Heuristics for Cells and I/O Pads Partitioning Targeting 3D VLSI Circuits

Sandro Sawicki, Marcelo Johann, Ricardo Reis
UFRGS, Federal University of Rio Grande do Sul, Institute of Informatics
Porto Alegre, BrazilI

Abstract: A 3D circuit is the stacking of regular 2D circuits. The advances on the fabrication and packaging technologies allow interconnection of stacked 2D circuits. However, 3D-vias can impose significant obstacles and constraints to the 3D placement problem. Most of the existing placement algorithms completely ignore this fact, but they do optimize the number of vias using a min-cut partitioning applied to a generic graph partitioning problem. This work proposes a new approach for I/O pads and cells partitioning addressing 3D-vias reduction and its impact on the 3D circuit design. The approach presents two distinct strategies: the first one is based on circuit structure analyses and the second one reducing the number of connections between non-adjacent tiers. The strategies outperformed a state-of-the-art hypergraph partitioner, hMetis [8] in the number of 3D-vias 19%, 17%, 12% and 16% using two, three, four and five tiers.

I. Introduction and Background

The design of 3D circuits is becoming a reality in the VLSI industry and academia. While the most recent manufacturing technologies introduce many wire related issues due to process shrinking (such as signal integrity, power, delay and manufacturability), the 3D technology seems to significantly aid the reduction of wire lengths [1-3] consequently reducing these problems. However, 3D technology also introduces its own issues. One of them is the thermal dissipation problem, which is well studied at the floorplanning level [4] as well as in placement level [3]. Another important issue introduced by 3D circuits is how to address the insertion of the inter-tier communication mechanism, i.e. a 3D-Via, since it introduces significant limitations to 3D VLSI design. This problem has not been properly addressed so far since there some aspects of the 3D via insertion problem that seem to be ignored by the literature: 1) all face-to-back integration of tiers imply that the communication elements occupy active area, limiting the placement of active cells/blocks; 2) the 3D-Via maximum density is considerably small compared to regular vias, which won't allow as many vertical connections as could be desired by EDA tools; 3) timing of those elements can be bad specially considering that a vertical connection needs to cross all metal layers in order to get to the other tier ; 4) 3D-Vias impose yield and electrical problems not only because of their recent and complex manufacturing process but also because they consume extra routing resources.

The 3D integration can happen in many granularity levels, ranging from transistor level to core level. While core level and block level integration are well accepted in the literature, there seem to exist some resistance to the idea of placing cells in 3D [6]. One of the reasons is that finer granularity demands higher 3D-vias, which might fail to meet the physical constraints imposed by them. On the other hand, the evolution of the via size is going fast and is already viable (for most designs) to perform such integration [2, 5] since we already can build 0.5 µm pitch face-to-face vias [6] and 2.4 µm pitch on face-to-back [5]; we believe that this limitation is more in the design tools side, since those are still not ready to cope with the many issues of 3D-vias [7].

The number of 3D-vias required in a design is determined by the tier assignment of each cell, which is performed during the cell partitioning. The cell partitioning is usually performed by an hypergraph partitioning tools (since it is straightforward to map a netlist into a hypergraph) such as hMetis [8] as done in [2]. On the other hand, hypergraph tools do not understand the distribution of partitions in the space (in 3D circuits they are distributed along in a single dimension) and fail to provide optimal results. It is important to understand that the amount of resources used is proportional to of the vertical distance of the tiers; in fact, considering that the path from a tier to an adjacent involves regular vias going through all metal layers plus one 3D-via, it is clear that any vertical connection larger than adjacency might be too expensive in terms of routing resources and delay.

II. Problem formulation

Consider a random logic circuit netlist and a target 3D circuit floorplan (including area and number of tiers), compute the partitioning of the I/O pins as well as the partitioning of cells into tiers such that the 3D-vias count is minimized; be constrained by keeping a reasonable balance of both, I/Os and cells, along the tiers.

