High-Performance, Cost-Effective Heterogeneous 3D FPGA Architectures

Roto Le
Division of Engineering
Brown University
Providence, RI 02912
ro-to_le@brown.edu

Sherief Reda
Division of Engineering
Brown University
Providence, RI 02912
sherief_reda@brown.edu

R. Iris Bahar
Division of Engineering
Brown University
Providence, RI 02912
iris_bahar@brown.edu

ABSTRACT

In this paper, we propose novel architectural and design techniques for three-dimensional field-programmable gate arrays (3D FPGAs) with Through-Silicon Vias (TSVs). We develop a novel design partitioning methodology that maps the heterogeneous computational resources of an FPGA into a number of die such that the total die area is minimized and the FPGA performance is maximized. Minimizing the total die area leads to direct manufacturing cost savings which is an important incentive to bring 3D technology to the fab and onto the market. An estimation framework is developed to assess the impact of silicon area utilized by 3D interconnect resources while taking into account the large area occupied by TSVs which is crucial to total die area of 3D FPGAs. In order to improve area and performance of 3D FPGAs, we design a novel 3D switch box with *bypass* TSVs. We also analyze the impact of different partitioning strategies on die area and find the optimal number of die that gives the largest reductions in total die area while maximizing the performance. Using a well-developed simulation infrastructure, we show that our methodologies can achieve an average reduction of 27.7% in total die area with a reduced interconnect path delay of about 58%.

Categories and Subject Descriptors: B.7.1 [INTEGRATED CIRCUITS]: Types and Design Styles—*Advanced technologies*.
General Terms: Economics, Performance
Keywords: Heterogeneous FPGA Design, 3D Integrated Circuits

1. INTRODUCTION

Field Programmable Gate Arrays (FPGAs) have become a viable alternative to custom Integrated Circuits (ICs) by providing flexible computing platforms with improved costs and shorter time-to-market. In an FPGA-based system, a design is mapped onto an array of reconfigurable logic blocks and communicated by re-programmable interconnections composed of wire segments and switch boxes. While the re-programmable capability provides flexibility, it also leads to area and performance overheads in comparison to custom chips. Thus, to benefit from advantages of both FPGAs and custom chips, *heterogeneous FPGAs* have emerged as an attractive choice for system-on-a-chip implementations. Beside the traditional reconfigurable fabric, heterogeneous FPGAs include dedicated full-custom design components such as digital signal processors (DSP), multipliers, on-chip memory blocks, and entire processors. Examples of such heterogeneous FPGAs include Xilinx Spartan 3, Virtex 4, 5 and Altera Cyclone II, Stratix II, III and Lattice ECP2 family.

To provide the required reconfigurable functionality, FPGAs provide a large amount of programmable interconnect resources in the form of wire segments, switches, and signal repeaters. These programmable interconnect resources typically consume a large portion of the FPGA silicon die area. A number of recent studies show that programmable interconnect fabric consumes about $70-80\%$ of the total FPGA area [8, 6]. Since die area is one of the main factors that determine manufacturing costs, reducing the silicon footprint of the programmable fabric can lead to significant improvements in the manufacturing costs of FPGAs. Reducing the length of interconnects will also bring performance improvements to the typical interconnect-delay dominated FPGAs.

Three-dimensional (3D) Integrated Circuits (ICs) with through-silicon vias is an new technology that will increase the functionality, scale of integration, and performance of integrated systems [1, 2]. Increasing the scale of integration is particularly attractive considering that optical lithography is approaching its natural limits. In 3D integration, multiple die or *layers* are integrated and interconnected with *through-silicon vias* (TSVs). Three-dimensional integration can lead to significant reduction in wire length and interconnect delay through the use of TSVs. A number of recent publications propose novel 3D architectures and physical design techniques that lead to FPGAs with better performance than existing planar FPGAs [3, 8, 9, 10, 6, 11]. For example, Alexander *et al.* [3] developed 3D island-style based FPGAs that extend four directional 2D switch boxes to six directional 3D switch boxes. This 3D switch architecture allows logic blocks to have six immediate neighbors including four on the die or plane where the switch box is placed, and two others above and below the die. In another work, Lin *et al.* propose a 3D FPGA architecture that partitions homogeneous FPGAs components such that configuration SRAM memory cells and switch transistors can be moved to other 3D layers [6]. In addition to devising new 3D FPGA architectures, a number of recent studies develop placement and routing models to support and assess 3D FPGA architectures (e.g., [7, 8, 9, 10, 11]).

