Early Estimation of TSV Area for Power Delivery in 3-D Integrated Circuits

Nauman H. Khan
Department of Computer Science
Tufts University
Medford, MA 02155
Email: nauman@cs.tufts.edu

Sherief Reda
School of Engineering
Brown University
Providence, RI 02912
Email: sherief_reda@brown.edu

Soha Hassoun
Department of Computer Science
Tufts University
Medford, MA 02155
Email: soha@cs.tufts.edu

***Abstract*—To harness the full potential of 3-D integrated circuits, analysis tools for early design space exploration are needed. Such tools, targeting multiple design facets and cost trade-off analysis, would allow designers to arrive at major decisions regarding architecture and implementations fabrics. We focus in this paper on the efficient estimation of on-chip power delivery requirements consistent with supply noise limits. We propose a number of algorithms to find the minimum number of through-silicon vias (TSVs) that deliver power with acceptable IR drops. Minimizing the number of TSVs reduces the total silicon die area which is the main recurring cost during fabrication. To compute the TSV requirements realistically, we utilize power traces derived from benchmark-based functional behavior of processors. To speed-up our simulations, we develop a trace selection technique that utilizes the relevant portion of power traces representing the worst load for IR drops. The trace selection scheme reduces the number of simulations by 51×. Using these traces we find the best spatial allocation of TSVs for a 3-D implementation of a processor. The iterative algorithm can be run in approximately one hour on a 40-processor cluster.**

I. Introduction

Power delivery network (PDN) design for a regular 2-D integrated circuit (IC) is a known challenging task. For a 3-D IC, increased device-density and package asymmetry make PDN design even more challenging. For example, conversion of a 2-D IC to an N-layer 3-D IC increases the power density by a factor of $N^{1/2}$ [10]. 3-D stacking using through-silicon via (TSV) in a flip-chip package provides higher interconnect-density and better performance for PDN than the wire-bond stacking technology [3]. However, these TSVs pass through the silicon substrate and occupy valuable silicon real estate. For instance, ITRS predicts the minimum TSV area for the global interconnect to be $16\mu m^2$ [1] whereas the area of a 6T SRAM cell for 45nm Hi-K Metal-Gate technology is 0.346 μm^2 [4]. As a result, the substrate area dedicated to TSVs needs to be minimized to reduce the manufacturing costs.

TSVs create blockages (≈46× the SRAM cell area) and neither a device can be fabricated nor a signal can be routed through the area occupied by a TSV. TSVs therefore impact the total number of realized devices, die floor-plan, and interconnect routing. TSVs are used to route inter-die signals (signal TSVs), to deliver power (power TSVs) to each die, and have been investigated for heat extraction (thermal TSVs). An early estimation of total TSV area helps to budget the overall penalty and manage the device and interconnect resources more efficiently.

We provide in this paper a set of algorithms to estimate and minimize the substrate area dedicated to power delivery. These algorithms can be applied early in the design stages when only functional block-level behaviors and a floorplan are available. Planning for signal TSVs clearly requires detailed circuit layout. Studies supporting the use of TSVs for heat removal are still in their early stages and do not consider realistic manufacturing constraints. Our proposed work is in contrast with recent TSV optimization techniques that utilize circuit-level information and are applied later in the design cycle [13] [11]. The highlights of our contributions are:

- We investigate the problem of PDN design at early design stages to determine area estimates for the TSV resources required for the PDN. By combining these estimates with other layout area estimates, designers can arrive at total area and cost estimations for potential 3-D implementations.
- We propose four techniques (Reduce Maximum Slack (RMS), Reduce Somewhat Arbitrary Slack (RSAS), Reduce Slack Locally (RSL), and Improve Worst Violation (IWV)) within an iterative framework to minimize the number of TSVs required for power delivery.
- To speed lengthy simulations associated with PDN analysis, we develop a methodology for selecting a subset of benchmark traces that are representative of the whole benchmark. This methodology results in a 51× reduction in the runtime of PDN IR analysis and still covers the workload range.
- Our problem formulation is generic and applicable to any TSV-based PDN design. We use a practical design consisting of three dies to compare the performance of these algorithms.

