chip, multiprocessor System-on-Chips, run-time incremental mapping

1 Introduction

Advance in CMOS technologies keeps driving up the number of processing cores that can be integrated on a single chip. To efficiently interconnect the large number of processing cores in embedded multiprocessor System-on-Chips (MPSoCs), Networks-on-Chip (NoC) have emerged as the mainstream on-chip interconnect architectures. As the number of processing cores continues to increase in such a rapid pace, NoC soon have to explore the third dimension (practically feasible due to the development of 3-D integration) to help meet the demanding integration challenges[1].

Apart from the architectural advantages of 3D NoC, 3-D integration itself alone can considerably reduce the global interconnection length, resulting in shorter interconnection delay, lower power consumption and smaller overall chip area[2]. Of the many existing approaches for 3-D integration, the through silicon via (TSV) technology is particularly suitable for 3D NoC due to its high density of vertical interconnects[1]. Fig.1(a) shows a face-to-back stacked IC enabled by TSV technology, while Fig.1(b) shows a possible floorplan where the tiles are stacked together to form a multi-layered homogeneous MPSoC system. The tiles with the same X and Y coordinates but different Z coordinates (i.e., they

Regular Paper

This work is supported by the National Natural Science Foundation of China under Grant Nos. 60873112 and 61028004, the National Natural Science Foundation of USA under Grant No. CNS-1126688.

The preliminary version of the paper was published in the Proceedings of NPC 2011.

*Corresponding Author

are located at the same corner of different layers) belong to the same column (we call them are in the same pillar). Two tasks are said vertically mapped, if all the tiles that the tasks are mapped onto are from the same column.

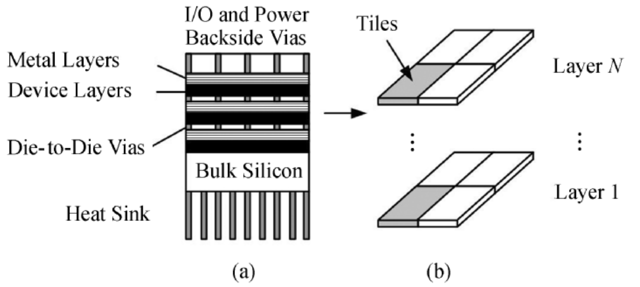


Fig.1. (a) Example of 3-D MPSoC stacking. (b) Floorplan of stacked homogenous dies.

A 3-D NoC-based MPSoC typically has a large number of processing cores available for high level of parallelism. To better utilize these vastly available computation resources, virtualization[3] is applied to allow a single MPSoC to be shared by multiple applications which can be mapped to different regions of the chip online. However, three major challenges are faced by run-time incremental mapping of applications to 3-D NoC.

- The vertical dimension in 3-D NoC needs to be fully explored, as this added dimension helps increase the degree of each router by one and tends to reduce the global wire length. As a result, the energy consumption caused by communications (i.e., energy consumed by the routers and links) can be reduced for a carefully designed mapping algorithm.
- By nature, the behavior of applications running on the NoC is not known a priori. That is, the time of the multiple applications and their run-time behaviors cannot be determined before they actually run on the system. Run-time mapping of applications with no a priori behaviors may cause fragmentation. Unfortunately, these fragmented tiles cannot be utilized by an incoming application if a conventional mapping flow is followed. As a result, any efficient run-time incremental mapping method must reduce the negative impacts of the mapped applications on incoming applications with random time.
- Thermal constraints on components both horizontally aligned at the same layer and vertically aligned across layers have to be considered. Increase in power density and temperature may adversely affect circuit reliability, and this effect becomes more noticeable in 3-D NoC, where the thermal correlation between vertically aligned components is stronger than that between components within the same layer[4]. As a result, increase of power or energy consumption at one single layer (die) may lead to the temperature increase of the whole vertical stacks. In another word, the run-time mapping should also consider the thermal constraint in a vertically aligned column so that the maximum temperature of the chip should be kept below an acceptable threshold.

Unfortunately, there is very little work on run-time 3-D NoC mapping to address the above mentioned challenges. Existing offline 3-D NoC mapping approaches[5] are not suitable for run-time mapping as they do not consider the shapes of the mapped regions and the impact of these mapped regions on the incoming applications that need to be mapped. The 2-D NoC run-time incremental mapping algorithms[6-7] also cannot be directly applied to 3-D NoC because the properties of vertical links available in 3-D NoC are not considered at all.

In this paper, we propose a novel energy efficient run-time incremental mapping (ERIM) framework which maps applications that arrive and exit a 3-D NoC-based MPSoC system in a truly random fashion. ERIM first finds a cuboid region to reduce the impact of mapped applications (i.e., fragmentation) on incoming applications to be mapped, after which the ERIM diverges for communication- and computation-centric applications, as defined by their respective task graph characteristics. For communication-centric applications, ERIM aims to reduce the energy consumption of communication by exploring the vertical dimension. For computation-centric applications, ERIM aims to balance the energy consumption of processors (with running tasks) allocated on each pillar. To improve the system utilization, ERIM also tries to map the tasks with low power/energy consumption to those fragmented tiles while satisfying all the relevant thermal constraints.

To the best of our knowledge, this is the first work on run-time incremental application mapping targeting 3-D NoC-based MPSoCs. The main contributions of this work are two-fold.

- This proposed mapping framework explores the vertical dimension to help minimize the total energy consumption, taking into account of distinct application characteristics.
- A method is developed to minimize the impact of the mapped applications on incoming applications by reducing the possibility of resource fragmentation.

The rest of this paper is organized as follows. Section 2 surveys the related work. Section 3 provides the preliminaries and formally defines the problem to be addressed in this paper. Section 4 presents the observations and a decomposition of the problem. Following the observations obtained from a motivating example, Section 5 presents the run-time mapping algorithm in detail. Section 6 presents the validation methodology, and the results are reported in Section 7. Finally, Section 8 concludes the paper.

2 Related Work

In this section, three relevant topics are reviewed: 1) design and analysis of 3-D NoC, 2) application mapping algorithms for NoC, and 3) thermal management approaches for 3-D NoC.

2.1 Design and Analysis of 3-D NoC

Research in 3-D NoC falls into four major areas: 1) topologies of 3-D NoC, 2) for 3-D NoC router designs, 3) 3-D NoC design methodologies, and 4) analysis of 3-D NoC. A quick survey of relevant topics is provided in this subsection.

Topologies. Matsutani *et al.*[8] proposed a three-layer NoC where each layer has a customized topology to meet its distinct cost-performance requests, and routers in the same pillar but at different layers are connected by crossbar switches. Feero *et al.*[9] compared and analyzed both 3-D mesh-based architectures (symmetric 3-D mesh, stacked mesh, and ciliated mesh) and 3-D tree-based architectures in terms of network performance and energy dissipation.

Router Designs. Kim *et al.*[10] proposed the 3-D dimensionally-decomposed (DimDe) router. The DimDe router features a true 3-D crossbar with two vertical interconnects across all the layers. Short vertical links are preferred when there is only one hop for vertical transmissions among different layers. In addition, the crossbar is decomposed to reduce complexity.

Design Methodologies. Seiculescu *et al.*[11] proposed a synthesis design flow and the SunFloor 3-D tool for 3-D NoC. SunFloor 3-D can take the applications' task graphs as its input and generate a customized topology with optimized energy consumption or hardware area. Pavlidis *et al.*[12] provided an analytical model of zero-load latency and a power model for 3-D NoC.

2.2 Application Mapping Algorithms for NoC

In the literature, a number of IP mapping algorithms have been proposed for 2-D NoC with the objective of minimizing the overall communication power. A summary of these algorithms was provided by Wang *et al.*[13]. However, most of these existing mapping algorithms were actually designed for 2-D NoC and they do not lend themselves well to tackle the run-time incremental mapping for 3-D NoC. Although the branch-and-bound algorithm[14-15] can be extended to 3-D NoC, the extremely long running time required renders its impracticability for run-time mapping. The high complexity of the thermal-aware 3-D offline mapping proposed by Addo-Quaye *et al.*[16] also prohibits its applicability for run-time 3-D NoC mapping.

In terms of run-time incremental mapping for 2-D NoC, several schemes have been proposed. Smit *et al.*[17] proposed a run-time task assignment algorithm for heterogeneous processors where the task graphs are limited to only a small number of tasks. Carvalho *et al.* proposed[18] a dynamic task mapping scheme which aims to improve the performance by minimizing the channel load. Chou *et al.*[6-7] proposed a number of incremental mapping algorithms. The general idea of these run-time mapping algorithms is to find a convex tile region first, after which the incoming application is mapped to that convex region. The convexity of the tile region helps to reduce the distance of communication paths within the region, and it also helps minimize the mapping impact on incoming applications as the remaining tiles form a continuous shape. Note that these mapping algorithms proposed by Chou *et al.*[6-7] may be extended for 3-D NoC. Run-time mapping algorithms in traditional parallel and distributed systems[19] do not suit for the on-chip application mapping as they do not consider the on-chip power consumption.

2.3 Thermal Management Approaches for 3-D NoC

Thermal management for 3-D NoC systems is essential to keep the maximum temperature of the chip under a threshold to avoid thermal hotspots. The existing thermal management techniques can be broadly classified into static thermal control and Dynamic thermal management (DTM) methods.

Static thermal control methods can allocate tasks offline by predicting the run-time temperature of the components, and as so, they can be used for thermal optimization in a global sense.

DTM methods on the other hand, monitor the temperature of each component online and accordingly adjust the frequency or voltage of the circuits for temperature control purposes. Arjomand *et al.*[20] proposed a static task mapping and voltage island planning method, Temperature-Aware Low Power Mapping (TL). TL sorts communication edges by assigning priorities according to the communication volumes (or bandwidths). For communication-centric applications, TL maps the edges with larger communication volumes to vertical links, while for computation-centric applications, TL attempts to map the tiles to the bottom layer. Most DTM methods, like those proposed in [4, 21-22], use dynamic voltage/frequency scaling (DVFS) to control the voltage/frequency of the components so that the temperature can be controlled eventually. However, DVFS requires additional circuits which might incur additional hardware cost. In one study, it has been shown that the DVFS circuit may account for up to 12% of the entire processor area[23].

