

Vertical Interconnects Squeezing in Symmetric 3D Mesh Network-on-Chip *

Cheng Liu, Lei Zhang, Yinhe Han, Xiaowei Li
Key Laboratory of Computer System and Architecture
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
e-mail: {liucheng, zlei, yinhes, lxw}@ict.ac.cn

Abstract— Three-dimensional (3D) integration and Network-on-Chip (NoC) are both proposed to tackle the on-chip interconnect scaling problems, and extensive research efforts have been devoted to the design challenges of combining both. Through-silicon via (TSV) is considered to be the most promising technology for 3D integration, however, TSV pads distributed across planar layers occupy significant chip area and result in routing congestions. In addition, the yield of 3D integrated circuits decreased dramatically as the number of TSVs increases. For symmetric 3D mesh NoC, we observe that the TSVs' utilization is pretty low and adjacent routers rarely transmit packets via their vertical channels (i.e. TSVs) at the same time. Based on this observation, we propose a novel TSV squeezing scheme to share TSVs among neighboring router in a time division multiplex mode, which greatly improves the utilization of TSVs. Experimental results show that the proposed method can save significant TSV footprint with negligible performance overhead.

1 Introduction

With continuously shrinking feature size, more and more processing cores and memories are integrated on a single chip. Two major trends, i.e. communication-centric interconnect architecture based on network-on-chip (NoC) that addresses scalability challenges as well as bandwidth bottleneck [1][2] and three dimensional integrated circuits (3D ICs) that alleviate interconnect latency pressure as well as heterogeneous integration problems [3] [4] [5] [6], are emerging for such complex integrated systems. 3D NoCs combining both the benefits soon become one of the most promising on-chip communication techniques in complex System-on-Chip (SoC) [7][8].

Several prior work has investigated the architecture design of 3D NoCs. Feero et al. compared both 2D Mesh and 2D Torus to 3D Mesh and 3D stacked Mesh [7]. Results showed that 3D NoCs offer more competitive performance and energy consumption for communication. Li et al. introduced a 3D NoC by hybridizing a common NoC router with a bus link in vertical dimension. The hybrid system allows single-hop communication between nodes connected by the vertical bus, and it provides both performance and area benefits [9]. Kim et al. proposed a dimension decomposition method to optimize 3D NoC area

cost and operation frequency as well [10]. Although 3D NoC designs vary, they fall in three categories, i.e. symmetric, hybrid and true 3D fabric NoC [10]. Symmetric 3D NoC is a simple extension of 2D NoC by adding two additional ports in 2D router design, i.e., upwards and downwards i/o ports. A two-layer symmetric 3D Mesh NoC is shown in Fig.1. There are two active silicon layers in this example and each layer can be potentially placed with multiple processing elements (PEs) such as processing cores, memories and some other user defined logic, which are attached to routers. Inter-layer communication channels are composed of TSVs which cut across thinned silicon substrates to build connectivity after bounding [11]. To make the best use of existing IPs and take least design efforts to migrate from 2D to 3D, symmetric 3D NoC is preferred for the time being. This paper focuses on the symmetric 3D mesh structure.

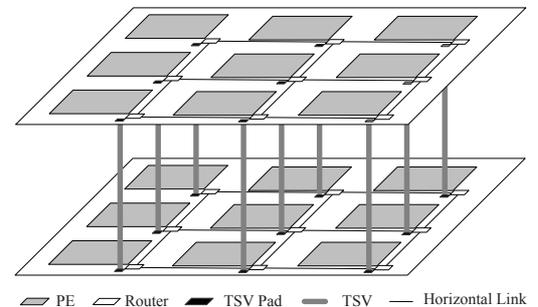


Figure 1: A Two-layer Symmetric 3D Mesh NoC

The vertical interconnects, i.e. TSVs in 3D ICs alleviate the interconnect problems that 2D ICs design encounters [9] [12] [13] [14]. TSVs also allow good compatibility with standard CMOS process and have been proved to be efficient in 3D integration [15][16][17]. However, TSVs also bring challenging problems that can not be ignored. First, as shown in Fig.1, TSV pads on the planar layer are needed for bounding purpose, which will consume chip area in each layer [18]. For example, assume the vertical link data width is 64-bit and there are 5×5 nodes in each layer. It is easy to see that we need 3200 simplex TSVs between two layers. For the time being, low density TSV pitch is more than 50um [19], thus total TSV pads will occupy at least 8.00 mm² on each layer. Even high density TSVs with 16um pitch is employed [19], this area overhead will 0.82 mm², which is almost equivalent to the size of an embedded core under 65nm technique. TSV technology is quite different from planar metal wires, and it is not expected to scale with

