Architectural Exploration of 3D FPGAs towards a Better Balance between Area and Delay

Chia-I Chen, Bau-Cheng Lee, and Juinn-Dar Huang
Department of Electronics Engineering and Institute of Electronics
National Chiao Tung University, Hsinchu, Taiwan
cichen.ee94g@nctu.edu.tw, bclee@adar.ee.nctu.edu.tw, jdhuang@mail.nctu.edu.tw

***Abstract*—The emerging 3D technology, which stacks multiple dies within a single chip and utilizes through-silicon vias (TSVs) as vertical connections, is considered a promising solution for achieving better performance and easy integration. Similarly, a generic 2D FPGA architecture can evolve into a 3D one by extending its signal switching scheme from 2D to 3D by means of TSVs. However, replacing all 2D switch boxes (SBs) by 3D ones with full vertical connectivity is found both area-consuming and resource-squandering. Therefore, it is possible to greatly reduce the footprint with only minor delay increase by properly tailoring the structure and deployment strategy of 3D SB. In this paper, we perform a comprehensive architectural exploration of 3D FPGAs. Various architectural alternatives are proposed and then evaluated thoroughly to pick out the most appropriate ones with a better balance between area and delay. Finally, we recommend several configurations for generic 3D FPGA architectures, which can save up to 52% area with virtually no delay penalty.**

Keywords-3D ICs*; *3D FPGAs*; *architectural exploration*; *area/delay trade-off

I. INTRODUCTION

To keep up with the Moore's law, the emerging three-dimensional (3D) technology seems to provide a promising solution while the manufacturing technology for feature size shrinking is facing the ultimate physical limitations. Hence the topics regarding 3D ICs become very popular both in academia and industry in recent years. The through-silicon via (TSV) technology is considered one of the most promising solutions for 3D integration. A TSV-based 3D IC stacks multiple dies on a single chip and uses inter-die vias for vertical connections, which provide shorter global interconnects, lower interconnect power, smaller footprint and better heterogeneous integration [1]. Therefore, in addition to those advanced processor and memory architectures, ASIC and FPGA designs are also stepping into the 3D era.

There are two major types of 3D FPGA architectures found in the literature. The first one is constructed by monolithically stacking distinct dies [2]. The second one is evolved from the original 2D structure by extending the 2D switch boxes (SBs) to 3D ones [3]–[5]. So far, there are two synthesis frameworks targeting the latter 3D FPGA architecture: the three-dimensional place and route (TPR) [3][4] and 3D MEANDER [5]. To the best of our knowledge, TPR should be the first synthesizer for 3D FPGAs. In TPR, all SBs are assumed 3D-SBs and the number of available TSVs in a 3D-SB is assumed unlimited, which is surely impractical. Meanwhile, according to [3][4][6], the SB has already been the most area-consuming unit compared to the other elements in 2D FPGAs for a long time. The situation is becoming even worse in 3D FPGAs because the 3D-SB is exactly where those TSVs locate. As shown in Fig. 1, as manufacturing technology keeps scaling down, the area share of the 3D-SB is getting more dominant, which is mainly because TSVs are not scaled well. Moreover, it is found that the TSV utilization is actually quite low if the 3D-SB with full vertical connectivity is in use. As depicted in Fig. 2, the TSV utilization is still less than 10% even in the 8-layer 3D FPGA. Therefore, there is a strong motive to optimize the 3D-SB structure and the 3D-SB deployment strategy for area saving. Meanwhile, 3D MEANDER is another design framework for 3D FPGAs. It further studies the impact of different deployment strategies for 3D-SBs. It proposes a family of 3D FPGA architectures in which 2D-SBs and 3D-SBs are mixed up in certain regular spatial patterns. However, the number of available TSVs within a 3D-SB is assumed fixed in 3D MEANDER. That is, it does not investigate what the impact of the different number of TSVs in a 3D-SB is.

