IPSJ Transactions on System LSI Design Methodology Vol.8 116–122 (Aug. 2015)

[DOI: 10.2197/ipsjtsldm.8.116]

Short Paper

A 3D FPGA Architecture to Realize Simple Die Stacking

Motoki Amagasaki[1,a)] Qian Zhao[1] Masahiro Iida[1] Morihiro Kuga[1] Toshinori Sueyoshi[1]

Received: December 5, 2014, Revised: March 13, 2015,
Accepted: April 29, 2015, Released: August 1, 2015

Abstract: To balance between cost and performance, and to explore 3D field-programmable gate array (FPGA) with realistic 3D integration processes, we propose spatially distributed and functionally distributed types of 3D FPGA architectures. The functionally distributed architecture consists of two wafers, a logic layer and a routing layer, and is stacked by a face-down process technology. Since vertical wires pass through microbumps, no TSVs are needed. In contrast, the spatially distributed architecture is divided into multiple layers with the same structure, unlike in the functionally distributed type. This architecture can be expanded to more than two layers by stacking multiples of the same die. The goal of this paper is to elucidate the advantages and disadvantages of these two types of 3D FPGAs. According to our evaluation, when only two layers are used, the functionally distributed architecture is more effective. When higher performance is achieved by using more than two layers, the spatially distributed architecture achieves better performance.

Keywords: 3D FPGA, spatially distributed type, functionally distributed type

1. Introduction

With the scaling down of device dimensions on semiconductor wafers, the speeds of integrated circuits have been improved greatly. However, as process shrinking has proceeded into scales much smaller than micrometers, the problems of circuit delay and power leakage have become critical. This is especially true for field-programmable gate arrays (FPGAs), where the routing resources account for the majority of chip area and circuit-delay performance. Employing three-dimensional (3D) integration technologies to vertically stack several silicon dies is considered as one way to solve this problem.

Conventional 3D FPGAs are classified into spatially distributed types and functionally distributed types on the basis of the distribution of die stacking. Spatially distributed 3D FPGAs are realized by stacking a set of similar silicon dies. Such 3D FPGAs employ a number of through-silicon vias (TSVs) in 3D switch boxes (SBs) to ensure high routability [1], [2]. The relation between TSVs and SBs is such that when the channel width (CW) is 50, 100 inter-layer connections will be necessary in each SB. In light of the size of microbumps and TSVs, such architectures will be infeasible to scale down in the near future due to the area overhead. In contrast, functionally distributed types specialize each layer to one function. Existing FPGAs [3] have a structure in which logic circuits and configuration memory bits are placed on different layers. In this type, circuits on each layer are optimized separately. However, the connections between layers are specialized to each design, and so the generalization of connections is difficult. Therefore, although 3D stacking technology is very attractive, effective 3D FPGA architectures with good cost and performance are yet to be introduced.

The development of computer-aided design (CAD) tools for 3D FPGA is also crucial. The design flow differences between 2D and 3D FPGAs occur in processes after logic clustering. First, logic blocks (LBs) are distributed into several layers. Then, intra- and inter-layer placement and routing are performed. In Ref. [4], TPR, an open-source 3D FPGA placement and routing tool, was developed to handle second-step processes. TPR is based on VPR 4.0. This tool was published over a decade ago, however, and therefore does not support some recent FPGA architectures. A design flow to explore minimal vertical connections is proposed in Ref. [2], but the details of their novel tools are not mentioned.

To balance cost and performance, and to explore 3D FPGA architectures with realistic 3D LSI processes, we present two facile 3D FPGA architectures to distinguish between the features of functionally distributed and spatially distributed approaches. **(1) Functionally distributed approach**: This FPGA consists of two wafers (a logic layer and a routing layer) and mitigates the side effects of vertical wires [5]. Since vertical wires pass through microbumps by using face-down stacking, no TSVs are needed. By dividing routing resources into two layers, a smaller tile can be achieved. Smaller tile sizes result in shorter routing wires and faster signal transport, which improves routing performance. **(2) Spatially distributed approach**: This FPGA is divided into multiple layers that have the same structure, unlike in the functionally distributed type. This architecture can be expanded to more than two layers. To decrease total number of vertical connections, vertical wire between layers is limited to outputs of logic cluster.

