Economizing TSV Resources in 3-D Network-on-Chip Design

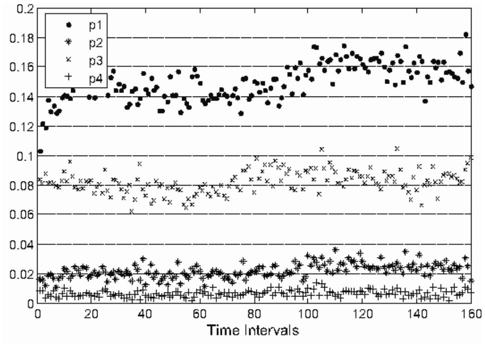
Ying Wang, *Student Member, IEEE*, Yin-He Han, *Member, IEEE*, Lei Zhang, *Member, IEEE*, Bin-Zhang Fu, *Member, IEEE*, Cheng Liu, *Student Member, IEEE*, Hua-Wei Li, *Senior Member, IEEE*, and Xiaowei Li, *Senior Member, IEEE*

***Abstract*— The confluence of 3-D integration and network-on-chip (NoC) provides an effective solution to the scalability problem of on-chip interconnects. In 3-D integration, through-silicon via (TSV) is considered to be the most promising bonding technology. However, TSVs are also precious link resources because they consume significant chip area and possibly lead to routing congestion in the physical design stage. In addition, TSVs suffer from serious yield losses that shrink the effective TSV density. Thus, it is necessary to implement a TSV-economical 3-D NoC architecture in cost-effective design. For symmetric 3-D mesh NoCs, we observe that the TSVs bandwidth utilization is low and they rarely become the contention spots in networks as planar links. Based on this observation, we propose the TSV sharing (TS) scheme to save TSVs in 3-D NoC by enabling neighboring routers to share the vertical channels in a time division multiplexing way. We also investigate different TS implementation alternatives and show how TS improves TSV-effectiveness (TE) in multicore processors through a design space exploration. In experiments, we comprehensively evaluate TSs influence on all layers of system. It is shown that the proposed method significantly promotes TE with negligible performance overhead.**

***Index Terms*— 3-D integration, multicore, network-on-chip (NoC), through-silicon via (TSV).**

I. Introduction

LARGE-SCALE computing systems are putting more and more emphasis on interconnects because communication plays an increasingly essential role in their performance and power efficiency. Therefore, network-on-chip (NoC) is proposed to address the communication scalability challenges in single-chip computing systems [1], [2]. As another promising solution to mitigate the interconnect problem at physical layer, the emerging 3-D integration technology has recently been extensively investigated. The 3-D integrated circuits (3-D ICs) not only provide shorter link latency, but also enable the compact integration of devices manufactured in incompatible process technology [3], [4]. Therefore, with the benefits of both, the combination of 3-D integration and NoC soon becomes one of the most effective ways to tailor the on-chip communication fabrics for multicore designs like CMPs [5]–[7]. Much work has been done on the architecture of 3-D NoCs to better exploit the 3-D integration in high performance interconnections [5], [8], [9].



(a)

Fig. 1. TSV utilization and conflict probability.

In 3-D chip stacking, TSV interconnection is considered a competitive solution because it provides much denser and shorter interconnects compared with alternatives, such as wire bonding, microbump, and contactless [10]. Although TSVs provide a short wiring distance and are able to boost the performance of NoC-based systems, they also bring some significant challenges. First, TSV pads are needed for bonding purpose, which consume considerable chip area in all layers [11]. For example, low density TSV pitch is >50 um [12]. Even when high density TSVs with 16-um pitch are employed [12], the aggregated area overhead could be very high for a typical 3-D NoC. Besides, TSV is quite different from planar metal wires, and it is not expected to scale with feature size at the same rate [13]. Thus, the relative cost could possibly increase faster when the size of transistors and wires keep shrinking. Second, the footprint occupied by TSVs can potentially extend the circuit wiring and induce physical design troubles like routing congestion in layout [14], [15]. It is beneficial to reduce the TSV consumption at early design stage to ease the pressure in physical design flow of placement and routing. Finally, TSVs must be saved for higher utilization since their fabrication in current technology still suffers a relatively low yield, which decreases sharply with the manufacture density [16], [17]. For all these reasons, it is essential to economize TSV resources in 3-D designs to achieve better utilization without compromising performance.

Manuscript received September 24, 2012; revised April 14, 2013 and October 19, 2013; accepted January 8, 2014. Date of publication April 11, 2014; date of current version February 20, 2015. This work was supported in part by the National Basic Research Program of China (973) under Grant 2011CB302503, in part by the National Natural Science Foundation of China (NSFC) under Grant 61376043, Grant 61202056, Grant 61173006, and Grant 61221062, and in part by the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant XDA06010401.

The authors are with the SKL Computer Architecture, ICT, Chinese Academy of Sciences, Beijing 100190, China, and also with the University of Chinese Academy of Sciences, Beijing, 100190, China (e-mail: wangying2009@ict.ac.cn; yinhes@ict.ac.cn; zlei@ict.ac.cn; fubinzhang@ict.ac.cn; liucheng@ict.ac.cn; lihuawei@ict.ac.cn; lxw@ict.ac.cn).

Digital Object Identifier 10.1109/TVLSI.2014.2311835

Though for traditional 2-D NoCs, cost-effective NoC design approaches have been intensively investigated in prior work, they mainly focus on reducing router area or planar links to deliver better silicon and power efficiency through customization, heterogeneity, time division multiplexing (TDM), or topology concentration [18]–[21]. Our design methodology aims to effectively allocate the TSV resources to NoCs without raising noticeable performance degradation and other side effects on routing algorithms. Prior techniques like link serialization can potentially save the TSV resources [11]. However, serialization will greatly decrease vertical bandwidth and degrade light-load latency, especially when an aggressive serialization policy is applied.

In this paper, we observe that in symmetric 3-D mesh NoCs with a lower vertical dimension than horizontal dimensions, treating TSVs like planar links and partitioning them equally among all routers will not lead to an efficient resource usage. We observe that a typical 3-D NoC mostly has a smaller depth than its width and length because of design factors like thermal issue, cost-effectiveness concerns of chip providers, and power delivery constraint [22]–[24]. In such flattend 3-D NoCs, TSVs are underutilized and less likely to be the bandwidth bottleneck compared with planar links. This is a key observation and assumption in this paper.

Based on the observation, we propose a TSV sharing (TS) scheme to economize TSV resources. The basic idea is clustering adjacent routers together to share a common TSV bundle in a TDM way. With the router-clustering mechanism, we propose a heterogeneous TS method for NoC-based many core processors. The TS is not only able to reduce planar chip area occupied by TSV pads, but also improve TSVs utilization significantly with negligible performance degradation in terms of network latency and system throughput. Specifically, we make the following contributions.

1) We propose TS as a supporting substrate to economize TSV resources. The detailed μ-architecture and circuit design are described and evaluated for performance and hardware overhead.
2) We explore the design space to study the performance potential of TS, and investigate the feasibility of varied design configurations with different routing algorithms, sharing degrees, and heterogeneous clustering methods.
3) We integrate TS into a full-system simulator and evaluate the system performance to investigate its impacts on upper layers of system. By weighing the effects of TSV saving against possible performance loss, we evaluate a variety of design choices. In analysis, we also prove that the TS will not impair the performance symmetry of mesh NoCs and keep transparent to high levels.

The rest of this paper is organized as follows. Section II presents preliminaries, motivations, and related work about 3-D NoCs. Section III discusses the TSV economization methods and implementation. Section IV shows the experimental mechanism, evaluation results, and analysis. Finally, conclusions are drawn in Section V.

II. Background and Motivation

A. 3-D NoC Architecture

As the evolution of router microarchitecture and NoC topology slows down, the emerging 3-D stacking technology opens up a new horizon for on-chip interconnects design. Several prior work have investigated the architecture of 3-D NoCs. Feero and Pande [8] compared a variety of 3-D architectures to their 2-D counterparts. Results show that 3-D NoCs offer more competitive energy-efficiency for communication. Li *et al.* [5] introduced a 3-D NoC design by hybriding a common NoC router with a bus link in vertical dimension. Kim *et al.* [9] proposed a dimension decomposition method to optimize 3-D NoC area cost and performance as well.

Although many different 3-D NoC architectures have been proposed, the well-studied paradigms mainly fall into three categories, i.e., symmetric, hybrid, and true 3-D fabric NoC [9]. Symmetric 3-D NoC is a simple extension of 2-D NoC like those in mainstream CMPs by adding two additional ports in 2-D router design [25], i.e., upward and downward ingress/outgress ports. Because such a NoC is symmetric in all routing direction, symmetric 3-D mesh NoC is easy to implement, test, and verify [26]. Besides, it directly supports the legacy IPs and classic routing algorithms in 2-D ICs. Therefore, the symmetric 3-D NoC is preferred for many cases [8], [27]. This paper focuses on the symmetric 3-D mesh structure.

B. 3-D Integration Technology and Through Silicon Via Manufacture

In 3-D integration, through silicon via (TSV) is known as a promising enabling technology for chip bonding [3], [4], [10]. TSV manufacture process undergoes several process stages from via etching, electroplating, bonding, and assembly. There are two typical bonding methods: 1) face-to-back and 2) back-to-back. After the die/wafer bonding stage, TSVs etched in multiple chip layers are connected by the metal pads on substrates. Because of the complex process technology, the fabricated TSVs not only face common open or bridge defects caused by process variation, but also have particular defects caused by pads-misalignment or mechanical stress [28]. Thus, the fabrication of TSVs faces severe yield losses in different manufacture stages [17], [29]. Particularly, the yield of TSVs drops dramatically with the increasing manufacture density. Worse still, testing the integrity of TSVs is more expensive than conventional testing [30], [31]. Technologies from Tezzaron, IMEC, MIT, and IBM are all reported to be challenged by the yield loss of TSVs [16]. It is revealed that for the latest IBM process technologies, in chip bonding stage only, the chip yield is only about 23% with 1.00E+5 TSVs or 87% with 1.00E+4 TSVs [16]. To boost the yield of 3-D chips, there are a variety of solutions proposed by researchers, most of which save 3-D chips from TSV failures at the expenses of nominal TSV density by applying redundancy or remapping [32]–[34].

C. Necessity of TSV-Economization

It has been discussed that the manufacture yield loss decreases the actual TSV density in 3-D NoCs. Besides, the mentioned manufacture losses, TSVs also suffer from critical off-line defects induced by mechanical stress [28], thermal stress at the metal–dielectric interface, and other wear-out mechanisms [35], [36]. If we rule out those TSVs used as redundancy for reliability purpose, the available TSVs that can be used as functioning links in NoCs are not so abundant given that the planar area intentionally left out for TSV manufacture is limited. In addition, NoC links have to contend for space with the vertical wirings used for other purposes, i.e., power delivery pillars, clock tree wirings, or even thermal vias inserted to increase thermal conductivity of stacks, which make the resources even more precious [37], [38].