III. Proposed Algorithm Based on Circuit Structure

For this approach, we analyse the random logic block structure and create an I/O partitioning flow. The algorithm firstly calculates the logical distance between pair of I/O pins. Next, it creates a complete graph of I/O pins considering the logical distance as a weight. Finally, it partitions the graph using hMetis and considering the logical distance between I/O pins. The I/O pins are locked to its partitions. Based on the I/O pins location, the cells are partitioned. In the end, simulated annealing is applied to find the best stacking arrangement. The I/O placement preserves the same I/O pins orientation, whitespaces and aspect ratio of the original netlist. This method was named I/O pins. More details can be found in [9] and [11].

IV. Reducing Non-Adjacent 3D-vias

The algorithm presented here is called Refinement and it picks an initial solution and improves it iteratively using random perturbations of the existing solution without any penalty performance. The perturbations might be accepted or rejected depending

on the cost variation. Any perturbation that improves the current state is accepted and all perturbations that increase the cost are rejected. The cost function is divided into three distinct parts: a cost $\boldsymbol{v}$ associated to the usage of 3D-vias resources, a value $\boldsymbol{a}$ for the area balance and finally a cost $\boldsymbol{p}$ for the I/O pins balance. The cost reported is a combination of the three parcels; to be able to add them together, we normalize each parcel by dividing them from their initial values v_i, a_i and p_i (computed before the first perturbation). In addition, we also impose weights (w_v, w_a and w_p) in order to fine tune the cost, as shown in equation 1.

$$c = \frac{(w_v \times v)}{v_1} + \frac{(w_a \times a)}{a_1} + \frac{(w_p \times p)}{p_1} \qquad (1)$$

V. Experimental Results

We used benchmarks IBM ISPD 2004 [10] for our experiments. The proposed algorithms, I/O Pins and Refinement were compared with the state-of-the-art partitioning algorithm, called hMetis (the same approach of [2]).

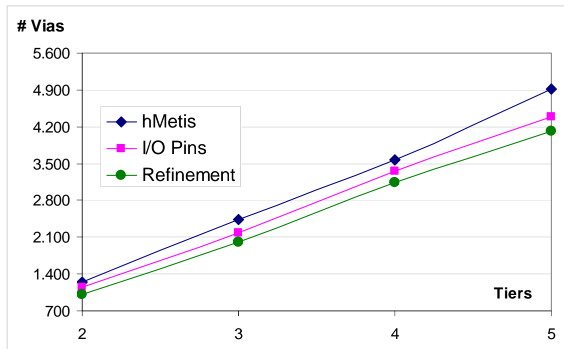

Figure 1: Number of 3D-vias

The Figure 1 shows the average 3D-vias count comparison between the methods. The benchmark circuits were partitioned into two, three, four and five tiers using the evaluated algorithms. The Refinement algorithm obtains the average least amount of 3D-vias. More specifically, Refinement lead to an average 3D-vias count improvement of 19% and 11% compared to hMetis and I/O pins respectively for 2 tiers, 17% and 8% for 3 tiers, 12% and 6% for four and finally 16% and 7% for 5 tiers. For a larger number of tiers Refinement presents a larger improvement when compared to hMetis. This is a direct consequence of the partitioning refinement step that targets at reducing the number of vias in long connections (i. e., connection between non-adjacent tiers), therefore, the larger is the number of tiers the larger is the number of long connections and the improvement obtained by this algorithm. Since partitioning refinement step is done after the partitions have been assigned to the tiers, it takes advantages from the knowledge of the actual partition locations, reducing the number of connections between non-adjacent tiers and increasing the number of connections between adjacent ones. This strategy yields a smaller number of 3D-Vias.

The Figure 2 presents a more detailed look into the vias distribution among the different tiers for the 5-tier configuration. Each bar represents the total number of 3D-vias obtained by each algorithm. The bars are divided into four parts. The lower part represents the number of vias that belong to adjacent connections between tiers, while the others represent 3D-vias in non-adjacent connections.

