3D-Stacked Integrated Circuits: How Fine Should System Partitioning Be?

Quentin DELHAYE, Dragomir MILOJEVIC
BEAMS, École polytechnique de Bruxelles, ULB
CP165/56, Av. F. Roosevelt, B-1050 Bruxelles, Belgium
qudelhay@ulb.ac.be, dragomir.milojevic@ulb.ac.be

Joël GOOSSENS
PARTS, Faculté des Sciences, ULB
CP212, 50 Av. F. Roosevelt, B-1050 Bruxelles, Belgium
joel.goossens@ulb.ac.be

Abstract—**3D stacked ICs package multiple, independently manufactured dies to reduce total system wire-length, improve timing, and reduce area and power. When designing stacked 3D-ICs, arises the question of the grain at which one should consider system partitioning to optimize the gains. This work uses known MAX-CUT graph partitioning algorithms to split designs from 42k up to 800k gates, with gates clustered from 8 and up to 32768 partitions. It has been found that with 2048 clusters, i.e. 20 to 400 gates per cluster depending on the design, a partitioning of the system allows on average to cut 35% of the nets that account for 73% of the total wire-length in 3D.**

I. INTRODUCTION

While CMOS technologies continue to scale as we enter the era of 3nm transistors [1], the question of Moore's law sustainability still remains open, and is even undermined by the second largest foundry dropping the 7nm node [2]. Many facts could support the previous claim, the following paragraphs will cover just a few.

At device level, transistor gate pitch is not expected to scale much further due to photo-lithography limitations. To still enable area scaling, new type of transistor devices have been proposed (FinFet, nano-wire, etc.) with huge number of device options. These options impact in a great deal the final Power, Performance, Area (PPA) of a design and thus require careful optimization during device selection/configuration and standard cell design. Due to the number of options involved, this optimization, known as Design-Technology Co-Optimization (DTCO) [3], is becoming more and more complex, requiring a lot of research and development effort.

At logic level, alternative gate architectures have been explored, all aiming at reducing the height of standard cells. While this might look appealing at first sight (area reduction of a gate with less scaled transistor area), this also means that the number of routing tracks per gate decreases. Unfortunately, the number of pins per gate remains constant: cell inputs, outputs and power pins are all still needed. Reduced cell height will cause the overall pin density of a design to increase, which in turn will cause increased congestion, causing routability problems. To solve congestion, options are limited: re-design, increase area and/or improve metal layers used for routing. All these solutions come with an extra cost.

Surely it is known that advanced CMOS nodes are becoming evermore expensive, mainly because of multi-patterning techniques used for circuit manufacturing. Furthermore, manufacturing yields can not reach desired figures, thus limiting the die area and preventing cost-effective manufacturing of big dies used in high-performance computing, network processors, graphical processing units, etc.

To address issues linked to 2D CMOS scaling, 3D integration technologies have been proposed in which multiple transistor/gate layers are combined together in the same package. Such integration can be applied on independently manufactured dies (known as 3D-stacked circuits) or at transistor level (monolithic 3D integration).

When compared to 2D, 3D integrated circuits offer: reduced footprint, less wire-length — meaning better system routability — less interconnect parasitics, less interconnect power, less buffer insertion during timing optimization and therefore less total silicon area, less power, better timing, less IR-drop, etc.

Today, 3D stacking technology is mature and already widely used in memories (stacked DRAMs), image circuits (smartphones), high-performance computing (Wide-IO DRAMs) and re-configurable computing (FPGAs). From a technology perspective, high-density 3D interconnects are readily available. Ultimately, monolithic 3D integration will enable even more 3D interconnect in the near future.

Systems that already use 3D technology have a very important characteristic: *system partitioning decision* (i.e. what block should go where) is known in advance as it happens at coarse functional level (coarse-grain partitioning). As the dimension and pitch of 3D structures scales, the system partitioning question becomes more complex since it can happen at lower functional levels, i.e. smaller sets of gates called gate-clusters. Ultimately, in monolithic 3D integration the gate-cluster can be a single logic gate. For such systems, automation of partitioning decision becomes mandatory.