In this paper, we first point out that the utilization of TSVs is actually very low in a 3D FPGA architecture with full vertical connectivity, which inspires us to discover new architectures that can achieve a better balance between area and delay. There are basically two approaches for area reduction:

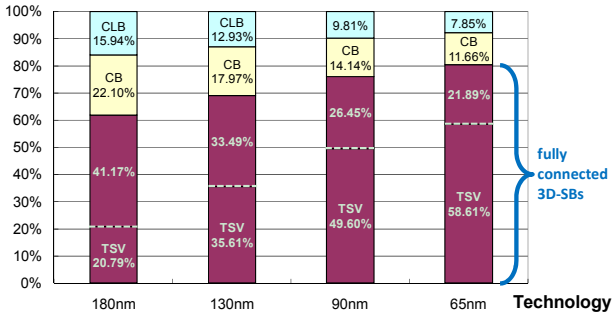

Figure 1. The area ratio for each component in a tile.

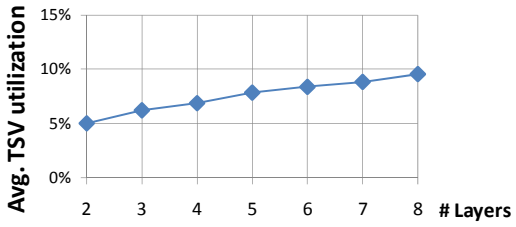

Figure 2. The average TSV utilization in a 3D FPGA architecture with full vertical connectivity.

This work was supported in part by the National Science Council of Taiwan under Grant NSC 99-2220-E-009-037

978-3-9810801-7-9/DATE11/©2011 EDAA

reducing the number of TSVs in a 3D-SB, and reducing the number of 3D-SBs. In combination of the above two ideas, we propose two major families of architectures and extensively evaluate numerous instances through thorough and systematic comparisons to pick out better ones. Note that minimizing area only is one thing, but minimizing area without sacrificing delay is completely another thing.

The rest of this paper is organized as follows. In Section II, the preliminaries including evaluation environment are briefly introduced. Section III and IV present our two major families of 3D FPGA architectures. The selected architectures with a better area/delay balance are thus recommended in Section V. Finally, the concluding remarks are given in Section VI.

II. PRELIMINARIES

A. 3D FPGA Architectures and the Synthesis Framework

A classic 2D FPGA is composed of logic elements (configurable logic blocks, CLBs) and routing resources (switch boxes, SBs; connection boxes, CBs). The basic unit is called a *tile*, which consists of a CLB, an SB, and associated CBs. The generic 3D FGPA architecture in this work is similar to the one used in TPR and 3D MEANDER, which basically stacks multiple identical 2D FPGA layers and then extends the structure of 2D-SB to provide inter-layer connectivity.

The reference synthesis flow used in this work is shown in Fig. 3. A given design (in blif format) is first packed into a netlist file composed of CLBs by T-VPack, which is a part of the 2D FPGA synthesis framework VPR [7]. The netlist file is then fed into a 3D FPGA synthesizer, which includes three steps: initial layering [8], timing-driven 3D placement and 3D routing, where the latter two are adapted from TPR. The framework also takes another input file, setting the architectural parameters of the target 3D FPGA, and generates the placement and routing results in the end.

B. Evaluation Environment

The architectural parameters are mostly set according to existing commercial FPGAs, well-known FPGA synthesizers, and related research works. The number of LUTs (lookup tables) in an CLB (N) is set to 2 instead of 1 in the previous works [3][4], which is considered more realistic. The channel widths in the X-Y directions are both set to 32; the multi-segment routing structure is adopted: the possible wire lengths are *L1*, *L2*, *L4* and *L8*; and the numbers of channels are 12, 12, 4 and 4, respectively. In the vertical direction (Z), only *L1* is available in order to maximize the routability. Finally, I/O pads are located only at the bottom-most layer.

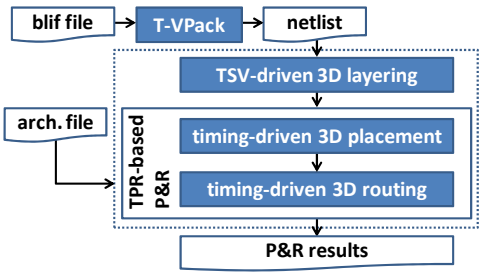

Figure 3. The reference synthesis framework.