3 Problem Formulation

In this paper, our study is focused on run-time incremental mapping for crossbar switch-based 3-D NoC due to their superior energy efficiency over symmetrical 3-D structures[8,10]. Multi-clock domains are also assumed for the NoC systems[8]. Fig.2 shows the 3-D NoC architecture, which is composed of N layers of IP cores and routers. Each IP core is indexed by (x, y, z) where $0 \leqslant x \leqslant M_x$, $0 \leqslant y \leqslant M_y$ and $0 \leqslant z \leqslant N-1$; each layer is composed of IP cores interconnected as an $M_x \times M_y$ mesh. At each layer, each IP core is located on a single physical plane and connected to its router through a horizontal link. Assume all the vertical links of the 3-D routers are connected by crossbar switches as the case in [8] and [10]. As shown in Fig.2, each router has six ports, including five horizontal ports, i.e., E, W, N, S and local, and one port connecting to the crossbar. The deterministic XYZ routing is assumed. A tile is defined to be an IP core with the corresponding part of the router on the same layer. The heat sink is placed close to the bottom layer (layer 1).

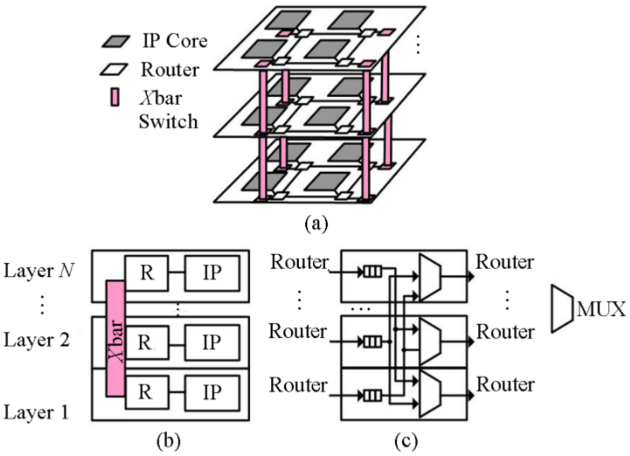


Fig.2. (a) Assumed 3-D NoC architecture. (b) Side view of different layers. (c) Implementation of inter-layer crossbars[8].

3.1 Power Models

In this subsection, the power/energy models for determining dynamic and leakage power for various components of the 3-D NoC systems, including the IP cores, routers and links, are introduced.

3.1.1 Dynamic Power/Energy

The dynamic power of the 3-D NoC systems is composed of both power consumed in computation (i.e., by IP cores) and power consumed in communications (i.e., by routers and links).

1) The dynamic power of IP core Pp can be calculated as[24]

$$Pp = \theta C_p f V_{\rm dd}^2, \tag{1}$$

where θ is the switching activity, C_p is the load capacitance, f is the frequency, and $V_{\rm dd}$ is the supply voltage.

2) The dynamic energy consumption of sending one flit from a source tile to a destination tile comes from two sources: the energy consumed at the routers and the energy consumed on the interconnection links. The average energy consumption of sending one bit of data from tile t_i to tile t_j can be represented as

$$E_{\rm bit}^{t_i,t_j} = \eta E_{\rm Rbit} + \eta_H E_{\rm LHbit} + \eta_V E_{\rm LVbit}, \tag{2}$$

where $E_{\rm Rbit}$ is the energy consumed when transporting one flit through the router, η is the number of routers traversed from tile t_i to tile t_j, η_H and η_V are the numbers of horizontal and vertical links in the communication path, respectively, and $E_{\rm LHbit}$ and $E_{\rm LVbit}$ are the respective energy consumed on the horizontal and vertical links. Following the wire model in [8], one can see that $E_{\rm LHbit} = d_H V_{\rm dd}^2 C_{{\rm wire}H}/2$ and $E_{\rm LVbit} = d_V V_{\rm dd}^2 C_{{\rm wire}V}/2$, where d_H and d_V are the respective lengths of the horizontal and vertical links, $V_{\rm dd}$ is the supply voltage, and $C_{{\rm wire}H}$ and $C_{{\rm wire}V}$ are the wire capacitances of horizontal and vertical links, respectively.

3.1.2 Leakage Power

Leakage power of any component is due to the subthreshold current I_{sub0} and the gate leakage current I_{g0}. The leakage power of the j-th component can be represented as

$$\begin{aligned} P_j^l(t) &= I_{sub0j} + I_{g0j} = A_j \alpha e^{\beta(T(\tau)-T_{\rm ref})} \\ &= Re^{\beta(T(\tau)-T_{\rm ref})}, \end{aligned} \tag{3}$$

where j is the index of a component (i.e., IP cores, routers, and links), A_j is the area of the j-th component, $T(\tau)$ is the temperature of the component, $T_{\rm ref}$ is the reference temperature (for example, 383 K), R is the leakage power at the reference temperature, α and β are the scaling factors between CMOS technology nodes[25].

3.2 Thermal Model

As indicated in [26], heat flow can be modeled as

$$\boldsymbol{C}\frac{d\boldsymbol{T}(\tau)}{d\tau} + \boldsymbol{A}\boldsymbol{T}(\tau) = \boldsymbol{P}(\tau), \tag{4}$$

where τ is the discrete time unit, $\boldsymbol{C}$ is the thermal capacitance matrix of components (routers, processors, etc), $\boldsymbol{A}$ is the thermal conductance matrix, $\boldsymbol{T}(\tau)$ is the temperature vector with $T(\tau)[j]$ representing the temperature of the j-th component, and $\boldsymbol{P}(\tau)$ is the power vector including both dynamic and leakage power. When τ approaches the infinity, we have $\boldsymbol{T} = \boldsymbol{PA}^{-1}$.

Noticeably, 3-D NoC demonstrates thermal heterogeneity at different layers. In general, the layer closer to the heat sink has better cooling efficiency[4]. In addition, as stated in Section 1, 3-D NoC have stronger thermal correlation among components in the same pillar.

3.3 Application and Architectural Models

Each incoming application is represented by its communication trace graph defined below. Virtualization and traffic isolation are assumed[3]. Applications are not allowed to be overlapped.

Definition 1. *A communication trace graph (CTG), $G = (V, E)$ is an undirected graph, where*

- *A vertex/node $v_k \in V$ represents a set of tasks. Tasks are partitioned offline. $EX(v_k)$ is the worst case execution time of vertex v_k and can be found by profiling the program with different input sets. The application is assigned with a completion deadline D. Deadline is set by the user and is application-specific. $P(v_k)$ is the computation power by vertex v_k, which can be obtained from (1). Both $EX(v_k)$ and $P(v_k)$ can be profiled offline.*
- *An edge $e_i = (v_j, v_k) \in E$ represents the communication trace between vertices v_j and v_k. For edge $e_i, \omega(e_i)$ defines the communication volume between vertices v_j and v_k in bits. $\omega(e_i)$ can be obtained by profiling or offline analysis. The data transmission time related to the task can be approximated by the zero load latency model*[27].

A 3-D NoC architecture is modeled with its architecture characterization graph.

Definition 2. *An architecture characterization graph (ACG) $G' = (TI, L)$ is an undirected graph representing a 3-D NoC architecture with N layers (as shown in Fig.2), where TI is the set of free tiles and L is the links between these tiles. In G', each vertex $t_i \in TI$ represents a tile, which has a coordinate (x, y, z) and the index is $z \times M_x \times M_y + y \times M_x + x$. Each edge $l_i \in L = (t_j, t_k)$ represents a link between adjacent tiles t_j and t_k. For link l_i,*

- *$\omega(l_i)$ defines the bandwidth provided on link l_i between adjacent tiles t_j and t_k;*
- *h_{t_j,t_k} is the set of links forming one of the shortest paths from tile t_j to tile $t_k (h_{t_j,t_k} \subseteq L)$.*

Without loss of generality, in this paper, we focus on NoC architectures with $bw(l_i) = B$. We also assume that the IP core with index (0, 0, 0) (referred as the global manager, GM) manages the whole mapping process.

3.4 Problem Description

Using the power, application and architectural models defined above, the 3-D NoC run-time mapping problem is described as: given the CTG $G = (V, E)$ of the incoming application and the ACG $G' = (TI, L)$ of the current 3-D NoC system, of n components (including IP cores, routers and links) in the mapped region, and the temperature constraint $T_{\rm MAX}$, to find a mapping function $M : V \to TI$, with minimum total energy consumption, i.e.,

$$\min\Big(\sum_{\substack{i=0 \\ e_i=(v_j,v_k)\in E}}^{|E|-1} \omega(e_i) \times E_{\rm bit}^{M(v_j),M(v_k)} + \sum_{i=0}^{|V|-1} p(v_k) \times EX(v_i) + \sum_{j=0}^{n-1} P_j^l(\tau) \times D\Big) \tag{5}$$

subject to

$$\forall v_j \in V, M(v_j) \in TI, \tag{6}$$

$$\forall v_j, v_k \in \text{ and } v_j \neq v_k, M(v_j) \neq M(v_k), \tag{7}$$

$$\forall l_m, D \times B \geqslant \sum_{e_i=(v_j,v_k)} \omega(e_i) \times f(l_m, h_{M(v_j),M(v_k)}), \tag{8}$$

$$\max_{i=0 \text{ to } |V|-1} (EX(v_i)) \leqslant D, \tag{9}$$

$$\max_{j=0 \text{ to } n-1} (T(\tau)[j]) \leqslant T_{\rm MAX}, \tag{10}$$

where

$$f(l_m, h_{M(v_j),M(v_k)}) = \begin{cases} 1, & \text{if } l_m \in h_{M(v_j),M(v_k)}, \\ 0, & \text{if } l_m \notin h_{M(v_j),M(v_k)}. \end{cases}$$

The cost function given in (5) has three terms (from left to right): dynamic communication energy, computation energy and energy caused by leakage. Since no DVFS is assumed and all the IP cores are identical (homogenous), each task's computation energy remains the same irrespective of which tile it is mapped to. As such, the second term is viewed as a constant. Conditions given by (6) and (7) ensure that each task should be mapped exactly to one tile and no tile can host more than one task. The inequities given in (8) specify the bandwidth constraint for every link. The inequities given in (9) ensure that the deadline of an application should be satisfied. The inequities given in (10) ensure that the highest temperature of all the components cannot exceed the threshold temperature.

4 Problem Decomposition

4.1 Observations

A number of observations shall be made before we show how to decompose the problem defined in the previous section.

First, the relative contribution of each term in (5) may vary with applications. For applications that involve heavy communications, the first term will dominate the overall energy consumption, and hence, these applications shall be treated differently from those applications involving small amount of communications.