A. Contributions and organization

Our main contributions are:

- The study of an optimal *partitioning grain*.
- Correlation between MAX-CUT partitioning and 3D nets.

While other approaches focus on TSV placement or heat-aware partitioning [4], this work studies at what size of gate-cluster system partitioning should be performed to maximize PPA gains of a 3D system. This will be called the *partitioning grain* further on. Using (hyper)graph partitioning algorithms and various designs, it will be shown that a very fine grain is not required to provide the best possible gains in 3D. The potential benefits are converging as gate-cluster size decreases, in a similar way and for different system architectures.

The paper is organized as follows: Section II is a description of the problem, prior work and introduces graph partitioning as a potential solution for 3D system design; Section III presents an experimental framework to study the relationship between gate-cluster size and the amount of intra- (2D) vs. inter-cluster (3D) wire-lengths; obtained results are analyzed in Section IV; finally conclusions are drawn in Section V.

II. Problems and solutions

A. Problem statement and existing solutions

As of today there is no full support for 3D circuit integration in commercial EDA tools, especially native 3D placement and routing (P&R). Rather, various extensions to existing 2D P&R have been proposed in academia and industrial R&D [5]. An extension of such approach with improved 3D PPA has been proposed in [6].

All the aforementioned design flows relay either on manual system partitioning, or they automate the partitioning decision for monolithic 3D integration (gate-cluster size of 1). What remains unclear is what happens between this ultimate partitioning grain and more coarser grain partitioning.

It has been argued that 3D-stacking was not practical for fine-grained 3D partitioning [7] due to the size of Through Silicon Vias (TSVs), and that a monolithic 3D-integration was the way to go. However, monolithic 3D still needs to solve some key showstoppers such as thermal budgets required for sequential Front End Of Line processing (i.e. transistor manufacturing). Meanwhile, Face-to-Face hybrid bonding seems to offer a nice compromise between the amount of 3D interconnect and potential PPA gains. With millions of 3D interconnects per mm^2 and no area penalty for the 3D structure (as opposed to Face-to-Back approach using TSVs) this technology seems like a serious contender for fine grain partitioned logic-on-logic systems.

B. Optimization objectives

When designing a 3D system, the obvious question is the partitioning decision: what should go where. To do this, a 3D system implementation tool could consider various optimization objectives: (1) number of 3D nets, (2) total 3D system wire-length, (3) total interconnect power, (4) longest 3D net and/or critical path, etc. Note that these objectives could be taken separately or all together in a multi-objective optimization approach.

Objective (1) is essential since it is driven by the pitch of the 3D connection given by the 3D technology: the finer the pitch, the more expensive the technology. Obviously, cost-effectiveness of the 3D design and potential PPA gains need to be matched. Objective (2) and (3) need to be analyzed to understand the overall gains of 3D, while objective (4) focuses on critical path and thus the system performance only. 3D implemented systems might reduce the total wire-length by bringing the gates closer to each other.

However, the 3D re-routed nets have to pass through the interface between the two gate layers and thus should not worsen the critical path. If not taken care of, particularly short nets could become longer in 3D, thus degrading their performance and requiring additional buffering and potentially killing all the benefits of 3D integration.

Section IV will study the impact of objectives (1) and (2).

C. Clustering

Clustering consists in grouping gates together to fulfill different objectives such as highlighting natural clusters [8], reduce the total amount of gates [9], or improve the 3D partitioning [10]. However thorough the literature is on the clustering algorithm, it scarcely shows interest for the optimal grain, or even its influence at all. The choice of the clustering method certainly is a major step in any EDA workflow, but it can be argued that its grain should matter all the same.

