Research Papers being presented:

Thanking the committee members:
1. Dr. Sudeep Pasricha (Academic Advisor)
2. Dr. Anura Jayasumana
3. Dr. H. J. Siegel
4. Dr. Michelle Strout
PM-COSYN: PE and Memory Co-Synthesis for MPSoCs

DATE 2010

Yi-Jung Chen, Chia-Lin Yang and Po-Han Wang

Department of Computer Science and Information Engineering
National Taiwan University

Presented by: Nishit Kapadia
Outline

- Background on Networks-on-chip (NoCs)
- Motivation
- Related Work
- Problem Formulation
- PE & Memory Co-synthesis Algorithm
- Victim PE Selection
- Data Block Assignment
- Experiments
- Paper Critique
Background on NoCs

- Networks-on-chip (NoCs) route ‘packets’

- **Why NoCs?** - With tens to hundreds of cores, Bus architectures are not scalable

- **Common Topologies:**
  1. **Mesh Topology:** All cores have to be of same size (homogeneous cores) and a simplified routing scheme with a regular structure
  2. **Custom (Application-Specific) Topology:** Heterogeneous cores used for better latency and area utilization, but design process more complex
Application-specific/ Mesh based Topologies for NoCs
Traditionally, memory system is synthesized with predetermined PE allocation and task assignment.

For Multi-Processor System-on-Chips (MPSoCs):

- **Main objective** is to exploit task-level parallelism for maximum throughput.
- But, **concurrent memory accesses** from multiple PEs may cause memory bottleneck.

Thus, to maximize system performance, simultaneously consider the PE and on-chip memory architecture design.
Motivation

- **MPSoc Design Problem:**
  - How to partition die resources between PEs and on-chip memory to achieve optimal performance?

- **Principle Design Trade-off:**
  - fully utilize the available task parallelism with multiple PEs
  - alleviate off-chip memory bandwidth bottleneck with on-chip memory modules
Related Work

- COSMECA: Co-synthesis of memory and bus-based communication architectures [S. Pasricha et al. DATE 2006]
  Branch and Bound Clustering algorithm to minimize # of buses and memory area

- Co-synthesis of memory and NoC architectures [I. Issenin et al. IESS 2007]
  Optimizes memory & communication energy

- Hardware/Software co-design of NoCs [Y.-J. Chen et al. JSA 2009]
  SA used for simultaneous selection of PEs/ task allocation & scheduling

Predefined memory subsystems optimized previously, simultaneous allocation of PEs and on-chip memories proposed for the first time in PM-COSYN
Problem Formulation

Given:
- A task-graph (DAG) $G = <V, E>$; $v_i \in V$
- $e_i \in E$ represents data block transfer
- A data block library $D = \{d_1, \ldots, d_k\}$ for the task set
- A tile-based Network-on-Chip (NoC)
- Architecture Graph $N = <T, L>$; $T = \{t_1, \ldots, t_m \times n\}$ is the set of tiles

Main Constraints:
- Each tile can contain a PE or a memory module
- Static XY routing scheme is assumed in the NoC
Problem Formulation

- The goal of PM-COSYN is to simultaneously allocate PEs and on-chip memory such that:
  - system performance is maximized
  - area constraint is met (fixed # of tiles available)

\[
\begin{align*}
\text{Given } G = \langle V, E \rangle, \quad D = \{d_1, \ldots, d_k\} \quad \text{and} \quad N = \langle T, L \rangle \\
\text{Find } P, M \text{ and } D', \text{ and the function } \phi \text{ and } \omega, \\
\text{such that } \text{Time}_G \text{ is minimized} \\
\text{Subject to } |P| + |M| = |T|
\end{align*}
\]

- P: set of allocated PEs
- M: set of on-chip memories
- D’: set of data blocks assigned to the on-chip memory modules, where D’ \subseteq D
- Time_G: execution time of the target task set
- \( \phi : V \rightarrow P \) is the task to PE mapping
- \( \omega : D' \rightarrow M \) is the data block to memory module mapping
Initial solution has the number of PEs equal to the degree of task parallelism.

In the greedy-based iterative refinement process, PEs are gradually replaced with on-chip memory modules.
Victim PE Selection (exhaustive)

**Victim PE Selection:**

**Input:** \( G = \langle V, E \rangle, D = \{d_1, ..., d_k\}, \)
\( N = \langle T, L \rangle \) and selected PE \( P = \{P_1, ..., P_l\} \)

**Output:** Victim PE \( P_{vic} \) and new assignment of tasks which are originally assigned to \( P_{vic} \)

1. **for each** PE \( P_i \in P \)
2. **for each** task \( v_j \) assigned to \( P_i \)
3. **for each** PE \( P_k \in P - P_i \)
4. Insert \( v_j \) to \( P_k \)
5. Evaluate execution time \( T(v_j, P_k) \) of system with \( v_j \) assigned to \( P_k \)
6. Find the smallest \( T(v_j, P_k) \) for all \( P_k \in P - P_i \)
7. if only one PE \( P_k \) has the smallest exe. time \( P_i(v_j) = P_k \)
8. else
9. Choose PE \( P_k \) with the most shared data with \( v_j \) \( P_i(v_j) = P_k \)
10. Evaluate execution time \( T(Vic(P_i)) \) of the system taking \( P_i \) as the victim PE
11. Find the smallest \( T(Vic(P_i)) \) among all \( P_i \in P \)
12. if only one \( P_i \) has the smallest \( T(Vic(P_i)) \)
13. \( P_{vic} = P_i \)
14. else
15. Randomly choose PE \( P_i \) with the smallest \( T(Vic(P_i)) \) \( P_{vic} = P_i \)
16. Assign all \( v_j \) on \( P_{vic} \) to \( P_k(v_j) \)
17. \( P = P - P_{vic} \)
Data Block Assignment:

Input: $G = \langle V, E \rangle, D = \{d_1, ..., d_k\}$ and $N = \langle T, L \rangle$

Output: Set of data blocks assigned to the new on-chip memory

1. $size\_left = Memory\_size$
2. Identify critical path of the task set in current configuration
3. while $size\_left > 0$
   
   and $\exists d_i \in D$ that is assigned to off-chip memory
   
   and $size(d_i) \leq size\_left$
4. Identify $d_{critical}$
5. Assign $d_{critical}$ to the new on-chip memory
6. Update the critical path
7. $size\_left = size\_left - size(d_{critical})$