In this paper our objective is to develop novel 3D FPGA architectures and designs that improve performance with lower costs than planar FPGAs. Our cost savings arise from significant reductions in total die area enabled by our methodology. We summarize the contributions of this paper as follows.

- We formulate the problem of resource partitioning for heterogeneous FPGAs into a number of die for 3D ICs to minimize the total die area and the fabrication costs of 3D FGPAs.

GLSVLSI'09, May 10–12, 2009, Boston, Massachusetts, USA.

- Using Rent-based statistical wirelength distribution models, we propose novel methods to estimate the area used by TSVs and interconnect resources of heterogeneous 3D FPGAs.
- We propose novel methods to minimize the total number of TSVs used in 3D stacking, and we also propose new techniques to estimate the performance of 3D heterogeneous FPGAs.
- We analyze the impact of different resource partitioning strategies on the performance of FPGAs as well as their total costs as measured by the total die area. We also analyze the relationship between the number of die in the 3D stack and the total die area, and show how to choose the optimal number of die that minimize the total die area of a 3D FPGA.
- Using a comprehensive experimental setup, we show that our method leads to a 27% reduction in direct total die area, and a 58% improvement in performance. The improvements in total die area lead to immediate cost savings.

The rest of this paper is organized as follows. Section 2 introduces our motivation and formulation for the problem of transforming a planar heterogeneous FPGA design into a 3D FPGA design. In Section 3, we propose how to calculate the die areas allocated for computation, TSVs, and wiring in 3D FPGAs. In Section 4, we discuss how to calculate the improvement in performance attained by using 3D FPGAs. Section 5 presents the results and observations from our experimental evaluation. Finally, Section 6 summarizes the main conclusions of this work.

2. MOTIVATION AND FORMULATION

One of the crucial properties of FPGAs is that the reconfigurable interconnect resources consume a large portion (up to 80%) of the total silicon die area [8, 6]. By bringing the computational components closer together in three dimensions, 3D ICs have the potential to reduce the size of the programmable interconnect fabric required for routing in FPGAs which lead to significant reductions in the silicon area, thereby directly reducing the cost of fabrication. Cost savings are important incentives for the industry to offset any cost increases required for TSV creation (either using laser drilling or bonding) and 3D bonding.

Designing heterogeneous 3D FPGAs involves a number of challenges. We outline and tackle the following challenges in this paper.

1. Typical TSVs occupy remarkably large silicon area ($4 \times 4 \mu m^2$ to $5 \times 15 \mu m^2$ [13]). Thus, introducing TSVs will lead to increases in total die area. Therefore, in 3D FPGAs, it is important to assess die area savings attained from reductions in interconnect resources against the increase in die area due to TSVs.

2. Introducing TSVs as a part of the programmable wiring fabric requires a new design for the reconfigurable switch boxes. Instead of just achieving connectivity in 2D as in planar 3D ICs, a switch box has to include extra switches to allow incoming lateral wires to connect to vertical TSVs, and even incoming TSVs to connect to outgoing TSVs since the 3D stack might include more than two die.

3. The number of die in the 3D stack should be chosen to maximize the performance and minimize the costs. Once the number of die in the 3D stack is determined, partitioning the computational resources of the FPGA among the die should be carried out in a way to minimize the total demand on the interconnect resources and the required number of TSVs.

In this paper, our objective is to tackle these challenges and develop a realistic design methodology for 3D FPGAs that delivers the expected 3D performance benefits while minimizes any incurred costs. The overarching goals of our objectives can be summarized with the following problem formulation.

Given: A planar FPGA that has a total area A and contains a set of heterogeneous computational resources $R = \{r_1, \dots r_N\}$.
Output: Find the optimal number of die, m, and a partition of R into the m die such that the total die area of the 3D FPGA is minimized compared to A and performance is maximized.

As an example, a set of heterogeneous computational resources R for an FPGA might have 4000 logic blocks, 1000 4K memory blocks, 200 DSPs, and 2 processors for a total of $n = 5202$ computational components. We seek to find the optimal number of die m and a partition that maps each computational resource into exactly one die.

We proceed by first proposing a novel approach to estimate the total die area of 3D FPGAs and determine the optimal number of die (Section 3). Our area estimation includes total logic area, total TSV area, and total programmable interconnect area.

3. ESTIMATING DIE AREA SAVINGS

The total die area of a 3D heterogeneous FPGA is the sum of the areas of the die or layers that constitute the 3D IC. The die area is comprised of (1) the total computational resource area, (2) the total TSV area, and (3) the area needed for the reconfigurable wiring fabric. It is expected that as the number of die in a 3D stack increases, the requirements for the *within-die lateral wiring* area will decrease while the TSV area required for the *inter-die vertical wiring* will increase.