For example, state-of-the-art process is able to produce ~8-μm^2 vias on a pitch of 16 μm [12]. When we consider a mediate $8 \times 10 \times 2$ mesh network with 39 flit width unidirection links, it costs about 8K TSVs to maintain the topology, which occupy the area equivalent to the size of a processing core. If we take into account of cooling pillars, vertical power delivery networks (consuming several thousands of TSVs [39]), clock trees, and redundant TSVs, and the area budget for vertical wirings will be at least doubled.

Conclusively, it is necessary to rebudget the links before we extend the planar interconnects to 3-D designs for better performance. The objects of this paper is to reduce the TSV consumed by processor interconnects without violating other design constraints, and make the TSVs sharing scheme totally transparent to upper system layers.

D. Motivation of TS in 3-D NoCs

To save TSVs in a 3-D network design, TSV serialization is potentially a feasible approach. For instance, a 4-to-1 serialization of TSV interconnects can save more than 70% of TSV area. However, serialization will decrease vertical link bandwidths, i.e., at least 75% for 4-to-1 serialization [11]. In such cases, network throughput decreases a lot, especially when the traffic pattern is nonuniform. Even worse, serialization inevitably increases the zero-load latency, and thus deteriorates network performance.

Instead of decreasing the bitwidth of vertical links, we seek to make the best of the TSVs and improve their utilization. In NoCs, bandwidth resources are considered overprovided in some cases [19]. Similarly, in our analysis, there is such a phenomenon that the vertical links in symmetrical 3-D NoCs are over-provisioned and thus underutilization of TSVs occurs in large networks with limited chip layers. To prove it, we first give an in-depth analysis on TSVs utilization in 3-D mesh NoC. The $4 \times 4 \times 2$ 3-D mesh NoC is implemented in our NoC simulator for investigation. The simulated design employs dimension order routing and wormhole flow control. For details, each router in the network has two stage pipelines and two virtual channels with 8-flit depth in each input port. To testify our assumption, we run a cycle-accurate simulation that lasts one million cycles. For each 5K-cycle interval, we record both the TSVs occupancy rate and the probability of conflicts between the transmission activities on different TSV bundles belonging to two adjacent routers. Please notice that the TSV conflicts only projects, the probability of event that both TSVs in neighboring routers is carrying flits at the same cycle. It is not a real flit conflict but an indication of temporal correlations between packet transmissions on different TSV bundles. The conflict probability $\Pr^{\text{tsv}}_{i,j}(t_k)$ of adjacent TSV-i and TSV-j can be defined as

$$\begin{cases}\Pr^{\text{tsv}}_{i,j}(t_k) = \Pr\{P^{\text{tsv}}_i(t_k) | P^{\text{tsv}}_j(t_k)\} \cdot P^{\text{tsv}}_j(t_k)\\ \qquad\qquad = P^{\text{tsv}}_i(t_k) \cdot P^{\text{tsv}}_j(t_k)\\ \text{Manh}(i, j) = 1\end{cases} \tag{1}$$

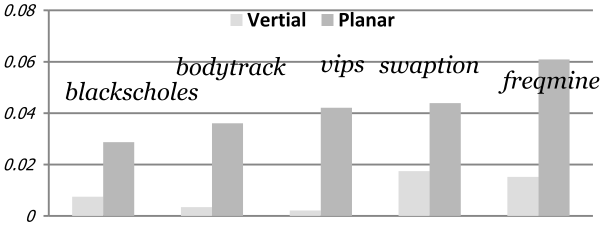


Fig. 2. Conflict probability of links of parsec workloads.

where $\Pr^{\text{tsv}}_{i,j}(t_k)$ denotes the probability of event that both TSV-i and TSV-j are transmitting flit at time t_k, and $\text{Manh}(i, j) = 1$ means the Manhattan distance between node-i and node-j is 1. $P^{\text{tsv}}_i(t_k)$ denotes the flit arrival rate on TSV-i. Also, the events on TSV-i and TSV-j are assumed to be independent.

Fig. 1 shows the average TSV utilization and conflict probabilities under uniform traffic. $p1$ and $p2$ indicate the TSV utilization rate and conflict rate, respectively, under average injection rate of 0.3, while $p3$ and $p4$ represent those values under average injection rate of 0.1. It is shown that average TSV utilization is low, especially under light network load. Even when the injection rate goes up to 0.3, the TSVs still keep idle more than 80% of the time. In addition, TSVs in adjacent nodes are seldom occupied by flits at the same time.

Fig. 2 plots the conflicting probability on adjacent TSV bundles and that of ordinary planar links for real workloads. The detailed simulation setup is described in Section IV. The conflict on planar links is defined in a similar way to indicate the probability of concurrent flit traversal on the planar links of two adjacent routers. It is shown that there is a wide gap between the conflicting rates of vertical and horizontal links. The gap makes it unsuitable to adopt traditional concentration topology that indiscriminately merges adjacent routers up for resources utilization purpose [20]. Based on above analysis, we conclude that TSV in the symmetric 3-D mesh NoC is underutilized significantly and data transmission on different TSVs also rarely concurs. Therefore, we develop a mechanism allowing neighboring nodes to share a single TSV bundle.

1) Explanation About the Conflict Gap: As we have said, TSV underutilization occurs in large network with limited chip layers, which is the basic assumption of this paper. For flattened 3-D NoCs like $W \times L \times D$ mesh (assuming $D < W < L$), the bisection width is $W \times D \times B$ (B is bandwidth of a single channel), which is the aggregate bandwidth of the cross section (X–Z or Y–Z plane) along Z dimension in the 3-D NoC. If we cut across the 3-D NoC along X/Y dimension, the aggregate bandwidth of the cross section is $W \times L \times B$, larger than bisection width. That means the vertical channels in Z dimension are not the bottleneck in such flattened 3-D NoCs considering that the bisection width is a good indication of communication capability in network.

A lower vertical dimension is a realistic assumption in a typical 3-D NoC, because there are a lot of design issues that limit the growth of layer counts [22]–[24]. It is surveyed that for the 36-core processors assumed in [24], two layers is the most effective solution for 3-D integrations in advanced technology. Thus, this paper focuses on 3-D NoCs with a lower vertical dimension than horizontal dimensions. When the network goes larger while layers cannot grow at the same

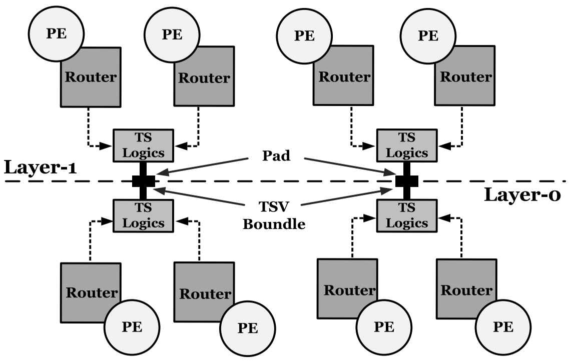


Fig. 3. 3-D mesh NoC with TS.

pace, or aggressive clock schemes are adopted on TSVs [9], this phenomenon of bandwidth utilization gap is even more evident.

To sum it up, we focus on improving the TSV utilization of typical flattened 3-D mesh NoCs, and we want to make sure that the NoC topology will not be ruined by our technology, which keeps the planar traffics and bandwidths uninfluenced. Second, we make sure that no complexity be introduced into the routing layer so that legacy routing algorithms and NoC μ-architectures can be adopted without any modification.

E. Discussion on Important Related Work

For traditional planar NoCs, some cost-effective NoC design approaches have been intensively investigated in prior work [18], [19]. Concentrated topology or high radix routers have also been proposed to better utilize the NoC resources by merging adjacent routers to serve multiple cores [20], [21]. Our method is enlightened by the idea of resources concentration. However, such techniques could influence the planar topologies of 3-D NoCs and induces hotspots by shrinking the bandwidth of planar channels. Comparatively, we witness the utilization gap between vertical and planar links, so we aim to merge only vertical links instead of other parts of network resources like entire routers or planar channels through high radix router, and to keep the planar topology and routing intact.

There are also some designs that enable PEs to share certain NoC components in a timing multiplexing mode (TDM) to deal with the resources shortage in interconnection [40], [41]. The TDM virtual circuits are often used to meet the QoS requirements of application. The TS also makes the routers reuse a single TSV bundle in different time slots. However, TS is parallel to the TDM techniques. The mature slot computation and assignment strategies in TDM virtual circuit can be applied to TSV arbitration for QoS purpose.

III. TS Architecture and Implementation

A. Basic TS Architecture

The general idea of TS scheme in 3-D NoC is shown in Fig. 3. Four routers in subregions of network are grouped together by the common links. Therefore, physical vertical channels become shared resources in a cluster and could be granted to any requesting router before transmission. Whenever there is a packet transmitted from one layer to the other, it has to acquire the grant from the TS arbiter at first. Then, the packet with the grant will traverse TS logics in current layer and flow through TSVs to other layers. Finally, the data packet will be forwarded to corresponding router through TS logics in destination layer.

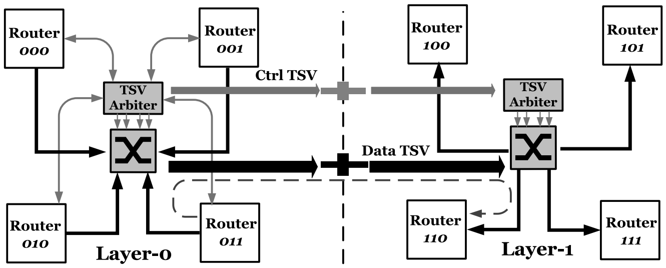


Fig. 4. TS architecture.

To achieve the effect described above, there are still other issues needed to be addressed. First, we do not want to change the original 3-D mesh topology or add special routing constraint to network, so that dimension order routing algorithm can be adopted directly in routers. This is to minimize the side effects to network brought by sharing scheme. Second, any data transmission from one layer to the other layer needs to traverse longer path to sharing logics before reaches the downward layer. Though there is clear evidence that considerable timing slacks exist in TSV traversal stage because of its timing and electrical characteristic [9], one single clock cycle may not be sufficient after the insertion of our sharing logics in a conservative design. Thus, to maintain the operation frequency of the router, an extra pipeline stage is inserted in vertical transmission. We will describe how the additional pipeline and the TS logics are integrated into router design in Section III-B.