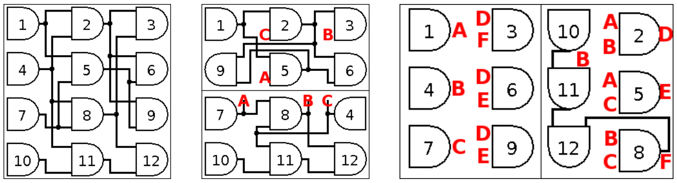
Section III-A will present the divisive hierarchical method used in this work and for which the clustering level impact will be studied.

D. Circuit partitioning using graphs

Digital integrated circuits are made out of logic gates placed in a 2D plane and connected by wires. Placed circuits can be represented with hypergraphs $\mathcal{H} = (V, H)$, where each hyperedge $h \in H$ (the gates or gate-clusters) is a subset of fully interconnected vertices $v \in V$ (the nets). Both V and H are weighted, e.g. with the cluster size and the net length, respectively. Since some algorithms can not be applied to hypergraphs, it is sometimes needed to transpose a hypergraph into its underlying graph $\mathcal{G}$. This is done by keeping the same set of vertices V, and by replacing each hyperedge with a complete subgraph in which all the vertices are connected to each other, yielding a set of edges E. This transposition gives a new graph $\mathcal{G} = (V, E)$. While such transformations have been criticized due to loss of accuracy [11], this approximation is acceptable for our purpose because the fan-out of the nets is small compared to total number of nets.

Graph partitioning algorithms have been exhaustively studied in the literature. The situation is stated as follows: For a given partitioned graph, let's call E_c the set of cut edges spanning across the partition. The problem is then to find a partition so that the cut size, the sum of the weights w_n of E_c, i.e. $\sum_{n \in E_c} w_n$, is minimized (MIN-CUT) or maximized (MAX-CUT). MIN-CUT has many implementations such as `hMETIS` [12] or `PaToH` [13]. As for MAX-CUT, [14] is an instance of its implementation and is packaged into `CirCut`.

MIN-CUT algorithms have already been used when designing 2D systems to limit the number of wires in the upper metal layers of the wiring stack during standard cell placement [15]. Therefore it could seem logical to apply the same methods to 3D. Whether this would make sense would depend on the partitioning grain and the pitch of the 3D connection. Our results showed that no matter the size of the gate-cluster,



(a) (b) (c)

Fig. 1. Partitioning of a mock design: (a) original 2D, (b) MIN-CUT partitions, (c) MAX-CUT partitions. The letters show the 3D connections: 3 for MIN-CUT and 6 for MAX-CUT.

MIN-CUT produces the same partition already decided by the 2D placement tool. However, if one goes the other way and applies a MAX-CUT algorithm on a placed 2D design, one can highlight interesting correlations between the amount of nets cut and the total 3D wire-length. Those results will be further presented in Section IV.

Both MIN-CUT and MAX-CUT produce balanced partitions on vertices weights. If the vertex weight is assumed to be its gate-cluster area, partitions will result in 50-50% die area split. This can be seen as a plus, since hybrid bonding can be executed at wafer level (Wafer-on-Wafer) for increased manufacturing throughput and thus lower price. Such 3D integration technique requires 50-50% die areas on both wafers to minimize losses.

Figure 1 shows the result of a MIN-CUT and a MAX-CUT balanced partitioning a simplistic design.

III. Experimental Setup

To study the impact of the partitioning grain on potential wire-length gains of the 3D design with respect to 2D implementation, a software tool chain has been developed to process placed and routed designs and partition netlists for 3D integration using the graph based methods described in Section II-D. Our tool takes as input a 2D placed and routed design (`DEF` file), and geometrical views of standard cells used (`LEF` file) to build a design data base. Following operations are then performed: (1) Input files are parsed to extract the design; (2) Standard cells are clustered, so that each cluster represents a graph vertex (V) where its weight represents the total cluster area (see Section III-A); (3) Inter-cluster nets are extracted to generate hyperedges (H) and a weight function is calculated for each (i.e. the amount of nets in each hyperedge); (4) hypergraph $\mathcal{H}$ is built; (5) and then partitioned using a) MIN-CUT partitioning (using `hMETIS` with various parameters) to minimize the cut size, or b) MAX-CUT (using `CirCut`) to maximize the number of inter-die (or 3D) connections. `CirCut` is used with the default parameters, i.e. a threshold at 10^{-4}, maximum 200 iterations and balanced partitions. As discussed in Section II-D, MAX-CUT results will be the focus further on.