3.1 Estimating Total Logic Area

In this section we present area estimation models for computational components in a generic heterogeneous island style SRAM-based FPGA. Such an FPGA will contain soft computational resources such as logic clusters, and hardcore computational resources such as embedded memory blocks, DSP blocks, and processors. We next describe the area estimation approach and assumptions for each of these components.

Logic Cluster: The reconfigurable logic cluster or block executes logic operations and is considered the main component in FPGAs. A generic logic cluster contains a number of Look-Up Tables (LUTs) and associated registers, I/O, multiplexers, clock, and reset units. The total area of a logic cluster may be computed by summing the area of all these components. In our study we use a cluster architecture and area model from [12]. The cluster consists of eight 4-input LUTs, 20 logic input pins, 8 output pins, 1 clock and 1 reset signal.

Embedded Memory and DSP Blocks: In contrast to homogeneous FPGAs, heterogeneous FPGAs contain dedicated hard memory and DSP blocks to obtain higher performance and power saving. A typical heterogeneous FPGA often contains SRAM memory blocks having different sizes to provide high flexibility in configuration and utilization. We use two different sized memory blocks, *Mem1* and *Mem2* sized similar to memory blocks in a realistic FPGA; *i.e.*, 64×16 bits and 128×32 bits respectively. We assume that these memory blocks are SRAM blocks and estimate their area by using the CACTI memory models [14, 15]. After getting the area estimation for memory blocks we estimate the area of DSP blocks based on the relative ratio between DSP and *Mem2* blocks from the Altera Stratix II handbook documentation ([16] p. 2-41).

Components	Capacity	Area
4-input LUT	1 LUT	1×
Cluster size 8	8 LUTs	28.5×
Cluster size 16	16 LUTs	76.7×
Mem1 block	32x16 bits	65.3×
Mem2 block	128x32 bits	365.3×
DSP block	four 16x16-bit multipliers	1461.2×

Table 1: Area estimation of computational components normalized to the area of a 4-input LUT (26598 λ^2).

Our area estimations for the different computational resources are summarized in Table 1, where the area of the components are normalized with respect to the area of an LUT. The area of a 4-input LUT is estimated to be equal to 26598 λ^2 [12].

3.2 Estimating Total TSV Area

In this section our objective is to compute the silicon area required by through-silicon vias. A typical TSV can occupy a remarkably large silicon area (*e.g.*, $4\times4\mu m^2$) with a pitch of $20\mu m$ [13], and thus it is important to calculate the expected area utilized by the TSVs in 3D FPGA designs. Figure 1 demonstrates a switch box for 3D ICs that is architecturally formed by extending a regular 2D switch box to include two vertical channels of TSVs in addition to the traditional lateral wiring channels. One key aspect in the design of a 3D switch box is determining the size of the *lateral wiring channel*, W_W, and the size of the *vertical TSV channel*, W_V. The sizes of these channels will play key roles in determining the routability, performance, and die area of the 3D FPGAs.

The size of the vertical TSV channel is determined by a number of factors, including: (1) the number of die in the 3D stack; (2) the allocation of computational resources across the different die; and (3) the expected inter-die communication which depends on the application circuit programmed in the FPGA as well as the placement and routing tool. The exact size of the vertical TSV channel is determined by using a graph-theoretic approach that we describe with the help of Figure 2. The figure shows a possible partitioning of a heterogeneous system into five parts, where each partition should be mapped to a different die. An edge that entirely falls within a partition will utilize within-die wiring fabric for routing, while an edge that straddles two partitions (or dies) will require wiring resources within the two die at its end points as well as a number of TSVs.

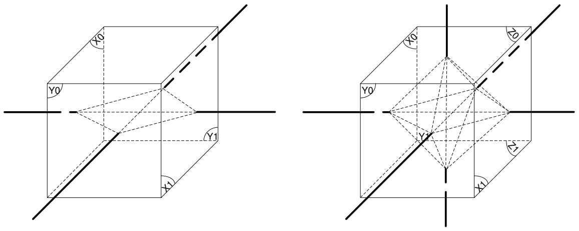


Figure 1: 2D switch vs. 3D switch.

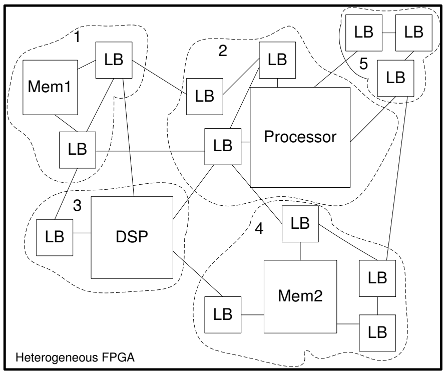


Figure 2: A partitioning of a heterogeneous FPGA.