- **data block $d_{critical}$**: resides in the off-chip memory and contributes the most data accesses ($num\ access \times size(d_{critical})$) to the critical path
Experiments: Task-Set, NoC Platform and # of mem. modules

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>NoC dimension</td>
<td>$3 \times 3$</td>
</tr>
<tr>
<td>Tile size</td>
<td>$1\text{mm}^2$</td>
</tr>
<tr>
<td>Router latency</td>
<td>5-cycle</td>
</tr>
<tr>
<td>Link width</td>
<td>32-bit</td>
</tr>
<tr>
<td>Off-chip memory access latency</td>
<td>64-cycle</td>
</tr>
<tr>
<td>On-chip memory module size</td>
<td>32KB</td>
</tr>
<tr>
<td>PE buffer size</td>
<td>16KB</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Task Set ID</th>
<th>Parallelism Degree</th>
<th>trans_bit/task_cycle</th>
<th>Memory Footprint (in bits)</th>
<th>Initial PE/MEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synthetic Task Sets</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t0</td>
<td>10</td>
<td>10.468301</td>
<td>276800</td>
<td>9/0</td>
</tr>
<tr>
<td>t1</td>
<td>10</td>
<td>1.20632</td>
<td>279100</td>
<td>9/0</td>
</tr>
<tr>
<td>t2</td>
<td>10</td>
<td>0.094262</td>
<td>276800</td>
<td>9/0</td>
</tr>
<tr>
<td>t3</td>
<td>10</td>
<td>0.051115</td>
<td>276800</td>
<td>9/0</td>
</tr>
<tr>
<td>Real-World Applications</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Consumer+Telecom</td>
<td>13</td>
<td>0.95139</td>
<td>9018000</td>
<td>9/0</td>
</tr>
<tr>
<td>Mpeg2_Enc+Mpeg2_Dec</td>
<td>12</td>
<td>0.03444</td>
<td>5455900</td>
<td>9/0</td>
</tr>
</tbody>
</table>
Experiments: Ex. times across # of mem. modules for synthetic BMs

<table>
<thead>
<tr>
<th>Task Set ID</th>
<th>Parallelism Degree</th>
<th>trans_bit/task_cycle</th>
<th>Memory Footprint (in bits)</th>
<th>Initial PE/MEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synthetic Task Sets</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t0</td>
<td>10</td>
<td>10.468301</td>
<td>276800</td>
<td>9/0</td>
</tr>
<tr>
<td>t1</td>
<td>10</td>
<td>1.20632</td>
<td>279100</td>
<td>9/0</td>
</tr>
<tr>
<td>t2</td>
<td>10</td>
<td>0.094262</td>
<td>276800</td>
<td>9/0</td>
</tr>
<tr>
<td>t3</td>
<td>10</td>
<td>0.051115</td>
<td>276800</td>
<td>9/0</td>
</tr>
<tr>
<td>Real-World Applications</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Consumer+Telecom</td>
<td>13</td>
<td>0.95139</td>
<td>9018000</td>
<td>9/0</td>
</tr>
<tr>
<td>Mpeg2_Enc+Mpeg2_Dec</td>
<td>12</td>
<td>0.03444</td>
<td>5455900</td>
<td>9/0</td>
</tr>
</tbody>
</table>

- **Task set t0**
  - Normalized Execution Time vs Number of On-Chip Memory Modules

- **Task set t3**
  - Normalized Execution Time vs Number of On-Chip Memory Modules
Experiments: Ex. times across # of mem. modules for real-world BMs

<table>
<thead>
<tr>
<th>Task Set ID</th>
<th>Parallelism Degree</th>
<th>trans_bit/task_cycle</th>
<th>Memory Footprint (in bits)</th>
<th>Initial PE/MEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synthetic Task Sets</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t0</td>
<td>10</td>
<td>10.468801</td>
<td>276800</td>
<td>9/0</td>
</tr>
<tr>
<td>t1</td>
<td>10</td>
<td>1.20632</td>
<td>279100</td>
<td>9/0</td>
</tr>
<tr>
<td>t2</td>
<td>10</td>
<td>0.094262</td>
<td>276800</td>
<td>9/0</td>
</tr>
<tr>
<td>t3</td>
<td>10</td>
<td>0.051115</td>
<td>276800</td>
<td>9/0</td>
</tr>
<tr>
<td>Real-World Applications</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Consumer+Telecom</td>
<td>13</td>
<td>0.95139</td>
<td>9018000</td>
<td>9/0</td>
</tr>
<tr>
<td>Mpeg2_Enc+Mpeg2_Dec</td>
<td>12</td>
<td>0.03444</td>
<td>5455900</td>
<td>9/0</td>
</tr>
</tbody>
</table>

![Graph](image1.png)

(a) Consumer+Telecom

![Graph](image2.png)

(b) Mpeg2_Enc+Mpeg2_Dec

3.2
Experiments: PM-COSYN Vs. SA

PM-COSYN uses at most 0.40% of SA CPU time
Solution quality is slightly better than SA

Example: for task set \( t_2 \), PMCOSYN uses only 0.07% of PM-SA CPU time, and the solution quality is 1.62% better than PM-SA.
Strengths:
- The right design trade-off is set-up given:
  - the worsening of memory wall
  - enhanced opportunities to exploit application task-parallelism

Key Insight:
Until the critical path is free of off-chip memory accesses, PE replacement continues.

Possible Weaknesses:
- A mechanism to trade-off # of off-chip accesses against parallel thread execution is lacking.
- Victim PE Selection and Data Block Assignment are disjoint: failing to capture the Processor/Memory interplay while determining the next Victim PE.
Chemical-Mechanical Polishing Aware Application-Specific 3D NoC Design

 ICCAD 2011

Wooyoung Jang, Ou He*, Jae-Seok Yang†, David Pan‡

SoC Platform Development Team, Samsung Electronics, Yongin-City, South Korea
*IBM System & Technology Group, Beijing, China
†CAE Team, Samsung Electronics, Hwaseung-City, South Korea
‡Department of Electrical and Computer Engineering, the University of Texas at Austin, Austin, USA

Presented by: Nishit Kapadia
Outline

- Background
  - Chemical-Mechanical Polishing (CMP) Process
  - TSV height variation
- Motivation
- Related Work
- Problem formulation
- Application-specific 3D NoC design flow
  - Core-to-layer assignment step
  - 3D topology decision step
  - 3D Floorplanning step
- Experiments
- Paper Critique
Chemical-Mechanical Polishing (CMP) Process

Process Steps:
1. Etched TSVs are filled with copper
2. Chemical reaction creates weak atomic-bonds
3. Mechanical surface abrasion for removal (fig. left)
4. Polished silicon surface plasma etched for TSVs to protrude (fig. right)
a) On another die, TSV landing pads fabricated on top metal layer (Cu-CMP)
b) Die with landing pads are bonded to the die with TSVs

[D. H. Kim et al. 2010, [9]]
Silicon-CMP involves simultaneous polishing of silicon and Cu.

