

# Chemical-Mechanical Polishing-Aware Application-Specific 3D NoC Design

Wooyoung Jang, *Member, IEEE*, and David Z. Pan, *Senior Member, IEEE*

**Abstract**—Three-dimensional (3D) integration with through-silicon vias (TSVs) is promising in the integration of many cores into a single chip. Network-on-chip (NoC) can efficiently manage the complicated 3D interconnections. However, irregular and dense TSV arrays used as vertical links in 3D NoC cause severe TSV height variation during silicon-thinning and chemical-mechanical polishing (CMP) processes. It may lead to TSV bonding failure between silicon layers. In this paper, we propose the first CMP-aware application-specific 3D NoC design that minimizes such TSV height variation and thus reduces the bonding failure, and meanwhile optimizes conventional NoC design objectives such as hop count, wirelength, power consumption, and area. Our 3D NoC design assigns cores to proper silicon layers, determines the 3D NoC topology, allocates routing paths, and then floorplans all cores, routers, and TSV arrays in a CMP-aware manner. The key idea behind this 3D NoC design flow is to determine the CMP-aware 3D NoC topology where TSV arrays with low and uniform metal density are inserted between adjacent layers. Experimental results show that our CMP-aware 3D NoC design achieves, on average, 17.9% lower TSV height variation, 15% lower hop count, 2.3% shorter total wirelength, and 7.8% lower power consumption than the previous state-of-the-art 3D NoC designs.

**Index Terms**—Chemical-mechanical polishing (CMP), Cu-Cu bonding, three-dimensional (3D) networks-on-chip, through-silicon via (TSV).

## I. INTRODUCTION

AS SHRINKING the horizontal feature size has critical limitations, vertically stacking silicon based on through-silicon vias (TSVs) has gained tremendous interest from academia and industry for the future integrated circuits (ICs). Network-on-chip (NoC) [3], [5] is a scalable on-chip communication solution for complex three-dimensional (3D) interconnections since it can control the number of TSVs necessary for various applications. However, 3D NoC is required to meet not only application performance and power constraints [27], [29], but also manufacturing constraints imposed by the 3D technologies. Therefore, the combination of the 3D technologies and the NoC offers new challenges and opportunities.

Manuscript received March 15, 2012; revised August 5, 2012 and November 26, 2012; accepted December 11, 2012. Date of current version May 15, 2013. This work is supported in part by NSF under grant CCF-1018750. This paper was recommended by Associate Editor Y. Xie.

W. Jang is with the Department of Electronics and Electrical Engineering, Dankook University, Gyeonggi 448-701, Korea (e-mail: wooyoung.jjang@gmail.com).

D. Z. Pan is with the Department of Electrical and Computer Engineering, University of Texas, Austin, TX 78712 USA (dpan@ece.utexas.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCAD.2013.2237771

So far, many researchers have addressed the issues of 3D floorplanning and NoC topology generation with consideration of thermal hot spots. For example, in 3D floorplanning, cores with high power density are assigned to the silicon layer attached to heat sinks and spread out at each silicon layer to reduce peak temperature and help mitigate reliability problems such as electromigration, stress, dielectric breakdown, leakage-thermal run-away, and speed of devices [4], [9]. Based on the 3D thermal-aware floorplanning, 3D NoC topology is then synthesized [17], [21], [25].

Besides the thermal and mechanical stress effects [2], [30], 3D-IC integration has a lot of manufacturability and layout challenges related to TSVs and landing pads [14], [19]. One particular challenge is that the wide range of area occupied by TSVs and landing pads increases nonuniform metal density distribution, and thus results in the critical variation of landing pad or wire thickness and TSV height during a chemical-mechanical polishing (CMP) process [6], [15], [22]. In 3D-IC fabrication, the CMP process is used for the removal of extra Cu on silicon after filling TSVs with Cu or depositing Cu for TSV landing pads, called Cu-CMP and silicon backside thinning, called silicon-CMP. The uneven Cu-wire thickness changes wire resistance and coupling capacitance between wires, and thus results in critical on-chip timing variation. Moreover, the irregular TSV height and landing pad thickness lead to bonding failure between TSVs and landing pads. In order to mitigate the Cu-CMP variation, dummy metal fill insertion is commonly used on empty metal spaces even though it may affect RC parasitics [11], [12]. Similarly, dummy TSV insertion may reduce the silicon-CMP variation, but it is rarely used since the dummy TSVs can not only result in severe TSV manufacturing stress, but also significantly reduce the usable silicon area of an entire chip. Since TSV height variation after the silicon-CMP strongly depends on the regularity and density of TSV distributions, 3D NoC designs with various vertical links composed of tens to hundreds of TSVs should consider TSV placement with low and uniform density.

In this paper, we propose a CMP-aware application-specific 3D NoC design that minimizes TSV height variation and thus reduces bonding failure, and meanwhile optimizes conventional NoC design objectives such as hop count, wirelength, power consumption, and die area. For NoC vertical links composed of tens to hundreds of TSVs, the layout of each individual TSV is undesirable since it may result in complex global routing and TSV manufacturing may stress affect more transistors [2], [30]. Therefore, TSVs should be placed as

an array type in 3D NoC. However, the array with dense TSVs is sensitive to the CMP that results in high TSV height variation, and thus leads to severe bonding failure. Moreover, if the arrays with different local TSV densities are placed in the same layer, bonding TSVs on landing pads is more difficult. NoC can include not only various link widths but also one- and two-way links of which the metal densities may be different. Therefore, the TSVs in an array should be placed with a pitch that results in low TSV height variation endured by a bonding technique, and the TSV arrays with the same density should be inserted in each layer. In addition, previous 3D NoC designs cannot handle TSV arrays during placement and routing stage since the size of TSV arrays is too large [17], [21], [25]. Therefore, TSV arrays should be handled during the floorplanning stage in physical design. Based on these motivations, the major contributions of this paper include the following.

- 1) We show that TSV height variation during the silicon-CMP process is severe in 3D NoC where dense TSV arrays are used as vertical links.
- 2) We propose a CMP-aware application-specific 3D NoC design that minimizes TSV height variation and optimizes conventional NoC design objectives.
- 3) We present CMP-aware 3D NoC techniques for core-to-layer assignment, topology synthesis, and floorplanning.
- 4) We show that the proposed 3D NoC design reduces TSV height variation with lower design cost, and meanwhile achieves less hop count, wirelength, and power consumption.

To the best of our knowledge, this is the first work that addresses TSV height variation by the CMP process in 3D NoC. The remainder of this paper is organized as follows. Section II reviews related works. Section III introduces CMP and Cu–Cu thermo-compression direct bonding, and then addresses various TSV layouts and their CMP variation. Section IV shows the problem formulation and proposed CMP-aware application-specific 3D NoC design flow. Section V presents detailed techniques of our proposed algorithms. Section VI shows experiment results and Section VII concludes this paper.

## II. RELATED WORKS

Several works for 3D ICs have explored thermal floorplanning [4], [9]. A fast thermal-driven floorplanning algorithm with a 3D floorplan representation and integrated compact resistive network thermal model was proposed in [4]. Hung *et al.* [9] presented interconnect and thermal-aware floorplanning for 3D microprocessors.

Recently, researchers have addressed the issues of application-specific 3D NoC topology synthesis with the limited number of TSVs [17], [21], [25]. Most of the previous 3D NoC designs have assumed that assigning cores to silicon layers and floorplanning the assigned cores were given as inputs. Then, the optimal location and interconnection of routers on each layer were determined based on the result of the 3D floorplanning. Yan *et al.* [25] presented a 3D NoC synthesis algorithm that made the use of accurate power and

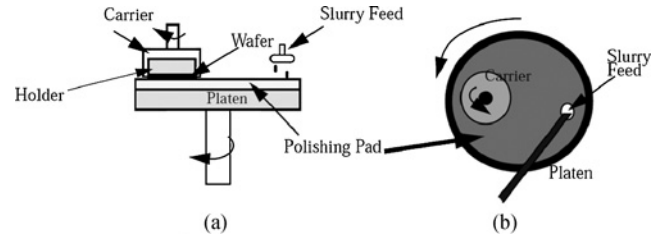


Fig. 1. Typical rotary CMP tool [6]. (a) Side view. (b) Top view.

delay models for 3D wiring with TSVs. In [17] and [21], 3D NoC topology synthesis algorithms based on the direct extension of 2D NoC synthesis procedures were proposed, where NoC was separately designed for each layer, and the connectivity of switches across layers was then determined with the number of TSVs being given.

However, these previous 3D NoC designs have not considered CMP variation that is highly dependent on the underlying TSV density, which may lead to bonding failures and timing variations. In addition, as routers and TSV arrays are becoming as large as other cores, the previous 3D NoC designs would achieve low design qualities. This is because 3D floorplanning was first performed without large routers [26], [28] and TSV arrays, and thus there might be not enough spaces to place the routers and TSV arrays [17], [21]. Even if 3D floorplanning is again performed after deciding a 3D NoC topology [25], their design qualities such as wirelength, power consumption, and area are severely limited.