As 3D P&R is not present in this work, the analysis will be limited to the inter-cluster connectivity.

TABLE I
Designs data base

Design	Gates	Nets	Wire-length (normalized)
D1	42471	49633	2101
D2	121580	137171	2050
D3	185777	200999	2860
D4	220587	234373	4318
D5	289812	306118	5312
D6	694082	773679	12606
D7	808199	883295	16722

A. Gate-clusters

To generate gate-clusters, a divisive hierarchical clustering [16] based on the geometry of the design is used: top-level design is first split vertically into two parts of the same size, then each part is split again, but horizontally. This process is repeated recursively — subsequent vertical and horizontal splits — until the targeted amount of clusters is reached. In these experiments, cluster sizes from 2 to 32768 in power-of-2 steps have been considered. The largest number of clusters (i.e. the smallest cluster size) has been picked so that the gate-count for the largest design does not go below 25 gates.

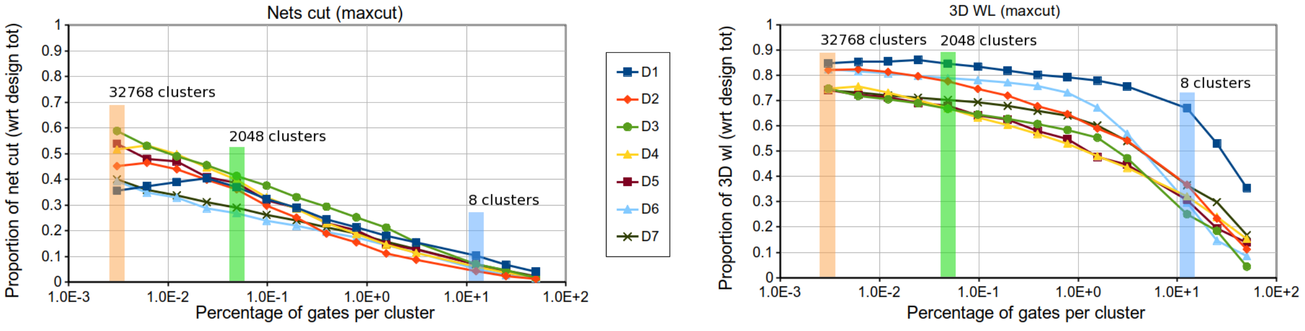
B. Designs data base

For these experiments, the following designs have been considered: D1) an open source implementation of low-parity density parity check; D2) BoomCore: *Berkeley Out-of-Order Machine*, an open source implementation of the RISC-V micro-processor; D4) MCC: a medium complexity core; D3) & D5) respectively a crossbar and core of the OpenSparc T2 SoC; D6) an SoC with 16 processing elements that are only locally connected (daisy chain); and D7) an SoC with 16 fully connected processing elements (each processing element is connected to all the others). Table I summarizes key design parameters: gate count, the total number of nets and total design wire-length after P&R using and open-source PDK. Note that the total wire-length is normalized to the design half-perimeter.

IV. Results

After graph partitioning with MAX-CUT, the number of nets that were cut (inter-cluster or 3D nets) and their length (Fig. 2) is analyzed. Measures are given for each level of clustering as a percentage of gates per cluster, e.g. for 100 clusters, each cluster hosts on average 1% of all the gates in the design. Since all clusters have the same size and the distribution of the gates after placement is roughly homogeneous, it seems fair to admit that all clusters host the same amount of gates. Consequentially, an increasing number of clusters, a finer clustering grain and a reduction of number of gates per cluster all have the same meaning (going from right to left in the graphs).