Removal rates (polishing times) differ with:
1. materials
2. TSV densities

After the CMP process, Cu is recessed w.r.t. the Si surface – more so for 64-bit TSV arrays compared to 128-bit arrays.

Increased TSV height variation (which leads to bonding failures) is induced by:
1. Dense TSVs
2. Variable TSV array sizes within the same layer
TSV Layouts and TSV Height Variation (contd ..)

- **Layout (b)** achieves shorter wirelength, but TSV height variation greatly increases.
- **Layout (c)** has globally uniform (also the least) TSV density – CMP variation is the least but NoC routing too complex and inefficient.
- **Layout (d)**: one-way and two-way links; TSVs in the small array fail to contact landing pads.
- **Layout (e)**: Small array size with uniform ‘local TSV’ density.
Motivation

- Imprecise TSV bonding on landing pads could result in significant yield loss

- TSVs laid out without any constraints imposed by 3D technology, could result in bonding failures due to TSV height variation

- **Key idea**: A 3D NoC design flow producing low and uniform metal density
Related Work

- **Thermal driven Floorplanning Algorithm** [J. Cong et al. ICCAD 2004]
  - thermal models, FP-algorithm with SA-based perturbations
- **Sunfloor-3D NoC topology synthesis** [Seiculescu et al. DATE 2009]
  - utilize min-cut partitioning for 3D topology generation
- **Rip-up and re-route based 3D NoC synthesis** [Yan et al. ICCD 2008]
  - for routing flows and optimizing network topology for power

- Previous 3D NoC designs have not considered CMP variation which may lead to bonding failures
- Unlike previous works; all cores, routers, and TSV arrays are **simultaneously** floorplanned in each layer.
Problem Formulation

**Inputs:**
- A core graph $G(V,E)$ with $n$ vertices is a directed graph, where:
  1. each vertex $v_i \in V$ represents a core
  2. each directed edge $e_{i,j} \in E$ represents communication relation $\{vol(e_{i,j})\}$ between $v_i$ and $v_j$
- Constraints: latency, # of layers $k$

**3D NoC Design Objectives:**
- assign cores to proper silicon layers
- determine 3D NoC topology, allocate routing paths
- floorplan cores, routers and TSV arrays in a CMP-aware manner.
Comparison between Application Specific 3D NoC Design Flows

(a) Conventional 3D NoC design flow

(b) CMP-aware 3D NoC design flow
CMP-Aware Core-to-Layer Assignment -- (Objectives)

- Set of cores $V=\{v_1, v_2, ..., v_n\}$ with areas $\{A_1, A_2, ..., A_n\}$ is assigned to $k$ layers $L=\{l_1, l_2, ..., l_k\}$, thus, $V=\{V^{l_1}, V^{l_2}, ..., V^{l_k}\}$ is obtained

$$\alpha_{\text{min}} \frac{\sum_{i=1}^{n} A_i}{k} < A_i < \alpha_{\text{max}} \frac{\sum_{i=1}^{n} A_i}{k}$$

- Objective is to minimize inter-layer communication and lower temperature

$$\min \left[ \beta_1 \sum \text{vol}(e_{i,j}) \cdot |u - v| + \beta_2 \left( \sum_{p=1}^{k} \sum_{q=1}^{p} P_p R_q + R_b \sum_{p=1}^{k} P_p \right) \right]$$

s.t. $\forall v_i \in V^l_u, \forall v_j \in V^l_v (u \neq v)$

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.
Core-to-Layer Assignment by Recursive Bi-Partitioning

Algorithm 1: Core-to-Layer Assignment by Recursive Bi-Partitioning

1: while the number of partitioned layers is not equal to the target number of layers do
2: Find bi-partitions of cores with min. cost computed by Eq. (2);
3: Compute 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

$Comm. Gain \{CG\} = (Inter\_layer\ comm.) - (Intra\_layer\ comm.)$
Step 1: A router communication graph $RCG(R,C)$ where each vertex $r_i \in R$. The objective of 2D topology is:

$$
\min \left[ \sum \text{vol}(e_{i,j}) \cdot \text{dist} \left( M(v_i), M(v_j) \right) \right]
$$

subject to

$$
\text{bw} \left( \text{link} \left( M(v_i), M(v_j) \right) \right) \geq \text{vol}(e_{i,j}), (\forall v_i, \forall v_j) \in V^lu
$$

where $\text{dist}(r_p, r_q)$ is distance (hop count) between $r_p$ and $r_q$ and $M()$ is a core-to-router mapping function.

Step 2: Objective of topology decision among layers is:

$$
\min \left[ \sum \text{vol}(e_{i,j}) \cdot \text{dist}(r_p, r_q) \right]
$$

subject to

$$
\text{link}_{TSV}(r_p, r_q) \neq \text{link}_{TSV}(r_q, r_p),
$$

$$
\text{bw} \left( \text{link} \left( r_p, r_q \right) \right) \geq \text{vol}(e_{i,j}), \forall v_i \in V^lu, \forall v_j \in V^lv (u \neq v)
$$
2D Router-to-router/core and Layer-to-layer interconnection

**Step 1:** Interconnections within the same layer
1. Router-to-core mapping is performed by i-way min-cut partitioning (i is # of routers)
2. Inter router communication graph (RCG) is built by MST (dist = 1/vol) or by P2P connections

**Step 2:** Layer-to-Layer Interconnections
Objective: minimum hop count; Constraint: 1-way links
The goal of the FP is to generate the layout that minimizes area, power consumption, and peak temperature.