*The work was supported in part by National Basic Research Program of China (973) under grant No. 2011CB302503, in part by National Natural Science Foundation of China (NSFC) under grant No.(60806014, 60831160526, 60633060, 60906018, 60921002, 61076037), and in part by Hi-Tech Research and Development Program of China (863) under grant No. 2009AA01Z126.

feature size [20]. Therefore, the above problem could become even severe when transistors and wires shrink. Secondly, a large number of distributed TSV pads across the whole network aggravate routing congestion [21] which is already a challenging problem for high performance IC designs. Finally, since TSV fabrication using current techniques still suffers a relatively low yield [15] [22] [17], more TSVs will result in lower product yield [11].

Vertical interconnect serialization technique is proposed as one way to address above challenges in [18]. For instance, a 4-to-1 serialization of TSV interconnects can save more than 70% of TSV area footprint. However, serialization will cause vertical link bandwidth decreases, for example at least 75% for 4-to-1 serialization. 3D NoC throughput decreases especially when the traffic is nonuniform. Even worse, serialization inevitably increases the zero-load latency, and thus even network latency under light workload deteriorates, particularly when aggressive serialization policy is applied. Although serialization reducing the number of TSV enable higher frequency TSV transmission, additional higher frequency clock domain may not be accepted by most synchronous design.

In this paper, we propose a TSV squeezing scheme among neighboring NoC routers. The idea is mainly based on a key observation that the TSVs' utilization is pretty low and adjacent routers rarely transmit packets via their TSVs at the same time in symmetric 3D mesh NoC as will be shown in the next section. Therefore, we propose to squeeze adjacent TSVs together, i.e. neighboring routers share the same vertical TSVs in a time division multiplex mode. TSV squeezing is not only able to greatly reduce planar chip area overhead but also improve TSVs' utilization thus NoC's performance in terms of latency and throughput significantly. Experimental results prove the efficiency of the proposed scheme.

The rest of the paper is organized as follows. Section 2 motivates this work. Section 3 presents the detailed design and implementation of TSV squeezing scheme. Section 4 shows experimental results. Finally, Sections 5 concludes this paper.

2 Motivation

As stated in above, the benefits of three dimensional integration are mainly brought by vertical interconnects, i.e., TSVs. However, TSVs also bring overhead. Therefore, we can not afford unlimited TSVs, while more importantly, we need to improve TSVs' utilizations. In this section, we firstly give in-depth analysis on TSVs' utilization in 3D mesh NoC. $4 \times 4 \times 2$ 3D mesh NoC is implemented in a cycle-accurate NoC simulator written in SystemC. Dimension order routing algorithm and wormhole flow control are employed in the network. Each router in the network has two stage pipelines and two virtual channels with 8-flit depth in each input port. Each packet contains 8-flit and is injected according to a poison process. Then we analyze both the average router TSVs' utilization and neighboring routers' TSV confliction probability every 5000 cycles. Note that TSV confliction means the occasion when both TSVs in neighboring routers transmit a flit at the same time. It is not a real confliction physically.

Fig.2(a) and Fig.2(b) show average TSV utilization and con-

fliction probabilities under uniform traffic and shuffle traffic respectively. p1 and p2 in the figure stand for average TSV utilization rate and TSV confliction rate correspondingly when average injection rate is 0.3. While p3 and p4 represent the same information when average injection rate is 0.1. It can be seen that average TSV utilization is quite low especially under light network load. Even when network load goes up to 0.3, TSV still keeps idle more than 80% of the time. Moreover, TSVs in neighboring nodes are seldom used at the same time. Particularly, when the traffic is unbalanced across the network, TSV confliction probability is almost negligible as shown in shuffle traffic pattern.