As shown in Fig. 3, another opportunity provided by TS is that the physical bypass in TSV crossbar can potentially boost the network performance. For example in Fig. 4, with the help of speculation, a packet from router-010 in upper layer is allocated the bypass channel two hops before the destination node in bottom. Then, it can skip the horizontal hop to router-011 and be directed to reach router-111 through TSV by correctly sending signal bits to demux in bottom layer. This bypass channel can potentially make a shorter routing path between nodes and its integration into mesh NoCs is well studied in work like EVC [42]. The impacts of such bypasses are going to be estimated in Section IV.

B. Router Modification and Implementation

Fig. 4 shows the proposed basic TS architecture. Since TSVs and control logics are symmetrical in all direction and layers, we only show the sharing logics and links for unidirection TSVs in this figure. In order to support TS scheme, we need to make minor modifications to conventional routers. A typical router usually consists of buffers, the routing computation unit (RC), the virtual channel allocator (VA), the switch allocator (SA), and the crossbar. When a flit arrives at a router, it will then be stored into the buffer. Then, RC detects data in buffer queue and calculates output port for each packet according to the routing algorithm. After that, VA simply receives the head flit and reserve buffer entries in next router based on its content. Finally, SA in the router

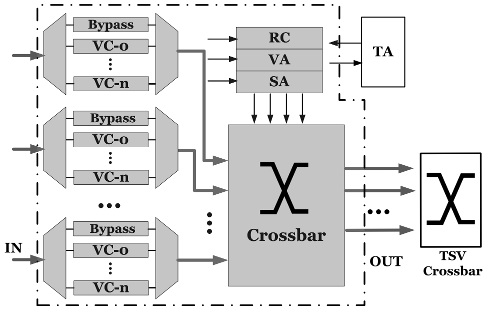


Fig. 5. Modified router architecture.

performs arbitration to decide which requesting flit can take over a particular output port.

For a TS-supported router, the vertical interconnects are considered as shared media instead of private links dedicated to each router. When data from a router is going to be sent down to the router in bottom layer, it has to be granted the vertical channel by TSV arbiter (TA) before it leaves the crossbar of router. Only when the data packet finally receives the TSV grant from TA and the switch grant from SA, it can traverse the data TSVs to the bottom layer. Meanwhile, the TA grant signal is also sent to the bottom layer through Ctrl TSVs to downstream crossbar for inputs multiplexing. With the sharing mechanism, the only TSV bundle in a router group is split into virtually independent links, preserving the routing topology.

Fig. 5 shows a typical router structure with TS logics. The parts in dashed line represent original router architecture. In addition, the only added component, TA, is an arbiter compatible with commonly used arbitration strategies, such as round robin, cyclic priority, and oldest first.

Our assumed two-stage router supports the widely used preallocation and speculation techniques proposed in [43]. In the first stage, router can perform look-ahead RC, preallocate the VCs, and switch in parallel. Fortunately, TSV arbitration can also be performed together with RC, VA, SA, crossbar traversing stages, moving TA off the critical path. Fig. 6 shows the critical data path of assumed router μ-architectures. The dash lines separate the components in different stages. Blue shaded blocks in the figure are the additional logics for TS.

It can be seen from Fig. 6 that the routed packet needs grant signals from VA, SA, and TA at the same time, while VA grant and SA grant are sufficient for data transmission in conventional design. Meanwhile, as preallocation results are stored in registers for next cycle, they are removed from potential critical path to ensure the concurrent operation of VA, SA, TA, and crossbar.

C. Influence of TS on Physical Design

In order to avoid IC routing and placement problems in physical layout design stage, TA is located in between the sharing nodes. In this case, the wiring latency from request to grant can be ignored. However, the data path is prolonged and additional pipeline stage might be needed in case of timing violation, as shown in Fig. 6. It is not easy to decide how much delay will be added, because the IC layout has significant influence on horizontal wiring latency. We assume that critical path latency for original router is t_0, the planar wiring latency between neighboring routers is t'_0, horizontal link latency from router to TA is $t_1(t_1 \leq t'_0/\sqrt[2]{2} < t_0)$, TSV latency is $t_2(t_2 \ll t_0)$, which is less than 10 ps [9], mux latency is $t_3(t_3 \ll t_0)$, demux latency is $t_4(t_4 \ll t_0)$, and buffer write latency is $t_5(t_5 < t_0)$. Then, we know the original link traversal latency is $\text{co} = (t'_0 + t_5)$ $(co < t_0)$, and the aggregated latency between adjacent routers of different layers is approximately $\text{cp} = (2 \cdot t_1 + t_2 + t_3 + t_4 + t_5)$. How many stages should be added depends on the footprints of PEs. In most cases, cp can be safely considered to be less than $2 \cdot t_0$ (according to the logic synthesis results of mux and demux), which means it takes one more clock cycle for a flit to traverse the TSVs than planar links. In our experiments, we assume that an extra cycle is inserted into the TSV traversal stage. At the same time, to avoid additional wiring latency (mainly to t_1), we restrict the sharing domain to be compact, e.g., no more than four adjacent routers, so that we have $t < cp \leq 2 \cdot 0$. TSV latency, t_2, depends on the selection of TSV technologies. Since there are many technologies under development, we choose only three representative TSVs that are described in Table I. With the data of resistance and capacitance values per um^3 from [36], we can use the Elmore-based delay model to safely prove that the latency of the chosen TSVs is under 120 ps, which is already a more conservative value than the values in [9] and [38]. However, the TSV transmission latency is still negligible compared with the clock period in typical NoCs. Therefore, we assume that the choice of TSV technologies has no direct influence on network performance.

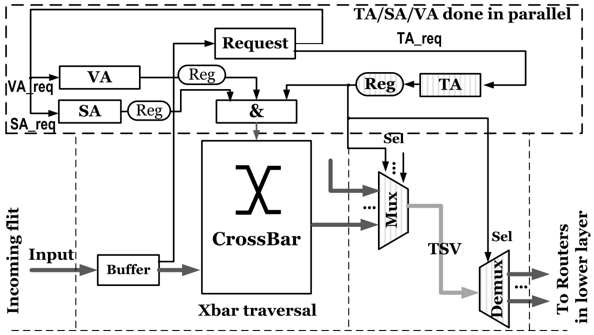


Fig. 6. Data path of modified router architecture with TS (registers for stages are not shown).

TABLE I
TSV CHARACTERISTICS

	Fermilab [44]	Tezzaron [12]	Amkor[45]
Diameter	*6um*	*5um*	*25un*
Pitch size	*24um*	*16um*	*40um*
Bonding type	*Back to face (b2f)*	*Back to face (b2f)*	*B2b/F2b*
Length	*85um*	*90um*	*100um*

D. Routing Algorithm

In TSV-shared NoCs, the vertical links are concentrated in hardware layer, and it is oblivious to routing layer. The TSV arbitration affairs before routing totally depend on TS logics. The TSV clusters are virtually separated links in routing

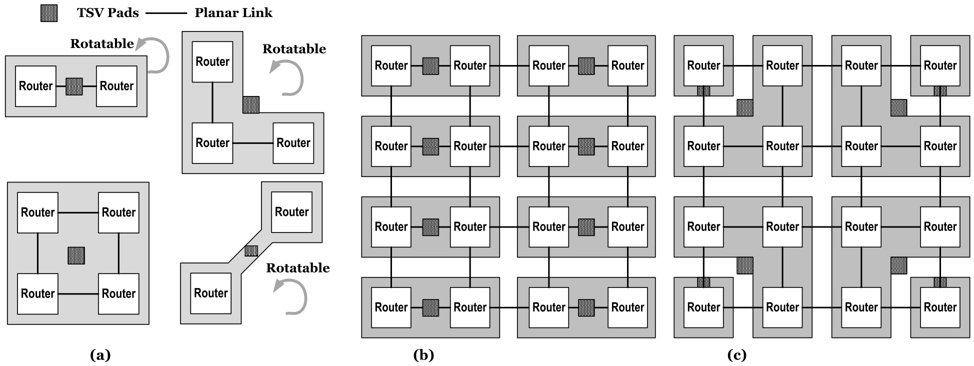


Fig. 7. Several sharing patterns and layouts. (a) Basic Patterns. (b) Homogeneous Layout. (c) Heterogeneous Layout.

stage because the signals are selected by sharers before they enter into and leave the physical links. Therefore, TSV-shared networks can adopt unmodified XYZ routing or any other routing algorithms, and inherit its merit of deadlock-immunity. In later discussion of routing algorithms, we can directly apply the well-known conclusions about adaptive algorithm or dimensional ordered algorithm in 3-D mesh to TS-mesh [46].

E. Global Configuration of TS

When we leverage TS to economize the TSVs in 3-D NoC, there are several parameters should be decided. The first one is sharing degree, which is defined as the number of routers in a group that shares a common TSV bundle. Although a higher sharing degree could reduce considerable TSVs consumption, it is not proper for two distant routers to share their TSVs as it will introduce quite long global wires and may degrade the performance of NoCs. In addition, large sharing degree will induce large crossbars, and thus increase hardware costs. In this paper, we allow routers to share their TSVs only if they can construct a 2 × 2 mesh or a subtopology within it. This condition can restrict routers in a cluster to be within two hops away from each other, thus guaranteeing a shorter horizontal latency $t1$. We have several design alternatives with distinct pads placement for a designated sharing degree, as shown in Fig. 7(a). These basic patterns can be distributed freely into NoC to form a particular layout.

In Fig. 7(b) and (c), as we can see from the two layouts composed of basic patterns, there are two viable options for global sharing and distribution configurations. The first one is to straightforwardly assign a common sharing degree to the network and then accordingly cluster the adjacent routers together as the layout of Fig. 7(b). This symmetrical sharing configuration is deemed as a homogeneous layout (though in some cases there may be a violation of homogeneity in boundary regions due to network scale). However, this is a worst-case design because the homogeneous configuration has to ensure that the hottest part of the NoC will be provided with enough TSV bandwidth to avoid congestion caused by load imbalance. Comparatively, there is also an alternative heterogeneous layout that may work better than worst-case design, which allows different sharing degrees in different subregions of the network like the layout in Fig. 7(c). This idea is enlightened by the notion that the link utilization is commonly nonuniform in most of network topologies [19]. For example, the centric network links of mesh NoC tend to exhibit higher occupancy rate than the edge links under various traffic patterns.