Cost function for the SA-based FP, here, maximum temperature difference in the same layer is minimized

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

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

where \( T(x,y,l_u) \) is the temperature of a tile and \( th_w \) is the maximum allowable wirelength.
Experiments – TSV height variation

- TSV Density and Predictive CMP Model:
  \[ hv = 0.8017 \ln \left( \frac{s}{p} \right) + 1.226 \]

- “Local TSV density” of CAS is more uniform and lower than that of [14]
- CAS has 17.9% lower TSV height variation than [14]

Table 1: TSV Height Variation Comparison (μm).

<table>
<thead>
<tr>
<th>Network protocol</th>
<th># of wire of one(two)-way link</th>
<th>[14]</th>
<th>CAS</th>
<th>Imp. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AHB [1]</td>
<td>137 (274)</td>
<td>1.651</td>
<td>1.372</td>
<td>16.9</td>
</tr>
<tr>
<td>AXI [1]</td>
<td>204 (408)</td>
<td>1.821</td>
<td>1.551</td>
<td>14.8</td>
</tr>
<tr>
<td>APB [1]</td>
<td>99 (198)</td>
<td>1.551</td>
<td>1.226</td>
<td>21.0</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>1.657</td>
<td>1.363</td>
<td>17.9</td>
</tr>
</tbody>
</table>

[14] is [Yan, Lin ICCD 2008]
‘CAS’ is CMP-Aware Application Specific 3D NoC
Experiments – hop-count, wire-length & power

- CAS improves **total hop count** as topology decisions are made before floor-planning
  - CAS achieves, on average, 15% lower hop count than [14]
- Using only one-way links may result in increasing **wirelength**
  - In worst case, CAS achieves just 0.3% longer total wirelength than [14]

**Power consumption** of CAS is 8.1% and 7.8% lower than that of [14] in MST and P2P, respectively
**Strengths:**
- Includes the routers and TSVs in the floorplanning step compared to disjoint floorplanning in conventional design flows
- One of the first works to address manufacturability issues in 3D NoC design

**Possible Weaknesses:**
- Local TSV density = \{size of TSV array/pitch\} is vague
  - **why does metal (TSV) density vary with array size??**
- “CMP-awareness” is restricted to 1-way TSV links, but placement of such arrays is not considered; e.g. min. distance between TSVs
- Algorithm for Router-to-router interconnections (for minimum hop count) across layers, is not explained
Thank You

- Open to more Questions !!
To insert a task $v_j$ into $P_k$, a simple insertion sort on task priorities of $P_k$: $O(|V| \log |V|)$

To evaluate system execution time, traverse all the nodes and edges on the task-graph (DAG): $O(|V| + |E|)$

Task insertion and system evaluation are performed at most $|V||P|^2$ times

Thus, $O(|V|^{2}|P|^2 + |V||P||E| + |V|^{2}|P|^2 \log |V|)$
Back ups – Thermal model (CBA-T)  
{Ref. [3] [J. Cong et al. ICCAD-2004]}

\[
\min \left[ \beta_1 \sum \text{vol}(e_{i,j}) \cdot |u - v| + \beta_2 \left\{ \sum_{p=1}^{k} \left( P_p \sum_{q=1}^{p} R_q \right) + R_b \sum_{p=1}^{k} P_p \right\} \right]
\]

s.t. \( \forall v_i \in V^u, \forall v_j \in V^v (u \neq v) \)

Thermal-Electrical Equivalents:
(i) Thermal Resistance \( <-> \) Resistance
(ii) Thermal current \( <-> \) Power density
(i) Thermal Voltage \( <-> \) Temperature

Figure 6 3D Compact Resistive Network Thermal Model
[Swinnen et al.]: within-die thickness variation after silicon-CMP was 1.5μm for a die size of $10.6 \times 10.6 \text{mm}^2$ when TSVs of which the diameter, pitch, and density are 5μm, 10μm, and 10k/mm$^2$, respectively, were evenly distributed over the whole chip.

The within-die thickness variation is directly related to TSV height variation.
Algorithm 2: Topology Decision within Layer

1: for \( i = \text{max}_\text{router} \) to \((\text{the number of core}/\text{max}_\text{int})\) do
2: \hspace{1em} Find \( i \)-way min-cut partitions under \( \text{max}_\text{int} \) constraints;
3: \hspace{1em} Assign each group to one router;
4: \hspace{1em} Interconnect routers by user’s design objective;
5: \hspace{1em} Build router communication graph (RCG);
6: \hspace{1em} Build prohibited turn set for RCG to avoid deadlocks;
7: \hspace{1em} Find paths for flows across different routers in each layer;
8: \hspace{1em} Evaluate latency and and bandwidth;
9: \hspace{1em} Go to line 2 if application constraints are not satisfied;
10: end for
11: Choose the best topology and design point;
Back-ups – Experiments

- Thickness of blue lines indicate communication volume
- Yellow rectangles - cores, red rectangles - TSV arrays, and green rectangles - routers
- [14] includes one-way and two-way links, CAS includes just one-way links
Back-ups - Experiments