Based on above analysis, we can conclude that TSV in the symmetric 3D mesh NoC is under utilized significantly. Meanwhile, data transmission in 3D NoC also exhibits a temporal characteristic that TSV in different nodes are rarely busy at the same time. In order to make full use of the vertical TSVs in 3D NoC, we develop a delicate scheme allowing neighboring nodes sharing a single TSV bundle. The idle time of TSVs can be seen as bubbles, and TSV squeezing get rid of these bubbles and maximize the utilization of TSVs, which not only save planar chip area but also maintain high performance.

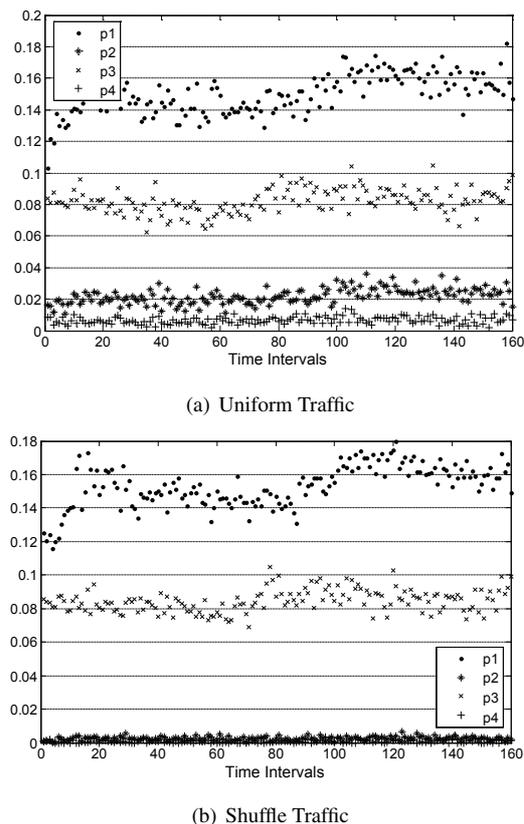


Figure 2: TSV Utilization and Confliction Probability

3 TSV Squeezing Scheme

3.1 General Idea

The general idea of TSV squeezing scheme in 3D NoC is shown in Fig.3 in which four nodes share a single TSV bundle. In each

silicon layer, router groups in upper layer and router groups in bottom layer are both coupled correspondingly. Therefore, physical vertical channels are seen as shared resources and should be granted for a transmission. Whenever there is data transmitting from one layer to the other, it will have to acquire grant of TSV sharing logic at first. Then data with grant will go straight to TSV sharing logic in current layer and flow through TSV later. Finally the data will be forwarded to corresponding router from TSV sharing logic in target layer.

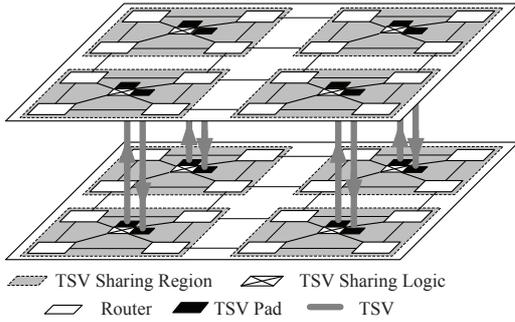


Figure 3: 3D Mesh NoC with TSV Squeezing

Despite of above transmission process, there are still another two issues needed to be explained in detail. Firstly, although routers in the group share TSV, we do not change original 3D mesh logic topology and we comply with strict dimension order algorithm instead. Therefore, there is still only one path available from a source router to a destination router. However, the new physical topology actually enables more flexible routing choices and shorter average network hops. For example, data in upper layer transmitting to nether layer doesn't have to reach the right node above destination before flowing through TSV. Namely, as long as the data arrives at the right group above destination, it can travel down to the destination. Nevertheless, new algorithm may be necessary in case of routing deadlock, and we leave it for future work.

Secondly, as any data transmitting from a router within one layer to a router within the other layer now need to traverse longer link i.e. upper horizontal link, and then vertical link, and finally bottom horizontal link. One single clock cycle will not be sufficient. To maintain the working frequency of the router, additional pipelines are inserted during vertical transmission. We will describe thoroughly how additional pipeline and the TSV squeezing logic are integrated in router architecture design in next subsection.