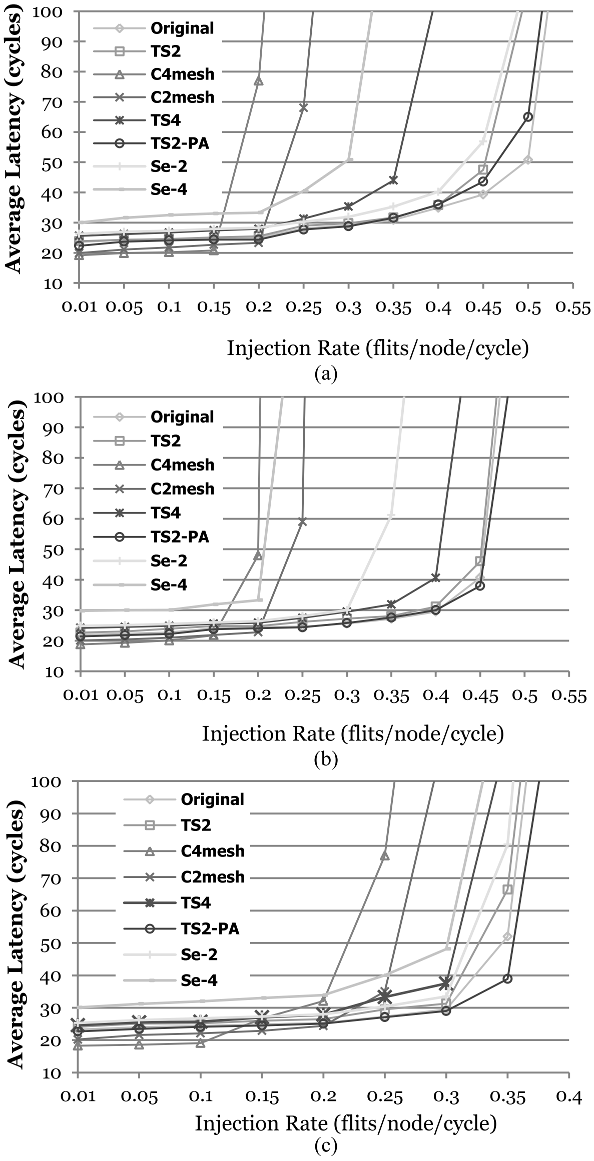


Fig. 8. Latency and throughput comparison (2 × 4 × 4). (a) Traffic type: uniform. (b) Traffic type: shuffle. (c) Traffic type: hotspot.

F. Using GSA to Explore the Design Space of Heterogeneous Layouts

To see the performance potential and how difference layouts affect the TSV utilization in TS, we are going to search the space of heterogeneous layouts. Before the design space exploration, we assume a qualified layout should exhibit a high TSV-effectiveness (TE) that is defined as the average cost-performance ratio of the layout under the target application set

$$\mathrm{TE}_i = \sum_{j=0}^{k} \mathrm{Area}_{\mathrm{tsv}} / \mathrm{Latency}_{\mathrm{inc}}^{j} \tag{2}$$

where $\mathrm{area}_{\mathrm{tsv}}$ denotes the equivalent planar area of saved TSVs, and $\mathrm{latency}_{\mathrm{inc}}^{j}$ is the average latency increased over the original NoC in network simulations under a range of injection rates and traffic patterns described in Table IV. k is the number of workloads we used in TE computation so that they can cover diverse traffic types without loss of generality. For domain-specific many core architectures that are designed for a particular group of applications, designers can choose their target benchmarks for TE calculation in design customization. Even for general purpose processors (GPPs), diverse representative benchmarks for GPPs are available for network parameters exploration [19], [47].

With the criteria decided, the next step is to search for the possibly best configuration by filling sharing patterns as shown in Fig. 7 into the chessboard of network. How to group TSV bundles together is a complicated problem because the selection and placement of different patterns would influence not only the bandwidth, but also the available bypasses of the related network regions.

First, we formalized the chessboard filling as a 2-D bin packing problem that can be easily reduced to integer programming problem, known to be NP-hard. The exploration space is as huge as $O(\exp(n))$.

1) Problem Description: Assuming the $N \times N$ network is a x–y plane, the layout vector $S = <s_0, s_1, s_2, \ldots, s_k>$ describes the k TSV clusters filled into the plane in sequence from the coordinate of (0, 0) in ascending order of x and y. (x_i, y_i) is the vertex coordinates of cluster-i and s_i is the shape index of the ith filled cluster in Fig. 7(a). Our target is to search for the optimal layout S with the maximum TE

$$\begin{aligned}
&\text{Maximize TE}(S = <s_0, s_1, s_2, \ldots, s_k>)\\
&\text{s.t.}\varphi_1 : 0 \le x_i(s_0, s_1, s_2, \ \ldots, s_{i-1}) \le N\\
&\qquad 0 \le y_i(s_0, s_1, s_2, \ \ldots, s_{i-1}) \le N, \quad i = 0, 1, 2, \ \ldots, n\\
&\varphi_2 : \left(x_i^{\min} > x_{i-1}^{\min} \&\& y_i^{\min} = y_{i-1}^{\min}\right) || \left(y_i^{\min} > y_{i-1}^{\min}\right)\\
&\varphi_3 : T(x_i, y_i, s_i) \cap T(x_j, y_j, s_j) \quad \equiv \emptyset
\end{aligned} \tag{3}$$

where (x_i, y_i, s_i) corresponds to the ith filled cluster in a layout S, and x_i, y_i are the vertex coordinates of cluster-i, and s_i is the shape index $(0 = s < 7)$. (x_i, y_i) determines the position of the cluster while the shape index relates to the chosen basic cluster pattern in Fig. 7(a). φ_1 also indicates that the x–y coordinates of cluster-i' vertex are determined by the shapes of $i - 1$ clusters filled before it. Constraint φ_2 claims the cluster filling sequence in the x–y plane of a network. $T(x_i, y_i, s_i)$ denotes the region covered by cluster-i, and constraint φ_3 makes sure that clusters do not overlap each other.

Since the search space is huge, especially for large-scale networks and estimating the qualities of potential solutions needs the feedbacks from our NoC simulator, we employ genetic algorithm (GA) to cover as many layouts as possible, which mimics the adaptive evolution of natural systems. Because GA has poor convergence properties, we also add simulated annealing procedure to the GA, which is known as GSA.

Simulated annealing uses a temperature parameter to control the acceptance of generated solutions. As the algorithm proceeds, temperature decreases in genetic reproduction and makes it probable to converge on a local optimal solution. The necessary steps to performance GSA in our algorithm includes the following.

2) String Representation: String $S = <s_0, s_1, s_2, \ldots, s_k>$ represents the chromosome of a potential solution.

3) Fitness Calculation: $f(i) = \mathrm{TE}_i$ is obtained from synthetic workloads simulation for each member string.

4) Crossover: Single point crossover is used to randomly exchange parts of two parental member strings, generating new solution strings.

5) Mutation: We randomly exchange the location of two code bits in the mutated member strings to generate new solution strings.

6) Reproduction and Annealing: Reproduction forms the new generation of n-sized population by accepting the member strings with higher fitness values. To survive into next generation, the member-i is accepted with probability P_i

$$P_i = \frac{f(i)}{\sum_{j=1}^{n} f(j)} \tag{4}$$

where fitness value $f(i)$ equals TE_i. To calculate P_i, TE_i of each produced configuration-i will be collected from our simulators and calculated as (4) for each generation of reproduction.

In GSA, the annealing temperature is used to compute the acceptance probability of parental genes (P') and produced genes $(1 - P')$ in selection phase as

$$P' = \frac{1}{1 + \exp\left[\frac{f_p - f_c}{t}\right]} \tag{5}$$

where f_p is the average fitness value of parental generation, and f_c is the average fitness value of the new generation. T is the annealing temperature. The initial temperature is set as large enough to make P' nearest to one. In each generation, T decreases and is applied to (5) to calculate the acceptance rate. So, the final acceptance rate for a member string-i is set as $P_i P'$ if it is a new string or $P_i(1 - P')$ if it is a parental string. GSA repeats until the average fitness difference between old generation and new generation is less than $\beta T\%$, and β is a selected constant factor.

We evaluate the performance of GSA from several aspects: successful rate of convergence (SRC), min/max/mean iterations, and rate of evolution leap, which are commonly used metrics to evaluate the performance of GA. The result is shown in Table II, which shows the effects of our optimization technique to the algorithm. In Table II, selected seeds mean the GSA with selected initial chromosomes that have the proper sharing degrees.

TABLE II

CONVERGENCE PERFORMANCE

	SRC	Min Iter	Max Iter	Mean Iter	REL
Random Seed	78%	*97*	*331*	*258*	*93.0%*
SelectedSeed	94%	*55*	*287*	*188*	*89.3%*
No Simulated Anealing	83%	*117*	*583*	*436*	*91%*

TABLE III

NETWORK, PROCESSOR, MEMORY HIERARCHY CONFIGURATIONS

Networks	
Topology	2×2× 2/4×4× 2/8 × 8 × 2, symmetric 3D-Mesh
Routers	7 ×7, 2 pipeline stages, wormhole flow control, 8-flit depth buffer, 2/3 Virtual Channels
Routing Algorithm	XYZ routing, Planar Adaptive(bypasses activated)
Cores	OoO, two issue, 2GHz, 32/16/8 cores
Base Memory Hierarchy Parameters	
L2 Cache	Shared, NUCA, Exclusive, 4 way, 64B lines, banked, 8MB, LRU replacement, 6 cycles hit
Memory	4G DRAM, 260 cycles access

IV. EVALUATION

A. Experimental Setup

In this section, the proposed TSV economization scheme is evaluated. There are two experimental environments to evaluate the impacts of TSV economization. The first one is a detailed cycle-accurate network simulator we developed using System-C. The NoC configuration corresponds to the network description in Table III. The second environment is the full-system cycle-accurate GEMS-2.1 simulator [48] integrating Garnet NoC model [49], so that real workloads can be used to evaluate our design. The basic system configurations are described in Table III and Garnet NoC model is configured to be consistent with the parameters used in network only simulator. The workloads used in simulations are described in Table IV.

B. Routing Algorithm

In this paper, we concentrate on XYZ routing to validate TS. However, the static XYZ routing does not fully exploit the performance potential in the TS-networks. TS logics enable more flexible routing choices and shorter average network latency. We also test the classic planar-adaptive (PA) in TS-network [50]. This algorithm is a minimal deadlock-free adaptive routing for n-dimensional NoCs. For example, it can detour the faulty node or regions to keep network connectivity or choose a less congested path to avoid hotspot. The idea is to provide adaptivity in two dimensions at all times, thus a packet is routed adaptively in a series of 2-D planes. Specifically, we use PA routing to get rid of the constraint that the packet is to be routed in planar-ordered way.