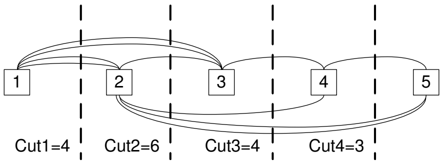


Figure 3: The maximum cut in the linear arrangement of the partitions determine the size of the TSV vertical channel width.

To determine the vertical channel size for the example of Figure 2, we create a graph, in Figure 3, composed of 5 nodes, where each node corresponds to one partition in Figure 2. The edges between the nodes in Figure 3 correspond to the edges that straddle across the partitions in Figure 2. The number of TSVs between two arbitrarily adjacent dies are different but the switch boxes should have a capability to handle maximum inter-die communication requirements (maximum cut) at any location. To estimate the maximum cut, consider the cut l between die l and die $l+1$.

$$E_l = \sum_{i=1}^{l} T_i \; - \sum_{i=1}^{l} \sum_{j=1, j\neq i}^{l} T_{ij}, \qquad (1)$$

where T_i is the number of TSVs connecting to die i and T_{ij}, $i \neq j$ is number of TSVs between die i and die j. For example, the *Cut2* between die 2 and die 3 is calculated as:

$$E_2 = T_1 + T_2 - T_{21} = 4 + 4 - 2 = 6 \qquad (2)$$

T_i and T_{ij} can be computed by using placed and routed benchmark designs or using statistical estimation based on Rent's rule. In this study, due to lack of placement and routing tools supporting 3D heterogeneous architectures, we utilize the heterogeneous Rent's rule [18] for estimation. Consequently the vertical channel width, W_V, should be equal to

$$W_V = \frac{\max\{E_1, E_2, \ldots, E_{m-1}\}}{\text{Number of switch boxes}}. \qquad (3)$$

Assuming that the pitch width of the TSVs is p then the area allocated for TSVs per switch box is equal to $p^2 W_V$.

3.3 Estimating Total Routing Area

To estimate the total chip area, in this section we present the estimation model derived from [12, 8] for the reconfigurable routing components, which includes the connection blocks and the switch boxes. The area occupied by these routing components depends on their architecture and the width of interconnect channels (*i.e.*, both the size of the within-die lateral wiring channels and the size of the vertical TSV channels). The areas for these routing components can be determined as follows:

Connection Blocks: The connection blocks consist of programmable switches that connect I/O pins of logic clusters to lateral channels, as shown in Figure 4. The size of the I/O connection blocks is determined by the *fan-in connection factor*, F_{ci}, and the *fan-out connection factor*, F_{co}, which gives the fraction of wiring tracks in a lateral channel to which each input pin and output pin can connect to, respectively. The area of a connection block can be calculated by first counting the number of buffers, pass transistors, and multiplexors required for it, and then summing the areas of these elements as outlined by [8].

Switch Boxes: The interconnect routing switch boxes (Figure 4) consist of switch points which can be implemented by programmable tri-state buffers or pass transistors to connect interconnect segments together. Therefore the size of a routing switch box depends on the size and number of switch points required for that switch box. In a switch box that consists of W_W wiring tracks, the number of switch points, S_{2D}, can be computed as:

$$S_{2D} = \frac{W_W F_s(F_s + 1)}{2}, \tag{4}$$

where F_s is the maximum allowable fanout for an incoming wire segment into the switch box [12, 8]. In contrast to a 2D switch box, a 3D switch box accommodates four lateral wiring channels (each consisting of W_W wiring tracks) and two vertical channels (each consisting of W_V TSVs), as shown in Figure 1. Since W_V is not necessarily equal to W_W, there could be only W_V tracks among the W_W tracks in each lateral wiring channel that can be connected to the W_V tracks of the TSVs. Furthermore, the maximum fanout of an incoming TSV, F_{s_v}, could be different from the maximum allowable fanout of in incoming wire segment, F_{s_w}. Thus, the number of switch points in a 3D switch box, S_{3D}, can be generally computed as

$$S_{3D} = \frac{(W_W - W_V)F_{s_w}(F_{s_w} + 1) + W_V F_{s_v}(F_{s_v} + 1)}{2}. \tag{5}$$

We determined vertical TSV channel width, W_V, using Equation (3). Thus, we need to estimate W_W in order to determine the number of switches and area of the 3D switch box. That is,

$$W_W = \frac{L_{total}}{N_{ch} \times e_t}. \tag{6}$$

This equation is based on an assumption that for any design, the total length of utilized wiring tracks $W_W \times N_{ch} \times e_t$ is equal to the required total wire length L_{total}. The value N_{ch} is the number of channels and e_t is the utilization factor of the wiring track. In island-style FPGAs, N_{ch} is the number of lateral wiring channels which depends on the number of logic blocks per die and e_t is a constant (typically around $0.4 - 0.5$) [8].