In this paper, we developed 3D FPGA computer-aided design (CAD) tools and used these to explore the architectures, comparing the two 3D FPGA architectures in terms of area and delay performance. The goal of this paper is to elucidate the advan-

[1] Kumamoto University, Kumamoto 860–8555, Japan
[a)] amagasaki@cs.kumamoto-u.ac.jp

tages and disadvantages of these two types of 3D FPGAs. The rest of the paper is organized as follows. Section 2 discusses the proposed 3D routing architecture. Section 3 describes the design flow and CAD tools. Section 4 describes the evaluation process and results. Conclusions and directions for future research are given in Section 5.

2. Proposed 3D FPGAs

2.1 3D Integration and Basic Tile Structure

Figure 1 shows the cross-sectional structure of a 3D LSI circuit fabricated by Koyanagi's 3D integration technology [6]. The thinned upper layers are stacked face-up onto a thick LSI wafer that is face-up (Fig. 1 (a)). Figure 1 (b) shows the arrangement when the thinned upper layers are stacked face-down onto the thick LSI wafer. If the number of layers is limited to two, no TSVs are needed.

It is important to balance cost and performance when deciding on a 3D FPGA architecture. To this end, we consider the homogeneous (uniform) tile structure [7]. To treat various array sizes in a similar manner, it is important to simplify the tile structure. In this paper, the proposed 3D FPGAs are based on a homogeneous tile structure (**Fig. 2**).

2.2 Functionally Distributed 3D FPGA Architecture

Figure 3 shows details of the structures in the functionally distributed 3D FPGA. There are two layers in the architecture examined in the current research: a logic layer and a routing layer. The tiles on the logic layer have an LB and a small part of the routing resources; the tiles on the routing layer have only routing resources. The difference between conventional and proposed 3D SBs is that the 3D connections are made at the LB input and output pins. The number of inter-layer connections within one tile is equal to the total number of LB input and output pins. The number of LB inputs I is determined by the following formula [8].

$$I = \frac{K}{2} \times (N + 1) \quad (1)$$

Here, K is the number of logic cell inputs, and N is the cluster size. The number of vertical wires per one tile are $I + N$. For example, when $K = 6$ and $N = 4$, the numbers of LB inputs and outputs are 15 and 4, respectively. Therefore, the number of vertical connections per tile is 19.

We next discuss a method for determining the minimum width of the routing track channel for the two layers. To find this, we first set the initial CW of the logic layer to 1.5 times the number of LB input pins. Then, the areas used by the small part of connection blocks (CBs) and SB on logic layer can be calculated as the routing area of the logic layer; the tile area of the logic layer is the sum of the logic and small routing resource to connect neighbor LBs. The CW of the routing layer is calculated by allocating the size of the routing area as the size of the logic layer tile area. We next perform routing. If the routing is successful, then the next trail of the logic layer will have its CW set to half the current one. If routing fails, the CW of the next trail of the logic layer will be set to twice the current one. This process is repeated until the minimum CW that can lead to successful routing is found.

By dividing routing resources into two layers, we can achieve a smaller tile. Smaller tiles allow shorter routing wires and thereby enable faster signal transport, which improves the routing performance. The router can choose a network route on the logic layer or the routing layer. Although the number of layers in this type of approach is limited to two, when face-down stacking is used, no TSVs are necessary.

2.3 Spatially Distributed 3D FPGA Architecture

Figure 4 shows details of the structures in the spatially dis-

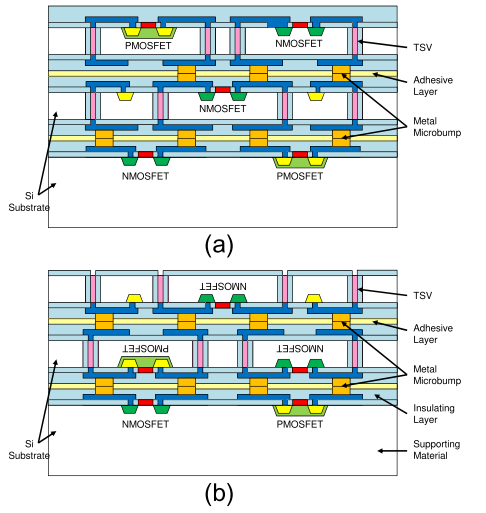

Fig. 1 Cross-sectional structure of 3D LSI architecture: (a) face-up stacking; (b) face-down stacking.