3.2 Router Architecture and Implementation

Fig.4 illustrates the proposed basic TSV squeezing architecture. Since TSV squeezing logic for up-to-down TSV and down-to-up TSV are almost the same, we only show squeezing logic for up-to-down TSV as an example in this figure. Corresponding down-to-up logic in planar layer is also ignored.

In order to support TSV squeezing scheme, we need to make minor modification based on conventional router. Conventional router usually consists of buffer, routing computing (RC), virtual channel allocator (VA), switch allocator (SA) and crossbar. When a flit from link arrives, it will then be stored in buffer. RC

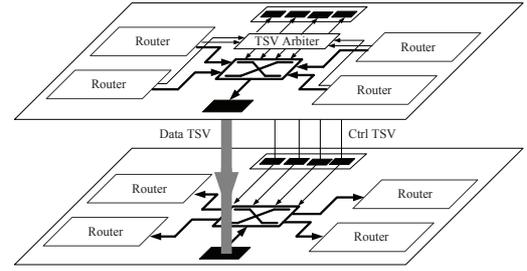


Figure 4: TSV Squeezing Architecture

detects data in buffer queue and calculates output port for each packet according to the routing algorithm. VA in the router simply serves for the head flit and it mainly aims to reserve buffer in next router. SA in the router performs the arbitration to decide the winner of all the candidates for each output port. A flit has to forward across the above components before it leaves a conventional router. For TSV squeezing router, however, the vertical interconnects are now considered as a shared resource instead of dedicated for each router. When data from router A is going to be sent to router a, it has to request grant of TSV Arbitrator (TA) in advance. Then when the data finally acquires TA grant as well as arbitration grants inner router, it can flow along the data TSV to the bottom layer. Meanwhile, TA grant is also sent to the bottom layer through Ctrl TSVs to guide crossbar forwarding the data to the destination node.

Fig.5 exhibits a conventional router architecture with TSV squeezing and logic in dashed line represents original router architecture. Additionally, TA is simply a normal arbiter, all the conventional arbitration strategies including round robin, cyclic priority and oldest first can be used.

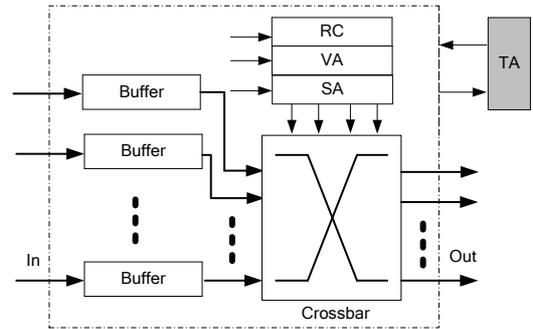


Figure 5: Modified Router Architecture

However, adding TA will increase the critical path of a conventional router. Fortunately, pre-allocation and speculation, which are widely applied to parallelize RC, VA, SA, crossbar operation and reduce critical path [23] [24], can also be adapted to TA design. Fig.6 shows a single data path of modified router microarchitecture with TSV squeezing. Shaded blocks in the figure is the additional logic for TSV squeezing. It can be seen that data now needs the grants 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 pre-allocation results are stored in registers for next cycle, they are removed from potential critical path and ensure parallel operations of

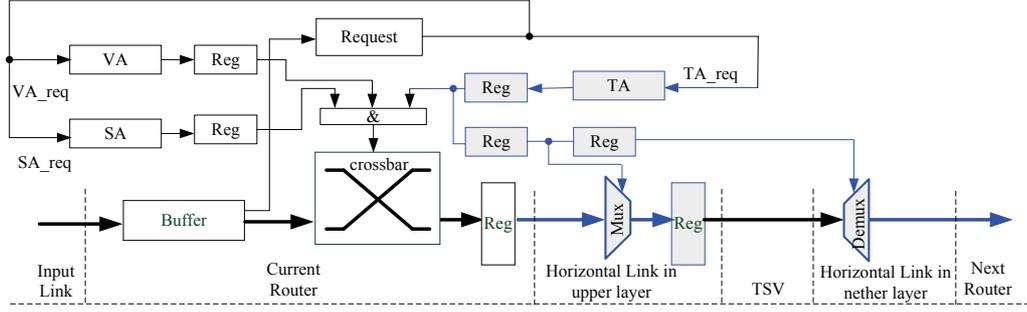


Figure 6: Data Path of Modified Router Architecture with TSV Squeezing

VA, SA, TA and crossbar.