There are 24 test cases in our benchmark set – 14 are from MCNC and the other 10 larger ones are from IWLS2005, ITC99 and Altera [9][10]. All test cases are preprocessed through T-VPack and the size of test case ranges from 3224 to 15390 CLBs.

In the baseline (BSL) architecture, every SB is a 3D-SB; and every 3D-SB is with *full vertical connectivity*, which means there are 32 TSVs (identical to the number of channels in the X-Y direction) within a fully connected 3D-SB. In the BSL, though the vertical routability is maximized (which is best for delay), the TSV utilization is extremely low (< 10%) as indicated in Fig. 2. This fact also suggests there is still a big room for further architecture improvements. Hereafter in this paper, the BSL would serve as a baseline for comparisons with the proposed architectures in the next two sections.

III. SPARSE ARCHITECTURES

Basically there are two ways to reduce the area occupied by 3D-SBs in a 3D FPGA. The first one is to decrease the number of TSVs in a 3D-SB. Such kinds of 3D-SBs are called *partially connected* 3D-SBs. The other one, also used in 3D MEANDER, is to reduce the number of 3D-SBs in a 3D FPGA. That is, some SBs become 3D ones, and others are still 2D ones. The *internally-spare* (*IS*) architectures are those utilizing the former method, while the *externally-sparse* (*ES*) ones are those utilizing the latter method. Different configurations of SBs, also referred as *patterns* in this work, are shown in Fig. 4. In the BSL, as shown in Fig. 4(a), all SBs are fully connected 3D-SBs. An IS architecture adopts the same type of partially connected 3D-SBs for all SBs instead, as depicted in Fig. 4(b). In an ES architecture, as shown in Fig. 4(c), fully connected 3D-SBs are partially distributed in a regular fashion. Note that the patterns are set identical for all layers in all proposed architectures. In addition to IS and ES architectures, the *sparse architecture* (*SP*), which is a hybrid of the above two, is proposed to further reduce the area. More details and evaluations are available in the following sub-sections.

A. Internally-Sparese Architecture (IS)

IS# represents an instance of the IS architecture, where the postfix # specifies the number of TSVs available in a 3D-SB. For example, every 3D-SB in *IS16* has only 16 TSVs inside, while *IS32* is actually equivalent to the BSL. As mentioned in Section II-B, the multi-segment routing structure is adopted

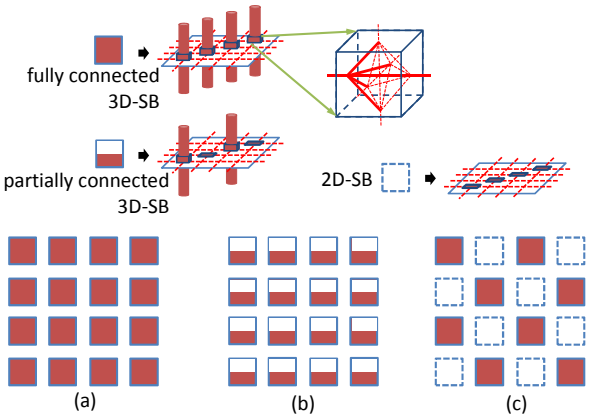

Figure 4. SB patterns of (a) BSL, (b) IS, and (c) ES.

and there are four different wire lengths with different amount of channels in the X-Y direction. In the BSL, it makes no differences since each X-Y channel has its own vertical connection. However, in an IS architecture, a part of TSVs in a 3D-SB are removed. Be more specific, the vertical connections (TSVs) are taken out from each type of X-Y channels proportionally whenever possible.

For *IS*, the sizes of an SB and a tile basically decrease linearly as the number of TSVs in a 3D-SB decreases. However, though reducing the number of TSVs in a 3D-SB can save area, it is likely to increase delay at the same time. To realize what the exact impact on delay is, the benchmark circuits are mapped onto different IS architectures and the BSL through the reference synthesis framework. The results show that some test cases fail to be mapped onto *IS12* due to insufficient vertical routing channels. Most of the test cases fail in *IS8*; one even succeeds, the delay increases significantly. Finally, since *IS20* achieves about 30% overall area reduction only with a delay penalty less than 1.5% as compared to the BSL, it should be the one with a better area/delay balance among all IS architectures.