The rest of the paper is organized as follows. Related work is presented in Section II. The problem of power TSV area minimization is formulated in Section III. Our proposed algorithms are presented in Section IV. We describe our trace selection methodology in Section V, and discuss our results in Section VI. We provide a conclusion in Section VII.

II. Background and Related Work

3-D stacking provides exciting opportunities for architecture and circuit design, not possible with traditional 2-D IC design. Optimal 3-D design can be achieved by co-optimizing the architecture and technology at each stage of design. Wire-bonding and TSV are the two common techniques to stack multiple dies forming a 3-D IC [9]. Although, wire-bonding is a cheap straightforward approach, it is only suitable for low-power and low-frequency ICs that need less inter-die connections. TSVs, on the other hand, provide high density and high speed inter-die connections.

Studies on 3-D power delivery can be divided into two major categories: system-level design and analysis, and TSV optimization. In the system-level PDN design, Khan *et al.* provide a system-level comparison of power delivery for 2-D and 3-D ICs and investigate various TSV technologies that impact PDN quality [10]. Kim *et al.* analyze a multi-story power delivery technique where a higher than nominal supply voltage is applied from the package and distributed differentially to subsequent power rails using level conversion [8]. Huang *et al.* propose an analytical physical model of 3-D PDN to capture the impact of power supply noise [7].

To optimize TSV area, Lee *et al.* investigated a co-optimization methodology for signal, power, and thermal TSVs based on design of experiments and response surface method, and they showed that careful tuning of response surface models can lead to reliable optimization results [11]. Yu *et al.* use I/O compression and structured and parametrized model order reduction to efficiently ensure dynamic power/thermal integrity [13]. Current TSV optimization techniques provide valuable insights but none include TSV manufacturing aspects while co-optimizing power and thermal TSVs. The following are a few manufacturing constraints associated with TSVs:

1) Each TSV needs to have a *liner* which is an insulating material filled around the TSV to provide isolation as well as stop metal diffusion into substrate. None of the previous models of thermal TSV included this insulation layer.
2) Mechanical stress associated with TSV calls for a keep-away area where no devices can be fabricated. This means that TSVs can not be directly connected to a hotspot or any floorplan tile as assumed in [13].
3) Finally, power TSVs only connect to just the top metal layers. They will not be efficient in extracting heat from surrounding volume.

While functional data early in the design cycle can be used to estimate thermal profiles, studies considering the factors above are needed to further explore the benefits of thermal TSVs. Therefore, there currently are no realistic studies of the benefits of co-optimizing thermal and power TSVs. We thus decided to investigate power TSVs independently from thermal TSVs.

III. Problem Formulation

Because the proposed TSV estimate occurs early in the design cycle, detailed device-level floorplans are unavailable. At such early stages, a functional model of each die is however available. Thus, a set of workloads (e.g., for the target design) can be executed using an architectural simulator and the power traces of each functional block are captured. Within each functional block, we assume uniform power consumption.

We assume a 3-D IC consisting of K number of dies in a flip chip package. Each die has its own on-chip power grid, each with M grid nodes. The bottom die is connected to an off-chip PDN via C4 bumps and the rest of the dies are interconnected using TSVs. Because our technique targets early design exploration, we assume uniform TSV sizing and that TSV insertion points have already been identified, each with an index, $1 \leq i \leq M$. Each TSV grid location is referred to as a TSV node t_i. Fig. 1 illustrates a 2×2 portion of an on-chip power grid.