In order to make sure TA is not far from any of the sharing nodes, TA is ideally located in the center of sharing nodes. In this case, wire latency from request to grant can't be ignored. Data path is lengthened and additional pipeline is added in case of this additional delay as shown in Fig.6. Additional data path pipelines have little control association, we simply store TA grant signal accordingly. However, it is not easy to decide how many additional pipelines should be added, because layout has quite a significant influence on horizontal link latency and TSV latency with different sizes also varies. Assume that critical path latency inner router is t_0 , horizontal link latency from router to TA is t_1 ($t_1 < t_0$), TSV latency is t_2 ($t_2 < t_0$), Mux latency is t_3 ($t_3 < t_0$), Demux latency is t_4 ($t_4 < t_0$), buffer write latency is t_5 ($t_5 < t_0$). Then original number of pipeline stages between two neighboring routers of different layers is approximately $op = \lceil (t_2 + t_5)/t_0 \rceil$, and current number of pipeline stages between neighboring routers of different layers is approximately $cp = \lceil (t_1 + t_2 + t_3 + t_4 + t_5)/t_0 \rceil$. op can be safely considered to be 1 that means it takes one clock cycle for a flit to traverse the TSVs, while cp depends on the relative location between TSV and routers. In order not to introduce too much overhead, we restrict the sharing domain to be small, e.g. no more than 4 adjacent routers, so that we need at most 2 cycles to transmit a flit across the TSV, i.e., $cp = 2$.

3.3 Global Organization for TSV Squeezing

Although TSV squeezing scheme could reduce considerable TSVs, 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 this work, we constrain routers that can share their TSVs only if they can construct a 2×2 mesh or a subpart within it. This can restrict routers sharing TSVs within 2 hops.

To save more TSVs, we first need to group routers into 2×2 meshes as many as possible, then however we should handle several corner cases as examples shown in Fig.7. For case A, we may have two design alternatives, the first one is "4+2" configuration and the other is "3+3". Both configurations need two vertical channels for the 6 routers, and save 66.7% TSV footprint when compared to conventional 3D NoC design. For case B, we have much more choices and two of them are shown in the figure. The differences among these configurations are the placement of TSV pads in the planar chip, as shown in Fig.7.

When to decide how to organize the routers to share TSVs, it greatly depends on the placement of other on-chip components to avoid congestions. In addition, we need to group as many as routers without letting the shared TSVs being far away from them. This routing and placement problem is out of the scope of this paper.

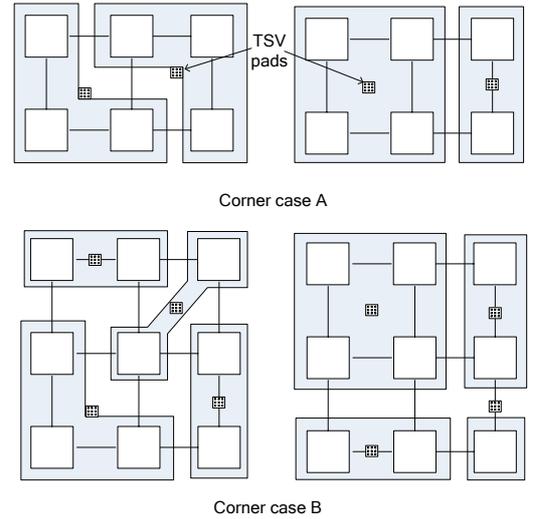


Figure 7: Three Detailed Network Partitions

4 Experimental Results

In this section, the proposed TSV squeezing scheme is evaluated through simulations in terms of performance penalty and area overhead respectively.