Second, in 3-D NoCs, whenever possible, vertical links are favored when mapping applications; in this way, energy consumption of communication tends to be reduced as opposed to map applications to horizontal links. In Fig.3(a), for instance, the given CTG is mapped to a 3-D NoC with three layers. In Fig.3(a), a 2-D oriented mapping result is shown, where the communication is mapped to the horizontal links. The communication energy cost of this mapping scheme is $E_{\rm Up} = (70 \times E_{\rm Rbit} + 40 \times E_{\rm LHbit})$. Fig.3(b) shows the mapping result which explores the vertical dimension. The communication energy of the lower one is $E_{\rm Low} = (60 \times E_{\rm Rbit} + 10 \times E_{\rm LHbit} + 20 \times E_{\rm LVbit})$. Using Orion simulator[28] (targeting a 90 nm CMOS technology), we have $E_{\rm Up} = 7.4\,{\rm nJ}$, while $E_{\rm Low} = 6.2\,{\rm nJ}$ which is about 0.83x of that of the upper one. From this comparison, we can see that, a mapping exploring the vertical dimension in 3-D NoC is more energy-efficient than a 2-D oriented mapping.

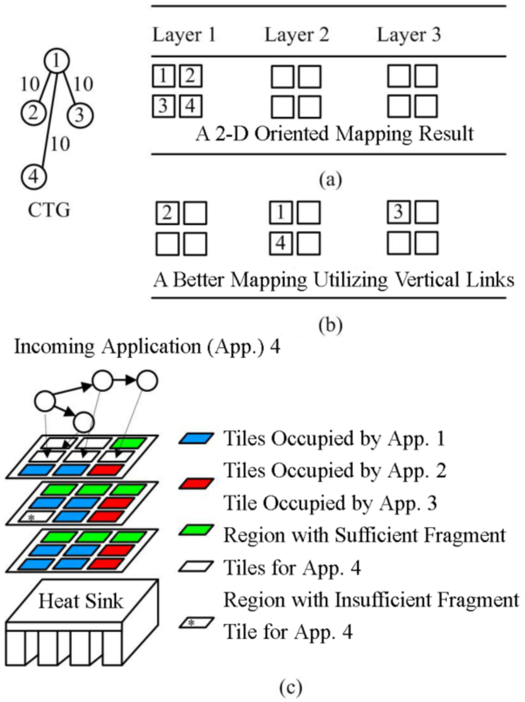


Fig.3. Motivating examples. (a) 2-D oriented mapping result. (b) Mapping result utilizing vertical links. (c) Utilization of fragmented tiles.

Third, if fragmented tiles can still be used, system utilization shall certainly be improved, as one can see in an example give in Fig.3(c). Suppose three applications are currently running in the system. The white boxes are fragmented tiles. When the fourth application arrives, a contiguous region of available tiles at the top layer is found. If the energy consumption of Application 4 is low and the temperature of each pillar does not exceed thermal constraint, mapping this new application to the contiguous tile region at the top layer would improve tile utilization and reduce application waiting time.

4.2 Problem Decomposition

Definition 3. *An application is communication-centric if*

$$\frac{(E_{\rm Rbit} + E_{\rm LHbit}) \times \sum\limits_{\substack{i=0 \\ e_i=(v_j,v_k)\in E}}^{|E|-1} \omega(e_i)}{\sum\limits_{k=0}^{|V|-1} EX(v_k) \times P(v_k)} \geqslant \Delta, \qquad (11)$$

where Δ *is a user-defined constant. Otherwise, the application is computation-centric. The left hand side of* (11) *is the approximated dynamic energy consumption of communication divided by the computation energy consumption of the application. Before mapping, the communication energy cost is not known. Thus, we use the energy consumption of communication where each communication takes only one hop to traverse to approximate the energy consumption of communication before mapping. Because only when the estimated energy consumption of communication accounts a certain percentage* (*e.g., over 20%*) *in the overall energy consumption, the minimization of the energy consumption of communication is meaningful. In the experiments,* Δ *is set to* 0.3.

Definition 4. *A fragment tile is such a tile that it is inside a cuboid tile region, but no task has been allocated to this tile.*

The 3-D NoC run-time mapping problem (defined in Subsection 3.4) is decomposed into five sub-problems as shown in Fig.4. When a new application arrives,

- A cuboid region which can help reduce fragmentation for incoming applications ($P1$) shall be found. Cuboid region is preferred as mapping applications to

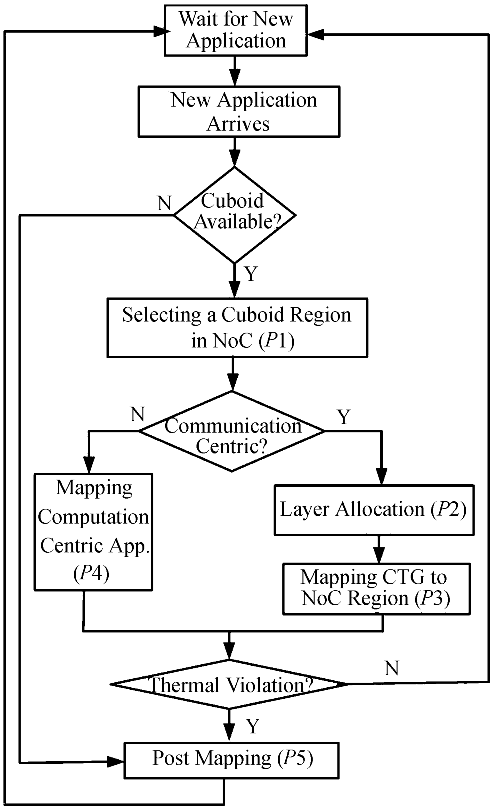


Fig.4. Overall run-time mapping algorithm flow.

processors lining in the vertical dimension can have lower communication energy and can also result in more regular shapes, which is desirable for minimal routing. Based on the nature of the applications (communication-centric or computation-centric), the tasks are allocated accordingly to different layers of the 3D NoC ($P2$). Then communications edges with high traffic volume are mapped to vertical dimension in order to reduce dynamic energy consumption of communication ($P3$).

• If the application is computation-centric, the tasks are mapped such that the energy consumption of processors (running tasks) in each pillar is balanced ($P4$).

• There are two special cases to consider:

Case 1. No cuboid region is available but there exist sufficient number of fragment tiles to form a contiguous region

Case 2. Thermal violation would occur after an initial mapping result is obtained. In both cases, a thermal compliance post-mapping is necessary to: 1) map the application to the fragment tiles to improve resource utilization under the thermal constraints or, 2) map the application to the bottom layer to satisfy the thermal constraint ($P5$).

In the rest of Section 4, these sub-problems, $P1$ through $P5$, will be formulated and the optimization metric for each sub-problem will be presented. The solutions to these sub-problems will be presented in Section 5.

4.2.1 NoC Region Selection ($P1$)

When an application arrives, a cuboid region made of free tiles from multiple layers is identified as the region that this application can be mapped onto. Without loss of generality, this region is assumed to have N sub-regions, one on each layer. To reduce the mapping impact on incoming applications, these sub-regions should be rectangle in shape and each of the sub-regions shall have identical size as possible.

Given the CTG of the incoming application, the ACG of the 3-D NoC and the free tile regions, the goal is to find a cuboid region $SR = \{SR_0, \ldots, SR_{N-1}\}$, where SR_w represents the sub-region of unallocated tiles on layer w, with the objective of

$$\min(|SR_w| \times N - |V|), \text{ for each } w = 0, \ldots, N-1, \quad (12)$$

subject to

$$|SR_w| \times N \geqslant |V|, \quad (13)$$

$$|SR_w| = |SR_{w+1}|, \text{ for } w = 0, \ldots, N-2. \quad (14)$$

Here $|V|$ gives the number of vertices in G and $|SR_w|$ gives the number of tiles in sub-region SR_w. (12) sets the objective to find a cuboid tile region with the minimal number of tiles that are sufficient to handle all the tasks of the incoming application. (13) ensures that the number of free tiles in the cuboid tile region should be larger than or equal to the minimum number of tiles required for the incoming application. (14) ensures that the tile region has cuboid shape.

4.2.2 Layer Allocation ($P2$)

To map a communication-centric application, the basic idea of layer allocation is to find the layers for those tasks which have high communication volumes to be mapped onto. Communication edges with high communication volumes shall be allocated vertically whenever possible in order to reduce energy consumption.

Assume the edges in G are sorted in non-increasing order of their communication volumes (i.e., $\omega(e_i)$). Let $SG \subset G$ be the sub-graph formed with the first $\zeta\%$ edges (ζ is set to 50 in the experiments) in the sorted edge list[13]. The objective is to partition all the vertices in SG into N subsets corresponding to the N sub-regions found in $P1$, such that the communications with large volumes are mapped vertically, i.e., vertices with large communication volumes are better to be placed into separate layers (sub-regions). This layer allocation sub-problem now is defined as follows.

Given the sub-graph SG of the CTG and the sub-regions $SR_0, \ldots, SR_{N-1}$ obtained from $P1$, partition the vertices in SG into N disjoint sets, $SG_0, \ldots$, and SG_{N-1}, with the objective of:

$$\max \sum_{\substack{e_i=(v_j,v_k)\\ v_j \in SG_m\\ v_k \in SG_n\\ m \neq n}} \omega(e_i), \tag{15}$$

subject to

$$SG_m \cap SG_n = \phi, \quad \text{for } m \neq n, \tag{16}$$

$$|SG_m| \leqslant |SR_w|, \quad \text{for each } m = 0, \ldots, N-1. \tag{17}$$

Here we target to maximize the inter-layer communication, such that communications with higher bandwidths will be mapped vertically. (16) ensures that the sets are disjoint. Because the sets correspond to the layers, so they must be disjoint. (17) ensures that the number of tasks in the sub-graph should be less than the tiles in the sub-region.

4.2.3 *CTG to NoC Region Mapping* ($P3$)

With the sub-regions and sets obtained in $P1$ and $P2$, for communication-centric applications, they are then mapped in the way that the total energy consumption is minimized. Leakage power model can be updated if a more precise physical model is available without a major modification of the flow or optimization cost function. As such, in this sub-problem, the goal is focused on minimizing the dynamic communication. This sub-problem is defined as follows: Given the CTG of an incoming application, N vertex sets, $SG_0, \ldots, SG_{N-1}$ and a set of sub-regions $SR = \{SR_0, \ldots, SR_{N-1}\}$, the goal is to find a mapping function $M : V \rightarrow SR$ so that:

$$\min\Bigg(\sum_{\substack{i=0\\ e_i=(v_j,v_k)\in E}}^{|E|} \omega(e_i) \times [\eta E_{\text{Rbit}} + \eta_H E_{\text{LHbit}} + \eta_V E_{\text{LVbit}}] \Bigg), \tag{18}$$

subject to the same conditions given in (6) to (10) as well as

$$\forall v_j \in V, M(v_j) \in SR, \tag{19}$$

which ensures one-to-one task to tile mapping.