The size of a power TSV depends on several factors including fabrication process, power delivery requirements [10], stress minimization [12], heat removal requirements [13], and layout constraints. 3-D power grid design studies recommend larger power TSVs (with lower resistance) to reduce the voltage drops and to meet the current density requirements [7] [10]. However, larger TSV sizes directly impact the keep-away area requirement. A keep-away area is required around each TSV where no devices can be fabricated. Bart *et al.* suggest that the keep-away area increases with the increase in TSV area [12]. A simple calculation based on their findings shows that for 1.7× increase in the TSV area (increase in diameter from 6 μm to 8 μm), a 2.4× increase in keep-away area is required. So, instead of using few larger TSVs, we use multiple smaller TSV arrangements connected in parallel between two TSV grid points. For this paper, we assume a TSV diameter of 5 μm. Each location t_i is assigned a number of TSVs, n_i.

We assume that power TSVs supply neighboring devices as shown in Fig. 2. Each device node, $g^k_{i,j}$ connects the two closest TSV grid locations, t_i and t_j, and the device node is located on die k. TSVs at locations t_i and t_j are thus directly connected to $g^k_{i,j}$. We define the neighborhood of a device node as the set of 6 TSV locations closest to $g^k_{i,j}$. This is illustrated simply in die 3 of Fig. 2.

We denote the voltage at any node $g^k_{i,j}$ as $v^k_{i,j}$. Voltage violation is defined as the difference between $(V_{ref} - V_b)$ and

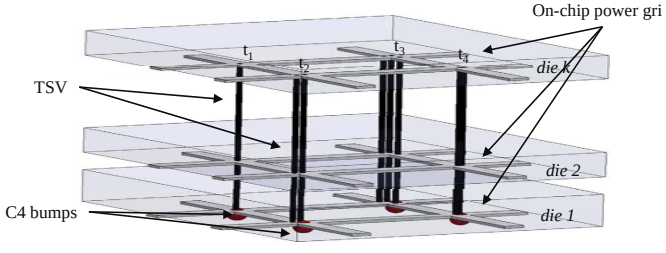

Fig. 1. An illustrative 2×2 power grid.

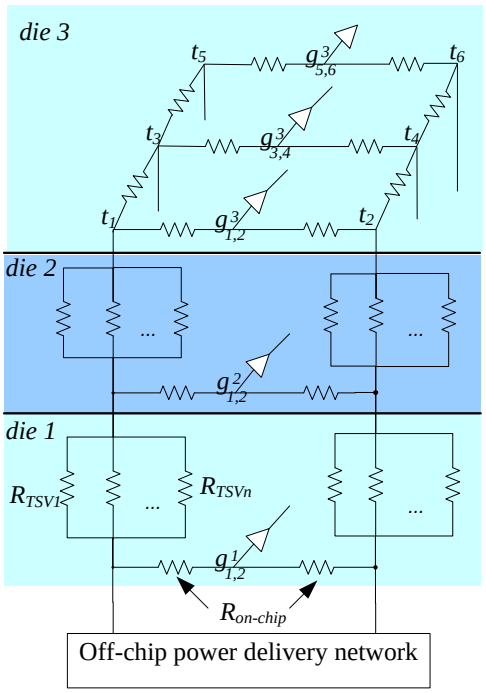

Fig. 2. Side and top view of select power delivery nodes in a stack of three dies, illustrating a device node's neighborhood.

$v^k_{i,j}$, where V_{ref} is the reference (input) voltage and V_b is the maximum allowable noise voltage. Voltage slack, on the other hand, is defined as the difference between $v^k_{i,j}$ and $(V_{ref} - V_b)$.

The total power TSV area can be calculated as $S \times \sum n_i$, where S is the TSV size and n_i is the number of TSVs at each node t_i. Our objective is to minimize the total number of power TSVs under the constraint that power integrity is maintained.