To calculate W_W using Equation (6) we need to compute the total required wirelength L_{total}. In this paper we utilize the estimation model for the global wiring requirement for heterogeneous networks developed by Zarkesh-Ha *et al.* [18], where the total wire length is computed as

$$L_{total} = \sum_{d=1}^{q} \bar{L}_d \times I_d, \tag{7}$$

where q is the maximum fan-out of the netlist, $\bar{L}_d$ is the average length of a net having fanout of d and I_d is the number of nets having fanout of d. Hence, to estimate the total wirelength of a heterogeneous system, $\bar{L}_d$ and I_d have to be computed for every $d = 1 \ldots q$. I_d and $\bar{L}_d$ can be computed using the following two Rent-rule based approximations (derivations of these formulas can be found in [18]):

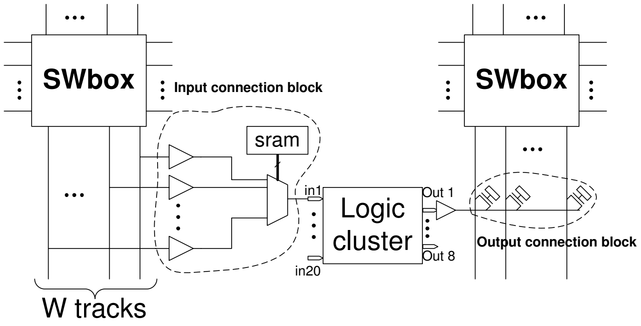


Figure 4: A typical SRAM-based Island Style FPGA

$$I_d = k_{eq} N \frac{(d-1)^{p_{eq}-1} - d^{p_{eq}-1}}{d} \tag{8}$$

The average length of a net having fanout of d is then

$$\bar{L}_d \approx 2(\alpha.d^{\gamma} - \beta + 1)\sqrt{\frac{A}{N}[d\eta_p + N(1 - \eta_p)]}, \tag{9}$$

where keq and p_eq are equivalent Rent's parameters of the system; α, β and γ are the empirical coefficients (we use $\alpha = 1.1, \beta = 2.0$ and $\gamma = 0.5$ in our experiments); η_p is the placement efficiency parameter that depends on the placement tool; and A is total die area [18]. To compute the total interconnect area using Equation (9), one needs to know the total die area A. However, to calculate the total area A one needs to first calculate the interconnect area! To break this circular dependency, we use a numerical search to find the smallest A that, when plugged in Equation (9), eventually gives an interconnect area that leads back to the same total die area A.

3.4 Interconnect Channel Width vs. Number of 3D Dies

In the previous subsections we have discussed how to calculate the lateral wiring channel width W_W and vertical TSV channel width W_V. We next explore the relationship between these channel parameters and the number of 3D dies and the impact of this relationship on total die area and performance.

We explain our methodology by means of an example. Consider the makeup of a heterogeneous FPGA that consists of 2000 logic clusters, 18 DSP blocks, 144 Mem1 memory blocks, and 202 Mem2 memory blocks. Assume a partitioning configuration that allocates the components equally among the different dies. For now, we do not discuss the details of our experimental setup (e.g., technology assumed, etc.); we disclose this information in Section 5. The result in Figure 5 shows that the lateral channel width decreases as the number of die increases, while the vertical channel width increases as number of die increases. As expected, the increase in number of 3D dies leads to a higher portion of lateral wiring segments being replaced by TSV; thus, the number of TSVs increases as number of dies increases. The increase in number of TSVs initially yields benefits in terms of performance and die area saving; however, when the number of TSVs increases too much, these benefits might not be further realized due to the TSV area overhead. This issue will be discussed more in Section 5.