Fig. 2(a) shows the proportion of nets cut by the partitioning with respect to the total amount of nets in the design. Except occasional outliers, especially for design D1 at 1.0E-2 point, as



(a)

(b)

Fig. 2. MAX-CUT partitioning results: Number of cut (3D) nets (a) and Ratio of total 3D wire-length (b) as function of gate-cluster size; Vertical bars corresponding to 8, 2048 and 32768 gate-clusters mark different zones in the graph

TABLE II
MAX-CUT RESULTS FOR THREE CLUSTER SIZES

	Percentage of nets cut			Percentage of 3D WL		
Clusters	8	2048	32768	8	2048	32768
Average	7%	35%	49%	37%	73%	78%
Std dev	2%	5%	7%	13%	6%	4%

the gate-count in clusters reduces, the growth is exponential from a very coarse to the finest clustering grain. Fig. 2(b) shows the proportion of 3D wire-length with respect to the total wire-length of the design. Two stages can clearly be highlighted: a first exponential growth of 3D wire-length when there are few clusters, and then a second, this time linear growth leading towards a *convergence region*.

An increasing number of 3D nets will translate into tighter, if not unfeasible, 3D pitch. Therefore it is essential to find a good compromise between the number of 3D nets and the wire-length they represent. Because of the *convergence region*, it is possible to define a sweet spot where the 3D wire-length is maximized, for a minimum number of 3D nets.

Such sweet spot can be set around the 0.05% of gates-per-cluster mark. This value corresponds to 2048 clusters and means that by cutting from 27% to 41% of all available nets, the total inter-cluster wire-length is in the range between 67% and 84% of the total wire-length, depending on the considered design. Table II summarizes this result for all designs in our data base and this is the region where the 3D partitioning should occur. Beyond this sweet spot, the total inter-cluster wire-length does not change much. At this level of clustering, it would be possible to capture most of the inter-cluster wire-length. Refining the grain further to eventually reach a single gate per cluster (i.e. monolithic 3D-integration) will not necessarily improve the system PPA.
Figure 3 shows that for most designs, the nets cut by the partition have a length between 0.1% to 10% of the longest net cut. In particular it highlights the fact that the shortest nets are a minority, hence limiting the drop of performance when rerouting those in 3D.

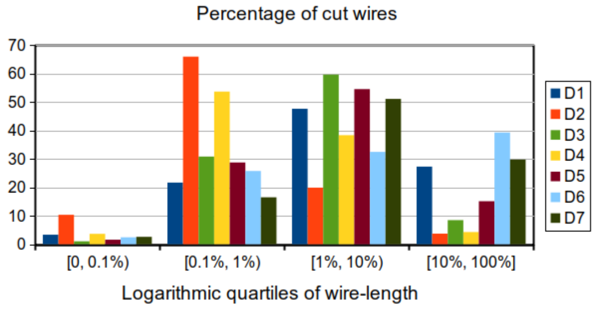


Fig. 3. Distribution of the percentage of cut wires per logarithmic quartile of their wire-length relative to the longest net, for 2048 clusters.

V. CONCLUSION

For quite some time 3D integrated circuits have been considered as a potential solution to extend Moore's law. As the pitch of the 3D structures scales allowing millions of connections per mm^2, the question on how, and especially at what grain, system partition should be executed when building a 3D system is becoming critical. This paper has highlighted the existence of a sweet spot for the clustering grain (around 2048 clusters) which maximizes the amount of inter-cluster wire-length with respect to the total system wire-length, while limiting the inter-tier nets. The existence of this sweet spot reveals that in order to maximize the PPA benefits of 3D integration technology, it is not needed to consider the circuit partitioning at very fine grain.