4.1 Performance

We developed a cycle-accurate NoC simulator using SystemC. And $4 \times 4 \times 2$ 3D mesh NoC is implemented based on the simulator. The proposed router design in the network adopts two-staged pipeline design employing wormhole flow control, dimensional order routing algorithm and round robin arbitration strategy. Furthermore, there are two 8-flit buffers in each input port.

To get a comprehensive view of the proposed design, performance of the original symmetric 3D NoC, 3D NoC with

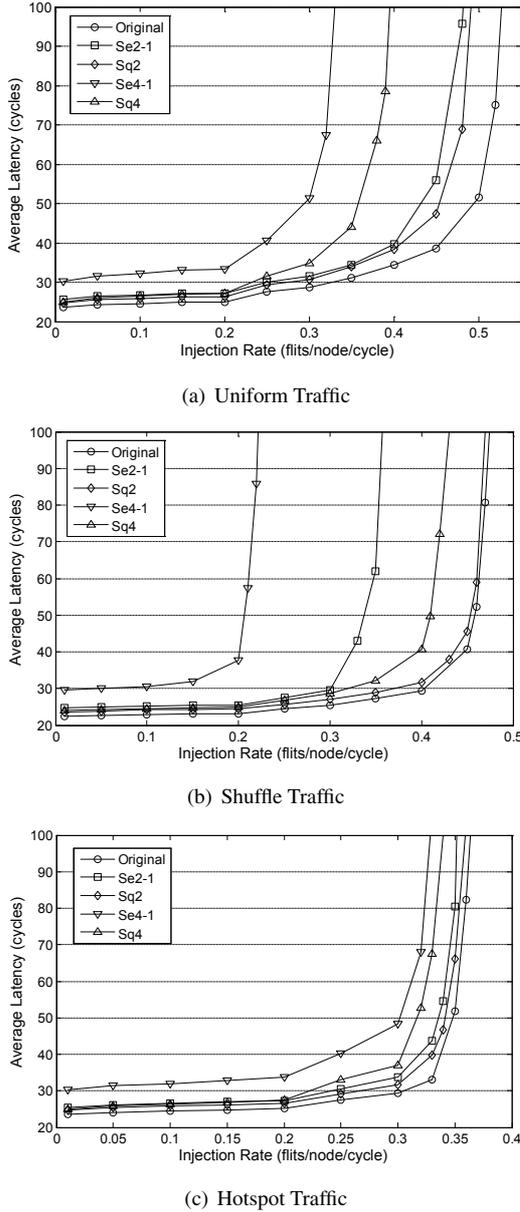


Figure 8: Latency and Throughput Comparison under Synthetic Traffic Patterns

serialized vertical channels and the proposed TSV squeezing NoC are all compared under various traffic patterns including uniform, shuffle, and hotspot. The comparison is presented in Fig.8. Se2-1 and Se4-1 in the figure represent NoC using 2:1 TSV serialization and NoC using 4:1 TSV serialization accordingly. While Sq2 and Sq4 stand for NoC with at most two neighboring routers sharing a single bundle of TSV and NoC with at most four neighboring routers sharing a single bundle of TSV respectively. It is obviously that zero-load latency of Sq4, which is close to that of original design, decreases by about 20% compared with that of Se4-1 under all these traffic patterns. The reason mainly lies in that Se4-1 brings in additional three-cycle latency induced by vertical link transmission while Sq4 adds only a single cycle penalty and pipeline transmission further alleviates the penalty. Moreover, throughput of Sq4 is approximate 30% higher than that of Se4-1 under uniform and

shuffle (Assume injection rate to be throughput when network latency reaches 100 cycles).

For Se2-1 and Sq2, since vertical link transmission penalty are almost the same except that the latter can be pipelined, thus Sq2 only performs slightly better than Se2-1. We can also observe that for shuffle traffic which results in significant unbalance in TSV utilization, it is quite suitable to apply the proposed squeezing scheme. While serialization scheme suffers more network throughput loss, as some busy while serialized TSVs become the bottleneck. In this case, as can be seen from Fig.8(b), Se4-1 and Se2-1 show significant lower throughput when compared to Sq2 and Sq4.

Finally, we consider the impact of TSV arbitration on NoC performance. Fig.9 shows performance of Sq4 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 and are omitted due to page limits. The results indicate that arbitration strategy has little influence on NoC performance, which again proves that TSV contention rarely happens. Therefore, we can employ simple arbitration method that can reduce chip area overhead.