4.2.4 *Mapping Computation-Centric Application* ($P4$)

If the application is computation-centric, it would be beneficial to 1) balance temperature and energy consumption of each pillar, and 2) to allocate tasks with higher power consumption to layers closer to the heat sink.

Given the CTG of the incoming application and a set of sub-regions $SR = \{SR_0, \ldots, SR_{N-1}\}$, find a mapping function $M : V \rightarrow SR$ with the objective of,

$$\min\left(\max_{\substack{\text{Vertical}\\ \text{stacks in}\\ SR}} \sum_{\substack{M(v_k)\text{ has the}\\ \text{same } Z\\ \text{coordinate in}\\ SR}} P(v_k) - \min_{\substack{\text{Vertical}\\ \text{stacks in}\\ SR}} \sum_{\substack{M(v_k)\text{ has the}\\ \text{same } Z\\ \text{coordinate in}\\ SR}} P(v_k) \right), \tag{20}$$

subject to the same conditions given in (6) to (10) and (19).

4.2.5 *Post Mapping* ($P5$)

Consider two special cases shown in Fig.4: 1) case 1: no cuboid region is available but there are sufficient number of fragmented tiles to form a contiguous region; and 2) case 2: initial mapping result violates the applicable thermal constraints. The goal of this sub-problem thus is to map/remap the applications such that the thermal constraints have to be satisfied.

Given the CTG of the incoming application and a region of contiguous tiles RG, find a mapping $M : V \rightarrow RG$, with the objective of minimizing the highest temperatures of all the components given below:

$$\min(T(\tau)[i]), \tag{21}$$

subject to the same conditions given in (7) to (11) as well as

$$\forall v_j \in V,\ M(v_j) \in RG, \tag{22}$$

which ensures one-to-one mapping in the region. When filling fragmentation, RG is a contiguous region of fragment tiles. When the thermal constraint is violated, RG represents a contiguous region of tiles at the bottom layer.

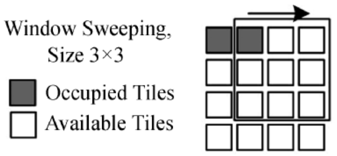
5 Algorithms of ERIM

To solve the run-time mapping problem for 3-D NoC, we propose an energy efficient run-time incremental mapping framework which consists of a set of algorithms, one for solving each sub-problem formulated in Section 4.

5.1 NoC Region Selection Algorithm for $P1$

As discussed in Section 4, a cuboid of NoC tiles has to be found for an incoming application. This problem can be further divided into two steps. First, given the incoming application's CTG G, a window of size $l \times w$ should be found such that $N \times l \times w \geqslant |G|$ and $N \times l \times w - |G|$ is minimized. That is, the shape and size of the sub-region should be chosen to minimize fragmentation. Second, after finding the $l \times w$ matrix, available tiles which form the $N \times l \times w$ cuboid should be found (these tile regions are called sub-regions). A window moving process is used as detailed in Fig.5(a). A window of size $l \times w$ starts from tile (0, 0, 1) and sweeps the first layer. The window sweeps the whole layer until all of the tiles inside the window are found to be not allocated yet. The NoC region selection algorithm is listed in Fig.5(b). The complexity of the algorithm is

$$O(M_x \times M_y + (|TI|/|N| \times M_x \times M_y)) = O(|TI|/|N|)^2.$$



(a)

```
NoC_region_selection($G, Q, G'$)
Input: 1) $G$: the CTG of the incoming application
       2) $Q$: current unallocated tiles in NOC
       3) $G'$: the ACG of the NoC architecture
Output: 1) $SR$: a cuboid the region allocated for the in-
           coming application
Function: find the sub-regions of tiles for the incoming
          application
Procedure body:
{ var: $size\_each\_layer = \lceil |G|/N \rceil$;
  //1. find the window size
  for ($l = 1$; $l < size\_each\_layer/2$; $l++$)
      for ($w = 1; w < size\_each\_layer/2$; $w++$)
        if ($l \times w > size\_each\_layer$){
          break from the $l$-indexed loop; }
  //2. find an $N \times l \times w$ cuboid in $Q$ for $G$,
  //use window sweeping at the bottom layer
  for (each tile $t$ in $Q_0$) {
    //suppose $Q_0$ is the free tile region in the bottom layer
      of 3-D NOC
    //suppose the coordinate of $ti$ is $(i, j, 0)$
    //let $R$ be the rec tangle tile region whose top left and
      bottom right ranges are $(i, j, 0)$ and $(i + l, j + w, 0)$
    //let $R'$ be the rectangle tile region whose top left and
      bottom right ranges are $(i, j, 0)$ and $i + w, j + l, 0)$
    //$m = 0, 1, \ldots, N - 1$
    if (all the tiles in $R$ are available) {
        $SR_m = R$;
        break; }
    else if (all the tiles in $R'$ are available) {
        $SR_m = R'$;
        break; } }
    $Q = Q - SR$}
```

(b)

Fig.5. (a) Window sweeping process. (b) NoC region selection algorithm for $P1$.

Fig.6 shows an example of applying this algorithm. The CTG shown in Fig.6(a) has 16 vertices. The black tiles in Fig.6(b) are the ones that are currently running previously mapped applications. A $3 \times 2 \times 3$ cuboid is selected based on the window sweeping process (Fig.6(b)).

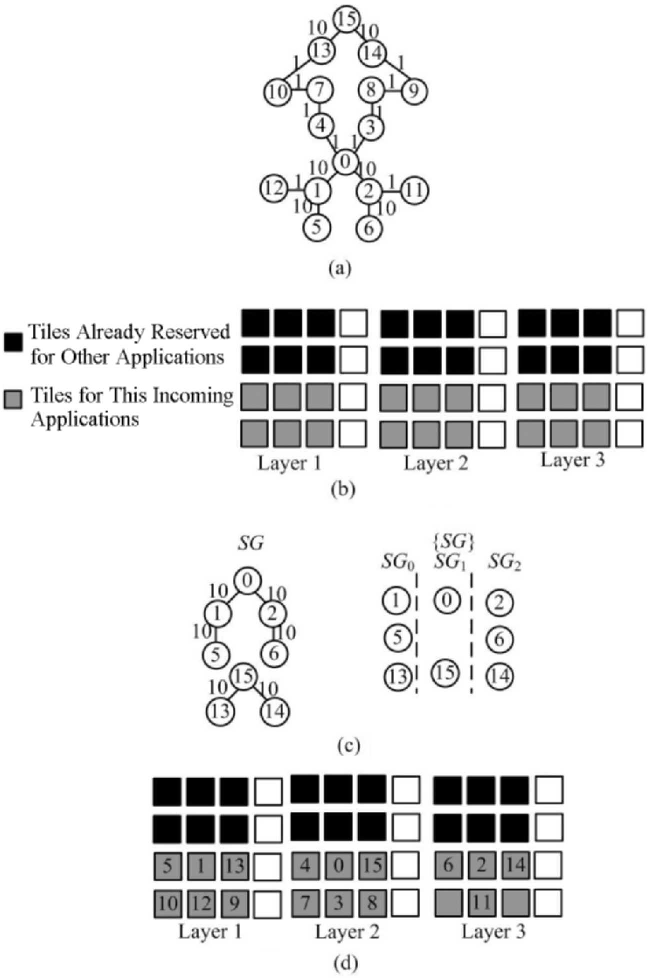
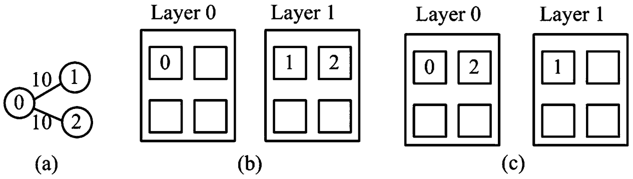

(a)
Tiles Already Reserved for Other Applications
Tiles for This Incoming Applications
Layer 1
Layer 2
Layer 3
(b)
SG
{SG}
SG0
SG1
SG2
(c)
Layer 1
Layer 2
Layer 3
(d)

Fig.6. Example of NoC region selection algorithm for $P1$. (a) CTG. (b) NoC region selection. (c) Sub-graph of CTG and set matching. (d) CTG to NoC region mapping.

5.2 Layer Allocation Algorithm for P2

In the second step of mapping communication-centric applications, the sub-graph of the CTG, i.e., SG, which includes the vertices with high communication volumes, should be partitioned into N sets, corresponding to N sub-regions of the layers obtained from $P1$. The sub-graph SG is formed by the first $\zeta\%$ edges (denoted as SE') in SE (SE is a descending sorted edge list by communication volume). Edges with high communication volumes are preferred to be mapped vertically. There is one exception to this strategy, though, as there is only one vertical link between two tiles (x, y, z) and $(x, y, z + 1)$, $z = 0, \ldots, N - 2$. Consider the following two cases: case 1: a task is allocated to tile (x, y, z) and has two neighbors in layer $z + 1$; case 2: one of the task' neighbors is mapped to the same layer z. The energy consumption of case 1 might be higher than that of case 2.