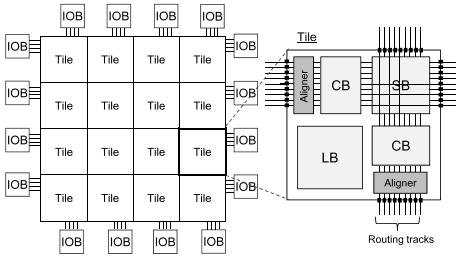

Fig. 2 Completely homogeneous routing architecture.

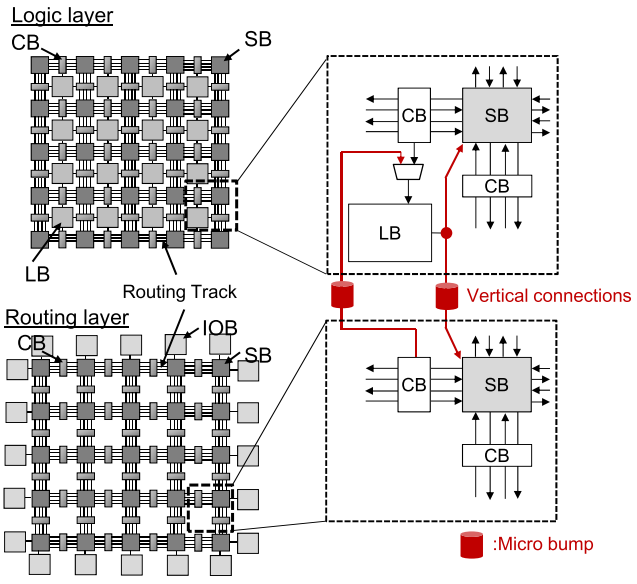

Fig. 3 Functionally distributed 3D architecture with face-down stacking.

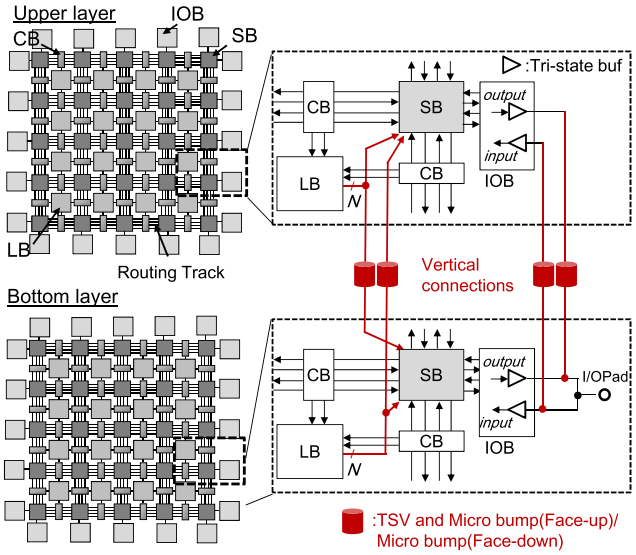

Fig. 4 Proposed 3D routing structure.

tributed 3D FPGA. There are two layers in the architecture examined in this example: an upper layer and a lower layer. Both layers have the same structure, which allows using a uniform mask set. The IOBs are connected to the two layers by vertical wires. One difference between this structure and that in our functionally distributed 3D FPGA (Fig. 3) is that vertical wires for LB inputs have been eliminated. The number of inter-layer connections within each tile is equal to twice the number of LB output pins. The total number of vertical wire connections, T_{VC}, is determined by the following formula.

$$T_{VC} = (Arraysize)^2 \times 2N + 4 \cdot Arraysize \times 2C_{IO} \qquad (2)$$

In this formula, $Arraysize$ is the side length of the FPGA array, N is the cluster size, and C_{IO} is the IO capacity. Compared with the functionally distributed architecture, which requires $\frac{LUTinputs}{2} \times (N+1) + N$ vertical connections per tile, this architecture reduces the required number of inter-layer connections. In addition, this architecture can be expanded to more than two layers by stacking dies of the same type. When the number of layers are 3 and 4, the maximum number of TSVs per one tile are $4N$ and $8N$, respectively.

3. Design Flow and CAD tools

The design of functionally distributed architectures can use the same CAD tools as are used for the design of traditional 2D FPGAs. In this section, we introduce a design flow and CAD tools that can be used for designing spatially distributed architectures. We developed the 3D FPGA design flow (**Fig. 5**) by using VTR 7.0 [9], which is the most current version as of this writing. As is done with 2D FPGAs, the circuit (in 'BLIF' format) is first technology-mapped with ABC [10] and then clustered with AA-Pack, which is included as part of VPR 7.0.