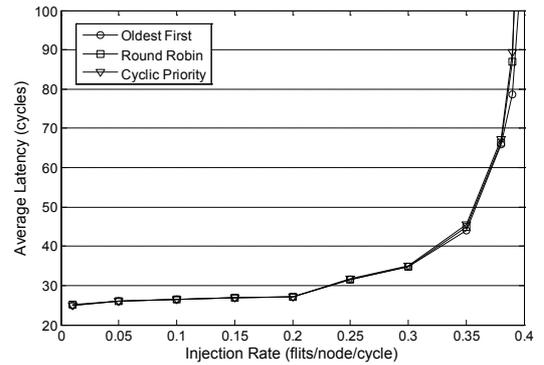


Figure 9: TSV Arbitration Impact under Uniform Traffic

4.2 Planar Chip Area

To quantify the impact using the proposed design on TSV area footprint, TSV squeezing and serialization schemes are implemented at RTL level and synthesized with Design Compiler [25]. The synthesis was performed using TSMC standard cell libraries. Fig.10 shows detailed comparison of planar chip area savings. We assume 16nm TSV pitch and 64-bits width vertical TSVs in this experiment, then we calculate the maximum planar chip area saving due to TSV reductions. For example, if 4 routers share one vertical channels, i.e., Sq4 as above, maximum 75% amount of TSVs can be eliminated. However, the additional logic, such as TSV arbiters, serializer, deserializer etc shrink the benefits of TSV reductions, indicated as overhead in this figure.

It can be seen from the figure, Sq2 achieves a little less area savings than Se2-1, while Sq4 is much superior than Se4-1. Sq4 can save more than 60% TSV planar footprint when compared to a conventional 3D mesh NoC. The reason is that additional pipeline registers are the major source for additional chip area cost. As four routers share a pipeline register in Sq4, each

router will have less area overhead than Sq2 in which only two routers share a pipeline register. Furthermore, when the technique scales down, logic area decreases dramatically, total area savings of both TSV schemes get close to maximum.

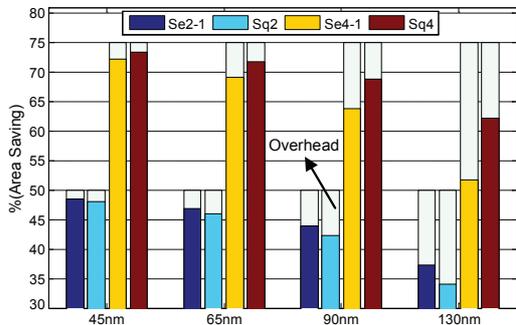


Figure 10: Planar Chip Area Saving

5 Conclusion

In this paper, we propose a TSV squeezing scheme that shares vertical interconnects among adjacent NoC routers. The idea is mainly based on the observation that TSVs' utilization in 3D symmetric mesh NoC is pretty low and adjacent routers rarely transmit data via their vertical channels at the same time. When compared to prior TSV serialization method, the proposed solution can achieve much better performance in terms of network latency and throughput. The squeezing router has quite small chip area overhead while the reduced TSV can save much planar chip area. Experimental results show that the proposed scheme is able to save more than 60% TSV planar footprint, while network latency penalty is negligible. The throughput of TSV squeezing scheme is approximate 30% higher than that of serialization scheme under certain unbalanced traffic patterns.