Fig.7 shows one example. Given the CTG shown in Fig.7(a), if vertices 1 and 2 are mapped to layer 1 (Fig.7(b)), the energy consumption is $(50 \times E_{\text{Rbit}} + 10 \times E_{\text{LHbit}} + 20 \times E_{\text{LVbit}})$. If vertex 2 is mapped to the same layer of vertex 0 (Fig.7(c)), the energy consumption drops to $(40 \times E_{\text{Rbit}} + 10 \times E_{\text{LHbit}} + 10 \times E_{\text{LVbit}})$. This example illustrates, if a task has neighbors already allocated to adjacent layers, it might be beneficial to allocate the remaining neighbors to the same layer. Fig.7(d) lists the layer allocation algorithm with a complexity of $O(|E||N|^2)$.



```
Layer_Allocation($SG, SE', SR$)
Input: 1) $SG$: the sub-graph of the incoming application's
          CTG
       2) $SE'$: the sorted edge list of the first $\zeta$% edge
       3) $SR$: the set of sub-regions for the incoming
          application
Output: 1) $SG_0, \ldots, SG_{N-1}$: the $N$ sets containing the
           vertices in $SG$
Function: Allocate the vertices in $SG$ into the $N$ sub-regions
Procedure body:
{ for (each edge $e_i = (v_j, v_k) \in SE'$) {
    if (neigher $v_j$ nor $v_k$ is allocated to any sub-region) {
    //suppose $v_j$ is the vertex with high degree in $SG_u$,
    $u = 0, \ldots, N-1$
      if ($v_j$ has more than 1 neighbors in $SE'$ and
         $|SG_{\lceil N/2 \rceil}| < SR_0|$)
         Add $v_j$ to $SG_{\lceil N/2 \rceil}$;
      else
         Add $v_j$ to $SG_u$ such that $|SG_u < |SR_0|$, for
         $u = 0, \ldots N-1$; }
    else if (one of the two vertices is allocated) {
    //suppose $v_j$ is allocated to $SG_u$
      if ($v_j$ has neighbors in $SE'$ and is allocated to $SG_{u-1}$ or
         $SG_{u+1}$)
         Add $v_k$ to $SG_u$;
      else
         Add $v_k$ to $SG_m$ such that $|SG_m| < |SR|_0$ with $m \neq u$,
         $|u - m|$ minimized and $v_j$ has no neighbor allocated
         to $SG_m$; } } }
```

(d)

Fig.7. Layer allocation algorithm ($P2$). (a) Sub-graph of an application. (b) Mapping with vertices 1 and 2 on layer 1. (c) Mapping with vertex 1 on layer 1 and vertex 2 on layer 0. (d) Layer allocation algorithm for P2.

Fig.6(c) shows an example of the layer allocation procedure based on the CTG given in Fig.6(a). The sub-graph SG of this CTG is given in Fig.6(a) and the size of sub-regions on each layer is 6. The sorted edge list of the sub-graph is $\{(0,1), (0,2), (1,5), (2,6), (13,15), (14,15)\}$. The final result of the layer allocation for the vertices in SG is given in Fig.6(c).

5.3 CTG to NoC Region Mapping Algorithm for $P3$

After finding the set of tile regions SR and the allocation of the vertices in SG to the N sets, $SR_0, \ldots, SR_{N-1}$, all of the vertices in the communication-centric application should be mapped to SR. In this algorithm, a metric $Dist()$ is used to find the weighted distance of two tiles. The $Dist()$ metric is defined as:

$$Dist(t_a, t_b) = |x_a - x_b| \times E_{\text{LHbit}} + |y_a - y_b| \times E_{\text{LHbit}} + |z_a - z_b| \times E_{\text{LVbit}},$$

where (x_a, y_a, z_a) and (x_b, y_b, z_b) represent the coordinates of tiles t_a and t_b, respectively. As discussed in Section 3, $Dist()$ reflects the energy consumption of each communication edge after the two vertices are mapped. This algorithm examines each edge in SE of the CTG and maps the vertices to minimize the distances between any two.

Fig.8 shows the CTG to NoC region mapping algorithm whose complexity is $O(|E||TI|^2)$. Thus, the total complexity of the run-time incremental mapping algorithm for communication-centric applications is $O(|E||TI|^2)$.

Fig.6(d) shows the mapping result given the communication-centric CTG (Fig.6(a)) based on the results of NoC region selection (Fig.6(b) and 6(c)).

5.4 Mapping Algorithm for Computation-Centric Applications for $P4$

As stated in Subsection 4.2.4, if the incoming application is computation-centric, to balance the temperatures at different layers, the difference of the aggregated energy consumption of the tiles in each pillar should be minimized. This problem is inherently NP-hard[29]. To solve this sub-problem, a heuristics is applied. The vertices are sorted by the energy/power consumption in decreasing order. In each iteration, a vertex from the sorted list is mapped to a pillar where the total energy consumption is minimized. The size of the cycles represents the average energy consumption of the vertices and the bins represent the tiles in the pillars as shown in Fig.9. The content of the bins represents the sum of the energy consumption in the pillar. This algorithm is listed in Fig.10, and the complexity of the algorithm is

```
CTG_to_NoC_Mapping($G, SE, SR$)
Input: 1) $G$: the CTG of the incoming application
       2) $SE$: the sorted edge list of the CTG
       3) $SR$: the set of sub-regions for the incoming
          application
Output: 1) $MAP$: mapping table for each vertex
Function: map each vertex to a tile in $SR$
Procedure body:
{ for (each edge $e_i = (v_j, v_k) \in SE$) {
    if (neither $v_j$ nor $v_k$ are mapped) {
      case 1: $v_j \in SG_u$ and $v_k \in SG_m$ {
            $//u, m = 0, \ldots, N-1$
            $MAP[v_j] = t_a$ s.t. $t_a \in SR_u$;
            //if $v_j$ has more than two neighbors in $SG$,
            //$t_a$ should also have more than two
            //fiee neighbor tiles
            $MAP[v_k] = t_b$ s.t. $t_b \in SR_m$ and $Dist(t_a, t_b)$
            is minimized; }
      case 2: $v_j \in SG_u\{//u = 0, \ldots, N-1$
            $MAP[v_j] = t_c$ s.t. $t_c \in SR_u$;
            //if $v_j$ has more than two neighbors in $SG$,
            //$t_c$ should also have more than two free
              neighbor tiles
            $MAP[v_k] = t_d$ s.t. $t_d \in SR$ and $Dist(t_c, t_d)$
            is minimized; }
      case 3: $\{MAP[v_j] = t_e$ s.t. $t_e \in SR$;
            //if $v_j$ has more than two neighbors in $SG$,
            //$t_e$ should also has more than two free
              neighbor tiles
            $MAP[v_k] = t_f$ s.t. $t_d \in SR$ and $Dist(t_e, t_f)$
            is minimized; } }
    else if (only one vertex is mapped) {
      //suppose $v_j$ is mapped to the $t$
      case 1: $v_k \in SG_m\{//m = 0, \ldots, N-1$
            $MAP[v_k] = t_g$ s.t. $t_g \in SR_v$ and $Dist(t, t_g)$
            is minimized; }
      case 2: $\{MAP[v_k] = t_h$ s.t. $Dist(t, t_h)$ is minimized; }
    } } }
```

Fig.8. CTG to NoC region mapping algorithm for $P3$.

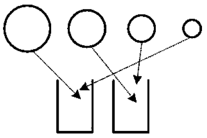
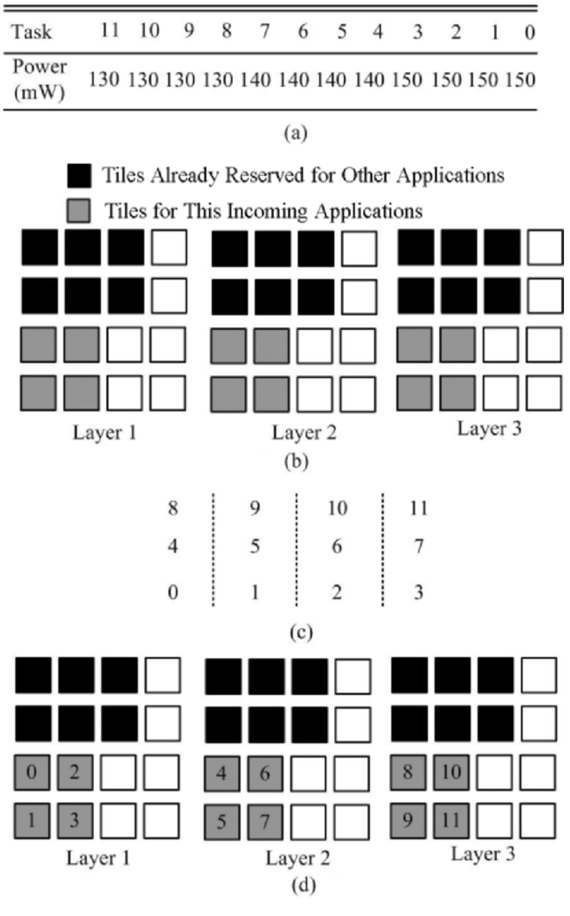


Fig.9. Analogue of energy balance.

$O(|V||TI|)$, assuming $|V| = |TI|$. Thus, the overall complexity of the run-time incremental mapping algorithm for computation-centric applications is $O(|TI|/|N|)^2$.

```
Map_Computation_Centric_Tasks($G, SR$)
Input: 1) $G$: the CTG of the incoming application
       2) $SR$: the set of sub-regions for the incoming
          application
Output: 1) $MAP$: mapping table for each vertex
Function: map each vertex to a tile in $SR$
Procedure body:
{ var: $SV$: sorted vertex list in decreasing order by average
             enery consumption
  for (each vertex $v_i \in SV$) {
    $MAP[v_i] = t_a$ s.t. $t_a \in SR$ and the task energy
    consumption of $t_a$'s vertical column is minimal; } }
```

Fig.10. Map computation centric tasks algorithm ($P4$).



Task	11	10	9	8	7	6	5	4	3	2	1	0
Power (mW)	130	130	130	130	140	140	140	140	150	150	150	150

(a)
■ Tiles Already Reserved for Other Applications
▨ Tiles for This Incoming Applications
Layer 1
Layer 2
Layer 3
(b)

8	9	10	11
4	5	6	7
0	1	2	3

(c)
0
2
1
3
4
6
5
7
8
10
9
11
Layer 1
Layer 2
Layer 3
(d)

Fig.11. (a) Average task power consumption. (b) Tile regions selected for the incoming application. (c) Allocation of tasks to pillars with balanced power/temperature. (d) Mapping of the computation-centric application.

Fig.11 shows an example of mapping a computation-centric application. Fig.11(a) shows the average power consumption of the tasks. Fig.11(b) shows the regions returned by NoC region selection by $P2$. Fig.11(c) shows the allocation of tasks to pillars with balanced power/temperature. In this example, the total average power consumption of the tasks in each column is the same. Fig.11(d) shows the mapping of tasks to the tiles found by $P4$.

5.5 Thermal Compliant Post Mapping Algorithm for $P5$

Thermal compliant post mapping considers two special cases: 1) If there is no cuboid region but a contiguous region with sufficient fragment tiles is found, then the incoming application is mapped to the fragment tiles under thermal constraint; 2) if the thermal constraint is violated after the above mapping processes, a contiguous region is found with available tiles at the layer closest to the heat sink and the application is mapped to the region under thermal constraint.