To investigate the performance of TS-NoC, we take the bypasses provided by TS into account and combine it with PA algorithm. The packet could skip some routing hops by taking the bypasses in TS crossbar. Because we adopt a speculation router architecture, the vertical shortcut is enabled when the output port is predicted to be bypassable in TSV sharer, which means the packet is two hops away from the destination node in difference layer. It can send a lookahead signal in advance to instruct the TSV crossbar and the downstream router to accept the packet. In this case, the packet is allocated with a special buffer slot associated to the bypass.

TABLE IV

WORKLOADS DESCRIPTIONS

Parsec	Multithreaded For CMP	*canneal, dedup, vips, bodytrace, ferret, swaptions, freqmine, x264, raytrace, fluidanimate, facesim, streamcluster, blackscholes*
Traffics	Sythetic Workloads	*Uniform: injection rate from 0-0.5, step by 0.05* *Shuffle: injection rate from 0-0.5, step by 0.05* *Hotspot: injection rate from 0-0.4, step by 0.05* *Transpose: injection rate from 0-0.4, step by 0.05*

To follow the rules of PA routing, we restricted the packet to traverse within the 2-D plane at one hop. In this way, bypasses can be compatible with planar adaptive routing. For example, packets are forbidden to turn from router-010 to router-101 through the TSV crossbar.

C. Network Only Analysis Under Synthetic Traffic

1) Performance of Homogeneous Layouts: For the performance evaluation under synthetic traffic, 4 × 4 × 2 3-D mesh NoC is chosen and implemented in network simulator. The proposed router design matches the description in Table III and TA uses round-robin arbitration strategy for TSV allocation. To get a comprehensive estimate of the proposed design, the performance of the original symmetric 3-D NoC, 3-D NoC with serialized vertical channels, and the NoC with proposed TS is all compared under various synthetic traffic patterns, including uniform, shuffle, and hotspot. The comparison results are shown in Fig. 8. In Fig. 8, Se-2 and Se-4 in the figure represent NoC using 2:1 TSV serialization and NoC using 4:1 TSV serialization, respectively, while TS2 and TS4 represent homogeneous TSV clustering configuration with sharing degrees of two and four, respectively. These four designs adopt XYZ routing algorithm while TS2-PA represents 3-D NoC with PA routing and a sharing degree of two. The TSV bypasses are only enabled in TS2-PA. It is obviously that zero-load latency of TS4, which is close to that of original design, decreases by about 20% compared with that of Se-4 under all these traffic patterns. The reason mainly lies in that Se-4 brings in additional three-cycle latency induced by vertical link transmission while TS-4 adds only a single cycle penalty and pipeline transmission further alleviates the penalty. Basically, TS2-PA has a very close latency to that of original and even outperforms original, especially for some heavy workloads because the enabled TSV bypasses and routing adaptivity in TS2-PA can alleviate the congestion-induced latency.

In addition, throughput of TS-4 is ∼30% higher than Se-4 under uniform and shuffle (assuming injection rate to be throughput when network latency reaches 100 cycles). In most cases, TS-2 only performs slightly better than Se-2 except shuffle, which results in significantly unbalanced TSV utilization and makes TS far more efficient. In this case, serialization suffers from more network throughput loss because its busy TSV bundle easily becomes the network bottleneck. In the case of Fig. 8(b), Se-4 and Se-2 show lower throughputs when compared with TS-2 and TS-4.

In Fig. 9, we show the performance results of an 8 × 8 × 3 3-D mesh NoC under two traffic patterns. The trend of network performance for different configurations keeps

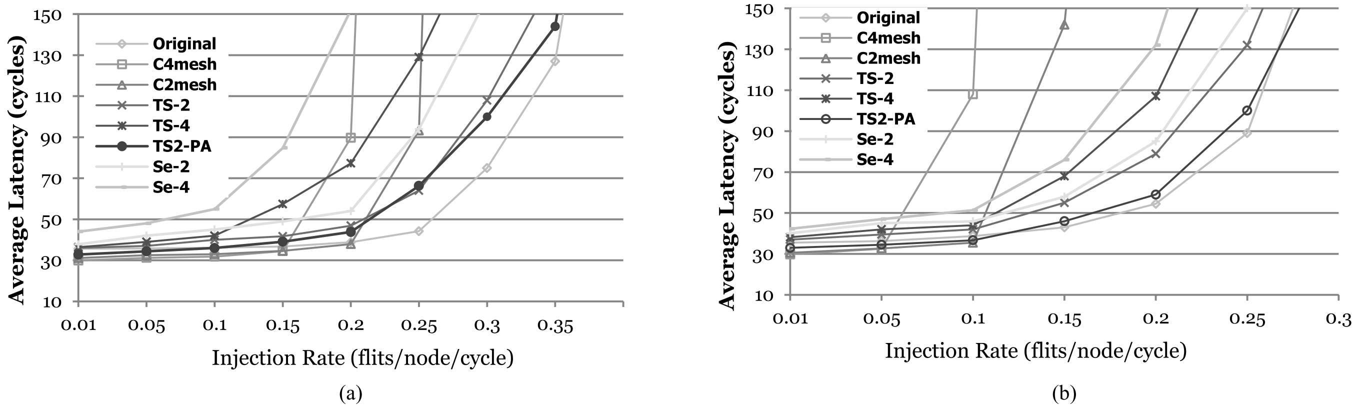


Fig. 9. Latency and throughput comparison (8 × 8 × 3).

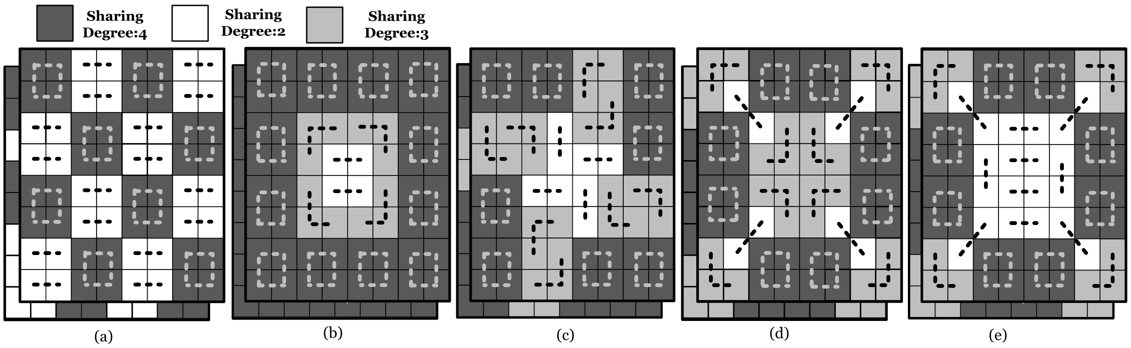


Fig. 10. Five configuration of TSV clustering in network (double layered, 2 × 8 × 8 mesh). (a) Interleaved. (b) Ring. (c) Swirl. (d) Thrifty-Diag. (e) Diagonal.

alike with those of the 4 × 4 × 3 network. The average zero-load latency of TS scheme is still lower than Se in most points even when large networks saturate earlier than smaller ones. The throughput improvement of TS-2 is about 15% over Se-2 while the throughput of TS-4 is about 30% higher than that of Se-4 under the uniform traffic.

2) Comparison With High-Radix Router: Using high-radix router and concentrated mesh (CMesh) to reduce TSV consumption is also a feasible option, so we implemented C4mesh and C2mesh in our simulator to show their performance. In C4mesh, three ports are added to each router to connect four local PEs so that the PEs are concentrated to share a single upstream/downstream TSV bundle. Similarly, C2mesh uses 7-ported high-radix router for topology concentration. As shown in Figs. 8 and 9, such a scheme shows the lowest zero-load latency compared with all other solutions thanks to the high-radix router. However, either C4mesh or C2mesh has a poor throughput because the planar bandwidths are limited by topology concentration.

3) TE of Heterogeneous Configurations: The TE is used for high-level evaluation of different TS configurations, which is defined as the ratio of saved TSV quantity to the performance loss. It can be calculated according to (2). For the network under synthetic traffics, the performance loss is measured as the average packet latency increment. For full system analysis in following sections, the performance loss is measured in terms of a system throughput metric, instructions per cycle (IPC). The latency increment is estimated by comparing the averaged network latency of TS-enabled networks with original networks with full TSVs, under a wide range of injection rates (0.1–0.5, before saturation point) and traffic patterns. The values are normalized to the TE of serialization scheme that removes 3/4 of TSVs. To show the performance of different heterogeneous configurations, we list several clustering layouts with the best quality in terms of TE, which are generated by GSA. Though only the T-Diag is the final optimal output of the algorithm, some of the other suboptimal configurations are also intentionally selected and shown. These regular patterns confirm with some of our intuitions for heterogeneous layout design. As shown in Fig. 10, these five types of layouts are, respectively, Interleaved, Ring, Swirl, Diagonal, and Thrifty Diagonal. Fig. 10 shows that the averaged TE of these five configurations is over 5.2× greater than that of Se-4 and 2.5× that of TS-4.

Interestingly, for most of the configurations in Fig. 10, the center regions in networks have lower sharing degree than boundary regions. One reason is the well-known fact that the center of a planar mesh or other nonedge symmetric topology usually gets more congested because it handles more traffic compared with the edge links [19]. Take TS-2 as an example, in simulation under uniform traffic of 0.3 injection rate for PA routing, we can see some of the centric pressure is spilt onto the vertical bypasses provided by TS, therefore indirectly leading to higher buffer occupancy rate at the ports of centric TSVs. As a result, both horizontal and vertical links show much more load pressure than the edge ones. When PA routing is adopted, the pressure on centric planar links can be shared by vertical links because of congestion-aware adaptivity.

Among the five layouts in Fig. 11, T-diag shows the best TE. It reaches a balance between the bandwidth demand

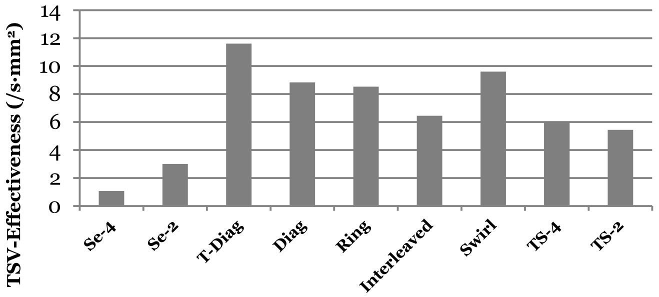


Fig. 11. TSV-effectiveness (8 × 8 × 2).