B. *Externally-Sparse Architecture (ES)*

An ES architecture replaces a part of fully connected 3D-SBs with 2D-SBs for area saving. There are various ways, or *patterns*, for mixing 2D-SBs and 3D-SBs. The pattern used in the ES architecture is *oblique stripes*, which is the same as the one used in 3D MEANDER. *ES#* represents an instance of the ES architecture, where the postfix # specifies the maximum *distance* between two adjacent 3D-SBs in either X or Y direction. Therefore, *ES1* is actually equivalent to the BSL, and *ES2* is the one shown in Fig. 4(c).

For *ES*, as the distance increases, the area reduction is significant at first and then becomes flat gradually, appearing like a harmonic sequence. As the distance increases, the delay is increased badly. As a result, *ES2*, *ES3* and *ES4* are all regarded as good instances since they can achieve around 35%~55% area reduction only with a delay penalty less than 5%. Moreover, if the delay penalty is constrained below 3%, *ES2* becomes the best choice because it achieves an area reduction of 35% with a delay penalty less than 1.5%.

C. *Sparse Architecture (SP)*

For IS and ES architectures, we try to reduce the area with just a single strategy – either purely reducing the number of TSVs in each 3D-SB or purely reducing the number of 3D-SBs. Here we present the *sparse architecture*, which takes above two strategies simultaneously. *SP*($\#_1$, $\#_2$) is used to name a specific instance of this hybrid architecture, which is the combination of *IS*$\#_1$ and *ES*$\#_2$. This notation can be generalized to represent IS and ES architectures as well. For example, *SP*(32, 1) is equivalent to the BSL, *SP*(16, 1) implies *IS16*, and *SP*(32, 2) is actually *ES2*. Meanwhile, the *TSV density* of an architecture is defined as the average number of TSVs in a tile, and can be calculated through $\#_1$ divided by $\#_2$.

As mentioned, *ES4*=*SP*(32, 4) with TSV density of 8 is still regarded as a good instance. Note that, the lower the TSV density is, the worse the delay is likely to be. Hence, only those SP architectures with TSV density no less than 8 are selected for evaluation. Fig. 5 shows the area of selected SP architectures, in which all values are normalized to that of the BSL. As expected, the lower the TSV density is, the smaller the area is. Fig. 6 reports the delay of same selected SP architectures, in which every value represents the average of normalized-to-the-BSL delay over the entire benchmark set, and conforms that the lower the TSV density is, the worse the delay is. Finally, *SP*(28, 2), *SP*(24, 2), and *SP*(20, 2), are our recommended SP architectures under the constraint that the delay penalty cannot exceed 3%.

IV. SUNNY EGG ARCHITECTURE

Though *SP*(20, 2) has already achieved an area reduction of 50% with minor delay loss, there is still room for improvement. Fig. 7 shows the average TSV distribution in a 6-layer BSL over 6 test cases. The results suggest that the TSV demand is much bigger in the central region than that in the peripheral zone; and thus inspire us to further propose the *sunny egg* (*SE*) architecture.

The sunny egg architecture divides a horizontal plane into two regions – center (egg yolk) and periphery (egg white). Two regions are implemented using different SP architectures – the

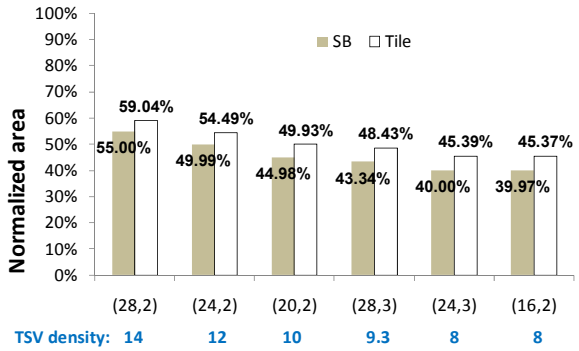

Figure 5. The normalized area of different SP architectures.