## III. PRELIMINARIES

### A. Chemical-Mechanical Polishing

One of the most potential sources of yield loss and timing variation in 3D technologies is TSV bonding on land pads. In a typical industrial bonding procedure [23], [24], a TSV wafer is ground down to target thickness slightly above TSV depth (keeping TSVs unexposed) and further thinned by CMP. The CMP process uses both chemical and mechanical means to polish the surface of a wafer. In a typical rotary CMP tool, a wafer is held on a rotating holder, as shown in Fig. 1. The surface of the wafer polished is pressed against a polishing pad that is mounted on a rotating disk. Slurry composed of particles suspended in a chemical solution is also deposited on the pad as a chemical abrasive. The material-removal mechanism of CMP is similar to the removal found in glass polishing. First, a chemical reaction softens the surface of materials to be removed later as it creates a hydroxylated-form material that has weaker atomic bonds. Then, a mechanical surface abrasion aided by slurry particles removes the material during a polishing process. Fig. 2(a) shows the uneven surface on the reverse side of a wafer after CMP. Subsequently, the polished silicon surface is plasma etched such that the TSVs protrude from the wafer as shown in Fig. 2(b). On the contrary, TSV landing pads are commonly fabricated on the top metal layer in a damascene process and designed to be larger than TSVs to prevent overlay errors. A CMP process is also used to remove overburden Cu on the top metal layer with TSV

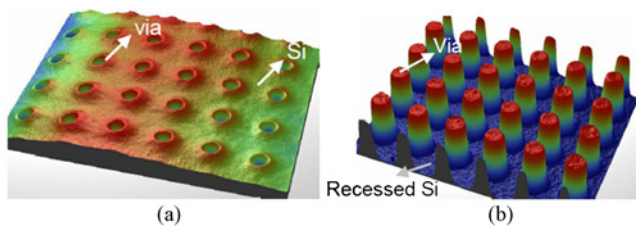


Fig. 2. Local topography on the reverse side of wafer after (a) CMP and (b) Si-recess etch following CMP [23].

land pads. Finally, the wafer or die with TSV landing pads is bonded with a different wafer or die with TSVs.

### B. Cu–Cu Thermo-Compression Direct Bonding

Recently, a micro-bump-less Cu–Cu direct bonding technique has attracted great attention in 3D die integration since the same bonding medium can prevent the formation of an intermetallic at the interface between TSVs and landing pads [23], [24]. In addition, the Cu–Cu direct bonding is further favored than solder-based connections since: 1) the Cu–Cu bond is easily scalable and ultra-fine pitch can be achieved; 2) Cu has excellent electrical and thermal conductivities; and 3) Cu has low electromigration resistance for high current density. The direct Cu–Cu bonding has been demonstrated from thermo-compression bonding via parallel applications of heat and pressure (typically  $\sim 300\text{--}400^\circ\text{C}$  and  $\sim 200\text{ kPa}$ ). The bonding mechanism is based on interdiffusion of Cu atoms and grain growth, and hence it is also widely known as diffusion bonding.

### C. TSV Layouts and TSV Height Variation

The goal in Cu-CMP is to polish a barrier and remove overburden Cu on silicon after filling TSVs with Cu and depositing Cu for landing pads. Cu-CMP involves simultaneous polishing of three materials: Cu, dielectric (oxide), and barrier (Tan, Ti, etc.). However, due to different chemical effects on the materials and their pattern differences in terms of pattern density, Cu line width, and oxide line space, the removal rates of the materials are quite different. Thus, the difference in removal rates results in different polish times across a wafer. For example, in Fig. 3(a), by the time the excess Cu and barrier on TSVs used for a 64-b link are cleared at a point on the die, those on TSVs used for 128-b links might, at another point, have already been cleared, where chemicals used for Cu-CMP react well on Cu rather than silicon. Hence, either the 128-b TSVs are overpolished at the time the excess Cu and barrier on the 64-b TSVs are cleared or the excess Cu and barrier on the 64-b TSVs are not cleared at the time the excess Cu and barrier on TSVs used for a 128-b link are cleared. Fig. 3(a) shows that the 128-b TSVs are overpolished after Cu-CMP. The uneven polishing problem in Cu-CMP can be solved by metal fill synthesis where dummy metals grounded or floating are inserted in the empty spaces of the top metal layer [11], [12].

Silicon-CMP is performed for finely thinning the reverse side of silicon after roughly grinding it since the processing time of silicon-CMP is too long. As silicon-CMP also

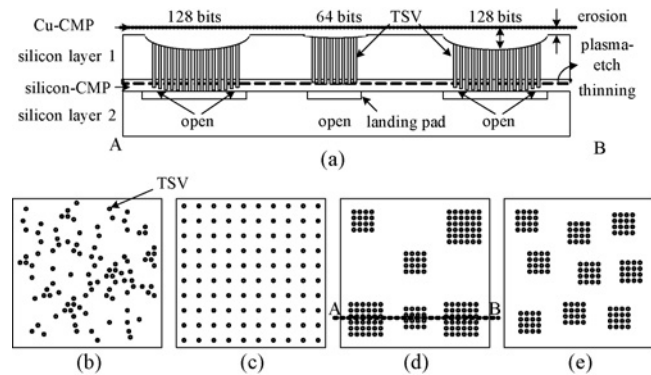


Fig. 3. TSV layouts and TSV height variation induced by the CMP process. (a) Cross section of wafer side. (b) Uneven TSVs. (c) Uniform TSVs. (d) Uneven TSV arrays. (e) Uniform TSV arrays.

involves simultaneous polishing of silicon, Cu, and barrier, their removal rates are different according to various chemical effects on the materials and uneven TSV densities in terms of the diameter of TSV, the pitch of TSV, and the size of TSV array. The uneven removal rates of the materials result in different polish times across the reverse side of a wafer. For example, in Fig. 3(a), by the time the silicon and barrier under TSVs used for a 64-b link are cleared at a point, the silicon and barrier under TSVs used for 128-b links might have been not cleared yet, where chemicals used for silicon-CMP react well on silicon rather than Cu. Hence, either the silicon and barrier under the 128-b TSVs are underpolished at the time the silicon and barrier under the 64-b TSVs are cleared or the 64-b TSVs are overpolished at the time the silicon and barrier under the 128-b TSVs are cleared. Fig. 3(a) shows the 128-b TSVs are underpolished after silicon-CMP. In [23], IMEC TSV technology showed that within-die thickness variation after silicon-CMP was  $1.5\ \mu\text{m}$  for a die size of  $10.6 \times 10.6\ \text{mm}^2$  when TSVs of which the diameter, pitch, and density are  $5\ \mu\text{m}$ ,  $10\ \mu\text{m}$ , and  $10\ \text{k/mm}^2$ , respectively, were evenly distributed over the whole wafer, as shown in Fig. 2. The within-die thickness variation is more sensitive to irregular and high TSV density and directly related to TSV height variation. Consequently, the uneven TSV height variation can induce severe TSV bonding failure, as shown in Fig. 3(a). In particular, the bonding failure will be more severe in Cu–Cu direct thermo-compression bonding widely used in 3D IC since TSVs must be directly contacted to landing pads without any microbump. Unlike Cu-CMP, metal fill synthesis is not an efficient solution for silicon-CMP since dummy TSV insertion would significantly increase the area of a chip.

TSVs can be placed with different schemes during placement and routing [2], [14], [19]. If TSVs are laid out without any constraints imposed by 3D technologies, they are distributed as shown in Fig. 3(b). Whereas such a layout achieves the shortest wirelength, TSV height variation induced by silicon-CMP increases due to uneven TSV density. In Fig. 3(c), TSVs are placed with globally uniform density distributions. The TSV distribution provides the least TSV height variation to 3D ICs. However, such a TSV layout is not suitable for NoC vertical links composed of tens to hundreds of TSVs since it results in so complex global routing that any wire in

the same vertical link may detour with a long path. The long wires detoured make system performance degraded or timing closure difficult. In addition, the layout of each individual TSV causes its manufacturing stress to more transistors [2], [30]. Therefore, grouping TSVs to an array and then laying out the array is more desirable for 3D NoC.

In Fig. 3(d), there exist two kinds of TSV arrays. The small array represents a single one-way NoC link and the large array represents a single two-way NoC link that includes two times more TSVs than the one-way NoC link. TSVs in the small array fail to contact landing pads since TSVs in the large array are less cleared than those in the small array during silicon-CMP such that the surface of a die is uneven. Fig. 3(a) shows the cross section of AB in Fig. 3(d), the 64-b TSV array which has the strong possibility of failing to contact landing pads on silicon layer 2 since the 128-b TSV array is underpolished. In addition, since the metal density of the 128-b TSV is high, its own silicon-CMP variation is so high that TSVs in the array have the possibility of failing to contact landing pads. We can control the local TSV density defined as the size of a TSV array divided by a TSV pitch. If the 128-b TSV array has a wider TSV pitch, its density can be as low as that of the 64-b TSV array. However, since it has the penalty of area, we focus on using a single type of a TSV array of which the metal density is sufficiently low, as shown in Fig. 3(e).