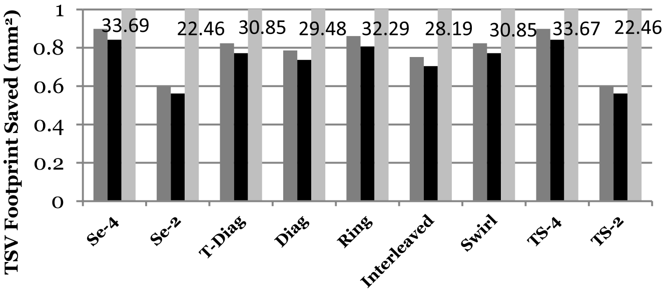


Fig. 12. TSV saving in different technologies (8 × 8 × 2).

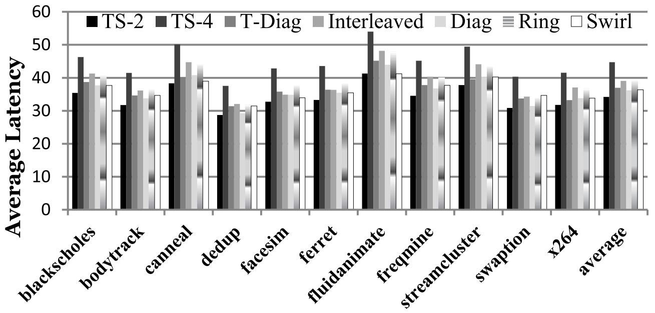


Fig. 13. Network performance of different TS topologies under real-traces.

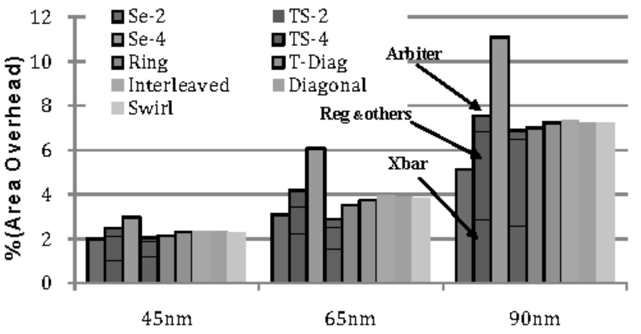


Fig. 14. Chip area overhead raised by TS.

(lower sharing degree) and a lower average latency (more bypasses) in the centric region of a network. The centric region is assigned sharing degree of three as a compromise between bandwidth and latency while peripheral routers are assigned a sharing degree of four for more utilization. The other layouts basically follow the rules of traffic centralization have shown different placement of basic patterns to suite different communication patterns. Ring and Swirl reduce the TSV bandwidths from inner to peripheral routers while Diagonal tends to distribute high bandwidth sharing patterns along the diagonals of mesh with a special consideration to the network corners in contrast to Swirl. Interleaved places basic sharing patterns of high bandwidth and those of low bandwidth next to one another. Also, from the results, we can see layouts with an averaged sharing degree around three have the best performance.

The improved effectiveness comes from the TSV reduction in network. We estimate the equivalent area saved by TSV reduction, as shown in Fig. 12. In the evaluation, we still use the three types of TSV technologies as described in Table I. As shown, TS can save considerable TSVs-related area.

4) Performance Analysis of Selected Topology: To justify that the generated layouts are superior to homogeneous layouts in different situations, we study the performance of the generated heterogeneous layouts under different workloads other than the training workloads in GSA. Fig. 13 shows the performance of each topology when we replace the synthetic workloads with the traces of real applications, The simulated traces of parsec benchmarks are extracted from GEMS simulator with the configurations in Table III, and ported to our system-C NoC simulator for network only evaluation. Even when we change the traces set, the five heterogeneous layouts only cause a little performance degradation compared with TS-2 while they save far more TSVs than TS-2, for example, T-Diag only increases 7% latency over TS-2, and reduces 22% latency over TS-4, which exhibits the applicability of these heterogeneous configurations under different workloads.

5) Overhead of TS: To quantify the hardware overhead brought by TS and serialization schemes are both implemented and evaluated. The hardware synthesis was performed using TSMC standard cell libraries. For all the TSV sharing configurations listed in Fig. 10, the relative hardware overheads raised by TS logics range from 1.8% to 7.5% of the whole network (wiring overheads not included), lower than that of the TSV serialization scheme. If we take into account the considerable planar area saved by TSV reduction the cost is negligible. As measured in experiments, planar adaptive routing algorithm and bypasses are enabled for heterogeneous layouts. Compared with original 3-D NoC adopting PA, our TS-PA adds only two bypass slots except the three original VCs in routers, so the overheads are also acceptable as shown in Fig. 14.

In Fig. 14, we count the overhead of the components, such as TAs, serializer, and deserializer in serialization scheme for comparison. It can be seen from the figure, TS-2, TS-4, and heterogeneous layouts add only about 4.2% of the network area to the chip. When feature size decreases step by step, the extra overhead shrinks to only about 2%. It can be seen from Fig. 14 that TS-2 costs slightly more area than Se-2, but the gap becomes closed in 45-nm technology. Comparatively, TS-4 is superior to Se-4 in terms of hardware cost though they can save the same number of TSVs. The reason is most of the area cost in TS is induced by the pipeline registers in the added stage. When four routers share the additional pipeline registers in TS-4, each router will have less area overhead than TS-2 in which only two routers share the pipeline registers. Furthermore, when process technology scales down, total area overhead of TS schemes decreases sharply.

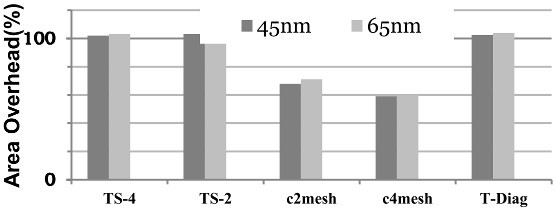


Fig. 15. Area comparison of different routers.

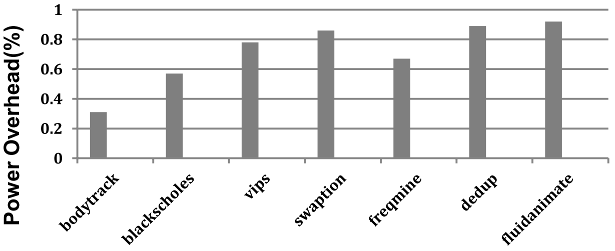


Fig. 16. Total power added to TS-4 network.

Power is another important issue, especially for 3-D chips. From the synthetic results of design compiler, the added hardware implemented in TSMC 90-nm technology increases about 4%–6% static power to routers. We use our analysis model inserted in simulator to roughly estimate the power consumption. The estimated power overhead includes the dynamic power in TS logics and the link transmission power based on the IR-model and TSV power model [36]. The dynamic power is computed as the switching frequency (obtained from GEMS) multiplied by the average per-switching power (from TSMC 90-nm typical synthesis library). The total NoC power consumption excluding the TS-induced portion is given by Orion2.0 integrated in GEMS. Compared with network power, the additional power attributed to TS logics and links are about 0.3%–0.9% for the most aggressive 4 × 4 × 2 TS-4 network (2.4–2.8 W). Because NoC only contributes to 5%–15% of multiprocessor chip power [25], it will not brought serious thermal and power issues to the processors.

For the scheme of C2mesh and C4mesh, it is well known that concentrated mesh can save some area dedicated to routers because the high-radix routers are shared by multiple PEs instead of one in normal topology. In Fig. 15, we show the area overhead of CMesh and TS. The area is estimated in both 45-nm and 65-nm process. Though C4mesh/C2mesh eliminates 75%/50% of the routers from network, they cannot save that much area compared with normal mesh. C4mesh/C2mesh can save about 30%–40% of the network area in total. After all, high-radix routers are much complex in circuit design.

6) Arbitration Strategy of TS: As our TA supports different arbitration policies to be compatible with application-aware flit scheduling or specific flow control policies in NoCs, we survey the impacts of TSV arbitration strategy on NoC performance in experiments. Fig. 16 shows the performance of TS-4 with various arbitration schemes, including oldest first, round robin, and cyclic priority under uniform traffic. For other traffic patterns, we can observe much similar results, which are omitted due to page limit. The results also indicate that arbitration strategy has little influence on NoC performance, which again proves that the TSV contention rarely happens. Therefore, we can just employ simple arbitration method for a further reduction of chip area overhead.

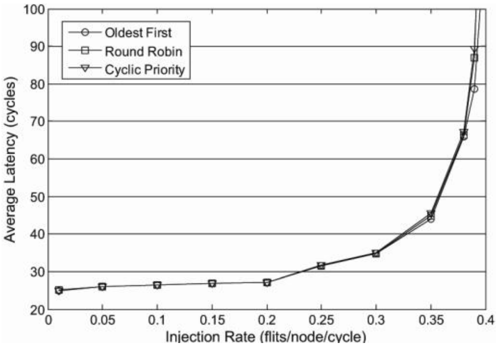


Fig. 17. Influence of added TSV arbitration stage under uniform traffic.

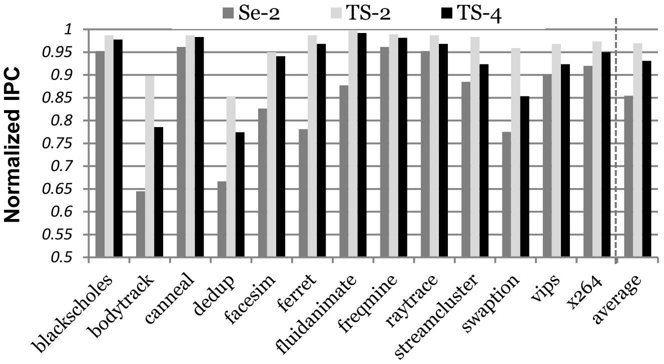


Fig. 18. System performance (4 × 4 × 2).

D. Putting It Into the System: Evaluate the Effectiveness of TS in Multicore Processors

It is shown TS only slightly degrades the network performance but greatly improves the TE. In this section, we integrate the TSV sharing scheme into processors to thoroughly estimate their cost-effectiveness and see whether it is worthwhile to sacrifice the system performance for the TSV resources saved. In the full system simulations, we redeem the metric TE (mm^2/IPC), as the ratio between equivalent planar area of saved TSVs and the IPC loss, and use it to measure the efficiency of TS.

1) Performance: First, we show the system performance of TS by executing the kernel codes of benchmarks from Parsec suite on our simulation platform. In simulation, 32-core CMP with symmetrical nonsharing NoC is set as baseline and the performance of TSV saving schemes are normalized to the baseline performance for comparison. System performance is measured in terms of IPC. For our simulator cannot support a CMP scale as large as 8 × 8 × 2, we do not obtain the performance results of heterogeneous layouts but can safely reason that the data would be comparable with that of 4 × 4 × 2.