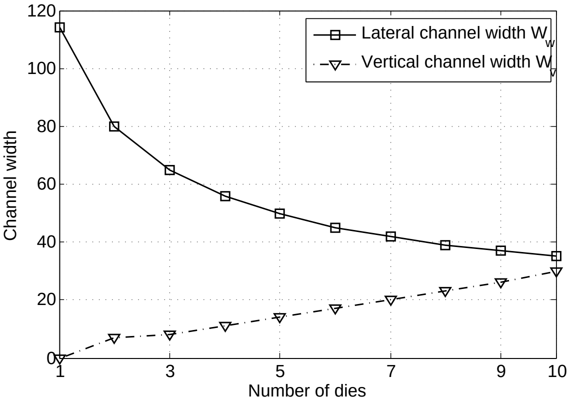
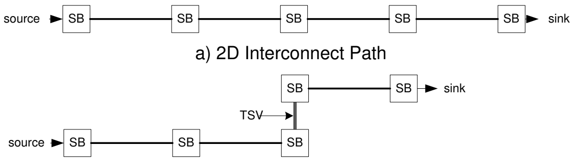


Figure 5: Interconnect Channel Width vs. Number of Dies



a) 2D Interconnect Path

b) 3D Interconnect Path

Figure 6: Interconnect path in 2D FPGAs vs. 3D FPGAs.

4. ESTIMATING IMPROVEMENTS IN PERFORMANCE

One of the important advantages of 3D technology is the general reduction in the average distance between the components of the computational system. Three-dimensional technology can substitute long interconnect paths by short ones that are "stitched" together using TSVs. This reduction in interconnect length improves the signal propagation delay between the computational resources improving the overall FPGA performance. The reductions in wire capacitance and resistance achieved from replacing long wires with TSVs are significant. The objective of this section is to estimate the improvement in signal propagation delay using our 3D FPGA design model.

To estimate the average interconnect path delay in 3D ICs, we first consider every pair of locations across all die and calculate the delay between the two locations and then calculate average delay as average of these point-to-point delays. For every pair of locations, we calculate the distance between them and then estimate the number of L4 and L16 wire segments that would be used to create an interconnect path between the two locations. If the two locations end up on the same die, then the delay of the path between them is calculated using a distributed RC delay model of its path constituents (i.e., the L4/L16 wire segments and the pass transistors in the intermediate switch boxes (*SB*), as shown in Figure 6(a)). If the two locations end up on different die, the delay is computed for the path shown in Figure 6(b) with TSV delay taken into account. The estimation result will be shown in Section 5.

To further improve the performance of 3D FPGAs, we propose incorporating *bypass TSVs* into the switch boxes. Bypass TSVs will be used to connect non-adjacent dies directly by passing through a switch box without any interaction with any intermediate switches. A bypass TSV will not eliminate the silicon area required for the in-series TSVs in the intermediate die, but it will eliminate the delay and area that would have been introduced by intermediate switches. For 3D FPGAs, our experiments in Section 5 show that using bypass TSVs can reduce the average interconnect path delay and the die area by significant amounts.

5. EXPERIMENTAL RESULTS

In this section we empirically assess the impact of our proposed 3D FPGA architecture on total die area and performance compared to a 2D FPGA architecture. For all experiments, we estimate the area of the computational resources according to Table 1, and estimate the routing area and TSV area according to the approach outlined in Sections 3.2 and 3.3. We use the TSMC 90nm library and assume a 10× pass transistor switch, 4× wire segment buffers, and a wire resistance and capacitance of $0.244\Omega/\mu m$ and $0.208fF/\mu m$ respectively. We also assume that the I/O connection factor of connection boxs are $F_{ci} = 0.5$ and $F_{co} = 0.125$. Based on data reported in [13] we choose typical 5×5 μm^2 TSVs with resistance and capacitance values of 43 $m\Omega$ and $40fF$ respectively. We consider two different experiments:

Experiment 1: Impact of choice of partitioning configuration on the total die area.

# of Dies	Config.	Die	Resource	Die Area (cm^2)	Total Area (cm^2)
1	B	1	logic + hardcores	0.700	0.70
2	A	1	logic	0.450	0.63
		2	hardcores	0.180	
	B	1	logic + hardcores	0.290	0.59
		2	logic + hardcores	0.290	
3	A	1	logic	0.170	0.52
		2	hardcores	0.180	
		3	logic	0.170	
	B	1	logic + hardcores	0.176	0.53
		2	logic + hardcores	0.176	
		3	logic + hardcores	0.176	

Table 2: Potential 3D partitioning configuration.

Experiment 2: Impact of the number of die in the 3D stack on the total area savings and performance.