#### IV. PROBLEM FORMULATION

In most of the previous application-specific 3D NoC designs [17], [21], [25], 3D floorplanning is first performed. Then, a 3D network topology is determined, depending on the result of the 3D floorplanning as shown in Fig. 4(a), where a 3D technology constraint is just the number of allowable TSVs. The constraint is not sufficient for robust and reliable 3D ICs, and CMP variation resulting in severe bonding failure is not considered. In addition, since the 3D floorplanning composed of assigning cores to layers and floorplanning the cores in each layer is first performed without any routers and TSV arrays, there may be no enough dead space where the routers and TSV arrays can be physically placed after deciding a 3D network topology [17], [21]. The area of the latest routers is no longer small since its complexity rapidly increases due to a virtual channel, a complex flow controller, and an adaptive routing path allocator. In order to prevent overlapping routers inserted and cores already floorplanned, additional floorplanning is performed in each layer after deciding a network topology [25]. However, such a conventional 3D NoC design flow is not efficient for reducing wirelength, hop count, and thus energy consumption as the routers get more and more complex. In addition, the previous 3D NoC designs have not considered the layout of TSVs since they assume that TSVs are laid out during placement and routing (PAR). Furthermore, the layout of each individual TSV without considering NoC architecture in the PAR stage worsens CMP variation, global routing, and manufacturing stress to more transistors. Finally, since TSV arrays for NoC links in Fig. 3(d) and (e) are much larger than other placement objects, it is not efficient that they are considered in the PAR stage.

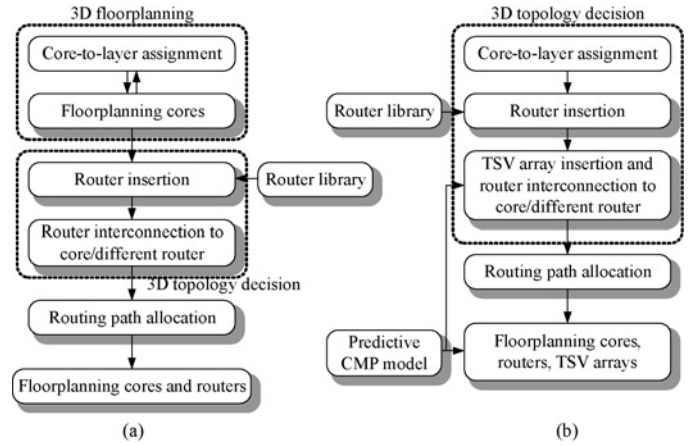


Fig. 4. Application-specific 3D NoC design flows. (a) Conventional 3D NoC design flow. (b) CMP-aware 3D NoC design flow.

Fig. 4(b) shows the proposed CMP-aware NoC design flow covering such issues. We first assign  $n$  cores to  $k$  layers with the purpose of minimizing communication between layers under a given area constraint, thus using the minimum TSVs. Based on communication relations between cores assigned to the same layer, the number of allowable routers is inserted in each layer, and then the routers are interconnected to the cores and different routers in the same layer, where the number of interconnecting a single router to cores and other routers is constrained. The goal of our network topology decision in each layer is to minimize hop count under the limited network resources. Then, the routers are interconnected to different routers in adjacent layers by only one-way vertical links. That is, any routers in different layers are not interconnected by two-way vertical links. Using only one-way links over a whole chip interconnects different layers with uniform and low local TSV density. Since the number of allowable TSVs between layers is also limited by a given area constraint, the best vertical interconnections minimizing the total hop count are selected. Then, routing paths without deadlock and livelock are allocated on the existing interconnections. We compute a TSV pitch applied in the one-way link and then a TSV array is composed. Finally, all cores, routers, and TSV arrays are simultaneously floorplanned in each layer.

Fig. 5 shows the application-specific 3D NoC designs where a dotted arrow represents the conventional 3D NoC design flow and a gray arrow represents the proposed CMP-aware 3D NoC design flow. We assume that six cores are assigned to an arbitrary layer, as shown in Fig. 5(a). The cores are first floorplanned without routers and TSVs as shown in Fig. 5(b), and then a NoC topology is determined as shown in Fig. 5(c) or (f). We assume that the number of allowable routers is 2 in this example. Fig. 5(c) shows the NoC topology optimized for the minimum hop count. However, since the wirelengths of  $c1-r1$  and  $c5-r1$  are too long, it may be difficult to satisfy target operating clock frequency. In Fig. 5(f), a core is interconnected to a router close to the core. Whereas such a topology makes the longest wires short, hop count is considerably increased and thus worsens dynamic energy consumption and communication latency. On the contrary, our

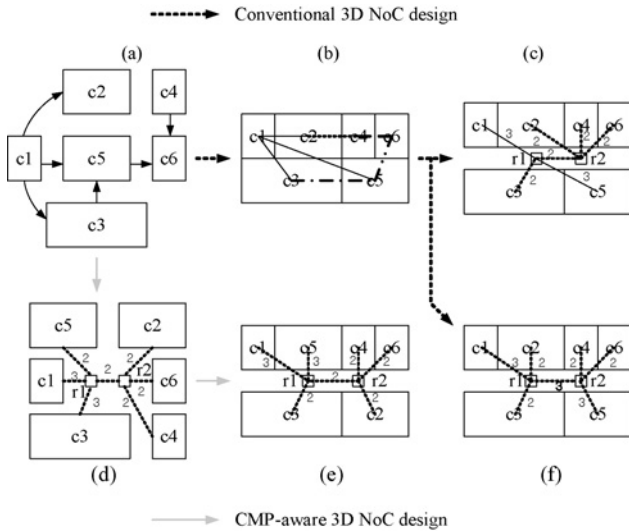


Fig. 5. Examples of application-specific 3D NoC design. (a) Core graph in layer. (b) Floorplanning. (c) Topology decision for minimum hop counts. (d) Topology decision. (e) Floorplanning. (f) Topology decision for minimum wirelength.

CMP-aware 3D NoC design first determines a 2D network topology. As shown in Fig. 5(d), two acceptable routers are inserted, and then interconnected to cores and different routers for the minimum hop count. Finally, six cores, two routers, and TSV arrays (not shown in this figure for simple example) are simultaneously floorplanned, as shown in Fig. 5(e). As a result, the proposed CMP-aware 3D NoC design achieves the same hop count as Fig. 5(c), and meanwhile similar length of the longest wire and total wirelength to Fig. 5(f).

We start to solve the CMP-aware application-specific 3D NoC issues from a core graph. A graph  $G(V, E)$  with  $n$  vertices is a directed graph, where each vertex  $v_i \in V$  represents a core, a router, a TSV array and each directed edge  $e_{i,j} \in E$  represents communication relation between  $v_i$  and  $v_j$ .  $vol(e_{i,j})$  represents communication volume between  $v_i$  and  $v_j$  and  $wl(e_{i,j})$  represents wirelength between  $v_i$  and  $v_j$ .

#### A. Core-to-Layer Assignment

Core-to-layer assignment allows cores to move from continuous space to discrete space, forcing each core to exactly occupy one layer. That is, a set of cores  $V = \{v_1, v_2, \dots, v_n\}$  is assigned to  $k$  layers  $L = \{l_1, l_2, \dots, l_k\}$ , and thus  $V = \{V^{l_1}, V^{l_2}, \dots, V^{l_k}\}$  is obtained, where  $V^{l_i} = \{v_1^{l_i}, v_2^{l_i}, \dots, v_n^{l_i}\}$ , where  $j < n$ . The area of cores is represented as  $\{A_1, A_2, \dots, A_n\}$ . To equally assign the area of cores to layers, an area constraint is defined as

$$\alpha_{\min} \sum_{i=1}^n \frac{A_i}{k} < A_l < \alpha_{\max} \sum_{i=1}^n \frac{A_i}{k} \quad (1)$$

where  $\alpha_{\min}$  and  $\alpha_{\max}$  are the acceptable minimum and maximum area coefficients ( $\alpha_{\min} < 1 < \alpha_{\max}$ ). We also consider thermal hot spots based on the thermal model proposed in [4]. The thermal model assigns a high power density core to a lower silicon layer attached to a heat sink. With the area constraint, the objective of our layer assignment is to minimize communication between different layers and temperature as

follows:

$$\min \left[ \beta_1 \sum vol(e_{i,j}) \cdot |u - v| + \beta_2 \left\{ \sum_{p=1}^k \left( P_p \sum_{q=1}^p R_q \right) + R_b \sum_{p=1}^k P_p \right\} \right] \text{ s.t. } \forall v_i \in V^{l_u}, \forall v_j \in V^{l_v} (u \neq v) \quad (2)$$

where  $\beta_1$  and  $\beta_2$  are weighting coefficients,  $R_q$  is a thermal resistor in layer  $q$ ,  $P_p$  is the sum of current source in layer  $p$ , and  $R_b$  is the thermal resistor of the bottom layer material.