### Table 3: Total Wirelength Comparison (mm).

<table>
<thead>
<tr>
<th>the number of layer</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>I (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>M\n</td>
<td>n100</td>
<td>9.6</td>
<td>8.4</td>
<td>7.5</td>
<td>7.1</td>
<td>6.5</td>
</tr>
<tr>
<td></td>
<td>CAS</td>
<td>9.6</td>
<td>8.5</td>
<td>7.4</td>
<td>7.1</td>
<td>6.6</td>
</tr>
<tr>
<td>S\n</td>
<td>n200</td>
<td>22.6</td>
<td>19.1</td>
<td>16.6</td>
<td>15.1</td>
<td>13.8</td>
</tr>
<tr>
<td></td>
<td>CAS</td>
<td>23.2</td>
<td>19.1</td>
<td>16.6</td>
<td>15.0</td>
<td>13.3</td>
</tr>
<tr>
<td>T\n</td>
<td>n300</td>
<td>46.5</td>
<td>39.5</td>
<td>34.4</td>
<td>30.5</td>
<td>27.2</td>
</tr>
<tr>
<td></td>
<td>CAS</td>
<td>47.0</td>
<td>40.0</td>
<td>34.1</td>
<td>30.7</td>
<td>27.3</td>
</tr>
<tr>
<td>Imp. (%)</td>
<td>-1.4</td>
<td>-0.9</td>
<td>0.7</td>
<td>-0.2</td>
<td>0.6</td>
<td>-0.3</td>
</tr>
<tr>
<td>P\n</td>
<td>n100</td>
<td>47.2</td>
<td>38.6</td>
<td>32.9</td>
<td>29.6</td>
<td>26.4</td>
</tr>
<tr>
<td></td>
<td>CAS</td>
<td>46.5</td>
<td>38.3</td>
<td>32.3</td>
<td>28.0</td>
<td>26.2</td>
</tr>
<tr>
<td>2\n</td>
<td>n200</td>
<td>95.4</td>
<td>77.7</td>
<td>67.8</td>
<td>61.3</td>
<td>55.2</td>
</tr>
<tr>
<td></td>
<td>CAS</td>
<td>89.6</td>
<td>75.1</td>
<td>65.0</td>
<td>58.5</td>
<td>52.3</td>
</tr>
<tr>
<td>P\n</td>
<td>n300</td>
<td>144.6</td>
<td>129.9</td>
<td>115.5</td>
<td>102.6</td>
<td>94.5</td>
</tr>
<tr>
<td></td>
<td>CAS</td>
<td>132.1</td>
<td>119.6</td>
<td>108.6</td>
<td>99.4</td>
<td>86.1</td>
</tr>
<tr>
<td>Imp. (%)</td>
<td>7.5</td>
<td>5.4</td>
<td>4.8</td>
<td>3.9</td>
<td>6.5</td>
<td>4.6</td>
</tr>
</tbody>
</table>

### Table 2: Hop Count Comparison.

<table>
<thead>
<tr>
<th>the number of layer</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>I (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>M\n</td>
<td>n100</td>
<td>1410</td>
<td>1470</td>
<td>1625</td>
<td>1671</td>
<td>1927</td>
</tr>
<tr>
<td></td>
<td>CAS</td>
<td>1201</td>
<td>1254</td>
<td>1299</td>
<td>1361</td>
<td>1650</td>
</tr>
<tr>
<td>S\n</td>
<td>n200</td>
<td>3341</td>
<td>3366</td>
<td>3459</td>
<td>3737</td>
<td>3927</td>
</tr>
<tr>
<td></td>
<td>CAS</td>
<td>2654</td>
<td>2931</td>
<td>2698</td>
<td>2934</td>
<td>3043</td>
</tr>
<tr>
<td>T\n</td>
<td>n300</td>
<td>5211</td>
<td>5158</td>
<td>5178</td>
<td>5257</td>
<td>5230</td>
</tr>
<tr>
<td></td>
<td>CAS</td>
<td>4065</td>
<td>3912</td>
<td>4077</td>
<td>3984</td>
<td>4118</td>
</tr>
<tr>
<td>Imp. (%)</td>
<td>20.5</td>
<td>19.0</td>
<td>21.3</td>
<td>22.4</td>
<td>20.5</td>
<td>19.7</td>
</tr>
<tr>
<td>P\n</td>
<td>n100</td>
<td>1193</td>
<td>1336</td>
<td>1488</td>
<td>1629</td>
<td>1799</td>
</tr>
<tr>
<td></td>
<td>CAS</td>
<td>1077</td>
<td>1194</td>
<td>1207</td>
<td>1416</td>
<td>1575</td>
</tr>
<tr>
<td>2\n</td>
<td>n200</td>
<td>2051</td>
<td>2487</td>
<td>2744</td>
<td>3136</td>
<td>3285</td>
</tr>
<tr>
<td></td>
<td>CAS</td>
<td>2041</td>
<td>2278</td>
<td>2441</td>
<td>2788</td>
<td>2968</td>
</tr>
<tr>
<td>P\n</td>
<td>n300</td>
<td>2638</td>
<td>3279</td>
<td>3433</td>
<td>4163</td>
<td>4371</td>
</tr>
<tr>
<td></td>
<td>CAS</td>
<td>2626</td>
<td>2943</td>
<td>3405</td>
<td>3234</td>
<td>3695</td>
</tr>
<tr>
<td>Imp. (%)</td>
<td>2.3</td>
<td>9.7</td>
<td>8.0</td>
<td>16.7</td>
<td>12.9</td>
<td>11.0</td>
</tr>
</tbody>
</table>

![Graph showing the relationship between metal density and wire variation](image-url)
Chemical-Mechanical Polishing Aware Application-Specific 3D NoC Design
Wooyoung Jang, Ou He†, Jae-Seok Yang‡, and David Z. Pan†
SoC Platform Development Team, Samsung Electronics, Yongin-City, South Korea
†IBM System & Technology Group, Beijing, China
‡CAE Team, Samsung Electronics, Hwaseong-City, South Korea
*Department of Electrical and Computer Engineering, the University of Texas at Austin, Austin, USA
wooyoung.jang@samsung.com, heou@cn.ibm.com, js.yang@samsung.com, dpan@ece.utexas.edu

ABSTRACT
In this paper, we propose the first chemical-mechanical polishing (CMP) aware application-specific three-dimensional (3D) network-on-chip (NoC) design that minimizes through-silicon-via (TSV) height variation, thus reduces its bonding failure, and meanwhile optimizes conventional NoC design objectives. Our 3D NoC design assigns cores to proper silicon layers, determines the 3D NoC topology, allocates routing paths, and then floorplans cores, routers and TSV arrays by 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 can achieve lower TSV height variation, higher performance and lower power consumption than the previous state-of-the-art 3D NoC designs.

1. INTRODUCTION
Network-on-chip (NoC) is an effective solution for on-chip communication in three-dimensional (3D) interconnections. However, the 3D NoC must meet not only application constraints, but also manufacturing constraints imposed by the 3D technologies. So far, many researchers have addressed the issues of 3D floorplanning and 3D NoC topology generation with the consideration of thermal hot spots. Besides the thermal and related thermal-mechanic stress effects [2], one particular challenge is that the wide range of the metal area by through-silicon-vias (TSVs) and landing pads increases non-uniform metal density, and thus results in the critical variation of wire thickness and TSV height during a chemical-mechanical polishing (CMP) process [8]. The CMP processes can be used for the removal of extra Cu on silicon after filling TSVs with Cu or depositing Cu on 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 timing variation. Moreover, the uneven TSV height leads to bonding failure between TSVs and landing pads. To mitigate the non-uniform metal density, dummy metal or TSV can be filled in empty spaces, but it may affect RC parasitics [8][9] and significantly reduce usable silicon area.