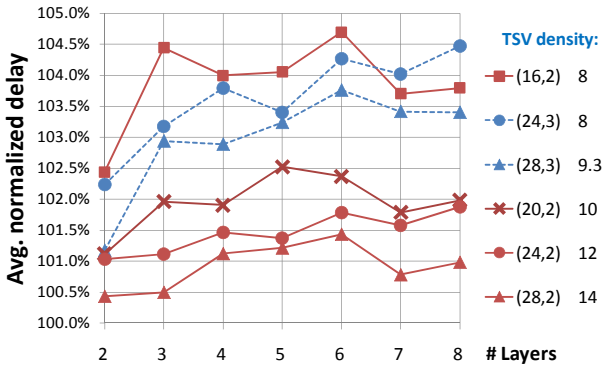

Figure 6. The average normalized delay of different SP architectures.

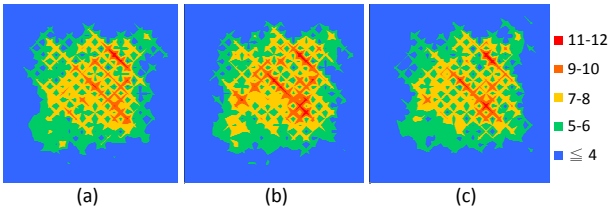

Figure 7. The average TSV utilization in a 6-layer BSL. Between: (a) Layer 2 and 3, (b) Layer 3 and 4, and (c) Layer 4 and 5.

TSV density in the center is set larger than that in the periphery. *SE*(IS_C#, ES_C#, *R*, IS_P#, ES_P#) indicates a specific SE architecture, where *SP*(IS_C# and ES_C#)/*SP*(IS_P# and ES_P#) is for the center/periphery respectively, and *R* is the ratio between the dimension of the center and X-Y plane.

A total of 20 different SE architectures are evaluated in this study. They are generated in the following way: 2 SP instances with higher TSV densities (*SP*(32, 2), *SP*(16, 1)) for the center, 2 SP instances with lower TSV densities (*SP*(16, 4), *SP*(8, 2)) for the periphery, and *R* ranges from 0.3 to 0.7. Meanwhile, given the same TSV density, *SP*(16, 1) distributes TSVs more *evenly* than *SP*(32, 2) in the center. Similarly, *SP*(8, 2) distributes TSVs more evenly than *SP*(16, 4) in the periphery. Fig. 8 demonstrates how the different TSV distributions can impact the delay. First, it suggests that it is better to distribute TSVs *unevenly* in the center, i.e., *SP*(32, 2) is preferred. Second, the delay is slightly better if TSVs are evenly distributed in the periphery, i.e., *SP*(8, 2) is preferred. The ratio *R* also does matter. As mentioned previously, the TSV density in the center is always larger than that in the periphery. Hence, the bigger the ratio *R* is, the larger the footprint is. Meanwhile, the delay gets better as *R* increases due to richer routing resources. Consequently, as usual, it is still a trade-off between delay and area. Again, to keep the delay penalty less than 3% compared to the BSL, *R* must be set to higher than 0.6. As a result, the best choice here should be *SE*(32, 2, 0.6, 8, 2).

V. Recommended Architectures

In this section, all the architecture styles mentioned previously are reviewed. In each architecture style, the one with the smallest area while still satisfying the delay constraint (i.e., the delay penalty must be less than 3% as compared to the BSL) is selected as the representative. Fig. 9 shows the area of these four representative architectures (*IS20*, *ES2*, *SP*(20, 2), *SE*(32, 2, 0.6, 8, 2)), in which all values are normalized to that of *IS20*. The average normalized delay to the BSL is given in Fig. 10. It is found that *SP*(20, 2) can save 31% area only with about 2% delay increase as compared to *IS20*. In addition, *SE*(32, 2, 0.6, 8, 2) can improve the area by another 4% and even with better delay as compared to *SP*(20, 2). In summary, above two are the most recommended architectures at last.

VI. Conclusion

In this paper, we first show that the utilization of TSVs can be extremely low (< 10%) in the baseline architecture where all SBs are fully connected 3D-SBs. Then we present two architectures for area (TSV) reduction: the internally-sparse (IS) architecture through reducing the number of TSVs in each 3D-SB, and the externally-sparse (ES) one through reducing the number of available 3D-SBs. In combination of IS and ES, we further develop the hybrid sparse (SP) architecture. In the end, we propose the ultimate sunny egg (SE) architecture, which incorporates two different SP architectures with different TSV densities in the central and peripheral region respectively. After extensive evaluations over all architecture styles, two generic 3D FPGA architectures are most recommended, which maximize the area reduction (up to 52%) with an acceptable delay penalty (< 3%). Therefore, we believe that all the ideas and architecture styles revealed in this paper can serve as a robust foundation for developing even more practical 3D FPGA architectures.