Experiment 1: Impact of Partitioning Configuration. In this experiment we consider an FPGA with a computational resource makeup based on an *Altera Stratix II EP2S30* FPGA device that consists of 2000 logic clusters, 18 DSP blocks, 144 *Mem1* and 202 *Mem2* memory blocks. We consider two partition configuration: in Configuration A, each die contains only one type of computational resources (either reconfigurable logic or hardcore units), and in Configuration B, every die contains both reconfigurable and hardcore units. In all configurations, all die have reconfigurable interconnects and switch boxes. In Table 2, we consider the application of these two partitioning configurations using either 1, 2 or 3 die in the 3D FPGA stack. The results show that configurations that lead to more balanced die areas are the ones that lead to the largest savings in the total die area. This result is not surprising; a non-balanced area distribution would lead to some die with relatively large areas. The interconnects in these larger die will tend to be relatively long, which implies more die area allocated for the reconfigurable switching boxes, triggering an increase in the total die area.

Experiment 2: Impact of Number of Die. In this second experiment, we evaluate the impact of number of die in the 3D FPGA stack on both total die area and performance as the number of die in the stack increases. We assume the same FPGA makeup as in Experiment 1 with configuration B. Earlier in Figure 5, we showed the tradeoff between the width of the vertical TSV channel (dash line) and the lateral interconnect channel (solid line) as the number of die increased. In this experiment, we show in Figure 7 that the total die area initially decreases as the number of die increases, reaching the minimum value when four die are used. However, if the number of die is further increased, the total die area does not continue to decrease, but rather increases. This increase happens because when many die are used, a larger number of TSVs are required for intercommunication, and the area of these TSVs end up dominating the total die area. The plot of Figure 7 also shows that the average interconnection delay initially decreases as the number of die increases achieving a minimum at 7 die per stack. When the design is further mapped to a larger number of die, the average delay increases (or equivalently performance decreases) as the vertical TSV interconnects tend to replace the local and medium wires, increasing the average interconnection delay. Another reason not to increase the number of die in the stack beyond a certain point is to avoid potential thermal problems. Thus, for this FPGA makeup a 3D stack of four die would achieve silicon area savings of 26% (from 0.71 cm^2 to 0.52 cm^2) compared to the planar design, together with an improved performance of 61% (from 11.8ns to 4.6ns) as measured by the average interconnection delay.

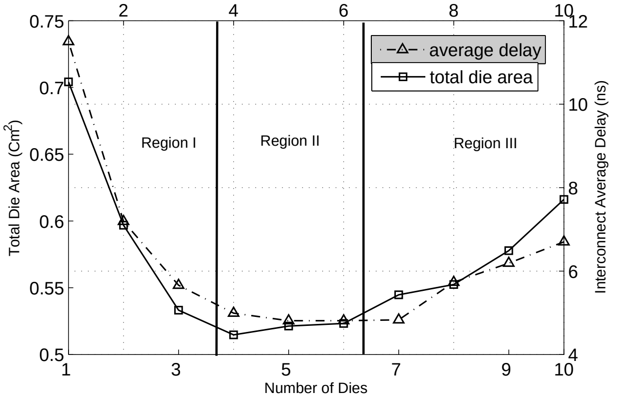


Figure 7: 3D total die area and average connection delay vs. number of die. The left y-axis gives the total area in cm^2 and the right y-axis gives the average connection delay in ns.

We also estimate the impact of using bypass TSVs between non-adjacent dies, as presented in Section 4. The result shows that by using bypass TSVs the reductions in die area and average delay can be improved more 4.63% and 9.78% respectively.

The common trends between the tested designs lead to an intuitive explanation for the impact of transforming planar FPGA designs to use 3D technology. If we denote the optimal number of die from a pure area savings perspective as m_a and the optimal number of die from a pure delay (or performance) perspective as m_p, then from our results we can identify three regions for 3D FPGA design.

Region I (less than m_a die) : For a small number of die in the stack, TSVs eliminate the long interconnections which significantly reduces delay and the reconfigurable switching logic overhead, leading to substantial area savings despite the silicon area overhead required for implementing the TSVs.

Region II (from m_a die to m_p die): If the number of die is further increased, the TSVs will start replacing the medium to long wires bringing only modest improvements to performance, while slightly increasing the total die area.

Region III (more than m_p die): If the number of die is unreasonably increased, then the TSVs will end up replacing the short to medium wires, increasing the average wire delay, which will lead to deterioration in performance. Furthermore, the area required for TSVs will start to dominate significantly over component and reconfigurable routing area, leading to an increase the total die area.

The experimental results show that by transforming existing planar FPGA designs to 3D technology, it is possible to attain significant total die area reductions, while attaining the expected performance benefits of 3D technology. The total die area reduction leads to direct cost savings that provide an important incentive for the industry to bring 3D IC technology into the market.