Next, the clustered netlist is partitioned by using hMETIS [11], which can efficiently group LBs into a specified number of layers of similar size and with a minimal number of interconnections. These constraints are important for 3D FPGA partitioning. Allocating a similar number of LBs on all layers ensures that the resources on each layer can be fully used. Simultaneously, in order to minimize the overhead from TSVs, partitions with fewer interconnections are preferable. We wrote a script that can generate hypergraphs of LBs from the clustered netlist. The hMETIS tool is used to perform partitioning on the hypergraph generated by the script. Finally, layer allocation information is added back to the clustered netlist as LB attributes.

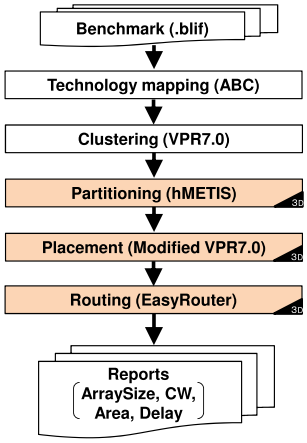

Fig. 5 Design flow of proposed 3D FPGA.

We created a 3D placement tool that uses VPR 7.0 placer as a base. The tool operates as follows. First, the layer allocation information is read from the clustered and partitioned netlist. Then, a conventional placement by simulated annealing is performed. During the placement process, logic modules are freely swapped within each layer. The algorithm used is the bounding box (BB) method, which focuses on minimizing the bounding-box wire length of the circuit. In order to evaluate the BBcost of a net across multiple layers, we calculate bb_x (BB distance on x direction) and bb_y (BB distance on y direction) of each layer, and then use the maximum bb_x and bb_y of all layers in the final BBcost. The vertical connection cost is not considered yet in the placement for two reasons: (1) A typical 3D-FPGA CAD flow processes the vertical allocation of logic modules in the partitioning step. During partition step, inter-layer-connections are minimized with algorithms like hMETIS. For the placement step, logic modules are placed within each layers, therefore, vertical connection cost is not necessary to be considered. (2) When the vertical connection cost is comparable with horizontal connection cost, we have to build a cost model for the partitioning algorithm. However, in this approach, the delay of the TSV is small compare to horizontal wire delay, and the number of layers is small, the vertical connection cost is negligible during partitioning and placement steps. However, the final delay derived from the router does certainly include the TSV delay. In order to handle 3D-FPGAs with different 3D-VLSI processes and more layers, we will improve 3D partitioning that considering vertical cost in the future work. We also plan to implement timing-driven algorithms.

Finally, routing was performed with our novel tool, the EasyRouter [12]. EasyRouter implements a pathfinder routing algorithm similar to that in VPR; however, EasyRouter simplifies the implementation of new FPGA architectures with various routing topologies. In addition, EasyRouter combined with VLSI CAD can provide highly accurate reports on area and critical path delays for FPGA designs that are based on standard cells.

Routing is performed in two main steps. First, the router explores the minimum channel-width for each circuit. Next, the CW is fixed at 1.2 times the CW of the circuit with the highest minimum CW (i.e., 1.2-fold the maximum width), and the results are evaluated. This method ensures that all circuits are fairly evaluated on the same device with sufficient resources.

4. Evaluation

This section compares the architectures of a functionally distributed 3D FPGA (type 1) with that of the proposed spatially distributed 3D FPGA (type 2), evaluating the area, the critical path delay, and the area delay product. An island-style 2D FPGA(2D_Island) and 2D homogeneous FPGA(2D_Homo) are used as the baseline for evaluations. An analysis of the proposed 3D FPGAs from evaluation results is also given in this section.

4.1 Evaluation Conditions

The parameter values for the target architectures are listed in **Table 1**. All implemented 3D FPGA architectures are homogeneous FPGAs with a lookup-table (LUT) size of 6 and a cluster size of 8. For the routing architecture, we used a Wilton-type SB with $Fs = 3$. The Fc_{in} parameter was set to 0.5, which means that half of the tracks in the routing channel are connected to an LB input through a CB. We set the area of each TSV to 96 μm^2 [*1] and the delay to 2.2 ps, taking these values from the report in Ref. [6]. The type 1 3D FPGA has only two layers, which is denoted by type 1_L2. The type 2 3D FPGAs were implemented with from 2 to 4 layers, and these are denoted by type 2_L2, type 2_L3 and type 2_L4 for 2, 3, and 4 layers, respectively. The face-up stacked architectures are marked as "(face-up)," and the face-down stacked architectures are noted as "(face-down)."