#### B. 3D NoC Topology Decision and Routing Path Allocation

Given the number of allowable routers and TSV arrays and the number of different routers interconnected to a single router in each layer, we interconnect a router to cores and different routers in the same layer. A router communication graph  $RCG(R, C)$  with  $m$  vertices is a directed graph, where each vertex  $r_i \in R$  represents a router, and each directed edge  $c_{i,j} \in C$  represents communication between  $r_i$  and  $r_j$ . The objective of our 2D topology decision in each layer is as follows:

$$\min \left[ \sum vol(e_{i,j}) \cdot dist(M(v_i), M(v_j)) \right] \text{ s.t. } bw(link(M(v_i), M(v_j))) \geq vol(e_{i,j}), (\forall v_i, \forall v_j) \in V^{l_u} \quad (3)$$

where  $dist(r_p, r_q)$  is distance (or hop count) between  $r_p$  and  $r_q$  and  $M()$  is a core-to-router mapping function, e.g.,  $r_p = M(v_i)$  and  $r_q = M(v_j)$ .  $link(r_p, r_q)$  is all links that any packet in  $r_p$  passes for reaching  $r_q$ . Then, we interconnect routers in adjacent layers, based on the RCG graphs. The objective of our topology decision among layers is as follows:

$$\min \left[ \sum vol(e_{i,j}) \cdot dist(r_p, r_q) \right] \text{ s.t. } link_{TSV}(r_p, r_q) \neq link_{TSV}(r_q, r_p), bw(link(r_p, r_q)) \geq vol(e_{i,j}), \forall v_i \in V^{l_u}, \forall v_j \in V^{l_v} (u \neq v) \quad (4)$$

where  $link_{TSV}(r_p, r_q) \in link(r_p, r_q)$  is a vertical link that any packet in  $r_p$  passes for reaching  $r_q$ . This equation indicates that routers in different layers are interconnected by only one-way links. Thus, CMP variation resulting in uneven TSV heights can greatly be reduced and the yield of TSV bonding can greatly be improved.

#### C. Floorplanning

We compute a TSV pitch where a bonding technique used should endure TSV height variation in the number of TSVs covering a one-way vertical link based on our predictive CMP model. Then, TSV arrays are composed and inserted between any routers in adjacent layers. As the inputs of our floorplanner, we take a set of cores, routers, and TSV arrays,  $\{v_1, v_2, \dots, v_n\}$ .  $v_i$  is a  $W_i \times H_i$  rectangle and aspect ratio  $H_i/W_i$ . Each block can be free to rotate and change the aspect ratio continuously in a given range  $[AR_{\min,i}, AR_{\max,i}]$ . A floorplan  $F$  is the assignment of  $(x_i, y_i)$  for each block  $v_i$  without any overlap of all cores, routers, and TSV arrays, where half-perimeter wire length estimation is used. In this

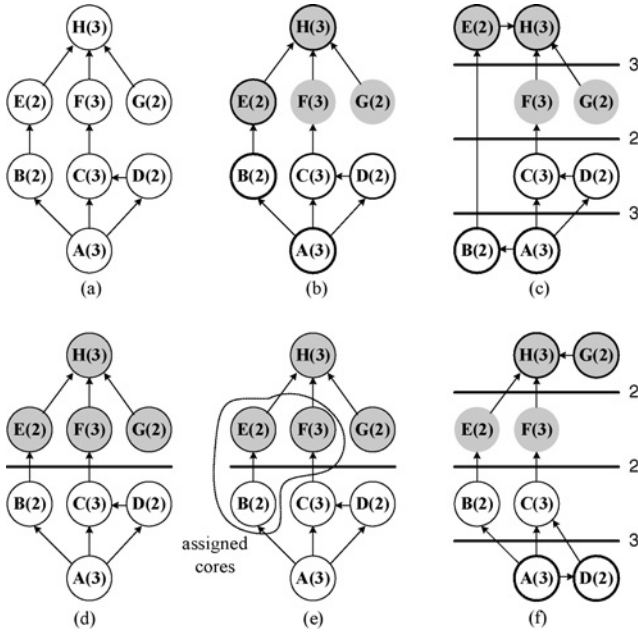


Fig. 6. Examples of assigning eight cores to four layers. (a) Core graph. (b) 4-way min-cut partitioning. (c) Layer assignment of (b). (d) First bipartitioning. (e) Layer assignment of cores B, E, and F. (f) Second bipartitioning.

step, we consider additional thermal hot spots based on the thermal model proposed in [4]. The thermal model minimizes the maximum temperature difference in the same layer. Therefore, the objective of our floorplan  $F$  is as follows:

$$\min \left[ \begin{array}{l} \gamma_1 \sum A_i + \gamma_2 \sum (wl(e_{p,q}) \times vol(e_{p,q})) \\ + \gamma_3 (\max(T(x, y, l_u)) - \min(T(x, y, l_u))) \end{array} \right] \quad (5)$$

s.t.  $(\forall v_i, \forall v_p, \forall v_q) \in V^{l_u}, \max(wl(e_{p,q})) < th_w$

where  $\gamma_1, \gamma_2$ , and  $\gamma_3$  are weighting factors.  $T(x, y, l_u)$  is the temperature of a tile in  $x, y$ , and  $l_u$  at  $x$ -axis,  $y$ -axis, and layer, respectively, and  $th_w$  is the maximum allowable wirelength.

## V. CMP-AWARE 3D NOC DESIGN

### A. CMP-Aware Core-to-Layer Assignment

Since the number of TSVs required depends on bandwidth between different layers, vertical communication volume should be minimized with thermal consideration. In addition, each layer should be smaller than the area constraint from (1) and network latency demanded by applications or cores should be satisfied. If there are any cores that are severely sensitive to network latency, it is more desirable to assign the cores to the same layer than different layers. Thus, as a preliminary work, the cores can be merged into one large core if the large core meets any other constraints. Fig. 6 shows two different core-to-layer approaches where eight cores are assigned to four layers. Let a core graph be given as shown in Fig. 6(a), where all edges have the same weight, all cores have the same power density, and the number is the area of a core for simple explanation.

The first approach is that 4-way minimum-cut area-balanced partitioning is performed, and then the partitioned subgroups are one-to-one assigned to different layers. For example, in Fig. 6(b), the cores are partitioned to  $\{A, B\}, \{C, D\}, \{F, G\}$ , and  $\{E, H\}$  that have the same area and the minimum cut. Then, the partitioned subgroups are one-to-one assigned to favorable layers, achieving the minimum hop as shown in Fig. 6(c).

### Algorithm 1 Core-to-Layer Assignment by Recursive Bipartitioning

- 1: **while** the number of partitioned layers is not equal to the target number of layers **do**
- 2: Find bipartitions of cores with the minimum cost computed by (2);
- 3: Compute the communication gain ( $CG_i$ ) of core  $i$  in layer  $k$ ;
- 4: **if**  $CG_i \geq 0$  **then**
- 5: Core  $i$  is assigned to layer  $k$ ;
- 6: **end if**
- 7: **end while**

The second approach that we propose in Algorithm 1 recursively performs area-balanced bipartitioning with the minimum cost computed from (2). Fig. 6(d) shows the result of the first bipartitioning where the same area and the minimum cut are obtained (line 2). Then, any core that communicates other cores in a different layer is assigned in advance, depending on their communication gain as shown in Fig. 6(e) (line 5). The communication gain is computed as the subtraction of the amount of intralayer communication from that of interlayer communication. If the communication gain of any core is greater than or equal to 0, the core is assigned to a current layer. In Fig. 6(e), cores B, C, E, and F communicate with cores in a different layer and their communication gains are 0, -1, 0, and 0, respectively. Thus, cores B, E, and F are assigned to a current layer. Then, the second bipartitioning in each subgroup is again performed for the minimum cut under the area constraint. Fig. 6(f) shows the final result where hop count between four layers is 7, whereas the hop count of the first approach between four layers is 8 in Fig. 6(c). Therefore, the second one is useful to reduce hop count between layers, and thus requires less TSVs.

Even if the number of a given layer is not a power of two, the basic idea of Algorithm 1 can easily be extended. For example, let eight cores, the total area of which is 20, be assigned to five layers in Fig. 6(a). When the first bipartitioning is performed, we make each partition not include the same area, i.e., one gets 8 and the other gets 12. Then, the first group with area 8 is again bipartitioned for the minimum cut and same area. On the contrary, the second group with area 12 is also bipartitioned for the minimum cut but different areas, i.e., one gets 4 and the other gets 8. Finally, the last subgroup with area 8 is again bipartitioned for the minimum cut and same area. Finally, all five subgroups get area 4. Similarly, this strategy can be applied to any number of a layer.