Fig.12 shows the thermal compliant post mapping algorithm whose complexity is $O(|V||TI|)$, assuming $|V| = |TI|$. In this algorithm, RG is a contiguous region of fragment tiles for the first case, or a contiguous region of tiles at the bottom layer for the second case.

```
Post_Mapping($G$, $RG$)
Input: 1) $G$: the CTG of the incoming application
       2) $RG$: the set of tiles forming a contiguous region
Output: 1) $MAP$: mapping table for each vertex
Function: map each vertex to a tile in $RG$
Procedure body:
{
  for (each vertex $v_i \in V$) {
      $MAP[v_i] = t_a$ s.t.
      1) $t_a \in RG$,
      2) the total power consumption of $t_a$'s vertical column
         is below $P_{\rm MAX}$,
      3) $t_a$ is closest to the tiles which host $v_i$'s neighbors;
      if (cannot find such a tile)
          wait until some tiles are released; }
}
```

Fig.12. Thermal compliant post mapping algorithm for $P5$.

Fig.13 shows an example of thermal compliant post mapping. Suppose the tiles found by previous communication-centric mapping algorithm for $P3$ fail to satisfy the thermal constraint, e.g., the temperature of at least one router exceeds the threshold temperature (by checking the total dynamic energy/power consumption of each vertical volume and comparing it with the power threshold $P_{\rm MAX}$). In Fig.13(c), a new contiguous tile region is selected until the mapping result satisfies the thermal constraint.

6 Validation Methodology

6.1 System Configuration

The 4 × 4 × 3 3-D NoC system is simulated using a SystemC-based cycle-accurate NoC simulator modified by the Noxim[①] simulator. Communication and task execution are separated in simulation. Each processor node in the simulator is replaced by a packet generator with the same execution time (treated as delay) and power trace of tasks profiled offline. The Hotspot[26] tool is integrated into the NoC simulator to perform thermal simulation. The 6 × 6 router model and the crossbar model (as shown in Fig.2) are implemented in hardware and synthesized using Synopsis Design Vision 2009.06 with TSMC 90 nm CMOS technology library. The power consumption of the crossbar model is evenly distributed to the routers at all layers. The router power consumption is 62 mW. In all the experiments, the value of Δ (Definition 3) is set to 0.3 in this paper and could be adjusted. $P_{\rm MAX}$ in Fig.12 is 3 W (calculated by running thermal simulations on Hotspot with the thermal threshold $T_{\rm MAX}$ set to 383 K). The value of β in (3) is 0.024 for 90 nm from the website[②].

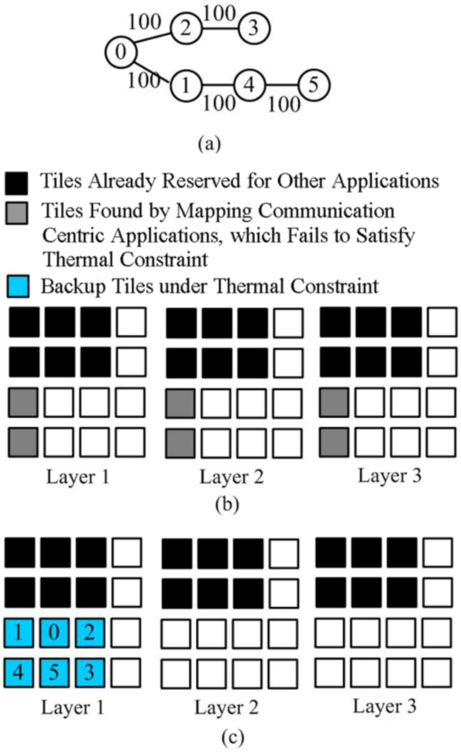

100
100
100
100
100
0
1
2
3
4
5
(a)
Tiles Already Reserved for Other Applications
Tiles Found by Mapping Communication Centric Applications, which Fails to Satisfy Thermal Constraint
Backup Tiles under Thermal Constraint
Layer 1
Layer 2
Layer 3
(b)
1
0
2
4
5
3
Layer 1
Layer 2
Layer 3
(c)

Fig.13. Thermal compliant post mapping algorithm when the thermal constraint is violated. (a) Task graph of a communication-centric application. (b) Tile region found by algorithm mapping communication-centric application. (c) Mapping of the cool application.

The 3-D NoC system parameters are listed in Table 1. Part of the parameters are from [4] and [30]. Fig.14(a) shows the package model assuming a face-to-back stacking. The package setup parameter is adopted from [4]. Fig.14(b) shows the floorplan of each individual tile. Each IP core (P) and local memory (M) are synthesized together (denoted as P+M). The area of router (R) and P+M are synthesized using Design

[①] Noxim. http://noxim.sourceforge.net, Nov. 2011.

[②] Managing power in 45 nm and 65 nm dDesigns. http://www.dianzichan.com/anonymous/ic/arm07conf_mgm_pwr45nm.pdf.

Vision tool 2009.06 with TSMC 90nm CMOS technology library. The effective thermal conductivity considering via density is given by [4],

$$K_{\text{eff}} = \rho_{\text{via}} K_{\text{via}} + (1 - \rho_{\text{via}}) K_{\text{layer}}, \tag{23}$$

where K_{via} and K_{layer} are the respective thermal conductivity values of the via material and region without any via, and ρ_{via} is the via density.

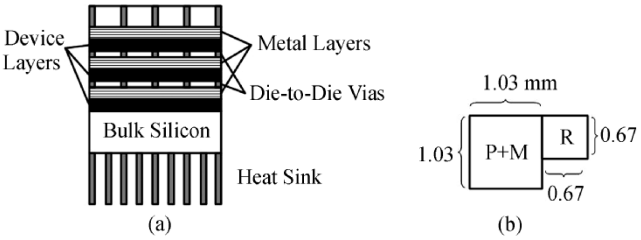


Fig.14. (a) Hotspot package model. (b) Floorplan of each tile at 90 nm of an embedded NoC system.

Table 1. Configuration of the System Characterization

Flit size			75 bits
Network size			4 × 4 × 3
Number of virtual channels			2
NoC buffer depth			8 flits
Local memory size			16 KB
Ambient temperature			318 K
NoC frequency (MHz)			500
Processor frequency (MHz)			500
Die size (mm^2)			0.774
C_{Hwire} (fF/mm)			212.12
C_{Vwire} (fF/mm)			600
Via size (um^2)			20 × 20
	Theral cond (W/mK)	Heat cap. (J/m^{3}K)	Depth (μm)
Active layer	160.11	1.66e+6	50
Interface layer	6.83	3.99e+6	10
Heat sink	400.00	3.55r+6	6 900

In the experiments, ERIM is compared against two known mapping algorithms.

- The first mapping algorithm is Temperature balance (TB), modified from [21]. In TB, the tasks are sorted in decreasing order by their power consumption. The tasks are mapped so that the power consumption of each pillar is balanced.
- The second mapping algorithm is Temperature-Aware Low Power Mapping (TL) in [20]. TL sorts communication edges by assigning priorities according to the communication volume (or bandwidth). For communication-centric applications, TL maps the edges with larger communication volume to vertical links. For computation-centric applications, TL maps the tiles to the bottom layer.

6.2 Benchmark Characteristics

The experiments are conducted with two sets of benchmarks. The first set consists of random applications with their task graphs generated from TGFF[31]. The performance metrics of ERIM, TB, and TL are compared, including 1) energy efficiency and temperature difference, 2) fragmentation results in terms of average response time and system utilization. Table 2 lists the parameters of the random benchmarks. For each random experiment, five experiments are conducted with the averaged results reported.

Table 2. Parameters of Random Benchmarks

Parameters	Value
Number of vertices	[3, 35]
Number of edges	[5, 40]
Average execution time (cycles)	[200, 1200]
Number of total applications	10
Task power consumption (mW)	[20, 120]
Communication volume (bits)	[100, 100 K]
CMOS technology point	90 nm

The second suite consists of five real applications, as tabulated in Table 3. Among the real applications, T264 decoder[③] is a typical video processing application. AES encoder/decoder are typical encryption applications[32]. The automotive and consumer from E3S[④] are used in literatures as [7]. The matrix multiplication is a kernel program for many scientific applications. OFDM is a typical streaming communication application[33].

Table 3. Characteristics of Real Benchmarks

Benchmark Name	Number of Task	Number of Edge	Template
T264 dec	4	7	Comp. centric
Automotive	24	22	Comp. centric
Consumer	12	12	Comp. centric
AES decoder	16	16	Comm. centric
AES encoder	16	16	Comm. centric
OFDM transmitter	10	9	Comp. centric
OFDM receiver	16	19	Comp. centric
Matrix multiplication	5	4	Comm. centric

For all these applications, tasks are partitioned offline. The execution time and power consumption of the tasks from applications are profiled offline using Wattch/SimpleScaler[34] (configured as a PISA architecture) which can report power traces of the tasks.

[③] T264. http://www.codeforge.cn/article/96237, Nov. 2012

[④] E3S. http://ziyang.eecs.umich.edu/~dickrp/e3s/, Nov. 2011.

The communication volume among the tasks is manually analyzed.

The T264 decoder has four pipeline stages, and the input to the decoder is a frame of foreman with quarter common intermediate format (QCIF). We follow the task partition in [32] and profile the corresponding tasks in the code of the AES decoder and encoder, which are partitioned into four pipeline stages and duplicated to 16 tasks. The tasks of automotive and consumer from E3S are obtained from MiBench⑤. The matrix multiplication is partitioned into five tasks.

We also include the task graphs of OFDM transmitter and receiver from [33] into our benchmark suites. The execution time of tasks of OFDM is from [33] and the power consumption trace is set to a uniform random sequence with the average set to be 120 mW.

7 Experimental Results

To evaluate the proposed ERIM framework, extensive experiments have been conducted.

7.1 Experiments on Random Benchmarks Testing Energy Efficiency

To show the effects on energy efficiency (both dynamic and leakage energy) of the three mapping schemes, the results on single application and multiple applications are reported.

7.1.1 *Mapping a Single Application*

In this set of experiments, only one single application is mapped. An optimal mapping algorithm, Branch and Bound (BNB)[15], is used as the baseline algorithm. The normalized energy savings of BNB over ERIM, TB, and TL are compared using random applications with different numbers of task vertices. The parameters of the experiments are listed in Table 2 except that only a single application is generated.

Fig.15 shows that overall the degradation of ERIM over BNB is less than 10%. Even with the application with a large number of task vertices, the mapping quality of ERIM is still quite close to what is obtained from BNB. As a contrast, the mapping result of TB over BNB declines sharply for the applications with larger number of task vertices. The reason is due to the fact that TB does not consider the communications among tasks in its mapping process. When the number of task vertices increases, the number of communication edges also increases. Failing to explore this feature, as the case in TB, will produce poor mapping results. Also, TL has higher energy consumption than ERIM. Because TL does not maximize the inter-layer communication, i.e., communications might be mapped horizontally so that the distance of communication paths increases.