6. CONCLUSION AND FUTURE WORK

In this paper we have proposed new architectures and design methodologies for heterogeneous 3D FPGAs with TSVs. The performance benefits as well as the cost savings incurred from using such 3D systems have been analyzed. We have also estimated the impact of 3D integration on both the total TSV area and the total area of the reconfigurable routing resources. We showed that as the number of die in a 3D stack increases, the total interconnect area reduces and the total TSV area increases. We have investigated the optimal number of die that gives the greatest savings in die area. We have estimated the improvement in performance that will be attained by switching to 3D technology, and we have analyzed the performance benefits of using heterogeneous FPGAs with regular TSVs and bypass TSVs. Using Rent-based statistical analysis, we have shown that 3D FPGAs can reduce die area by about 27% while simultaneously improving performance by up to 58%. Though statistical-based estimation might cause variations compared with realistic benchmark designs, the experimental results are consistent with theoretical analyses.

Finally, for future work, we would like to develop a 3D heterogeneous placement and routing tool to conduct experiments on benchmark designs to evaluate our statistical estimation model. Analyzing the impact of 3D stacking on thermal distribution of 3D heterogeneous FPGAs also would be considered.

7. REFERENCES

[1] K. Banerjee, *et. al.*, "3-D ICs: A Novel Chip Design for Deep-Submicrometer Interconnect Performance and Systems-on-Chip Integration," *Proc. of the IEEE*, vol. 89(5), pp. 602–633, 2001.

[2] A. W. Topol, *et. al.*, "Three-dimensional Integrated Circuits," *IBM Journal of Res. and Dev.*, vol. 50(4-5), pp. 491–506, 2006.

[3] M. Alexander, *et. al.*, "Three-dimensional field-programmable gate arrays," *ASIC Conference and Exhibit, 1995., Proc. of the Eighth Annual IEEE International*, pp. 253–256, Sep 1995.

[4] W. Meleis, *et. al.*, "Architectural design of a three dimensional FPGA," *Advanced Research in VLSI, 1997. Proc., Seventeenth Conference on*, pp. 256–268, Sep 1997.

[5] G. Borriello, *et. al.*, "The triptych FPGA architecture," *VLSI Systems, IEEE Transactions on*, vol. 3, no. 4, pp. 491–501, Dec 1995.

[6] M. Lin, *et. al.*, "Performance benefits of monolithically stacked 3D-FPGA," in *Proc. of the ACM/SIGDA 14th ISFPGA*. New York, NY, USA: ACM, 2006, pp. 113–122.

[7] A. J. Alexander, *et. al.*, "Placement and routing for three-dimensional FPGAs," in *Fourth Canadian Workshop on Field-Programmable Devices*, 1996, pp. 11–18.

[8] A. Rahman, *et. al.*, "Wiring requirement and three-dimensional integration technology for field programmable gate arrays," *VLSI Systems, IEEE Transactions on*, vol. 11, no. 1, pp. 44–54, Feb 2003.

[9] C. Ababei, *et. al.*, "Three-dimensional place and route for FPGAs," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 25, no. 6, pp. 1132–1140, June 2006.

[10] Y.-S. Kwon, *et. al.*, "A 3-D FPGA wire resource prediction model validated using a 3-D placement and routing tool," in *Proc. of SLIP '05*. New York, NY, USA: ACM, 2005, pp. 65–72.

[11] M. Lin, *et. al.*, "A routing fabric for monolithically stacked 3D-FPGA," in *Proc. of the ACM/SIGDA 15th ISFPGA*. New York, NY, USA: ACM, 2007, pp. 3–12.

[12] V. Betz, *et. al.*, *Architecture and CAD for Deep-Submicron FPGAs*. Norwell, MA, USA: Kluwer Academic Publishers, 1999.

[13] Vasilis F. Pavlidis, et. al, *Three Dimensional Integrated Circuit Design*. Morgan Kaufman Publishers, 2008.

[14] S. Wilton and N. Jouppi, "Cacti: an enhanced cache access and cycle time model," *Solid-State Circuits, IEEE Journal of*, vol. 31, no. 5, pp. 677–688, May 1996.

[15] Cacti 5.3, Online, available at: http://quid.hpl.hp.com:9081/cacti/index.y?new.

[16] "Altera stratix ii device handbook, volume 1," http://www.altera.com/literature/hb/stx2/stratix2_handbook.pdf.

[17] B. Landman and R. Russo, "On a pin versus block relationship for partitions of logic graphs," *Computers, IEEE Transactions on*, vol. C-20, no. 12, pp. 1469–1479, Dec. 1971.

[18] P. Zarkesh-Ha, *et. al.*, "Prediction of net-length distribution for global interconnects in a heterogeneous system-on-a-chip," *VLSI Systems, IEEE Transactions on*, vol. 8, no. 6, pp. 649–659, 2000.