### B. CMP-Aware 3D NoC Topology Decision

Since a 3D network topology decision problem is NP hard [20], we present efficient heuristics in this section. Furthermore, since the integrated problem makes it difficult to reach guaranteed quality bounds on the solution, we divide the 3D network topology decision problem into two distinct

**Algorithm 2** Topology Decision Within Layer

---

```

1: for  $i = \text{max\_router}$  to  $(\text{the number of core}/\text{max\_int})$  do
2:   Find  $i$ -way min-cut partitions under  $\text{max\_int}$  constraints;
3:   Assign partitioned group  $p_n$  to router  $r_n$  ( $0 < n < i$ );
4:   Interconnect  $r_p$  to  $r_q$  by user's design objective
   ( $0 < p < i, 0 < q < i, p \neq q$ );
5:   Build router communication graph (RCG);
6:   Build prohibited turn set on RCG to avoid deadlocks;
7:   Find paths for flows across different routers in each layer
8:   Evaluate latency  $lc(r_p, r_q)$  and bandwidth  $bw(r_p, r_q)$ ;
9:   Go to line 2 if application constraints are not satisfied;
10: end for
11: Choose the best topology and design point;

```

---

subproblems, called router-to-core or router interconnection in the same layer and router-to-router interconnection between different layers, and then we solve the respective subproblems. Whereas a bandwidth requirement can easily be satisfied by finding alternative routing paths or adding more interconnection resources, satisfying latency constraints is difficult if cores communicating each other are too wide apart. Therefore, any master core sensitive to latency should be interconnected to the same router as its slave core. A TSV array covering a one-way vertical link is used for interconnection between different layers and any router is not interconnected to routers in a different layer if it is already interconnected to the router with one direction as shown in (4), which minimizes TSV density variation, and thus reduces TSV height variation resulting in TSV bonding failure.

1) *2D Router-to-Router/Core Interconnection*: Given a core graph, the number of allowable routers ( $\text{max\_router}$ ), and the number of allowable interconnections to a router ( $\text{max\_int}$ ), our 2D network topology synthesis technique interconnects possible cores to any routers. The objective of our 2D network topology decision is to minimize power consumption in each layer. Varying the number of routers in NoC designs has a great impact on power consumption and communication latency. NoC using few routers leads to longer core-to-router interconnections and hence, higher interconnection power consumption. On the contrary, when a number of routers are used, data flows have to traverse more routers, leading to high router power consumption and increasing area. Thus, we need to explore NoC designs with the different number of routers to obtain the best solution, starting from a design point where each core is interconnected to the minimum routers to one where cores are connected to the maximum allowable routers ( $\text{max\_router}$ ) in each layer.

For example, we assume that there are 20 cores within any layer, the maximum number of allowable routers ( $\text{max\_router}$ ) is 6 and the maximum allowable interconnection to one router ( $\text{max\_int}$ ) is 5. We explore a 2D network topology with 4 (equal to the number of core/ $\text{max\_int}$ ) to six routers.

The objective of Algorithm 2 is to establish efficient physical links between a router and a router/core in each layer. First,  $i$ -way minimum-cut partitioning is performed for cores in the same layer under the  $\text{max\_int}$  constraint (line 2), and then each group is assigned to a single router (line 3). Next, network links between the routers are inserted according to user's design objective (line 4). In our implementation, we use the minimum spanning tree (MST) or point-to-point (P2P) interconnection. MST requires the minimum network resources even if it achieves the worst hop count. On the contrary, P2P achieves the best hop count whereas it requires the maximum network resources. MST generation requires distance information between routers. However, since floorplanning is not yet performed, we use different metrics instead of the distance information. MST first interconnects two vertices close to each other. Similarly, since the routers  $r_p$  and  $r_q$  heavily communicate with each other and should be interconnected with high priority, we use  $1/\text{vol}(c_{p,q})$  as the distance information. Then, the breadth-first-search or depth-first algorithms are used for searching MST. MST requiring only  $i - 1$  links decreases the length of physical wires, but increases hop count and total travel distance of traffics, where  $i$  is the number of routers. On the contrary, P2P decreases hop count and total travel distance of traffics, but increases the length of physical wires. Next, a new router communication graph (RCG) is generated and then a prohibited turn set for RCG is built to avoid deadlocks (lines 5 and 6). Paths for flows across different routers in the same layer are allocated on the inserted links, based on the Dijkstra's shortest path algorithm (line 7). Application constraints, such as hop count, communication latency, and bandwidth, are evaluated (line 8). If they are not satisfied, a different network topology in each layer is again synthesized (line 9). Finally, the best network topology and design point are selected among  $\{\text{max\_router} - (\# \text{ of core}/\text{max\_int}) + 1\}$  topologies, including the different number of routers in each layer (line 11).

2) *Layer-to-Layer Interconnection*: After deciding a 2D network topology in all layers, any layer must be interconnected to adjacent layers with TSV arrays. In Section V-A, we minimized hop count between different layers, which made few TSVs used. However, total hop count may increase depending on the location of the few TSVs. In addition, inserting either both one-way and two-way links in the same layer or a TSV array with high metal density results in severe TSV height variation during CMP. Thus, the objective of the proposed layer-to-layer interconnection is to insert only one-way links between adjacent layers for uniform TSV distribution and the minimum hop count under performance constraints.

Fig. 7(a) is a core graph assigned to two layers, where the weight of all edges is 1. After deciding the network topology in each layer, let TSV arrays be inserted for the minimum hop count as shown in Fig. 7(b) and (c). In Fig. 7(b), both one-way and two-way links are used between layers. If an industrial open core protocol (OCP) [18] and an AMBA advanced extensible interface (AXI) protocol [1] that have widely been used as on-chip network interfaces have a 32-b data bus and a 32-b address, the number of TSVs required



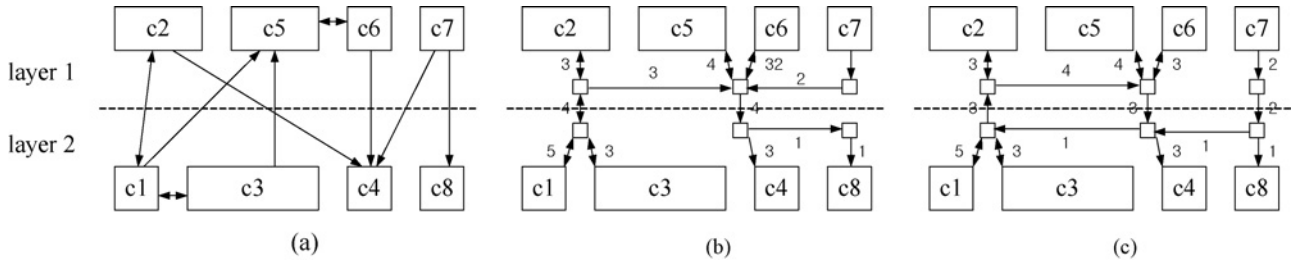


Fig. 7. CMP-aware router-to-router interconnection in adjacent layers. (a) Core graph assigned to layer. (b) Case 1: routing path allocation. (c) Case 2: routing path allocation.

for a one-way vertical link is 113 and 204, respectively. In addition, the number of TSVs required for a two-way vertical link is 226 and 408 in the OCP and the AMBA AXI protocol, respectively. If the pitch of the two-way link is equal to that of the one-way link, it has higher metal density than the one-way link. As a result, the two-way link is more sensitive to TSV height variation than the one-way link during CMP. In addition, a TSV array in the one-way link may have also the strong possibility of failing to contact landing pads since it is further polished than a TSV array in the two-way link, as shown in Fig. 3(a). Therefore, Fig. 7(c) is more desirable for low and uniform local TSV density if the hop count of case 2 is similar to that of case 1. In our technique, when a one-way vertical link for  $c_{p,q}$  is established, the opposite one-way link for  $c_{q,p}$  is removed in the list of TSV array insertion candidates, where  $r_p \in V^{lm}$  and  $r_q \in V^{ln}$ , ( $m \neq n$ ). As a result, all layers can be interconnected to adjacent layers with one-way links that have less TSV height variation.

### C. CMP-Aware Floorplanning

Before performing floorplanning in each layer, we first compute a TSV pitch for one-way links inserted for layer-to-layer interconnection, using the predictive CMP model. The pitch should result in low TSV height variation sufficiently endured by a used bonding technique. Then, the TSV array for one-way links is built and then simultaneously floorplanned with routers and cores in each layer. The goal of our floorplanning is to generate a layout that minimizes area, power consumption, and peak temperature. We modify an existing floorplanning technique [8] and invoke it with our unique cost function (5).

The power consumption on a given network architecture can be presented as the power required by point-to-point physical links between a core and a router or a router and a router. It is desirable to place cores, routers, and TSV arrays close to if they heavily communicate one another. This is because the power consumption of NoC is directly proportional to the volume of communication and the travel distance of traffics. Hence, we define the cost function as the product of communication volume  $vol(e_{i,j})$  and wirelength  $w_{i,j}$  in (5). In addition, it is necessary to floorplan cores, routers, and TSV arrays within allowable wirelength to guarantee timing closure at a target clock speed. If we assume that wafers are stacked in a face-to-back fashion and the bottom layer (layer 0) includes no TSV array, we start floorplanning from the top layer (layer  $n - 1$ ) with TSVs in decreasing order. After floorplanning any layer  $k$  (where  $0 < k \leq n - 1$ ), terminals with zero area are

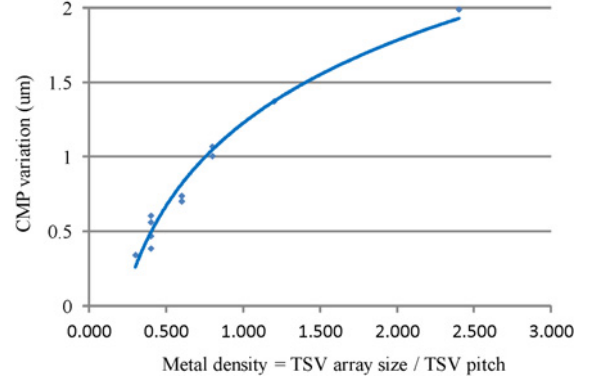


Fig. 8. Silicon-CMP variation based on IMEC wafer measurement [10].

inserted at the same XY location in the next floorplanned layer  $k - 1$  as TSV arrays in layer  $k$ . It makes cores in layer  $k - 1$  floorplanned nearby TSVs in layer  $k$  if the cores communicate with any cores in layer  $k$ . Consequently, total travel distance of traffics related to performance and power consumption is reduced. Since TSV arrays in each layer have no relation to one another, they are floorplanned wide apart such that a uniform local TSV density can easily be achieved.