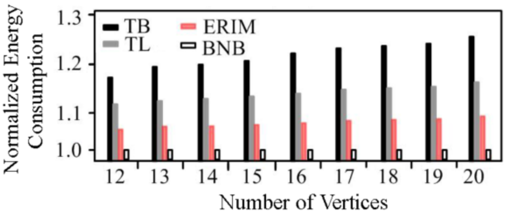


Fig.15. Comparison of energy consumption of BNB, ERIM, TB, and TL over different number of vertices. The energy consumption of ERIM, TB and TL is normalized over that of BNB.

7.1.2 *Mapping Multiple Applications*

In this set of experiments, multiple applications are mapped at run-time to the 3-D NoC system. In each experiment, five random applications are generated.

Fig.16 reports the average energy savings of ERIM over TB and TL for multiple applications with different average traffic volumes. The applications with communication volume less than or equal to 10 K bits are considered to be computation-centric. The application parameters are listed as in Table 2 except that the communication volume is ranged as shown in Fig.16. From Fig.16 we can see that, as the traffic volume increases, ERIM is more efficient over TB and TL. Due to the same reason for Fig.15, the mapping results generated by TB have high communication energy. TL is not efficient when the CTG becomes complex, i.e., the average energy consumption of communication is high and the number of communication edges is large (i.e., the average degree of vertices is large).

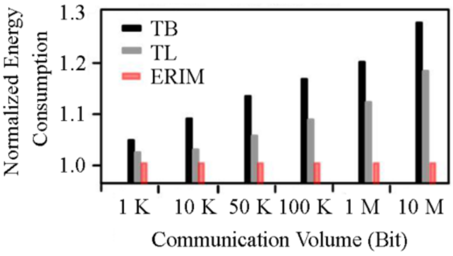


Fig.16. Comparison of energy consumption of ERIM, TB and TL, over different traffic volume. The energy consumption of TB and TL is normalized over that of ERIM. The applications with traffic volume less than or equal to 10 K bits are considered to be computation-centric.

To evaluate the impact of the number of communication- and computation-centric applications on the energy consumption, a set of experiments are

⑤MiBech Version 1.0. http://www.eecs.umich.edu/mibench/, Nov. 2011.

conducted. A sequence of 20 random applications are mapped, among which, 20%, 40%, 60%, 80% of the applications are computation-centric, respectively. The application parameters are listed in Table 2. From Fig.17, we can see that the more communication-centric applications, the lower energy consumption resulted from ERIM. The reason is similar to that of Fig.16, i.e., ERIM is more efficient in mapping communication-centric applications.

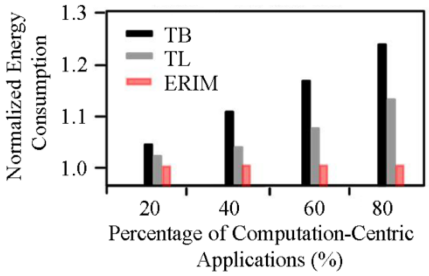


Fig.17. Comparison of energy consumption of ERIM, TB and TL, with different number of computation- and communication-centric applications. The energy consumption of TB and TL is normalized over that of ERIM.

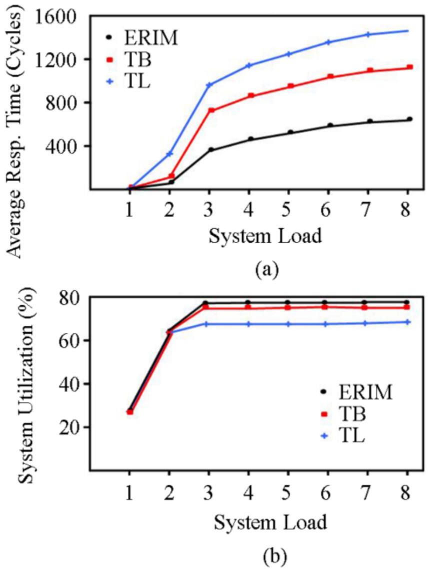
7.2 Experiments on Fragmentation Test with Random Benchmarks

In this set of experiments, the system load is used as the independent variable. System load is defined as the ratio of the mean service time divided by mean inter-arrival time of jobs[19]. In the experiments, we measure:

- *Average response time*: the time between an application arrival and the starting time when tasks are mapped to available tiles.
- *System utilization*: the percentage of tiles that are utilized over time.

The experiment parameters are shown in Table 2. In Fig.18(a), the average response time of the three mapping algorithms are compared. From this figure, we can see that ERIM has the least average response time while TL has the largest average response time. The reason can be that, ERIM and TB try to form cuboids for incoming applications, thus reducing the impact on incoming applications. With the thermal compliant post mapping algorithm, ERIM has less average response time than TB by filling fragmentation. TL, on the other hand, does not consider the impact on incoming applications, thus resulting in severe fragmentation.

In Fig.18(b), the system utilizations of the three mapping schemes are shown. The peak system utilization is reached when the system load is set to 3. Due to the similar reason, the system utilizations of ERIM and TB are close while the result of ERIM is slightly higher than that of TB. Again, TL has the worst system utilization due to fragmentation.



(a)

(b)

Fig.18. (a) Average response time of applications over different system loads. (b) System utilization over different system loads.

Fig.19 shows an example situation of fragmentation. Suppose that a new application arrives with 20 tasks. As there are only 9 tiles available, the incoming application has to wait until 20 tiles are available. For the system configuration in Fig.19, the system utilization is about 81%.

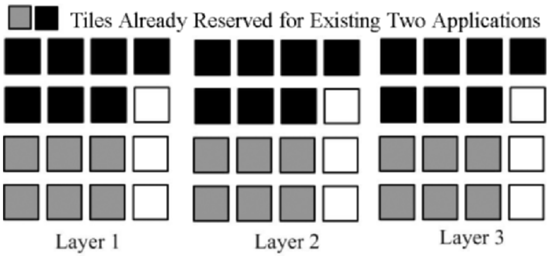


Fig.19. Example showing the fragmentation situation. Suppose an application with 20 tasks arrives. There are only 9 tiles available, so the incoming application has to wait until sufficient tiles are released.

To evaluate the effectiveness of the thermal compliant post mapping step ($P5$) and its impact on system utilization, a set of experiments are conducted. Three groups of random benchmarks are generated with average task power consumption set as 1 W, 2 W and 2.5 W, respectively. The system load is set to 5. The thermal threshold is set to 383 K[25]. Five applications are mapped incrementally to the systems. The percentage of violation is the number of cases that fail to conform to the thermal constraint divided by five.

From Table 4, we can see that, when the average

Table 4. Comparison of ERIM with and Without Thermal Compliant Post Mapping ($P5$)

Average	Highest Temperature (K)		Percentage	System
Task Power (W)	Without $P5$	With $P5$	of Violation (%)	Utilization (%)
1.0	363	363	0	79
2.0	404	383	85	63
2.5	425	383	100	47

task power consumption increases, if no thermal compliant post mapping is used, thermal violation occurs more frequently. For example, without $P5$, when the average task power is 2.5 W, the highest temperature is 425 K. However, with $P5$, the highest temperature is 383 K. This is because, without $P5$, the tasks are still mapped vertically, thus the power density increases. On the other hand, with $P5$, the tasks are mapped to the layer close to the heat sink, which helps reduce the power density. Table 5 also shows the impact of thermal violation on the overall system utilization. When the average task power is high, e.g., 2.5 W, thermal violation is observed in each single experiment. Thus, the system utilization decreases drastically. The reason is that, when thermal violated occurs, the tasks can only be mapped to the free tiles in the first layer. Once the tiles in the first layer are all occupied, the incoming application has to wait. Although there are available tiles in other layers, allocating tasks to these tiles will incur thermal violation. Thus, the inability to use these available tiles contributes to lower system utilization.

7.3 Experiments on Real Benchmarks

In this subsection, experiments using benchmarks from real applications are performed. The following performance metrics are accounted in the experiments: 1) running time of the mapping algorithms themselves, and 2) data used by the manage core controlling the other tiles. The system load is set to 5, with the following performance metrics redefined.

- The *average response time* is defined as the time between an application arrival and the starting time when tasks are mapped to available tiles plus the running time of the mapping algorithm.
- The *total energy* is defined to be the total energy consumption of the application plus the energy consumed by the mapping algorithm.

The proposed mapping algorithm runs at the global manager (GM) core where no other tasks shall be scheduled to it. The running time of ERIM, TB and TL is about 50 000, 45 000, and 46 000 cycles, respectively, obtained by running the code on a PISA-configured Wattch/SimpleScalar simulator. The power consumption of the three algorithms is 122 mW, 120 mW and 122 mW, respectively. A control message is composed of 14 bits, including 6 bits for tile index coding, 1 bit flag for indication of allocation status (1 for allocated and 0 for not allocated), and 7 bits representing the tile temperature value. Such communication overhead has been accounted in calculating the energy consumption.

First, the average response time and system utilization of the three mapping schemes are compared for real applications (Fig.20). The system load is set to 5. The average response time of ERIM, TB and TL is 56 ms, 64 ms and 113 ms, respectively. The system utilizations of the three schemes are 76%, 72% and 65%, respectively. When the power consumption, execution time, and communication volume of the mapping schemes are considered, ERIM still has less average response time than the other two.

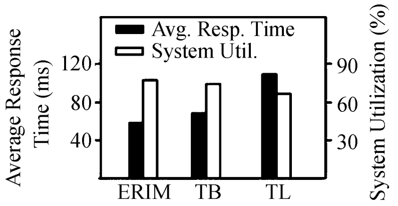


Fig.20. Average response time and system utilization of our algorithm over TB and TL on real applications at 90 nm technology.

Second, the energy savings of ERIM over TB and TL are reported in Table 5. In Table 5, we can see that, ERIM can be more energy efficient when mapping communication-centric applications. The energy savings of ERIM over TB and TL are over 12%. The maximum temperature of ERIM is observed to be 337 K.