In this paper, we propose the first CMP-aware application-specific 3D NoC design that minimizes TSV height variation, thus reduces bonding failure, and meanwhile optimizes conventional NoC design objectives. For NoC vertical links composed of tens to hundreds of TSVs, the layout of each individual TSV is not efficient since it results in complex global routing and TSV manufacturing stress affects more transistors [2]. Therefore, TSVs should be placed as an array type in 3D NoC.

However, the array with dense TSVs is sensitive to a CMP process which results in high TSV height variation, and thus leads to bonding failure. Moreover, if the arrays with different TSV density are used in the same layer, bonding TSVs on landing pads is more difficult. Therefore, TSVs in an array should be placed with a pitch resulting in low CMP variation endured by a bonding technique and TSV arrays with the same density should be inserted in each layer. In addition, since the size of the TSV arrays is too large, TSV arrays should be handled during the floorplanning stage in physical design. Based on these motivations, the major contributions of this paper include:
1) We show that TSV height variation during silicon-CMP process is more severe in 3D NoC.
2) We propose a CMP-aware application-specific 3D NoC design which consists of core-to-layer assignment, topology synthesis, and floorplanning.
3) 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.

The rest of this paper is organized as follows: Section 2 reviews related works. Section 3 introduces CMP and Cu-Cu thermal-compression direct bonding, and then addresses various TSV layouts and their CMP variation. Section 4 formulates CMP-aware application-specific 3D NoC design problems. Section 5 presents detailed techniques of the proposed algorithms. Section 6 shows experiment results and Section 7 concludes this paper.

2. RELATED WORKS
Several works for 3D ICs have explored thermal floorplanning [3][6]. A fast thermal-driven floorplanning algorithm with a 3D floorplan representation and integrated compact resistive network thermal model was proposed in [3]. Hung et al. presented interconnect and thermal-aware floorplanning for 3D microprocessors in [6]. Recently, researchers have synthesized a 3D NoC topology with the limited number of TSVs. Yan et al. presented a 3D NoC synthesis algorithm that made use of accurate power and delay models for 3D ICs [14]. In [10] and [12], 3D NoC topology synthesis algorithms based on a direct extension of the 2D NoC synthesis procedure was proposed.

However, these previous 3D NoC designs have not considered CMP variation 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 suffer low physical design quality. This is because 3D floorplanning was first performed without the large routers and the TSV arrays, and thus there might be not enough spaces to place the routers and TSV arrays [10][12]. Even if 3D floorplanning is again performed after deciding a 3D NoC topology [14], their design qualities such as wirelength, power consumption, and area are severely limited.

The work is done while Wooyoung Jang was a Ph.D. student at the University of Texas at Austin.
3. PRELIMINARIES

3.1 Chemical-Mechanical Polishing Process

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 [13], a TSV-wafer is ground down to a target thickness slightly above the TSV depth (keeping TSVs unexposed) and further thinned using a CMP process. The chemical reaction creates a hydroxilated-form material which has weaker atomic bonds. Then, a mechanical surface abrasion aided by slurry particles removes the material. Fig. 1(a) shows the uneven surface of wafer backside after grinding and CMP. Subsequently, the polished silicon surface is plasma-etched, such that the TSVs protrude from the wafer as shown in Fig. 1(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 error. The top metal layer with land pads is also polished by CMP to remove overburden Cu. Finally, a wafer or die with landing pads is bonded with a different wafer or die with TSVs.

Fig. 1: Local topography on backside of wafer after (a) grinding and CMP and (b) Si-recess etch following CMP [13].

3.2 TSV Layouts and TSV Height Variation

Silicon-CMP is just used for finely thinning silicon after grinding silicon backside since the processing time of CMP is too long. As silicon-CMP involves simultaneous polishing of silicon, Cu, and barrier, their removal rates are different according to both different chemical effects on the materials and different TSV densities. The different removal rates of these materials results in different polish times across the wafer backside. For example, in Fig. 2(a), by the time the silicon and barrier under TSVs used for a 64-bit link are cleared, the silicon and barrier under TSVs used for 128-bit links might have been not cleared yet. Hence, either the silicon and barrier under the 128-bit TSVs are underpolished at the time the silicon and barrier under the 64-bit TSVs are cleared or the 64-bit TSVs are overpolished at the time the silicon and barrier under the 128-bit TSVs are cleared. Fig. 2(a) shows the 128-bit TSVs are underpolished after silicon-CMP. In [13], IMEC TSV technology showed that within-die thickness variation after silicon-CMP was 1.5μm for a die size of 10.6×10.6mm² when TSVs of which the diameter, pitch, and density are 5μm, 10μm, and 10k/mm², respectively, were evenly distributed over the whole chip. The within-die thickness variation is sensitive to high and irregular TSV density and directly related to TSV height variation. Consequently, the uneven TSV height variation can induce TSV bonding failure as shown in Fig. 2(a). In particular, the bonding failure will be more severe in a Cu-Cu direct thermo-compression bonding technique since TSVs must be directly contacted to landing pads without any micro-bump. Metal fill synthesis is not an efficient solution for silicon-CMP since dummy TSV insertion would significantly increases the overall chip area.

Fig. 2: TSV layouts and TSV height variation induced by CMP process. TSVs can be placed with different schemes during placement and routing [2]. If TSVs are laid out without any constraints imposed by 3D technology, they can be distributed as shown in Fig. 2(b). Whereas such layout achieves much shorter wirelength, TSV height variation induced by silicon-CMP greatly increases due to uneven TSV density. In Fig. 2(c), TSVs are placed with globally uniform density distributions. The TSV distribution provides the least TSV height variation to 3D ICs. However, such 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 makes system performance degraded or timing closure difficult. In addition, the layout of each individual TSV causes manufacturing stresses to more devices [2]. Therefore, grouping TSVs to an array and then laying out the array is more desirable for 3D NoC.

In Fig. 2(d), there exist two kinds of TSV arrays. The small array includes a one-way link and the large array includes a two-way link which may have two times more TSVs than the one-way link. TSVs in the small array fail to contact landing pads since TSVs in the large array are less cleaned than those in the small array during silicon-CMP such that the surface in a die is uneven. In Fig. 2(a) that is the cross section of AB in Fig. 2(d), the 64-bit TSV array has the strong possibility of failing to contact landing pads on silicon layer 2 since the 128-bit TSV array is underpolished. In addition, since the metal density of the 128-bit TSV is high, its own silicon-CMP variation can be 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-bit TSV array has a wider TSV pitch, its density can be as low as that of the 64-bit TSV array. However, since it has the penalty of area, we focus on reducing the size of a TSV array as shown in Fig. 2(e).