## VI. EXPERIMENTAL RESULTS

### A. TSV Density and Predictive CMP Model

CMP is a complex process with a large number of input variables such as slurry flow rate, pressure, velocity, friction force, lubrication, pad, and wafer geometry and output variables such as polish rate, planarization rate, polish rate uniformity, and surface quality. While there is much research on modeling Cu-CMP variation [6], [15], there are few studies on modeling silicon-CMP variation. Fig. 8 shows CMP variation measured from the latest 3D ICs of IMEC after silicon-CMP, depending on metal density when a TSV diameter is  $5 \mu\text{m}$  [10]. With these industry measurement data, we model TSV height variation as follows:

$$hv = 0.8017 \ln \left( \frac{s}{p} \right) + 1.226 \quad (6)$$

where  $hv$  is TSV height variation,  $s$  is the size of a TSV array, and  $p$  is a TSV pitch in the TSV array. This equation shows that a small TSV array and a wide TSV pitch are desirable for low TSV height variation. Based on this model, a TSV



TABLE I  
TSV HEIGHT VARIATION COMPARISON

Network Protocol (Address Width and Data Width)	The Number of Wire		TSV Height Variation ( $\mu\text{m}$ )		Improvement (%)
	One-Way Link	Two-Way Link	[25]	CAS	
APB (16 and 16)	65	130	1.372	1.142	16.81
APB (32 and 32)	115	230	1.551	1.226	20.96
AHB (32 and 32)	137	274	1.651	1.372	16.91
AHB (64 and 64)	233	466	1.858	1.603	13.74
AXI (32 and 32)	204	408	1.821	1.551	14.81
AXI (64 and 64)	332	664	1.992	1.741	12.62
ACE (32 and 32)	306	612	1.961	1.697	13.43
ACE (64 and 64)	434	868	2.107	1.821	13.57
OCP (32 and 32)	113	266	1.651	1.302	21.13
OCP (64 and 64)	209	418	1.821	1.551	14.81
Average			1.779	1.501	15.88

pitch is computed for the size of a given TSV array. During a CMP process, the TSV pitch should guarantee low TSV height variation that a bonding technique endures. Then, TSV arrays with the pitch are built and then simultaneously floorplanned with cores and routers. For example, if the size of a TSV array including a one-way link (113 wires) for OCP is  $11 \times 11$ , the TSV pitch must be at least  $14.58 \mu\text{m}$  for TSV height variation less than  $1 \mu\text{m}$ . On the contrary, if the size of a TSV array including a two-way link (226 wires) for OCP is  $16 \times 16$ , the TSV pitch must be at least  $21.21 \mu\text{m}$  for TSV height variation less than  $1 \mu\text{m}$ . Thus, the widths of  $11 \times 11$  and  $16 \times 16$  TSV arrays are  $160 \mu\text{m}$  and  $339 \mu\text{m}$  and their areas are  $0.0256 \text{ mm}^2$  and  $0.1151 \text{ mm}^2$ , respectively. Whereas the TSV array for two-way links includes two times more TSVs than the TSV array for the one-way link, it occupies 4.5 times larger area than the TSV array for one-way links to get TSV height variation less than  $1 \mu\text{m}$ . Consequently, two TSV arrays for a one-way vertical link show lower CMP variation or smaller design area than a single TSV array for a two-way vertical link if the performance and energy constraints of a synthesized network are satisfied.

### B. CMP-Aware Application-Specific 3D NoC

We implement the proposed CMP-aware application-specific 3D NoC, called CAS with four to eight layers in C++. We repeat CAS for ten times on GSRC Benchmarks with 100, 200, and 300 modules [7] and compute their average to obtain reliable statistics. Wafers are stacked in a face-to-back fashion and we set the diameter and pitch of TSVs to  $5 \mu\text{m}$  and  $10 \mu\text{m}$ , respectively. Both [25] and CAS employ MST and P2P for 2D network topology generation. Since the goal of MST is extremely opposite to that of P2P, the performance and design cost improvement of CAS with other 2D network topologies will be within the gap of improvement of CAS with MST and P2P. Note that [17] and [21] are not suitable for comparison for 3D NoC with large routers and TSV arrays if there is less dead space after floorplanning.

Table I shows TSV height variation when various on-chip network protocols are used with different address and data widths. When a 3D network topology is decided, CAS inserts only one-way links between adjacent layers whereas [25] inserts both one-way and two-way links. Therefore, the local

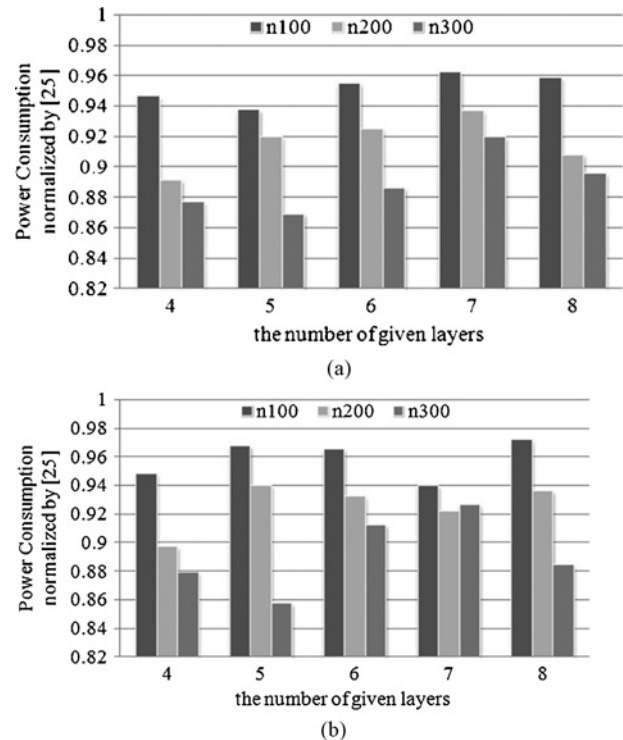


Fig. 9. Power consumption normalized by [25]. (a) MST. (b) P2P.

TSV density of CAS is more uniform and lower than that of [25], and thus CAS achieves 15.88% lower TSV height variation than [25], on average, after silicon-CMP. On the contrary, inserting only one-way links results in increasing hop count since such limited network resources may not provide the shortest path for any traffic across different wafers. However, the proposed 3D NoC design flow recovers the penalty of the hop count and even improves total hop count since a topology decision is first optimized.

Table II shows total hop count (THC) and average hop count (AHC) resulting from [25] and CAS. AHC is defined as THC divided by the total number of source/destination pairs. The proposed CAS achieves, on average, 19.7% and 11.0% lower hop count than [25] when MSP and P2P are applied for deciding 2D network topologies, respectively. In the case that

TABLE II  
TOTAL AND AVERAGE HOP COUNT COMPARISON

The Number of Layers			4		5		6		7		8		Avg.		Imp. (%)
			THC	AHC	THC	AHC	THC	AHC	THC	AHC	THC	AHC	THC	AHC	
MST	n100	[25]	1410	1.59	1470	1.66	1625	1.84	1671	1.89	1927	2.18	1621	1.83	16.5
		CAS	1201	1.36	1254	1.42	1299	1.47	1361	1.54	1650	1.86	1353	1.53	
	n200	[25]	3341	2.11	3366	2.12	3459	2.18	3737	2.36	3927	2.48	3566	2.25	20.0
		CAS	2654	1.67	2931	1.85	2698	1.70	2934	1.85	3043	1.92	2852	1.80	
	n300	[25]	5211	2.75	5158	2.72	5178	2.74	5257	2.78	5230	2.76	5207	2.75	22.6
		CAS	4065	2.15	3912	2.07	4077	2.15	3984	2.10	4118	2.18	4031	2.13	
Imp. (%)			20.5		19.0		21.3		22.4		20.5		20.74		-
P2P	n100	[25]	1193	1.35	1336	1.51	1488	1.68	1629	1.84	1799	2.03	1489	1.68	13.1
		CAS	1077	1.22	1194	1.35	1207	1.36	1416	1.60	1575	1.78	1294	1.46	
	n200	[25]	2051	1.29	2487	1.57	2744	1.73	3136	1.98	3285	2.07	2741	1.73	8.7
		CAS	2041	1.29	2278	1.44	2441	1.54	2788	1.76	2968	1.87	2503	1.58	
	n300	[25]	2638	1.39	3279	1.73	3433	1.81	4163	2.20	4371	2.31	3577	1.89	11.1
		CAS	2626	1.39	2943	1.55	3405	1.80	3234	1.71	3695	1.95	3181	1.68	
Imp. (%)			2.3		9.7		8.0		16.7		12.9		9.92		-

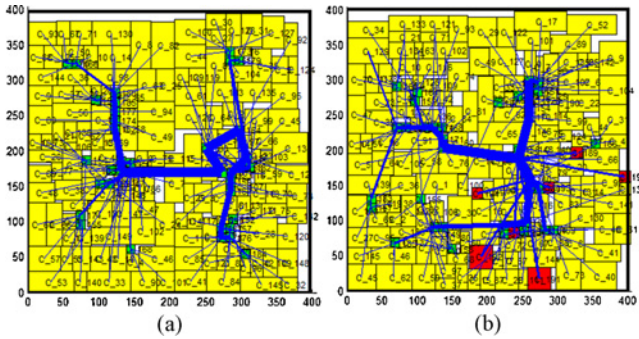


Fig. 10. Typical application-specific 3D NoC with two layers [25]. (a) Layer 1. (b) Layer 2.