Table 5. Energy Savings of ERIM over TB and TL on Real Applications

Algorithms	Total Energy Saving (%)	Max. Temperature Difference (K)
TB	16.8	2
TL	12.1	−1

8 Conclusions

In this paper, we have proposed an energy-efficient run-time incremental mapping (ERIM) framework for 3-D NoC-based MPSoCs. The characteristics of ERIM are threefold: 1) minimizing the total energy consumption, 2) reducing the impact on incoming applications, and 3) satisfying the thermal constraints when admitting a new application. ERIM distinguishes two types of applications, the communication-centric and computation-centric ones. For both types of applications, cuboid tile regions are selected. For communication-centric applications, the tasks are mapped vertically to reduce the energy consumption of communication. For computation-centric applications, the total energy consumption of each pillar is balanced. To improve system utilization, selected applications are mapped to fragmented tiles. Our experimental results show that ERIM achieves more significant energy saving than two other mapping schemes, TB and TL, for both synthetic and embedded MPSoC applications. The average response time and system utilization of ERIM are also much better than TB and TL.

References

[1] Pavlidis V F, Friedman E G. 3-D topologies for networks-on-chip. *IEEE Trans. Very Large Scale Integration Systems*, 2007, 15(10): 1081-1090

[2] Davis W R, Wilson J, Mick S *et al.* Demystifying 3D ICs: The pros and cons of going vertical. *IEEE Design and Test of Computers*, 2005, 22(6): 498-511.

[3] Triviño F, Sánchez J L, Alfaro F J, Flich J. Virtualizing network-on-chip resources in chip-multiprocessors. *Microprocessors and Microsystems*, 2011, 35(2): 230-245.

[4] Zhu C, Gu Z, Shang L, Dick R P, Joseph R. Three-dimensional chip-multiprocessor run-time thermal management. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 2008, 27(8): 1479-1492.

[5] Addo-Quaye C. Thermal-aware mapping and placement for 3-D NoC designs. In *Proc. Int. SoC Conf.*, Sept. 2005, pp.25-28.

[6] Chou C L, Marculescu R. Run-time task allocation considering user behavior in embedded multiprocessor networks-on-chip. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 2010, 29(1): 78-91.

[7] Chou C L, Ogras U Y, Marculescu R. Energy- and performance-aware incremental mapping for networks on chip with multiple voltage levels. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 2008, 27(10): 1866-1879.

[8] Matsutani H, Koibuchi M, Amano H. Tightly-coupled multi-layer topologies for 3-D NoCs, In *Proc. Int. Conf. Parallel Processing*, Sept. 2007, Article No.75.

[9] Feero B S, Pande P P. Networks-on-chip in a three-dimensional environment: A performance evaluation. *IEEE Trans. Computers*, 2009, 58(1): 32-45.

[10] Kim J, Nicopoulos C, Park D *et al.* A novel dimensionally-decomposed router for on-chip communication in 3D architectures. In *Proc. Int. Symp. Computer Architecture*, June 2007, pp.138-149.

[11] Seiculescu C, Murali S, Benini L, De Micheli G. Sunfloor 3D: A tool for networks on chip topology synthesis for 3-D systems on chips. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 2010, 29(12): 1987-2000.

[12] Pavlidis V F, Friedman E G. 3-D topologies for networks-on-chip. *IEEE Trans. Very Large Scale Integration Systems*, 2007, 15(10): 1081-1090.

[13] Wang X, Yang M, Jiang Y, Liu P. A power-aware mapping approach to map IP cores onto NoCs under bandwidth and latency constraints. *ACM Trans. Architecture and Code Optimization*, 2010, 7(1): 1-30.

[14] Land A H, Doig A G. An automatic method for solving discrete programming problems. *Econometrica*, 1960, 28(3): 497-520.

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

[16] Addo-Quaye C. Thermal-aware mapping and placement for 3-D NoC designs. In *Proc. Int. SoC Conf.*, Sept. 2005, pp.25-28.

[17] Smit L T, Smit G J M, Hurink J L *et al.* Run-time assignment of tasks to multiple heterogeneous processors. In *Proc. the 5th Progress Embedded System Symp.*, Oct. 2004, pp.185-192.

[18] Carvalho E, Calazans N, Moraes F. Heuristics for dynamic task mapping in NoC-based heterogeneous MPSoCs. In *Proc. the 18th IEEE/IFIP Int. Workshop. Rapid System Prototyping*, May 2007, pp.34-40.

[19] Lo V, Windisch K J, Liu W, Nitzberg B. Noncontiguous processor allocation algorithms for mesh-connected multicomputers. *Trans. Parallel and Distributed Systems*, 1997, 8(7): 712-726.

[20] Arjomand M, Sarbazi-Azad H. Voltage-frequency planning for thermal-aware, low-power design of regular 3-D NoCs. In *Proc. the 23rd Int. Conf. VLSI Design*, Jan. 2010, pp.57-62.

[21] Zhou X, Yang J, Xu Y, Zhang Y, Zhao J. Thermal-aware task scheduling for 3D multi-core processors. *IEEE Trans. Parallel and Distributed Systems*, 2010, 21(1): 60-71.

[22] Chao C H, Jheng K Y, Wang H Y *et al.* Traffic- and thermal-aware run-time thermal management scheme for 3D NoC systems. In *Proc. the 4th ACM/IEEE Int. Symp. Networks-on-Chip*, May 2010, pp.223-230.

[23] Truong D, Cheng W, Mohsenin T *et al.* A 167-processor 65 nm computational platform with per-processor dynamic supply voltage and dynamic clock frequency scaling. In *Proc. IEEE Symp. VLSI Circuits*, June 2008, pp.22-23.

[24] Liu Y, Yang H, Dick R P, Wang H, Shang L. Thermal vs energy optimization for DVFS-enabled processors in embedded systems. In *Proc. Int. Symp. Quality Electronic Design*, March 2007, pp.204-209.

[25] Srinivasan J, Adve S V, Bose P, Rivers J A. The impact of technology scaling on lifetime reliability. In *Proc. Int. Conf. Dependable Systems and Networks*, June 28-July 1, 2004, pp.177-186.

[26] Huang W, Ghosh S, Velusamy S *et al.* HotSpot: A compact thermal modeling methodology for early-stage VLSI design. *IEEE Trans. Very Large Scale Integration Systems*, 2006, 14(5): 501-513.

[27] Dally W J, Towles B. Principles and Practices of Interconnection Networks. San Francisco, USA: Morgan Kaufmann, 2004.

[28] Kahng A, Li B, Peh L S *et al.* Orion 2.0: A fast and accurate NoC power and area model for early-stage design space exploration. In *Proc. Conf. Design, Automation & Test in Europe*, Apr. 2009, pp.423-428.

[29] Garey M R, Johnson D S. Computers and Intractability: A Guide to the Theory of NP-Completeness. New York, USA: WH Freeman, 1979.

[30] Lin M, Gamal A E, Lu Y C, Wong S. Performance benefits of monolithically stacked 3D-FPGA. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 2007, 26(2): 216-229.

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

[32] Yang Y S, Bahn J H, Lee S E, Bagherzadeh N. Parallel and pipeline processing for block cipher algorithms on a network-on-chip. In *Proc. the 6th Int. Conf. Information Technology: New Generations*, Apr. 2009, pp.849-854.

[33] Delorme J, Houzet D. A complete 4G radiocommunication application mapping onto a 2D mesh NoC architecture. In *Proc. North-East Workshop. Circuits and Systems*, June 2006, pp.93-96.

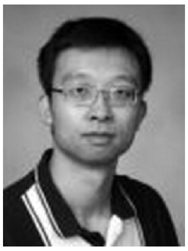
[34] Brooks D, Tiwari V, Martonosi M. Wattch: A framework for architectural-level power analysis and optimizations. In *Proc. the 27th Int. Symp. Computer Architecture*, June 2000, pp.83-94.

Xiao-Hang Wang received the B.Eng. and Ph.D. degrees in communication and electronic engineering from Zhejiang University, China, in 2006 and 2011 respectively. He is currently a postdoc in computer science at Zhejiang University. His research interests include NoC system simulation, routing algorithm, and parallel programming for NoC-based systems.

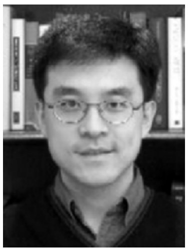
Peng Liu received the B.Eng. and M.Eng. degrees in optical engineering from Zhejiang University, in 1992 and 1996, respectively, and the Ph.D. degree in communication and electronic engineering from Zhejiang University, in 1999. He has been an associate professor with the Information Science and Electronic Engineering Department, Zhejiang University, since 2002. His research focuses on embedded processor microarchitecture, multiprocessor system-on-chip architectures, on-chip interconnection networks, parallel programming, and VLSI design.

Mei Yang received her Ph.D. degree in computer science from the University of Texas at Dallas, USA, in Aug. 2003. She has been an associate professor in the Department of Electrical and Computer Engineering, University of Nevada, Las Vegas since 2010. Her research interests include computer architectures, networking, and embedded systems.

Maurizio Palesi received the M.S. and Ph.D. degrees in computer engineering from the University of Catania, Italy, in 1999 and 2003, respectively. Since November 2010 he is an assistant professor at Kore University, Italy. Dr. Palesi serves on the editorial board of VLSI Design Journal as an associate editor since May 2007. He has served as a guest editor for the VLSI Design Journal, the International Journal of High Performance Systems Architecture, Elsevier MICPRO Journal and ACM Transactions on Embedded Computing Systems. He is in the technical program committee of several IEEE/ACM International Conferences. He has been the co-organizer of the four editions of the International Workshop on Network-on-Chip Architectures (from 2008 to 2011). Dr. Palesi is a member of the European Network of Excellence on High Performance and Embedded Architecture and Compilation (HiPEAC).



Ying-Tao Jiang received his Ph.D. degree in computer science from the University of Texas at Dallas in Aug. 2001. He joined the Department of Electrical and Computer Engineering, University of Nevada, Las Vegas in Aug. 2001. He has been an associate professor since Aug. 2007. His research interests include algorithms, computer architectures, VLSI, networking, nano technologies, etc.



Michael C Huang received the B.S. degree in computer science and engineering from Tsinghua University, Beijing, in 1994, the M.S. and the Ph.D. degrees in computer science from University of Illinois at Urbana-Champaign in 1999 and 2002, respectively. From 1994 to 1997, he was a lead architect in building a 32-processor hierarchical shared-memory multiprocessor research prototype. He joined the faculty of the Electrical and Computer Engineering Department at University of Rochester in 2002. His research interests include various aspects of high-performance computer architecture, such as processor micro-architecture, communication and memory substrate, reliability, and energy-efficient and complexity-effective design. He is a recipient of the NSF CAREER Award and IBM Faculty Award, and a member of the ACM and IEEE.