From Fig. 18, we can see both TS-2 and TS-4 only slightly reduce the IPCs of benchmarks. The averaged IPC reductions are only about 3.1% and 6.8% caused by TS-2 and TS-4, respectively. Even TS-4 outperforms Se-2 that brings about an average of 15% IPC reduction. This is mainly because that serialization scheme delivery a much worse zero-load

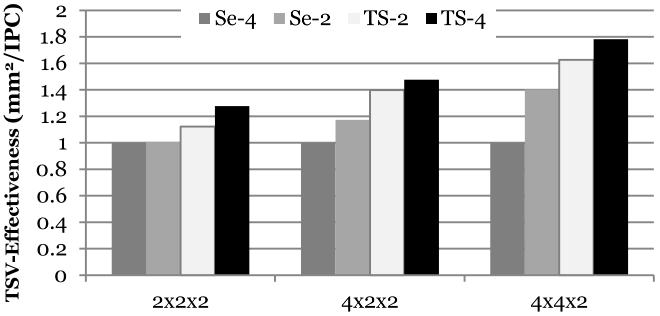


Fig. 19. TE results.

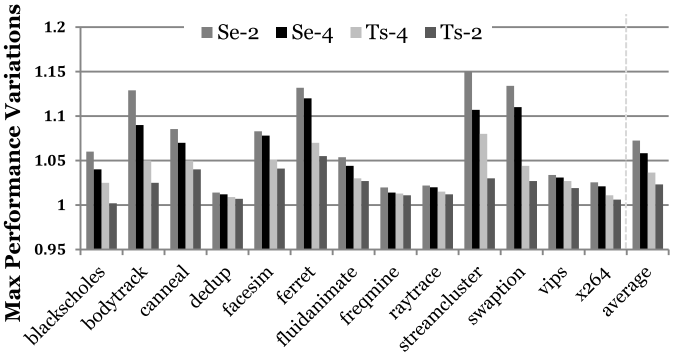


Fig. 20. Performance variation.

latency than TSV Sharing, especially for real workloads in memory coherent processors that mostly inject packets as long as 16 flits. Besides, most of the benchmarks from Parsec suite are computation-intensive and there are only a few bursts of memory requests witnessed, so the shared TSV links would encounter even less conflicts and perform almost like in full-TSV links in the network. In other words, the threads with less bandwidth demands like bodytrack or dedup often exhibit more latency-sensitivity as a common rule, and appear less tolerant to the on-chip network latency brought by serialization.

2) Cost-Effectiveness Analysis: To compute the TE of TS in full systems, we setup three types of simulated machines, $2 \times 2 \times 2$, $2 \times 4 \times 2$, and $4 \times 4 \times 2$. For each CMP, we execute the whole packets of Parsec suite and collect the averaged IPC reduction over the baseline CMP without TS. The TE values are shown in Fig. 19. We can see that the TE of TS-2 is about 1.5× that of the Se-2 schemes, less conspicuous than that achieved in network analysis under synthetic traffics.

3) System Transparency: Performance Symmetry Analysis: We deal with the TSV-utilization problem in hardware layer and attempt to hide its influence on mesh NoC from upper layers. Thus, we want to prove that TS do not cause much performance inconsistency in symmetrical networks.

It is meaningful to keep the system being viewed as a homogeneous platform at the application level though our TS ruins the symmetry of physical network [51], [52]. For example, when the OS repetitively schedules a threads group onto the cores and expects to get similar performance because of the homogeneity of cores and networks. However, the potential performance asymmetry in NoC locations would be probably caused by TSV sharing. For example, mapping a 4-threads program onto four nodes within a TSV-sharing group or nodes belonging to four different TSV-sharing groups is supposed to make a difference in the nominal network bandwidths allocated to this program. To examine whether it is the case in experiments, we repetitively execute benchmarks and randomly schedule threads group onto processor cores of $4 \times 4 \times 2$ CMP adopting aggressive TS-4, and record the IPC ratios between the best-case and worst-case performance as an indication of maximum variation. In Fig. 20, we can see random thread mapping policy brings performance variations to TS system in contrast to baseline multicore processor without TS (we configure the system parameters to exclude other factors inducing performance asymmetry, like memory controller placement). As Fig. 20 shows, serialization causes evident performance asymmetry in the network (the worst-case performance can be observed when the threads are separately mapped to different layers). In contrast, the average performance variations witnessed in simulation are about 3% for TS-4 and about 2% for TS-2. We can see that the performance variations are too marginal to be noticed by applications, and conclude that TS can be safely thought as transparent to upper levels of system. Thus, TS schemes can provide predictable performance for parallel applications regardless of the processing core on which a thread is scheduled.

E. Limitation of TS Methodology

The technique of TS aims to promote the TE of flattened 3-D mesh NoCs, which suffer from TSV underutilization. Because we add a traversal stage by the side of NoC router, the zero-load latency has been slightly increased compared with high-performance high-radix routers. Fortunately, the degradation can be compensated by the high-throughput and vertical bypasses provided by TS.

Meanwhile, to study the potential of heterogeneous TS, we use GSA to optimize the sharing topology of 3-D mesh. We have shown that heterogeneous TS increases TSV utilization compared with homogeneous TS. The purpose of genetic-based space exploration is to show the effectiveness and potential of heterogeneous TS. However, if the technique is used as a topology generator for NoC layout, there are some points should be noticed. First, because the results partly depend on the selection of workload training set, different application sets are encouraged to be used for application specific or domain specific platforms. Second, for general purpose computing system, a huge set of workloads may be necessary in topology training for statistical justification, which is the common issue in design space exploration phase for systems.

Last of all, because GSA is a stochastic algorithm, which cannot guarantee optimum in an each single run, Therefore, to achieve a quick convergence to a qualified solution needs some empirical tuning or optimization techniques as we used.

V. CONCLUSION

In this paper, we propose a TSV economization scheme that allows link sharing among the adjacent NoC routers. Compared with prior TSV serialization, the proposed solution can achieve better performance in terms of network latency and throughput. The TS-supported router only involves very low area overhead while the TSV reduction can save considerable chip area. Experimental results show that the proposed scheme is able to save more than 60% TSVs, and at the same time keeps the network latency penalty negligible. The network throughput of TS is ~30% higher than the serialization scheme under certain a variety of traffic patterns.

Besides, the evaluated homogeneous and heterogeneous TS layouts show 2× ~ 5× TE improvement than the serialization scheme for network evaluation under synthetic traffics. Last of all, it is proved that TS can bring negligible performance degradation and maintain the performance predictability of symmetrical CMPs with TS-oblivious task scheduling.

References

[1] W. Dally and B. Towles, “Route packets, not wires: On-chip interconnection networks,” in *Proc. DAC*, 2001, pp. 684–689.

[2] L. Benini and G. D. Micheli, “Networks on chips: A new SoC paradigm,” *IEEE Comput.*, vol. 35, no. 1, pp. 70–78, Jul. 2002.

[3] T. Vucurevich, “The long road to 3-D integration: Are we there yet,” in *Proc. Keynote Speech 3D Archit. Conf.*, 2007, pp. 1–3.

[4] G. Loh, Y. Xie, and B. Black, “Processor design in 3D die-stacking technologies,” *IEEE Micro*, vol. 27, no. 3, pp. 31–48, May/Jun. 2007.

[5] F. Li, C. Nicopoulos, T. Richardson, Y. Xie, V. Narayanan, and M. Kandemir, “Design and management of 3D chip multiprocessors using network-in-memory,” *ACM SIGARCH Comput. Archit. News*, vol. 34, no. 2, pp. 130–141, 2006.

[6] D. Park *et al.*, “MIRA: A multi-layered on-chip interconnect router architecture,” in *Proc. ISCA*, 2008, pp. 251–261.

[7] Y. Xu, Y. Du, B. Zhao, X. Zhou, Y. Zhang, and J. Yang, “A low-radix and low-diameter 3D interconnection network design,” in *Proc. HPCA*, 2009, pp. 30–42.

[8] B. Feero and P. Pande, “Performance evaluation for three-dimensional networks-on-chip,” in *Proc. Symp. VLSI*, 2007, pp. 305–310.

[9] J. Kim *et al.*, “A novel dimensionally-decomposed router for on-chip communication in 3D architectures,” *ACM SIGARCH Comput. Archit. News*, vol. 35, no. 2, pp. 138–149, 2007.

[10] W. R. Davis *et al.*, “Demystifying 3D ICs: The pros and cons of going vertical,” *IEEE Des., Test Comput.*, vol. 2, no. 5, pp. 498–510, Nov./Dec. 2005

[11] S. Pasricha, “Exploring serial vertical interconnects for 3D ICs,” in *Proc. DAC*, 2009, pp. 581–586.

[12] H. Sangki, “3D super-via for memory applications,” in *Proc. MSPI Packag. Workshop*, 2007, pp. 1–35.

[13] S. Das, A. Fan, K. Chen, C. Tan, N. Checka, and R. Reif, “Technology, performance, and computer-aided design of three-dimensional integrated circuits,” in *Proc. Int. Symp. Phys. Des.*, 2004, pp. 115–122.

[14] C. Ababei *et al.*, “Placement and routing in 3D integrated circuits,” *IEEE Des. Test*, vol. 22, no. 6, pp. 520–531, Nov./Dec. 2005.

[15] M. Healy *et al.*, “Multi-objective microarchitectural floorplanning for 2D and 3D ICs,” *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, vol. 26, no. 1, pp. 38–52, Jan. 2007.

[16] I. Loi, S. Mitra, T. Lee, S. Fujita, and L. Benini, “A low-overhead fault tolerance scheme for TSV-based 3D network on chip links,” in *Proc. ICCAD*, 2008, pp. 598–602.

[17] L. Jiang, Q. Xu, and B. Eklow, “On effective TSV repair for 3D-stacked ICs,” in *Proc. DATE*, 2012, pp. 793–798.

[18] Y. Jin, E. J. Kim, and K. H. Yum, “A domain-specific on-chip network design for large scale cache systems,” in *Proc. HPCA*, 2007, pp. 318–327.

[19] A. K. Mishra, N. Vijaykrishnan, and C. R. Das, “A case for heterogeneous on-chip interconnects for CMPs,” in *Proc. ISCA*, 2011, pp. 389–400.

[20] J. Balfour and W. J. Dally, “Design tradeoffs for tiled cmp on-chip networks,” in *Proc. Int. Conf. Supercomput.*, 2006, pp. 187–198.

[21] Z. Li, C. Zhu, L. Shang, R. Dick, and Y. Sun, “Transaction-aware network-on-chip resource reservation,” *IEEE Comput. Archit. Lett.*, vol. 7, no. 2, pp. 53–56, Jul./Dec. 2008.