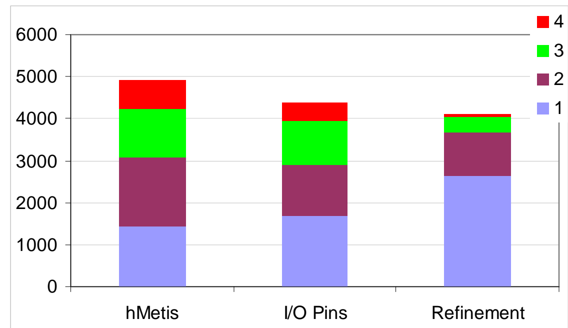

Figure 2: 3D-vias distribution for a 5-tier design

The block identified by the number 2 represents the amount of 3D-vias in connections that are one tier away, i. e., for each connection two 3D-vias are needed. Blocks identified by 3 and 4 represent 3D-vias introduced by connections that are 2 and 3 tiers away respectively. It should be noticed that Figure 2 shows the number of vias that belong to different types of connection, i. e., if a design presents three connections between tiers that are 3 tiers away the number reported in figure 2 is 12, since each connection requires 4 vias. The 3D-vias reduction using the Refinement algorithm was of 804 vias (16%) when compared to the hMetis approach and 280 (6%) when compared to I/O Pins. Experiments using, 2, 3, 4 and 5 tiers were performed and presented the same behavior.

VI. References

[1] W. R. Davis et al. Demystifying 3D ICs: The Pros and Cons of Going Vertical. *IEEE Design and Test of Computers – special issue on 3D integration*; pp 498-510, Nov.-Dec. 2005.

[2] C. Ababei, et al. Placement and Routing in 3D Integrated Circuits. *IEEE Design and Test of Computers – special issue on 3D integration;* pp 520-531, Nov.-Dec. 2005.

[3] B. Goplen; S. Sapatnekar; Efficient Thermal Placement of Standard Cells in 3D ICs using Forced Directed Approach. *In: ICCAD'03*, November, San Jose, California, USA, 2003.

[4] E. Wong; S. Lim. 3D Floorplanning with Thermal Vias. *In: DATE '06*, 2006. p.878–883.

[5] S. Das, et al. Technology, performance, and computer-aided design of three-dimensional integrated circuits. *In: ISPD'04, 2004*, New York, NY, USA. ACM Press, 2004. p.108–115.

[6] R. Patti, Three-dimensional integrated circuits and the future of system-on-chip designs. *Proceedings of IEEE, [S.l.]*, v.94, p.1214–1224, 2006.

[7] R. Hentschke, et al. 3D-Vias Aware Quadratic Placement for 3D VLSI Circuits. *In: IEEE Computer Society Anual Symposium on VLSI, ISVLSI, 2007*, Porto Alegre, Brazil. Los Alamitos: IEEE Computer Society, 2007. p.67–72.

[8] G. Karypis et al. Hypergraph Partitioning: Application in VLSI domain. *In Proceedings of 34th Annual Conference on. Design Automation, DAC 1997*, pages 526–529, 1997.

[9] S. Sawicki, et al. An Algorithm for I/O Pins Partitioning Targeting 3D VLSI Integrated Circuits. *In: 49th IEEE International Midwest Symposium on Circuits and Systems*, 2006, Porto Rico, 2006.

[10] ISPD 2004 - IBM Standard Cell Benchmarks with Pads. http://www.public.Iastate .edu/ ~nataraj /ISPD04 _Bench.html #Benchmark _Description. Access on Mar 2009.

[11] S. Sawicki, et al. "Unbalancing the I/O Pins Partitioning for Minimizing Inter-Vias in 3D VLSI Circuits". *In: IEEE International Conference on Electronics Circuits and Systems, 2006.*