The power TSV minimization problem can be formulated as follows:

$$\min \sum_{i=1}^{M} n_i$$

subject to:

$$\forall_{i,j,k} : \left| V_{ref} - v^k_{i,j} \right| \leq V_b$$

$$\forall_i : n_i \geq 0$$

IV. Power TSV Minimization Algorithms

We propose in this section four techniques (Reduce Maximum Slack (RMS) , Reduce Somewhat Arbitrary Slack (RSAS), Reduce Slack Locally (RSL), and Improve Worst Violation (IWV)) within an iterative framework to minimize the number of power TSVs. The techniques differ in their starting and stopping points, and in the decision made during each iterative step. In the first three techniques, we start with an abundance of TSVs allowing the circuit to comfortably meet the voltage budget constraint. The initial number of TSVs at each grid node is chosen by considering the floorplan and the power requirements of each functional block. For each of the three techniques, we selectively decrement the number of TSVs and continue till all efforts fail to further decrease the number of TSVs while maintaining power integrity.

In contrast to the first three techniques, Improve Worst Violation (IWV) starts with a scarcity of TSVs and judiciously increases the number of power TSVs during each iterative step. A small number of TSVs is initially assigned to each node, selected based on the designers' experiences, or simply with one TSV. During the iterative process, the number of TSVs at the node with the largest voltage violation is incremented. The process repeats until all nodes meet the required noise budget.

A. Reduce Maximum Slack (RMS)

We explore in this technique the effectiveness of decreasing the number of TSVs for a node with the largest slack. An initial circuit is generated assuming uniform n_i for each grid node. This initial estimate is made by considering the power demand of each functional block. We assign a small number of TSVs (one, if no other information is available), at the TSV grid nodes within the functional blocks with the smallest power demand. We then assign relative n_i to rest of the grid nodes in the other functional blocks. We run SPICE assuming power requirements (as explained in our trace selection methodology in Section V). If any node $g^k_{i,j}$ fails the voltage budget constraint, we increment the minimum n_i for all nodes. We reevaluate the circuit. We continue this process until all $g^k_{i,j}$ meet the voltage budget constraint.

Pseudo code for the RMS iterative algorithm is presented in Algorithm 1. After initialization, and at the the beginning of each iteration of the while loop, a device node $g^k_{i,j}$ with the maximum voltage is identified as g_s (line 5). Then, the direct TSV neighbor node with the largest voltage slack, t_x, is identified (voltage measurements taken from the furthest away from C4 bumps, typically the worst case voltages). The number of TSVs at t_x, n_x, is decremented by one (line 7), and SPICE is used to evaluate the results (line 8). If the circuit fails, then the algorithm attempts to reduce the number of TSVs at the other direct neighbor, t_y, while restoring the TSV count at t_x. If both attempts fail, then the identified node g_s is marked as "done" (line 16), indicating that the device node cannot withstand the downsizing of its direct TSV neighboring nodes. The algorithm stops when all the device nodes are marked as "done", and no further TSV decrease is possible.

B. Reduce Somewhat Arbitrary Slack (RSAS)

This technique is similar to RMS, but differs in which of g_s's direct neighbors will be selected for TSV decrementing first. In particular, the algorithm differs in line 6, where the code is changed to select a direct neighbor at random. The rational is to avoid a purely greedy technique as was the case in RMS.

Input: An initial design with an abundance of TSVs
Output: The minimum n_i for each grid node such that each node meets the voltage budget

```
mark all device nodes g^k_{i,j} "not done"
Initialize noOfNodesDone to zero
run SPICE
while noOfNodesDone < M do
    Let g_s denote the device node that is "not done" and has largest voltage
    pick node t_x which is g_s's direct TSV neighbor that has the largest slack
    decrement n_x by 1
    run SPICE
    if circuit fails then
        increment n_x by 1
        pick the second neighboring TSV node t_y
        decrement n_y by 1
        run SPICE
        if circuit fails then
            increment n_y by 1
            mark g_s "done"
            increment noOfNodesDone by 1;
        end
    end
end
```

Algorithm 1: REDUCE MAXIMUM SLACK (RMS)

C. Reduce Slack Locally (RSL)

To further move away from a greedy technique, we explore in this technique the impact of minimizing the TSV count at nodes other than direct neighbors, and the impact of not immediately selecting the nodes with the maximum slack. The algorithm initialization is similar to the one shown in Algorithm 1, but the while loop is different and is presented in Algorithm 2. A node whose local neighborhood was not thoroughly explored is randomly selected. Within the neighborhood, TSV nodes with the maximum voltage slack are chosen for decrementing successively until none of the neighborhood TSV nodes can be further decremented. Once a neighborhood is explored, a device node is marked as "done". The iterations stop when all the device nodes are marked as "done".