[22] X. Zhou, J. Yang, Y. Xu, Y. Zhang, and J. Zhao, “Thermal-aware task scheduling for 3D multicore processors,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 1, pp. 60–71, Jan. 2010.

[23] R. Weerasekera, D. Pamunuwa, L.-R. Zheng, and H. Tenhunen, “Extending systems-on-chip to the third dimension: Performance, cost and technological tradeoffs,” in *Proc. ICCAD*, 2007, pp. 212–219.

[24] J. Zhao, X. Dong, and Y. Xie, “Cost-aware three-dimensional (3D) many-core multiprocessor design,” in *Proc. ICCAD*, 2010, pp. 126–131.

[25] D Wentzlaff *et al.*, “On-chip interconnection architecture of the tile processor,” *IEEE Micro*, vol. 27, no. 5, pp. 15–31, Sep./Oct. 2007.

[26] M. Daneshtalab, M. Ebrahimi, P. Liljeberg, J. Plosila, and H. Tenhunen, “Cluster-based topologies for 3D stacked architectures,” in *Proc. Int. Conf. Comput. Frontiers*, 2011, pp. 1–14.

[27] M. Ebrahimi, M. Daneshtalab, P. Liljeberg, J. Plosila, and H. Tenhunen, “Exploring partitioning methods for 3D networks-on-chip utilizing adaptive routing model,” in *Proc. NOCS*, 2011, pp. 73–80.

[28] M. Jung, J. Mitra, D. Z. Pan, and S. K. Lim, “TSV stress-aware full-chip mechanical reliability analysis and optimization for 3D IC,” in *Proc. DAC*, 2011, pp. 188–193.

[29] G. Smith, L. Smith, S. Hosali, and S. Arkalgud, “Yield considerations in the choice of 3D technology,” in *Proc. Int. Symp. Semiconduct. Manuf.*, 2007, pp. 1–3.

[30] B. Noia and K. Chakrabarty, “Pre-bond testing of die logic and TSVs in high performance 3D-SICs,” in *Proc. DIC*, 2012, pp. 1–5.

[31] U. S. Syed, K. Chakrabarty, A. Chandra, and R. Kapur, “3D-scalanble adaptive scan (3D-SAS),” in *Proc. 3DIC*, 2012, pp. 1–6

[32] C. Ferri, S. Reda, and R. I. Bahar, “Strategies for improving the parametric yield and profits of 3D ICs,” in *Proc. ICCAD*, 2007, pp. 220–226.

[33] A. C. Hsieh and T. Hwang, “TSV redundancy: Architecture and design issues in 3-D IC,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 4, pp. 1–12, Apr. 2011.

[34] Y. Han, C. Liu, H. Lu, W. Li, L. Zhang, and X. Li, “RevivePath: Resilient network-on-chip design through data path salvaging of router,” *J. Comput. Sci. Technol.*, vol. 28, no. 6, pp. 1045–1053, Nov. 2013.

[35] F. Ye and K. Chakrabarty, “TSV open defects in 3D integrated circuits: Characterization, test, and optimal spare allocation,” in *Proc. 49th DAC*, 2012, pp. 1024–1030.

[36] T. Frank *et al.*, “Resistance increase due to electromigration induced depletion under TSV,” in *Proc. IRPS*, 2011, pp. 3F.4.1–3F.4.6.

[37] H. Yu, Y. Shi, L. He, and T. Karnik, “Thermal via allocation for 3D ICs considering temporally and spatially variant thermal power,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 16, no. 12, pp. 1609–1619, Dec. 2008.

[38] V. F. Pavlidis and E. G. Friedman, *Three-Dimensional Integrated Circuit Design*. San Francisco, CA, USA: Morgan Kaufmann, 2008.

[39] G. Huang, M. Bakir, A. Naeemi, H. Chen, and J. D. Meindl, “Power delivery for 3D chip stacks: Physical modeling and design implication,” in *Proc. Conf. Electr. Perform. Electron. Packag.*, 2007, pp. 205–208.

[40] Z. Lu and A. Jantsch, “TDM virtual-circuit configuration for network-on-chip,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 16, no. 8, pp. 1021–1034, Aug. 2008.

[41] R. Stefan, A. Molnos, and K. Goossens, “dAElite: A TDM NoC supporting QoS, multicast, and fast connection set-up,” in *Proc. DATE*, 2012, pp. 1283–1288.

[42] A. Kumar, L. Peh, P. Kundu, and N. K. Jha, “Express virtual channels: Towards the ideal interconnection fabric,” *SIGARCH Comput. Archit. News*, vol. 35, no. 2, pp. 150–161, 2007.

[43] R. Mullins, A. West, and S. Moore, “Low-latency virtual-channel routers for on-chip networks,” in *Proc. ISCA*, 2004, pp. 188–198.

[44] G. W. Deptuch *et al.*, “Vertically integrated circuits at fermilab,” *IEEE Trans. Nucl. Sci.*, vol. 57, no. 4, pp. 2178–2186, Sep. 2008.

[45] M. Kelly. (2011). *Through Silicon Via (TSV) Product Development* [Online]. Available: http://sites.amd.com/la/Documents/TFE2011_001AMC.pdf

[46] B. Fu, Y. Han, J. Ma, H. Li, and X. Li, “An abacus turn model for time/space-efficient reconfigurable routing,” in *Proc. ISCA*, 2011, pp. 259–270.

[47] S. Kang and R. Kumar, “Magellan: A search and machine learning-based framework for fast multi-core design space exploration and optimization,” in *Proc. DATE*, 2008, pp. 1432–1437.

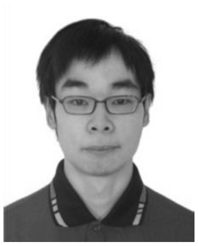
[48] M. M. K. Martin *et al.*, “Multifacet's general execution-driven multiprocessor simulator (GEMS) toolset,” *Comput. Archit. News Lett.*, vol. 33, no. 4, pp. 92–99, 2005.

[49] N. Agarwal, L. S. Peh, and N. K. Jha, “GARNET: A detailed interconnection network model inside a full-system simulation framework,” Dept. Electr. Eng., Princeton Univ., Princeton, NJ, USA, Tech. Rep. CE-P08-001, 2008.

[50] A. A. Chien and J. H. Kim, “Planar-adaptive routing: Low-cost adaptive networks for multiprocessors,” in *Proc. ISCA*, 1992, pp. 268–277.

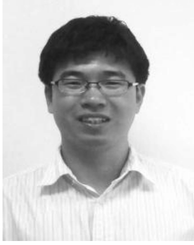
[51] L. Zhang, Y. Han, Q. Xu, and X. Li, “Defect tolerance in homogeneous manycore processors using core-level redundancy with unified topology,” in *Proc. DATE*, 2008, pp. 891–896.

[52] L. Zhang, Y. Han, Q. Xu, X. Li, and H. Li, “On topology reconfiguration for defect-tolerant NoC-based homogeneous manycore systems,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 17, no. 9, pp. 1173–1186, Sep. 2009.



Ying Wang (S’11) received the B.S. and M.S. degrees in electrical engineering from the Harbin Institute of Technology, Harbin, China, in 2007 and 2009, respectively. He is currently pursuing the Ph.D. degree with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China.

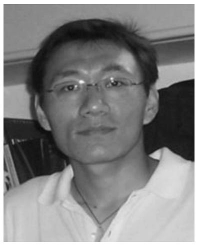
His current research interests include reconfigurable computing, interconnects, memory system, and fault-tolerance for many-core architectures.



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

He is currently an Associate Professor with the State Key Laboratory of Computer Architecture, ICT, CAS. His current research interests include very large scale integration architecture design and test, in particular, fault-tolerant and low-power architecture.

Dr. Han was a recipient of the Best Paper Award at ATS’2003. He is a Program Chair of ATS’2014, Finance Chair of HPCA’2013, Program Co-Chair of WRTLT 2009, and serves on the Technical Program Committees of several IEEE and ACM conferences, including HPCA’13, ASPDAC’13, Cool Chip’13, ATS’08–10, and GVLSI’09–10.

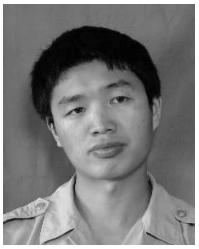


Lei Zhang (M’09) received the Ph.D. degree in computer architecture from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, China, in 2008.

He is currently an Associate Professor with the State Key Laboratory of Computer Architecture, ICT, CAS. His current research interests include network-on-chip, 3-D integration, fault-tolerant computing, and multicore/many-core processors.

Bin-Zhang Fu (M’09) received the B.Eng. degree in electronics and information engineering, and computer science and technology from the Huazhong University of Science and Technology, Wuhan, China, and the Ph.D. degree in computer science from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, China, in 2004 and 2011, respectively.

He is currently an Associate Professor at ICT, CAS. His current research interests include high-performance and high-reliable interconnection networks.



Cheng Liu (S’10) received the B.Eng. and M.Eng. degrees from the Harbin Institute of Technology, Harbin, China, and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2011. He is currently pursuing the Ph.D. degree with the University of Hong Kong, Hong Kong.

His current research interests include reconfigurable computing, hardware acceleration, and network-on-chip.

Hua-Wei Li (SM’09) received the B.S. degree in computer science from Xiangtan University, Xiangtan, China, and the M.S. and Ph.D. degrees from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, China, in 1996, 1999, and 2001, respectively.

She is currently a Professor at ICT, CAS. Her current research interests include very large scale integration/system-on-chip design verification and test generation, delay test, and dependable computing.

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

He was an Assistant Professor from 1991 to 2000 and has been an Associate Professor since 1993 with the Department of Computer Science, Peking University, Beijing. He joined the ICT, CAS, as a Professor in 2000. He is currently the Deputy (Executive) Director of the State Key Laboratory of Computer Architecture, ICT, CAS. He has co-authored more than 200 papers in academic journals and international conference, and holds 34 patents and 35 software copyrights. His current research interests include very large scale integration testing, design verification, and dependable computing.

Dr. Li has been served as the Chair of the China Computer Federation Technical Committee on Fault Tolerant Computing since 2008, the IEEE Asian Pacific Regional Test Technology Technical Council Vice Chair since 2004, and the Steering Committee Chair of the IEEE Asian Test Symposium since 2011. In addition, he serves on the Technical Program Committee of several IEEE and ACM conferences, including VTS, DATE, ASPDAC, and PRDC. He also serves as an Editorial Board Member of the *Journal of Computer Science and Technology*, the *Journal of Electronic Testing: Theory and Applications*, and the *Journal of Low Power Electronics*.