In future work, other partitioning algorithms or tools could considered, but the results can be expected to be very similar. Indeed, the partitioning scheme only influences the quality of the cut which can vary by a few percents without any lost of generality in our conclusion. However, the clustering plays a more fundamental role in this paper. As such, alternative clustering methods will be explored in order to further reduce the amount of short nets in the partitioning cut and further optimize the partitioning decision, including multi-criteria optimization methods.

REFERENCES

[1] "Imec and Cadence Tape Out Industry's First 3nm Test Chip," https://www.allaboutcircuits.com/news/world-first-3nm-tapeout-lithography-Cadence-Design-Systems-Imec/, accessed: 2018-10-15.

[2] "GlobalFoundries Halts 7-Nanometer Chip Development," https://spectrum.ieee.org/nanoclast/semiconductors/devices/globalfoundries-halts-7nm-chip-development/, accessed: 2018-10-15.

[3] L. Mattii, D. Milojevic, P. Debacker, Y. Sherazi, M. Berekovic, and P. Raghavan, "IR-drop aware Design & technology co-optimization for N5 node with different device and cell height options," *IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, ICCAD*, vol. 2017-November, pp. 89–94, 2017.

[4] K. Athikulwongse, M. Ekpanyapong, and S. K. Lim, "Exploiting die-to-die thermal coupling in 3-D IC placement," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 10, pp. 2145–2155, 2014.

[5] S. a. Panth, K. Samadi, Y. Du, and S. K. Lim, "Design and CAD methodologies for low power gate-level monolithic 3D ICs," *Proceedings of the 2014 international symposium on Low power electronics and design - ISLPED '14*, vol. 1, pp. 171–176, 2014.

[6] K. Chang, S. Sinha, B. Cline, R. Southerland, M. Doherty, G. Yeric, and S. K. Lim, "Cascade2D: A design-aware partitioning approach to monolithic 3D IC with 2D commercial tools," *Proceedings of the 35th International Conference on Computer-Aided Design - ICCAD '16*, pp. 1–8, 2016.

[7] S. K. Samal, D. Nayak, M. Ichihashi, S. Banna, and S. K. Lim, "Monolithic 3D IC vs. TSV-based 3D IC in 14nm FinFET technology," *2016 SOI-3D-Subthreshold Microelectronics Technology Unified Conference, S3S 2016*, pp. 2–3, 2017.

[8] J. Li and L. Behjat, "A Connectivity Based Clustering Algorithm With Application to VLSI Circuit Partitioning," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 53, no. 5, pp. 384–388, 2006.

[9] G. Moura, F. Pisoni, and R. Reis, "A Cell Clustering Technique to Reduce Transistor Count," in *Electronics, Circuits and Systems (ICECS), 2017 24th IEEE International Conference on*, 2017, pp. 186–189.

[10] K. Han, A. B. Kahng, J. Li, and U. C. S. Diego, "Improved Performance of 3DIC Implementations Through Inherent Awareness of Mix-and-Match Die Stacking," pp. 61–66, 2016.

[11] E. Ihler, D. Wagner, and F. Wagner, "Modeling hypergraphs by graphs with the same mincut properties," *Information Processing Letters*, vol. 45, no. 4, pp. 171–175, 1993.

[12] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar, "Multilevel hypergraph partitioning: Applications in VLSI domain," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 7, no. 1, pp. 69–79, 1999.

[13] Ü. Çatalyürek and C. Aykanat, "PaToH (Partitioning Tool for Hypergraphs)," pp. 1–9, 2011.

[14] S. Burer, R. D. C. Monteiro, and Y. Zhang, "Rank-Two Relaxation Heuristics for Max-Cut and Other Binary Quadratic Programs," *SIAM Journal on Optimization*, vol. 12, pp. 503–521, 2000.

[15] A. B. Kahng, J. Lienig, I. L. Markov, and J. Hu, *VLSI physical design: From graph partitioning to timing closure*. Springer Netherlands, 2011, vol. 25, no. 9.

[16] L. Rokach and O. Maimon, "Clustering methods," *Data mining and knowledge discovery handbook*, pp. 321–352, 2005.