D. Improve Worst Violation (IWV)

While the first three techniques attempted to lower an abundance of TSVs at each TSV node, IWV starts with few TSVs and increases the number of power TSVs during each iterative step. The pseudo code is shown in Algorithm 3. The initial circuit has a small number of TSVs at each grid node, selected based on experience, or simply with one TSV. During the iterative process, the number of TSVs at the node with the largest voltage violation is incremented. The process repeats until all nodes meet the required noise margins. This algorithm is greedy as well, as it always tries to improve the voltage

```
while noOfNodesDone < M do
    pick a random device node g_s that is "not done"
    Let S be a set of TSV nodes in g_s's neighborhood
    while S is not empty do
        Find t_s s.t. t_s ϵ S and t_s has maximum slack
        decrement n_s by 1
        run SPICE
        if circuit fails then
            increment n_s by 1
            remove t_s from S
        end
    end
    mark g_s "done"
    increment noOfNodesDone by 1;
end
```

Algorithm 2: REDUCE SLACK LOCALLY (RSL).

Input: An initial circuit with a low number of TSVs
Output: The minimum n_i for each grid node such that each node meets the voltage budget

```
run SPICE
Find the minimum voltage, v_s, in the circuit
while v_s < (V_ref − V_b) do
    Pick a device node g_s s.t. its voltage is equal to v_s
    Let S be a set of TSV nodes in g_s's neighborhood
    Find t_s s.t. t_s ϵ S and t_s has the largest noise violation
    increment n_s by 1
    run SPICE;
end
```

Algorithm 3: IMPROVE WORST VIOLATION (IWV)

of the most offending device node by incrementing the TSV count at the TSV node with the lowest voltage.

V. TRACE SELECTION

Predicting the power demand of functional blocks in a 3-D system is critical in determining the number of power TSVs in early design stages. Like our earlier work [10], we use current traces that represent a variety of current patterns: step, resonating, and pulsing patterns, and scale them appropriately to run in a 3-D stack composed of three dies in a flip-chip package. These patterns were derived based on the work of Meeta *et al.* [6], where four SPEC workloads (apsi, bzip, equake, and mcf) were run for 100 million instructions using Wattch [5], and 2048 cycle snippets (8192 total traces) representing the current patterns were then extracted. Such a power grid evaluation methodology replaces observing millions of instructions from a wide variety of benchmarks, thus significantly saving power grid simulation times.

A simple power grid design strategy is to identify the peak power requirement of each functional block across all cycles of the current traces, and then use these numbers to drive power grid design. Such a scenario however is pessimistic as blocks

typically do not operate at peak power at the same time, nor do they change current demand simultaneously. Relying on the peak power will therefore result in an unnecessarily large area for power TSVs. Another strategy is to use the current traces to estimate the number of power-TSVs for all functional blocks in each cycle, and then pick the most conservative estimate for each functional block. This method is ineffective as it requires performing our iterative algorithm for each of the 8192 cycle snippets. We propose to use a trace selection strategy that can speed up the estimation process when analyzing IR drops.

For each trace of a benchmark, and for every clock cycle, power dissipation of each functional block is normalized to total power dissipation The traces are then partitioned into bins. Each bin contains the traces with same distribution of normalized power dissipation over all the functional blocks. Some traces in one of the bins are illustrated in Fig. 3. Here, the relative power dissipation for each of the ten traces is similar, within a +/-5% among functional blocks, FB1-FB7.