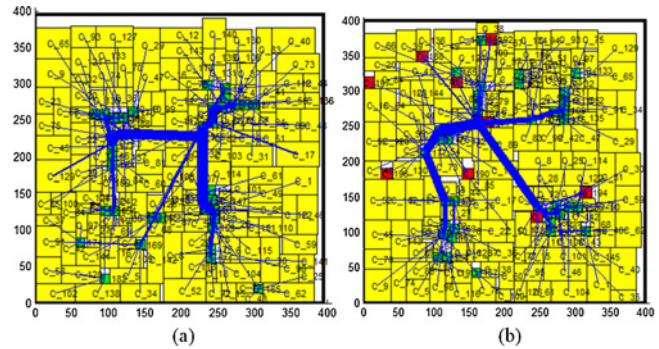


Fig. 11. CMP-aware application-specific 3D NoC with two layers. (a) Layer 1. (b) Layer 2.

a network is synthesized with limited resources such as MST, CAS achieves less hop count than [25]. In addition, CAS tends to further improve hop count in complex NoCs with a number of modules and layers.

In Table III, the total wirelength (TWL) and the longest wirelength (LWL) of CAS are compared with those of [25]. CAS achieves, on average, 0.24% longer total wirelength than [25] in MST, but 5.62% shorter total wirelength than [25] in P2P. On the contrary, the longest wirelength of CAS is, on average, 3.65% and 1.86% longer than [25] in MST and P2P, respectively. However, the longest wire is endurable and under control from (5). In addition, since our CAS makes the longest wire passed by few packets, the longest wire does not increase power consumption.

Fig. 9 shows power consumption normalized by [25]. The power consumption of CAS is, on average, 8.1% and 7.8% lower than that of [25] in MST and P2P, respectively. This is because CAS makes the travel distance of traffics shorter and hop count less than [25], even if the physical wirelength of CAS is similar to or slightly longer than that of [25]. In the figure, CAS tends to further improve power consumption in NoC with a lot of modules and few network resources such as MSP.

The total area of CAS is slightly smaller than [25] since CAS has a smaller total TSV array area than [25]. The runtime

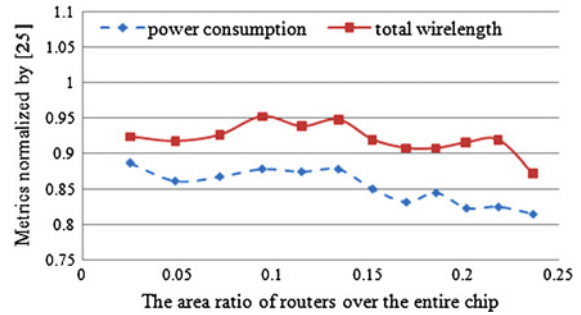


Fig. 12. Improvement according to the area of routers.

of CAS ranges from 48 to 99 s in n300, which is about three times faster than [25].

Figs. 10 and 11 show the layouts of n300 with two layers generated by [25] and CAS, respectively, when MST is used as a 2D network topology. In the figures, the blue lines are communication relations and their thickness represents communication volume. The yellow rectangles, red rectangles, and green rectangles are cores, TSV arrays, and routers, respectively. In particular, the small red rectangles represent TSV arrays used for one-way vertical links and the large red rectangles represent TSV arrays used for two-way vertical links. Whereas [25] generates layer 2 with both one-way and

TABLE III  
TOTAL AND LONGEST WIRELENGTH COMPARISON (UM)

The Number of Layers			4		5		6		7		8		Avg.		Imp (%)
			TWL	LWL	TWL	LWL	TWL	LWL	TWL	LWL	TWL	LWL	TWL	LWL	
M S T	n100	[25]	9631	165	8373	151	7532	124	7096	122	6487	108	7824	134	-0.3
		CAS	9577	166	8493	147	7412	129	7121	123	6584	107	7837	134	
	n200	[25]	22 605	183	19 093	168	16 565	149	15 076	140	13 830	128	17 434	154	0.0
		CAS	23 238	191	19 125	176	16 567	153	15 011	150	13 262	132	17 441	160	
	n300	[25]	46 530	238	39 526	205	34 353	187	30 504	172	27 194	164	35 621	193	-0.6
		CAS	47 019	245	39 966	194	34 080	202	30 673	185	27 289	179	35 805	211	
Imp. (%)			-1.4	-2.9	-0.9	1.5	0.7	-5.1	-0.2	-5.7	0.6	-4.6	-0.24	-3.65	-
P 2 P	n100	[25]	47 202	302	38 605	264	32 909	236	29 553	209	26 411	199	34 936	242	2.1
		CAS	46 482	314	38 270	271	32 339	240	27 957	211	26 231	204	34 256	248	
	n200	[25]	95 443	329	77 664	299	67 808	261	61 253	240	55 204	224	71 474	271	4.7
		CAS	89 612	334	75 086	297	65 040	271	58 477	241	52 276	219	68 098	272	
	n300	[25]	144 659	399	129 920	362	115 515	335	102 614	304	94 463	278	117 434	336	7.0
		CAS	132 131	404	119 611	387	108 630	338	99 395	311	86 102	277	109 174	349	
Imp. (%)			7.5	-2.3	5.4	-3.2	4.8	-2.1	3.9	-1.4	6.5	0.3	5.62	-1.86	-

two-way links as shown in Fig. 10(b), CAS generates layer 2 with just one-way links as shown in Fig. 11(b). Consequently, the uniform and low TSV density by CAS leads to less TSV height variation after silicon-CMP, and thus makes TSVs contact landing pads more easily than [25].

In Fig. 12, CAS proves more merits on 3D NoC with complex routers, where CAS further improves power consumption and total wirelength as the area ratio of routers over a whole chip increases. Since the previous 3D NoC designs first floorplan only cores without routers and TSV arrays before synthesizing network topology, neither is the dead space sufficient for complex routers and TSV arrays inserted nor wirelength and power consumption are well optimized even if floorplanning is again performed after synthesizing a network.

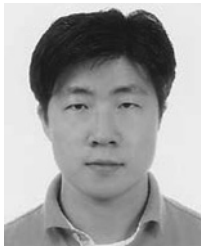
## VII. CONCLUSION

In this paper, we proposed the CMP-aware application-specific 3D NoC design. Our vertical integration managing architecture, physical design, and manufacturing issues together enabled a reliable and robust 3D NoC. In particular, our CMP-aware 3D NoC approach reduced TSV height variation after the CMP process, and thus prevented severe bonding failures and timing variation. Meanwhile, it also improved hop count, wirelength, power consumption, and area, compared to the previous state-of-the-art 3D NoCs.