References

- [1] W. Dally and B. Towles, "Route packets, not wires: on-chip interconnection networks," in *Design Automation Conference, 2001. Proceedings*, pp. 684–689, 2001.
- [2] L. Benini and G. De Micheli, "Networks on chips: A new SoC paradigm," *Computer*, vol. 35, no. 1, pp. 70–78, 2002.
- [3] K. Bernstein, "New Dimensions in Performance," *System*, vol. 2002, no. 2004, p. 2006, 2000.
- [4] T. Vucurevich, "The Long Road to 3-D Integration: Are We There Yet," in *Keynote speech at the 3D Architecture Conference*, 2007.
- [5] G. Loh, Y. Xie, and B. Black, "Processor design in 3D die-stacking technologies," *IEEE Micro*, vol. 27, no. 3, pp. 31–48, 2007.
- [6] Y. Xie, G. Loh, B. Black, and K. Bernstein, "Design space exploration for 3D architectures," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 2, no. 2, pp. 65–103, 2006.
- [7] B. Feero and P. Pande, "Performance evaluation for three-dimensional networks-on-chip," in *IEEE Computer Society Annual Symposium on VLSI, 2007. ISVLSI'07*, pp. 305–310, 2007.
- [8] I. Loi, F. Angiolini, and L. Benini, "Supporting vertical links for 3D networks-on-chip: toward an automated design and analysis flow," in *Proceedings of the 2nd international conference on Nano-Networks*, p. 15, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007.
- [9] 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 Computer Architecture News*, vol. 34, no. 2, p. 141, 2006.
- [10] J. Kim, C. Nicopoulos, D. Park, R. Das, Y. Xie, V. Narayanan, M. Yousif, and C. Das, "A novel dimensionally-decomposed router for on-chip communication in 3D architectures," *ACM SIGARCH Computer Architecture News*, vol. 35, no. 2, p. 149, 2007.
- [11] 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 *Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design*, pp. 598–602, IEEE Press, 2008.
- [12] D. Park, S. Eachempati, R. Das, A. Mishra, Y. Xie, N. Vijaykrishnan, and C. Das, "MIRA: A multi-layered on-chip interconnect router architecture," in *Proceedings of the 35th International Symposium on Computer Architecture*, pp. 251–261, IEEE Computer Society, 2008.
- [13] V. Pavlidis and E. Friedman, "3-D topologies for networks-on-chip," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 10, pp. 1081–1090, 2007.
- [14] Y. Xu, Y. Du, B. Zhao, X. Zhou, Y. Zhang, and J. Yang, "A low-radix and low-diameter 3D interconnection network design,"
- [15] A. Topol *et al.*, "Three-dimensional integrated circuits," *IBM Journal of Research and Development*, vol. 50, no. 4/5, p. 506, 2006.
- [16] K. Bernstein, P. Andry, J. Cann, P. Emma, D. Greenberg, W. Haensch, M. Ignatowski, S. Koester, J. Magerlein, R. Puri, *et al.*, "Interconnects in the third dimension: Design challenges for 3D ICs," in *Proceedings of the 44th annual Design Automation Conference*, p. 567, ACM, 2007.
- [17] R. Patti, "Three-dimensional integrated circuits and the future of system-on-chip designs," *Proceedings of the IEEE*, vol. 94, no. 6, 2006.
- [18] S. Pasricha, "Exploring serial vertical interconnects for 3D ICs," in *Proceedings of the 46th Annual Design Automation Conference*, pp. 581–586, ACM, 2009.
- [19] H. Sangki, "3D Super-Via for Memory Applications,"
- [20] 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 *Proceedings of the 2004 international symposium on Physical design*, p. 115, ACM, 2004.
- [21] C. Ababei, Y. Feng, B. Goplen, H. Mogal, T. Zhang, K. Bazargan, and S. Sapatnekar, "Placement and routing in 3D integrated circuits,"
- [22] B. Swinnen, W. Ruythooren, P. De Moor, L. Bogaerts, L. Carbonell, K. De Munck, B. Eyckens, S. Stoukatch, D. Tezcan, Z. Tokei, *et al.*, "3D integration by Cu-Cu thermo-compression bonding of extremely thinned bulk-Si die containing 10 μ m pitch through-Si vias," in *Electron Devices Meeting, 2006. IEDM'06. International*, pp. 1–4, 2006.
- [23] R. Mullins, A. West, and S. Moore, "Low-latency virtual-channel routers for on-chip networks," in *Proceedings of the 31st annual international symposium on Computer architecture*, IEEE Computer Society Washington, DC, USA, 2004.
- [24] R. Mullins, A. West, and S. Moore, "The design and implementation of a low-latency on-chip network," in *Proceedings of the 2006 Asia and South Pacific Design Automation Conference*, p. 169, IEEE Press, 2006.
- [25] S. Inc, "Design compiler," *Online*, <http://www.synopsys.com>, 2009.