The 20 largest MCNC benchmarks were used as the evaluation test suite. The device was designed to use 65-nm CMOS technology. The tile was synthesized with a Synopsys Design Compiler F-2011.09-SP2. For the area calculation, tile areas of all architectures were from synthesized results. The total area was calculated by multiplying the number of tiles with the area of one tile. For the delay evaluation, all CBs and SBs are considered to be composed of 2-to-1 multiplexers (MUXes) for both area and delay. The MUXes' physical parameters are incorporated by referenced to the standard cell library.

4.2 Preparation: Partitioning Results

The partitioning is performed by hMETIS with UBfactor set at 0.01, which allows an imbalance of up to 1% between the partitions in order to minimize inter-layer connections. All circuits are partitioned into layers of similar sizes. Because it is necessary to set the array size of all FPGA layers to the size of the largest partition, a good partitioning balance ensures that the resources on each layer can be fully used.

Table 1 Target architectures parameters.

Name	Value	Name	Value
LUT size	6	IO capacity	10
Cluster size	8	TSV area	96 μm^2
Fc_{in}	0.5	TSV delay	2.2 ps

[*1] In Ref. [6], four poly-Si TSVs with a size of $2 \times 12\,\mu$m^2 are connected in parallel in one vertical interconnection.

4.3 CW Exploration Results

Figure 6 shows the results of CW exploration for each architecture for the 2D and 3D FPGAs. According to the CW exploration results, the fixed CW result for each architecture was taken as 1.2 times the largest of explored CW of all benchmarks.

As reported in Ref. [7], the 2D_Homo layout uses slightly more channels than the 2D_Island layout. The CW of the routing layer of the type 1 3D FPGA was determined by examining the size of the logic layer tile. A CW of 82 (the total CW from the logic and routing layers) was the minimum CW of this architecture, which was sufficient to implement all circuits for this evaluation. The type 2 architectures permit narrower CWs as the number of layers increases. This is because the 3D allocation of LBs brings some logics closer together than in the 2D allocation, which reduces routing congestion. The type 2 3D FPGAs with 2, 3, and 4 layers have, respectively, 5.6%, 16.7%, and 19.4% narrower CWs, on average, than the 2D_Island does.

4.4 Area Results

The results of comparing chip areas between type 1 and type 2 3D FPGAs are shown in **Fig. 7**. We first implemented all benchmarks on each architecture and normalized the area results by the area of the 2D_Island FPGA. All results shown in Fig. 7 are an average of the normalized results for the corresponding architecture. We evaluated face-down and face-up 3D stacking methods

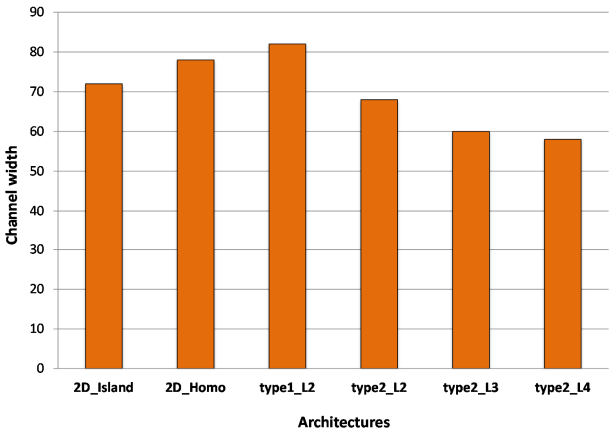

Fig. 6 Channel width.

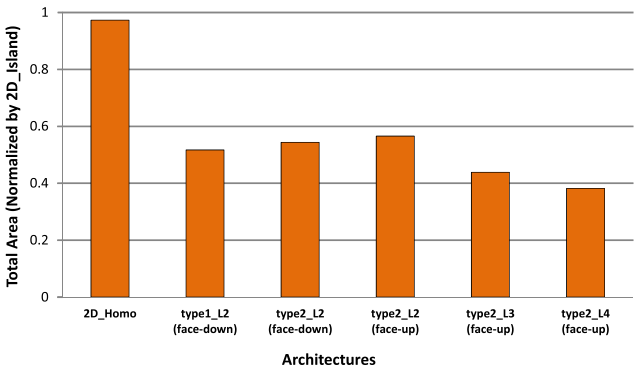

Fig. 7 Total area per layer.