4. PROBLEM FORMULATION

In most previous application-specific 3D NoC designs [10][12][14], 3D floorplanning is first performed and then a 3D topology is determined as shown in Fig. 3(a), where their 3D technology constraint is the number of allowable TSVs. However, there may be no enough dead space where routers and TSV arrays can be physically placed after deciding a 3D topology [10][12]. In order to prevent overlapping routers, TSV arrays, and cores already floorplanned, additional floorplanning is performed in each layer [14], but such 3D NoC design flow is not efficient for reducing wirelength, hop count, and thus energy consumption. Furthermore, the layout of TSV arrays without considering CMP variation leads to TSV bonding failure on landing pads.
Fig. 3: Application-specific 3D NoC design flows.

(a) Conventional 3D NoC design flow (b) CMP-aware 3D NoC design flow

The problem of the CMP-aware application-specific 3D NoC design is as follows:

\[
\min \left[ \sum \left( v_1^e \cdot \text{dist}(v_i, v_j) \right) \right] \\
\text{s.t.} \quad \text{bw}(\text{link}(v_i, v_j)) \geq v_1^e, \forall v_i \in V^L, \forall v_j \in V^L (u \neq v)
\]

where \(v_1^e\) is the maximum allowable wirelength. Then, we connect routers in adjacent layers, based on the RCG graphs. The objective of our topology decision among layers is as follows:

\[
\min \left[ \sum \left( \text{vol}(e_{ij}) \cdot \text{dist}(r_p, r_q) \right) \right] \\
\text{s.t.} \quad \text{bw}(\text{link}(r_p, r_q)) \geq \text{vol}(e_{ij}), \forall v_i \in V^L, \forall v_j \in V^L (u \neq v)
\]

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

4.4 Floorplanning

Based on the predictive CMP model, we compute a TSV pitch where a used boding technique must endure TSV height variation in the number of TSVs covering a one-way vertical link. Then TSV arrays are inserted between any routers in adjacent layers. As the inputs of our floorplanner, we take a set of cores, routers, and TSV arrays, \(\{v_1, v_2, ..., v_n\}\). Each block can be free to rotate and change the aspect ratio continuously in a given range \([AR_{min}, AR_{max}]\). A floorplan \(F\) is the assignment of \((x, y)\) for each block \(v_i\) without any overlap of all cores, routers and TSV arrays, where half-perimeter wire length estimation is used. We use the thermal model proposed in [3], which minimizes the maximum temperature difference in the same layer. Finally, the objective of our floorplan \(F\) is as follows:

\[
\min \left[ \gamma_1 \sum A_i + \gamma_2 \sum (\text{wl}(e_{pq}) \times \text{vol}(e_{pq})) + \gamma_3 \left( \max \left( T(x, y, l_i) \right) - \min \left( T(x, y, l_i) \right) \right) \right] \\
\text{s.t.} \quad \forall v_i \in V^L, \forall v_j \in V^L (u \neq v)
\]

where \(\gamma_1, \gamma_2, \text{ and } \gamma_3\) are weighting factors. \(T(x, y, l_i)\) is the temperature of a tile in \(x, y\), and \(l_i\) at \(x\)-axis, \(y\)-axis, and \(layer\) respectively, and \(th_u\) is the maximum allowable wirelength.
5. CMP-AWARE 3D NOC DESIGN

5.1 CMP-Aware Core-to-Layer Assignment

Since the number of TSVs required depends on communication volume between different layers, the communication volume should be minimized together with thermal consideration. In addition, the area of each layer should meet the area constraint, Eq. (1). Fig. 4 shows two different core-to-layer assignment approaches where eight cores are assigned to four layers. Let a core graph given as shown in Fig. 4(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. 4(b), the cores are partitioned to \{A, B\}, \{C, D\}, \{E, F\}, and \{G, H\} that have the same area and the minimum cut. Then, the partitioned subgroups are one-to-one assigned to any layers, achieving the minimum hops as shown in Fig. 4(c).

The second approach we propose in Algorithm 1 recursively performs area-balanced bi-partitioning with the minimum cost computed from Eq. (2). Fig. 4(d) shows the result of the first bi-partitioning where the same area and the minimum cut are obtained (line 2). Then, any core which communicates other cores in a different layer is assigned in advance, depending on their communication gain as shown in Fig. 4(e) (line 5). The communication gain is computed as the subtraction of the amount of intra-layer communication from that of inter-layer 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. 4(e), core B, C, E, and F communicate 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 boundary layers. Then, the second bi-partitioning in each sub-group is again performed for the minimum cut under the area constraint. Fig. 4(f) shows the final result where hop count between layers is 7 whereas the first approach is 8. Therefore, the second one can require fewer TSVs. The basic idea of Algorithm 1 can be easily extended even if the number of a given layer is not a power of two.

![Algorithm 1: Core-to-Layer Assignment by Recursive Bi-Partitioning](image)

5.2 CMP-Aware 3D NoC Topology Decision

Since a 3D network topology decision problem is NP-Hard, 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 subproblems, called router-to-core/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 be easily 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 Eq. (4), which minimizes TSV density variation, thus reduces TSV height variation resulting in TSV bonding failure.

5.2.1 2D Router-to-router/core interconnection

Given a core graph, the number of allowable routers (max_router), and the number of allowable interconnection to a router (max_int), our 2D topology synthesis technique interconnects possible cores to any routers. The objective of our 2D 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 (max_router) in each layer.