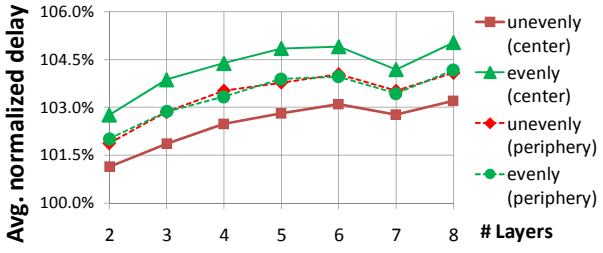

Figure 8. The average normalized delay with different TSV distributions.

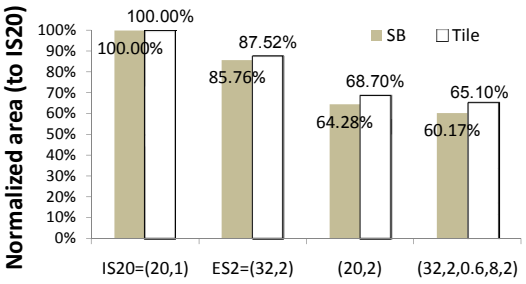

Figure 9. The normalized area of representative architectures.

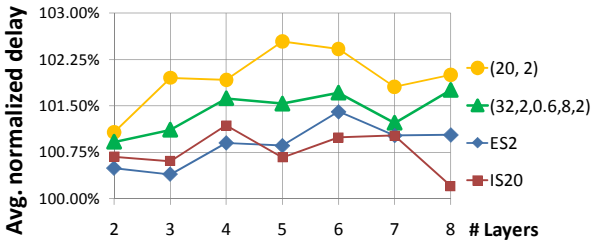

Figure 10. The average normalized delay of representative architectures.

References

[1] W. R. Davis et al., "Demystifying 3D ICs: the pros and cons of going vertical," *IEEE Design and Test of Computers*, vol. 22, no. 6, pp. 498–510, Nov.–Dec., 2005.

[2] M. Lin, A. El Gamal, Y.-C. Lu, and S. Wong, "Performance benefits of monolithically stacked 3-D FPGA," *IEEE Trans. Computer-Aided Design Integrated Circuits and Systems*, vol.26, no.2, pp.216–229, Feb. 2007.

[3] C. Ababei, H. Mogal, K. Bazargan, "Three-dimensional place and route for FPGAs," *IEEE Trans. Computer-Aided Design Integrated Circuits and Systems*, vol.25, no.6, pp.1132–1140, Jun. 2006.

[4] TPR: three-dimensional placement and route for FPGAs. [Online]. Available: http://www.ece.umn.edu/users/kia/mount/Download/

[5] K. Siozios, A. Bartzas, and D. Soudirs, "Architecture-level exploration of alternative interconnection schemes targeting to 3D FPGAs: a software-supported methodology," *Int'l Journal of Reconfigurable Computing*, vol. 2008, Article ID 764942, 2008.

[6] International Technology Roadmap for Semiconductor. Semiconductor Industry Association 2009.

[7] VPR: versatile packing, placement and routing for FPGAs. [Online]. Available: http://www.eecg.toronto.edu/~vaughn/vpr/vpr.html

[8] Y.-S. Huang, Y.-H. Liu, and J.-D. Huang, "Iterative 3D partitioning for through-silicon via minimization," *Proc. 16th Workshop on Synthesis and System Integration of Mixed Information Technologies*, 2010.

[9] S. Yang, "Logic synthesis and optimization benchmarks user guide," Technical Report 1991-IWLS-UG-Saeyang, Microelectronics Center of North Carolina, 1991.

[10] [Online]. Available: http://www.eecs.berkeley.edu/~alanmi/benchmarks/