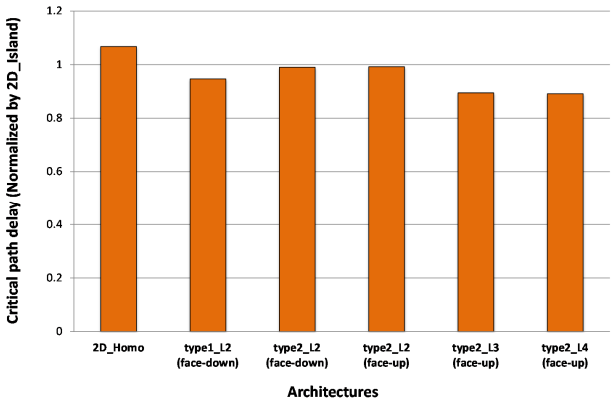

Fig. 8 Critical path delay.

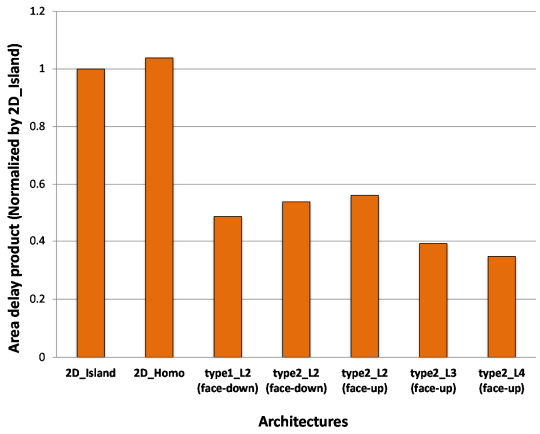

Fig. 9 Area delay product.

for 3D FPGAs. The face-up counts include TSV area in the total area, and the face-down counts do not (more specifically, the area used by TSVs is 0 for face-down stacking).

First, we compare two-layer 3D FPGAs. The face-down stacked type 1_L2 (face-down) and type 2_L2 (face-down) reduce area by about 48.2% and 45.6% from the area of 2D_Island. In contrast, the reduction from the face-up stacked type 2_L2 (face-up) is about 43.4%. Next, we compare type 2 3D FPGAs with different numbers of layers. We can see the trend of area reduction from the results of type 2 3D FPGAs having from 2 to 4 layers. Relative to the area of 2D_Island, the type 2_L2 (face-up), type 2_L3 (face-up), and type 2_L4 (face-up) designs reduce area by 43.4%, 56.1%, and 61.7%, respectively. The type 2_L4 (face-up) design offers the best performance of all examined architectures.

To summarize, for a two-layer 3D FPGA, type 1 with face-down stacking has the best performance on area. However, when implementing 3D FPGAs with more than two layers, where face-up stacking is necessary, the type 2 architecture offers a much smaller area.

4.5 Delay Results

The results of testing the type 1 and type 2 architectures for critical path delay are shown in **Fig. 8**, where the delays are normalized by the critical path delay of the 2D_Island architecture. Delay results are not significantly different between (all are within 10% of one another). This is because the MCNC benchmark circuits are not very large, and so the critical path delay is mainly from the LB rather than from routing. However, the results still show some trends. The type 1_L2 (face-down) has 5.3% better delay performance than the 2D_Island FPGA. The improvement is a result of the type 1 3D FPGA having more routing channels on the routing layer. When comparing the type 2_L2 (face-down) and type 2_L2 (face-up) designs, we can see that the TSV overhead in the critical path delay is very small. The TSV delay overhead is caused mainly by increased routing-wire delay due to the larger tile sizes and the delays of the TSVs themselves.

For the type 2 3D FPGA, the critical path delay was mainly affected by two factors. More LBs are allocated closer together as more layers are stacked, which improves the delay. Opposing this, the delay from MUXes of the SBs increase when stacking more layers. In the proposed type 2 3D FPGA, all LB outputs are connected to an SB of tiles at the same coordinate across all layers. When the number of MUX inputs per SB is m, each SB will have $(m - 1)$ 2-to-1 MUXes with logic depth $log_2(m)$. As a result, each type 2 SB MUX has more levels in its logic depth, which lengthens the total routing path. The performance of the type 2_L3 (face-up) architecture is similar to that of the type 2_L4 (face-up) architecture, which is because the increased SB MUX depth offsets the advantages of 3D allocation.