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 max_int constraint (line 2) and then each group is assigned to one router (line 3). Next, links between the routers are inserted according to user’s design objective (line 4). We implement the minimum spanning tree (MST) or point-to-point (P2P) interconnection. MST first interconnects two vertices nearby. Similarly, since two routers, \( r_p \) and \( r_q \) which heavily communicates each other 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. Next, a new router communication graph (RCG) is generated and then a prohibited turn set for the RCG is build to

![Fig. 4: Examples of assigning eight cores to four layers](image)
The technique, if a one-way vertical link for insertion candidates, where the minimum hop count under performance constraints. In our floorplanning, the objective of our layer-to-layer interconnection is to insert one-way links between layers for locally uniform TSV distribution in each layer. The goal of our floorplanning is to generate the layout that minimizes area, power consumption, and peak temperature. We modify an existing floorplanning technique [5] and invoke it with our unique cost function.

\[ hv = 0.8017 \ln \left( \frac{s}{p} \right) + 1.226 \]

where \( hv \) is TSV height variation, \( s \) is the size of TSV array, and \( p \) is a TSV pitch in the array. Based on this model, we can compute a TSV pitch for the size of a given TSV array, which guarantees TSV height variation endured by a bonding technique. For example, if the size of a TSV array including a one-way link (113 wires) for OCP is \( 11 \times 11 \), its TSV pitch must be at least 14.58\( \mu \)m for less than 1\% TSV height variation. On the contrary, if the size of a TSV array including a two-way link is \( 16 \times 16 \), its TSV pitch must be at least 21.21\( \mu \)m. Thus, their areas are 0.0256mm\(^2\) and 0.1151mm\(^2\), respectively. Consequently, two one-way vertical links can show lower CMP variation or smaller design area than a single two-way vertical link.

### 6.2 CMP-Aware Application-Specific 3D NoC

We implement the CMP-aware application-specific (CAS) 3D NoC and [14] on GSRC Benchmarks with 100, 200 and 300 modules [4]. Wafers are stacked in a face-to-back fashion and we set a diameter and pitch of TSV to 5\( \mu \)m and 10\( \mu \)m, respectively.

Table 1 shows TSV height variation when various network interfaces are used. The local TSV density of CAS is more uniform and lower than that of [14] and CAS has 17.9\% lower TSV height variation than [14]. Using only one-way links results in increasing hop count since it may not provide the shortest path. However, our 3D NoC design flow recovers the penalty of the hop count and even improves total hop count since a topology we defined the cost function as the product of communication volume \( vol(e) \) and wirelength \( w_i \) in Eq. (5). In addition, it is necessary to place cores, routers, and TSV arrays communicating within the allowable wirelength.

**Algorithm 2: Topology Decision within Layer**

1. **for** \( i = \text{max router to (the number of core/}(max \_ \text{int}) \) **do**
2. Find \( i \)-way min-cut partitions under \( max \_ \text{int} \) constraints;
3. Assign each group to one router;
4. Interconnect routers by user’s design objective;
5. Build router communication graph (RCG);
6. Build prohibited turn set for RCG to avoid deadlocks;
7. Find paths for flows across different routers in each layer;
8. Evaluate latency and and bandwidth;
9. Go to line 2 if application constraints are not satisfied;
10. Choose the best topology and design point;
11. **end for**

Fig. 5 shows TSV heights measured from the latest 3D ICs of IMEC after silicon-CMP, where the TSV diameter is 5\( \mu \)m [7]. With these industry measurement data, we model TSV height variation as follows:

**Table 1: TSV Height Variation Comparison (\( \mu \)m).**

<table>
<thead>
<tr>
<th>Network protocol</th>
<th>( # ) of wire of one(two)-way link</th>
<th>[14]</th>
<th>CAS</th>
<th>Imp. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AHB [1]</td>
<td>137 (274)</td>
<td>1.651</td>
<td>1.372</td>
<td>16.9</td>
</tr>
<tr>
<td>AXI [1]</td>
<td>204 (408)</td>
<td>1.821</td>
<td>1.551</td>
<td>14.8</td>
</tr>
<tr>
<td>APB [1]</td>
<td>99 (198)</td>
<td>1.551</td>
<td>1.226</td>
<td>21.0</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td></td>
<td>1.857</td>
<td>1.363</td>
<td>17.9</td>
</tr>
</tbody>
</table>

Fig. 6. Silicon-CMP variation based on IMEC wafer measurement [7].
decision is first performed. Table 2 shows total hop count. CAS achieves, on average, 15% lower hop count than [14]. CAS tends to further improve hop count in complex NoC with a number of modules and layers. In addition, when a network is synthesized with limited resources like MST, CAS further improves hop count. As shown in Table 3, CAS achieves just 0.3% longer total wirelength than [14] in MST and even 4.6% shorter total wirelength than [14] in P2P.

Table 2: Hop Count Comparison.

<table>
<thead>
<tr>
<th></th>
<th>n100</th>
<th>n200</th>
<th>n300</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>[14]</td>
<td>[14]</td>
<td>[14]</td>
</tr>
<tr>
<td>S</td>
<td>1193</td>
<td>1336</td>
<td>1488</td>
</tr>
<tr>
<td>T</td>
<td>1207</td>
<td>1416</td>
<td>1629</td>
</tr>
</tbody>
</table>

Table 3: Total Wirelength Comparison (mm).

<table>
<thead>
<tr>
<th></th>
<th>n100</th>
<th>n200</th>
<th>n300</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>[14]</td>
<td>[14]</td>
<td>[14]</td>
</tr>
<tr>
<td>S</td>
<td>1193</td>
<td>1336</td>
<td>1488</td>
</tr>
<tr>
<td>T</td>
<td>1207</td>
<td>1416</td>
<td>1629</td>
</tr>
</tbody>
</table>

Fig. 7 shows power consumption normalized by [14]. The power consumption of CAS is 8.1% and 7.8% lower than that of [14] in MST and P2P, respectively. The total area of CAS is slightly smaller than [14] since CAS has smaller total TSV array area than [14]. The runtime of CAS ranges from 48-99 seconds in n300, which is about three times faster than [14].

Fig. 8 shows the layouts generated by [14]+MST and CAS+MST, where blue lines show communication relations and their thickness indicates communication volume. Yellow rectangles, red rectangles, and green rectangles are cores, TSV arrays, and routers, respectively. Whereas Fig. 8(a) includes both one-way and two-way links, Fig. 8(b) includes just one-way links. Therefore, TSV heights are less variable, thus TSVs can directly contact landing pads easily.

7. CONCLUSION

In this paper, we proposed the first CMP-aware application-specific 3D NoC design. Our vertical integration managing architecture, physical design, and manufacturing issues together enables a reliable and robust 3D NoC with low power consumption and high performance. In particular, our CMP-aware 3D NoC approach reduces TSV height variation during the CMP process, and thus prevents bonding failures and timing variation.

8. REFERENCES