While each bin contains a variety of traces with different total power dissipation, the relative power demands for each functional block are similar. Relative power consumption of the functional blocks determines the capability of the PDN to share power for neighboring functional blocks. So, within each bin, we need to simulate only the trace with largest total power consumption; for example, trace 10 will be simulated representing the bin shown in Fig. 3. We used this approach on 8192 traces of four benchmarks and we were able to reduce the number of traces to 160. A similar approach to analyze changes in current from one cycle to the next or over multiple cycles may be applicable for di/dt analysis.

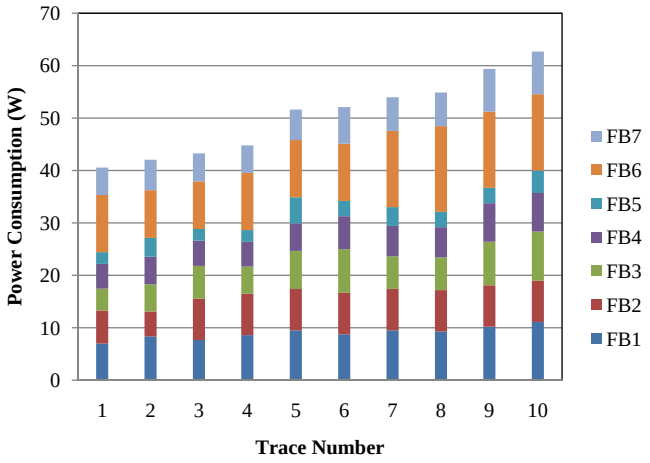

Fig. 3. Illustrative contents of a bin: the x-axis represents the trace number within a bin. The y-axis represents the total power dissipation in a particular clock cycle. The relative power distribution for functional blocks, FB1-FB7, is similar within each bin.

VI. Results and Discussion

We ran the four algorithms to estimate the number of TSVs for a block-level model of a 3-D IC that is similar to the one used in our earlier work [10], where we derive values of on-chip and off-chip power delivery components from published works and technical documentation. We utilize the electrical characterization approach by Alam et al. [2] to calculate the resistance of individual TSVs.

Fig. 4 shows the estimated number of TSVs for the selected traces of one of the benchmarks, bzip. The results obtained using IWV are always better than the other three algorithms. Each increase in the number of TSVs in IWV brings the device nodes closer to meeting the voltage requirement. The process is incremental. For the other three techniques, the minimization is a two-step process: a device node is selected and then a TSV node is chosen to decrease the number of TSVs. This strategy leads to a state where n_i for a node is decreased to an extent such that any decrement anywhere else in the circuit results in circuit failure. The greedy nature of the algorithms does not allow any significant backtracking to explore other options. Among the three reduction techniques, RMS and RSAS result in similar TSV numbers. The results of RSL however were inconsistent. The TSV estimates were sometimes much worse than the those obtained by RMS and RSAS, and sometimes better.

The estimated minimum required number of TSVs at each TSV node t_i is the maximum value of n_i across all benchmarks as the n_i choice must satisfy the demands of the worst-case benchmark. The total number of TSVs required by each benchmark is shown in Fig. 5. Clearly, IWV provides the best and smallest results for all four benchmarks. RMS and RSAS provide similar estimates. The performance of RSL is less consistent and dependent on the benchmark. For comparison, we ran these algorithms for a pessimistic power dissipation scenario, assuming worst case power dissipation of each functional block across all benchmarks. The total TSV count was $\approx 2.7\times$ the minimal TSV count found using select traces.

The run time of the iterative algorithms is dependent on the network details and number of insertion points, with SPICE as the bottleneck. Trace selection is a useful strategy to reduce the SPICE simulation load. We were able to compress 8192 traced of four benchmarks to 160 traces resulting in $\approx 51\times$ reduction. The TSV minimization using algorithm IWV on the benchmark apsi produced the exact number of required TSVs when using the select traces and all the traces. The iterative algorithm can be run for each of the 160 traces independently on a different machine. The run time thus becomes feasible. For a sample circuit with functional blocks of three dies (processor, memory, and accelerator chips) and 256 TSV grid points, the algorithm requires about one hour on a cluster/farm with 40 processing nodes, common in circuit design houses.