Additionally, partitioning and placement algorithms affect delay performance. In future work, we plan to develop a timing-driven partitioning and placement algorithm for the proposed 3D FPGA.

4.6 Area Delay Product Results

Finally, the area delay product results are shown in **Fig. 9**. For the face-down stacking method, the type 1_L2 (face-down) design performs 51.5% better than 2D_Island. For the face-up stacking method, the 4-layer type 2_L4 (face-up) performs 65.2% better than 2D_Island. When limiting the analysis to two layers, the functionally distributed type of architecture (type 1) is most effective. However, if we prioritize performance and use more than two layers, the spatially distributed architecture (type 2) offers better performance.

5. Conclusion

In this paper we proposed a functionally distributed type and a spatially distributed type of 3D FPGA architecture to allow simple die stacking. According to the evaluation results, the functionally distributed type is more effective when limiting designs to two layers, but the spatially distributed architecture with more than two layers is a better choice when performance is prioritized. In this research, we used relatively large TSVs. However, since more smaller TSVs are been developed in recent researches, functionally distributed type is more effective using these smaller TSVs. In future work, we are planning to stack multiple functionally distributed FPGAs with the face-up method and propose inter-layer high-speed communications that use TSV serial connections [13]. We also intend to improve the CAD toolsets to support algorithms that consider power consumption and timing in order to achieve better power and delay performance. At the same time, to show the effectiveness of 3D technology, it is nec-

essary to evaluate using more larger circuits, such as VTR benchmark sets [9].

References

[1] Alexander, M.J., Cohoon, J.P., Colflesh, J.L., Karro, J., Robins, G. and Science, C.: Three-Dimensional Field-Programmable Gate Arrays, *Proc. 8th Annual IEEE International ASIC Conference and Exhibit*, pp.253–256 (Sep. 1995).

[2] Gayasen, A., Narayanan, V., Kandemir, M. and Rahman, A.: Designing a 3-D FPGA: Switch Box Architecture and Thermal Issues, *IEEE Trans. VLSI Systems*, Vol.16, No.7, pp.882–893 (2008).

[3] Naoto, T., Ishida, T., Onoduka, T., Nishigoori, M., Nakayama, T., Ueno, Y., Ishimoto, Y., Suzuki, A., Chung, W., Madurawe, R., Wu, S., Ikeda, S. and Oyamatsu, H.: World's first monolithic 3D-FPGA with TFT SRAM over 90nm 9 layer Cu CMOS, *Proc. VLSIT*, pp.219–220 (June 2010).

[4] Ababei, C., Mogal, H. and Bazargan, K.: Three-dimensional Place and Route for FPGAs, *IEEE Trans. CAD of Integrated Circuits and Systems*, Vol.25, No.6, pp.1132–1140 (2006).

[5] Hamada, T., Zhao, Q., Amagasaki, M., Iida, M., Kuga, M. and Sueyoshi, T.: Three-Dimensional Stacking FPGA Architecture Using Face-to-Face Integration, *Proc. VLSI-SoC*, pp.196–201 (Oct. 2013).

[6] Koyanagi, M., Fukushima, T. and Tanaka, T., High-Density Through Silicon Vias for 3-D LSIs, *IEEE Journals & Magazines*, Vol.97, No.1, pp.49–59 (2009).

[7] Inoue, K., Yosho, H., Amagasaki, M., Iida, M. and Sueyoshi, T.: An Easily Testable Routing Architecture And Effecient Test Technique, *Proc. FPL*, pp.291–294 (Aug. 2011).

[8] Ahmed, E. and Rose, J.: The Effect of LUT and Cluster Size on Deep-Submicron FPGA Performance and Density, *Proc. FPGAs*, pp.3–12 (Feb. 2000).

[9] Luu, J., Goeders, J., Wainberg, M., Somerville, A., Yu, T., Nasartschuk, K., Nasr, M., Wang, S., Liu, T., Ahmed, N., Kent, K.B., Anderson, J., Rose, J. and Betz, V.: VTR 7.0: Next Generation Architecture and CAD System for FPGAs, *ACM TRETS*, Vol.7, No.2, pp.6:1–6:30 (June 2014).