## REFERENCES

- [1] ARM. (2011, Jun.). *AMBA AXI, AHB, and APB Protocol Specifications* [Online]. Available: <http://www.arm.com>
- [2] K. Athikulwongse, A. Chakraborty, J.-S. Yang, D. Z. Pan, and S. K. Lim, "Stress-driven 3D-IC placement with TSV keep-out zone and regularity study," in *Proc. Int. Conf. Comput.-Aided Des.*, Nov. 2010, pp. 669–674.
- [3] L. Benini and G. De Micheli, "Network on chips: A new SoC paradigm," *Computer*, vol. 35, no. 1, pp. 70–78, Jan. 2002.
- [4] J. Cong, J. Wei, and Y. Zhang, "A thermal-driven floorplanning algorithm," in *Proc. Int. Conf. Comput.-Aided Des.*, Nov. 2004, pp. 306–313.
- [5] W. J. Dally and B. Towles, "Route packets, not wires: on-chip interconnection networks," in *Proc. Des. Autom. Conf.*, Jun. 2001, pp. 684–689.
- [6] T. Gan, "Modeling of chemical mechanical polishing for shallow trench isolation," Ph.D. dissertation, Dept. Electr. Eng. Comput. Sci., Massachusetts Instit. Technol., Cambridge, MA, USA, 2000.
- [7] H. Wu. (2000, Dec.). *GSRC Floorplan Benchmarks* [Online]. Available: <http://vlsicad.eecs.umich.edu/BK/GSRCbench>
- [8] O. He, S. Dong, J. Bian, S. Goto, and C.-K. Cheng, "A novel fixed-outline floorplanner with zero dead space for hierarchical design," in *Proc. Int. Conf. Comput.-Aided Des.*, Nov. 2008, pp. 16–23.
- [9] W.-L. Hung, G. M. Link, Y. Xie, N. Vijaykrishnan, and M. J. Irwin, "Interconnect and thermal-aware floorplanning for 3D micro-processors," in *Proc. Int. Symp. Quality Electron. Des.*, Mar. 2006, pp. 98–104.
- [10] IMEC [Online]. Available: <http://www.imec.be>.
- [11] A. B. Kahng and K. Samadi, "CMP fill synthesis: A survey of recent studies," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 27, no. 1, pp. 3–19, Jan. 2008.
- [12] D. H. Kim, Y.-K. Wu, R. O. Topaloglu, and S. K. Lim, "Enabling 3D integration through optimal topograph," in *Proc. Int. Workshop Des. Manuf. Yield Workshop*, Jun. 2010, pp. 70–73.
- [13] W. Jang, O. He, J.-S. Yang, and D. P. Pan, "Chemical-mechanical polishing aware application-specific 3D NoC design," in *Proc. Int. Conf. Comput.-Aided Des.*, Nov. 2011, pp. 207–212.
- [14] S. K. Lim. (2010, Jun.). *TSV-Aware 3D Physical Design Toll Needs for Faster Minstream Acceptance of 3D ICs* [Online]. Available: <http://www.dac.com>
- [15] J. Luo and D. A. Dornfeld, "Material removal mechanism in chemical mechanical polishing: Theory and modeling," *IEEE Trans. Semicond. Manuf.*, vol. 14, no. 2, pp. 112–133, May 2001.
- [16] R. Marculescu, U. Y. Ogras, L.-S. Peh, N. E. Jerger, and Y. Hoskote, "Outstanding research problems in NoC design: system, microarchitecture, and circuit perspectives," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 28, no. 1, pp. 3–21, Jan. 2009.
- [17] S. Murali, C. Seiculescu, L. Benini, and G. De Micheli, "Synthesis of networks on chips for 3D systems on chips," in *Proc. Asia South Pacific Des. Autom. Conf.*, Jan. 2009, pp. 242–247.
- [18] OCP-IP. *Open Core Protocol Specification* [Online]. Available: <http://www.ocpip.org>
- [19] M. Pathak, Y.-J. Lee, T. Moon, and S. K. Lim, "Through-silicon-via management during 3D physical design: When to add and how many?" in *Proc. Int. Conf. Comput.-Aided Des.*, Nov. 2010, pp. 387–394.
- [20] A. Pinto, L. P. Carloni, and A. L. Sangiovanni-Vincentelli, "Efficient synthesis of networks on chip," in *Proc. Int. Conf. Comput. Des.*, Oct. 2003, p. 146.
- [21] C. Seiculescu, S. Murali, L. Benini, and G. De Micheli, "SunFloor 3D: A tool for networks on chip topology synthesis for 3D systems on chips," in *Proc. Des. Autom. Test Eur.*, Mar. 2009, pp. 9–14.
- [22] S. Sivaram, H. Bath, R. Legegett, A. Maury, K. Monning, and R. Tolles, "Planarizing interlevel dielectrics by chemical mechanical polishing," *Solid State Technol.*, vol. 35, no. 5, pp. 87–91, 1992.
- [23] B. Swinnen, W. Ruythooren, P. De Moor, L. Bogaerts, L. Carbonell, K. De Munck, B. Eyckens, S. Stoukatch, D. Sabuncuoglu Tezcan, Z. Tökei, J. Vaes, J. Van Aelst, and E. Beyne, "3D integration by Cu-Cu thermo-compression bonding of extremely thinned bulk-Si die containing 10  $\mu\text{m}$  pitch through-Si vias," in *Proc. IEDM*, 2006, pp. 1–4.

- [24] J. Van Olmen, A. Mercha, G. Katti, C. Huyghebaert, J. Van Aelst, E. Seppala, Z. Chao, S. Armini, J. Vaes, R. C. Teixeira, M. Van Cauwenbergh, P. Verdonck, K. Verhemeldonck, A. Jourdain, W. Ruythooren, M. de Potter de ten Broeck, A. Opdebeeck, T. Chiarella, B. Parvais, I. Debusschere, T. Y. Hoffmann, B. De Wachter, W. Dehaene, M. Stucchi, M. Rakowski, P. Soussan, R. Cartuyvels, E. Beyne, S. Biesemans, and B. Swinnen, "3D stacked IC demonstration using a through silicon via first approach," in *Proc. IEDM*, Dec. 2008, pp. 1–4.
- [25] S. Yan and B. Lin, "Design of application-specific 3D networks-on-chip architectures," in *Proc. Int. Conf. Comput. Des.*, Oct. 2008, pp. 142–149.
- [26] W. Jang and D. Z. Pan, "An SDRAM-aware router for networks-on-chip," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 29, no. 10, pp. 1572–1585, Oct. 2010.
- [27] W. Jang and D. Z. Pan, "A voltage-frequency island aware energy optimization framework for networks-on-chip," *IEEE J. Emerging Sel. Topics Circuits Syst.*, vol. 1, no. 3, pp. 420–432, Sep. 2011.
- [28] W. Jang and D. Z. Pan, "Application-aware NoC design for efficient SDRAM access," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 30, no. 10, pp. 1521–1533, Oct. 2011.
- [29] W. Jang and D. Z. Pan, "A3MAP: Architecture-aware analytic mapping for networks-on-chip," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 17, no. 3, pp. 1703–1726, 2012.
- [30] J.-S. Yang, K. Athikulwongse, Y.-J. Lee, S. K. Lim, and D. Z. Pan, "TSV stress aware timing analysis with applications to 3D-IC layout optimization," in *Proc. Des. Autom. Conf.*, Jun. 2010, pp. 803–806.



**Wooyoung Jang** (S'08–M'12) received the B.E. degree in radio science and technology from Kyung Hee University, Suwon, Korea, in 1998, the M.S. degree in electrical and computer engineering from Yonsei University, Seoul, Korea, in 2000, and the Ph.D. degree in electrical and computer engineering from the University of Texas, Austin, in 2011.

From 2000 to 2012, he was a Senior Engineer with the System Large Scale Integration Division, Samsung Electronics, Seoul. He is currently an Assistant Professor with the Department of Electronics and Electrical Engineering, Dankook University, Gyeonggi, Korea. He has published over 24 technical papers in refereed journals and conference proceedings. He holds four U.S. patents. His current research interests include interconnection networks, computer architectures, cloud computing, high-performance computing, nanometer physical design, and image processing.

Dr. Jang was a recipient of the SK Telecom Scholarship for 1994 to 1997, the Samsung Outstanding Achievement Award in 2005, and the Samsung Scholarship for 2006 to 2011.



**David Z. Pan** (S'97–M'00–SM'06) received the Ph.D. degree (Honors.) in computer science from the University of California, Los Angeles (UCLA), in 2000.

He is currently an Associate Professor with the Department of Electrical and Computer Engineering, University of Texas, Austin. From 2000 to 2003, he was a Research Staff Member with the IBM T. J. Watson Research Center, Austin. He has published over 170 technical papers in refereed journals and conference proceedings. He holds eight U.S.

patents.

Dr. Pan has served in the Editorial Boards of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION SYSTEMS, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I and II, *Science China Information Sciences*, *Journal of Computer Science and Technology*, and the IEEE CAS Society Newsletter. He has also served as the IEEE CAS/CEDA CANDE Committee Chair and the ACM/SIGDA Technical Committee Chair on Physical Design. He is the General Chair of ISPD 2008 and the Steering Committee Chair of ISPD 2009. He has served on technical program committees of many premier conferences, including the Program Chair of ISPD and the Subcommittee Chair of ASPDAC, DAC, ICCAD, ISLPEd, among many others. He was a recipient of a number of awards for his research contributions and professional services, including the ACM/SIGDA Outstanding New Faculty Award in 2005, the NSF CAREER Award in 2007, the UCLA Engineering Distinguished Young Alumnus Award in 2009, the SRC Inventor Recognition Award three times, the IBM Faculty Award four times, and nine Best Paper Awards (ASPAC 2012, ISPD 2011, IBM Research 2010 Pat Goldberg Memorial Best Paper Award in CS/EE/Math, ASPAC 2010, DATE 2009, ICICDT 2009, and SRC Techcon in 1998, 2007, and 2012). He was a recipient of the Dimitris Chorafas Foundation Research Award in 2000, the eASIC Placement Contest Grand Prize in 2009, the ICCAD'12 CAD Contest Award (Second Place), the ISPD Routing Contest Award in 2007, and the ACM Recognition of Service Award in 2007 and 2008. He was an IEEE CAS Society Distinguished Lecturer from 2008 to 2009.