VII. Conclusion

We have developed and evaluated several estimation techniques to evaluate area requirements for 3-D power delivery when only a functional model and a floorplan are available. We have shown that an iterative framework is feasible. Within this framework, we have shown that a greedy algorithm that

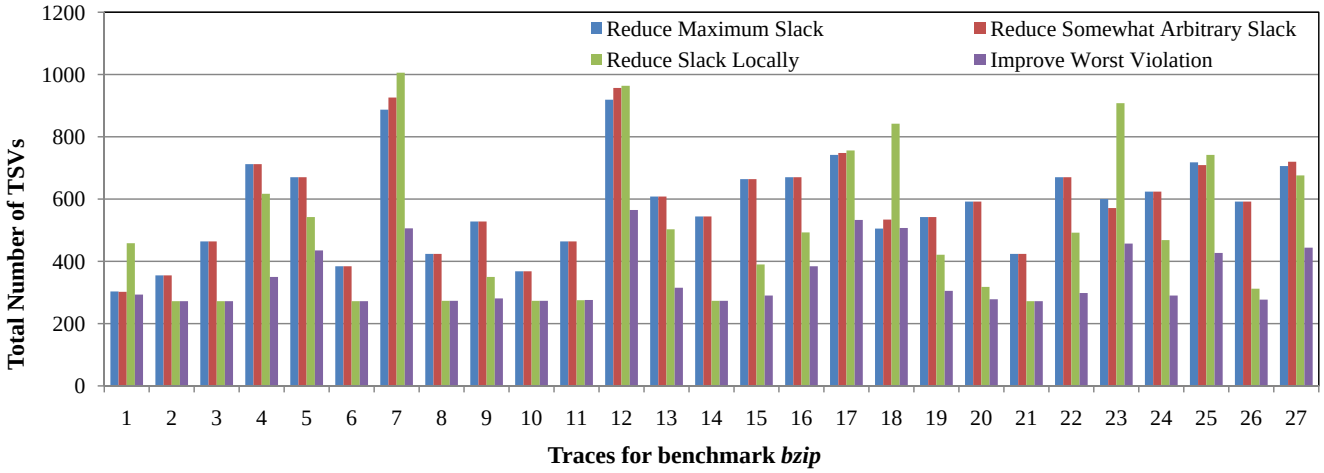

Fig. 4. Total number of TSVs required for each select trace for the bzip benchmark using the four minimization techniques.

gradually increments the number of TSVs at each grid node provides a minimal solution when compared to techniques that start with an abundance of TSVs and attempt to lower the TSV count. In addition, we developed a trace selection technique that allowed us to reduce the number of IR drop simulation traces by approximately 51×. This trace selection method can potentially be extended and applied for di/dt analysis. The run time of this iterative procedure is about one hour when dispatched on a cluster of 40 processing nodes.

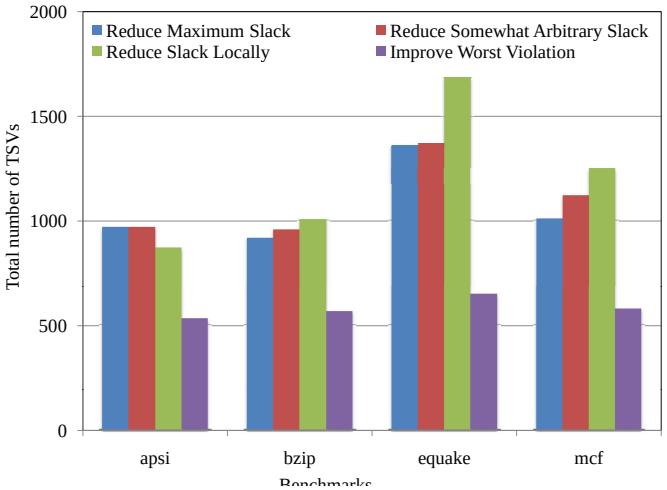

Fig. 5. Comparison of the four proposed techniques.