[10] Mishchenko, A. et al.: ABC: A System for Sequential Synthesis and Verification (2009), available from ⟨http://www.eecs.berkeley.edu/~alanmi/abc/⟩.

[11] Selvakkumaran, N. and Karypis, G.: Multi-Objective Hypergraph Partitioning Algorithms for Cut and Maximum Subdomain Degree Minimization, *IEEE Trans. CAD of Integrated Circuits and Systems*, Vol.25, No.3, pp.504–517 (Feb. 2006).

[12] Zhao, Q., Inoue, K., Amagasaki, M., Iida, M. and Sueyoshi, T.: FPGA Design Framework Combined with Commercial VLSI CAD, *IEICE Trans. Inf. Syst.*, Vol.E96-D, No.8, pp.1602–1612 (2013).

[13] Kajiwara, T., Zhao, Q., Amagasaki, M., Iida, M., Kuga, M. and Sueyoshi, T.: A Novel Three-Dimensional FPGA Architecture with High-speed Serial Communication Links, *Proc. ICFPT*, pp.306–309 (Dec. 2015).

Motoki Amagasaki received his B.E. and M.E., degrees in Control Engineering and Science from Kyushu Institute of Technology, Japan in 2000, 2002, respectively. He was a design engineer at NEC Micro Systems Co., Ltd. from 2002–2005. He received his D.E. degree from Kumamoto University, Japan, in 2007. He has been an assistant professor in the Department of Computer Science at Kumamoto University since October 2007. His research interests include reconfigurable system and VLSI design. He is a member of IPSJ, IEICE and IEEE.

Qian Zhao received his B.E. degree in Control Engineering received B.E. degree in the college of automation and electronic engineering from Qingdao University of Science and Technology, China, in 2007. Further, he received his M.E. and D.E. degrees in Computer Science and Electrical Engineering from Kumamoto University in 2011 and 2014, respectively. He is now a postdoctoral researcher at Kumamoto University. His research interest is reconfigurable computing systems.

Masahiro Iida received his B.E. degree in Electronic Engineering from Tokyo Denki University in 1988. He was a research engineer at Mitsubishi Electric Engineering Co., Ltd. from 1988 to 2003. He received his M.E. degree in Computer Science from Kyushu Institute of Technology in 1997. Further, he received his D.E. degree from Kumamoto University, Japan, in 2002. He has been an associate professor in the Department of Computer Science and Electrical Engineering at Kumamoto University since February 2003. During 2002–2005, he held an additional post as a researcher at PRESTO, Japan Science and Technology Corporation (JST). His current research interests include high-performance low-power computer architectures, reconfigurable computing systems, VLSI devices and design methodology. He is a member of IPSJ, IEICE and IEEE.

Morihiro Kuga received his B.E. degree in Electronics from Fukuoka University in 1987 and M.E. and D.Eng. degrees in Information Systems from Kyushu University in 1989 and 1992. From 1992 to 1998, he was a lecturer at the center for Microelectronic Systems, Kyushu Institute of Technology. He has been an associate professor of comuter science at Kumamoto University since 1998. His research interests include parallel processing, computer architecture, reconfigurable system, and VLSI system design. He is a member of IPSJ and IEICE.

Toshinori Sueyoshi received his B.E. and M.E. degrees in Computer Science and Communication Engineering from Kyushu University in 1976 and 1978 respectively. From 1978 to 1987, he was a research associate at Kyushu University, where he received his D.E. degree in 1986. From 1987 to 1989, he was an associate professor of Information System at Kyushu University. From 1989 to 1997, he was an associate professor of Artificial Intelligence at Kyushu Institute of Technology. Since 1997 he has been a professor of Computer Science at Kumamoto University. He is also a guest professor at Kyushu University. His primary research interests include computer architecture, reconfigurable systems, parallel processing. He served as Chair of the Technical Committee on Reconfigurable Systems of IEICE, Chair of the Technical Committee on Computer Systems of IEICE, Chair of IEEE Computer Society Fukuoka Chapter, Chair of IEEE Circuits and Systems Society Fukuoka Chapter, and Director of the IPSJ Kyushu Chapter. He is a fellow of IEICE and a senior member of IPSJ.

(Recommended by Associate Editor: *Toshinori Hosokawa*)