REFERENCES

[1] International Technology Roadmap for Semiconductors. *http://www.itrs.net/Links/2009ITRS/Home2009.htm*, 2009.

[2] S. M. Alam, R. E. Jones, S. Rauf, and R. Chatterjee. Inter-Strata Connection Characteristics and Signal Transmission in Three-Dimensional (3D) Integration Technology. In *ISQED '07: Proceedings of the 8th International Symposium on Quality Electronic Design*, pages 580–585. IEEE Computer Society, 2007.

[3] M. Bakir, C. King, D. Sekar, H. Thacker, B. Dang, G. Huang, A. Naeemi, and J. Meindl. 3D Heterogeneous Integrated Systems: Liquid Cooling, Power Delivery, and Implementation. In *Custom Integrated Circuits Conference, 2008*, pages 663–70, Piscataway, NJ, USA, 2008. IEEE.

[4] U. Bhattacharya, Y. Wang, F. Hamzaoglu, Y. Ng, L. Wei, Z. Chen, J. Rohlman, I. Young, and K. Zhang. 45nm SRAM Technology Development and Technology Lead Vehicle. *Intel Technology Journal*, 12(02), June 2008.

[5] D. Brooks, V. Tiwari, and M. Martonosi. Wattch: A Framework for Architectural-Level Power Analysis and Optimizations. In *Proceedings of the 27th annual international symposium on Computer architecture*, pages 83–94, Vancouver, British Columbia, Canada, 2000. ACM.

[6] M. S. Gupta, J. L. Oatley, R. Joseph, G. Wei, and D. M. Brooks. Understanding Voltage Variations in Chip Multiprocessors Using a Distributed Power-Delivery Network. In *Proceedings of the Conference on Design, Automation and Test in Europe*, pages 624–629, Nice, France, 2007. EDA Consortium.

[7] G. Huang, M. Bakir, A. Naeemi, H. Chen, and J. Meindl. Power Delivery for 3D Chip Stacks: Physical Modeling and Design Implication. In *Electrical Performance of Electronic Packaging, 2007 IEEE*, pages 205–208, Oct. 2007.

[8] P. Jain, T. Kim, J. Keane, and C. H. Kim. A Multi-Story Power Delivery Technique for 3D Integrated Circuits. In *13th International Symposium on Low Power Electronics and Design*, pages 57–62, Bangalore, India, 2008. ACM.

[9] T. Jiang and S. Luo. 3D Integration-Present and Future. 10th Electronics Packaging Technology Conference, EPTC 2008, pages 373–378, Singapore, Singapore, 2008. IEEE Computer Society.

[10] N. Khan, S. Alam, and S. Hassoun. Power Delivery Design for 3D ICs Using Different TSV Technologies. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 18(1), Jan. 2009.

[11] Y. Lee, R. Goel, and S. K. Lim. Multi-Functional Interconnect Co-Optimization for Fast and Reliable 3D Stacked ICs. In *2009 IEEE/ACM International Conference on Computer-Aided Design (ICCAD 2009), 2-5 Nov. 2009*, pages 645–51, Piscataway, NJ, USA, 2009. IEEE.

[12] B. Vandevelde, C. Okoro, M. Gonzalez, B. Swinnen, and E. Beyne. Thermo-Mechanics of 3D-Wafer Level and 3D Stacked IC Packaging Technologies. In *Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Micro-Systems, 2008. EuroSimE 2008. International Conference on*, pages 1–7, April 2008.

[13] H. Yu, J. Ho, and L. He. Allocating Power Ground Vias in 3D ICs for Simultaneous Power and Thermal Integrity. *ACM Trans. Des. Autom. Electron. Syst.*, 14(3):1–31, 2009.