



**HAL**  
open science

## Architecture level optimization of 3-dimensional tree-based FPGA

Vinod Pangracious, Emna Amouri, Zied MARRAKCHI, Habib Mehrez

► **To cite this version:**

Vinod Pangracious, Emna Amouri, Zied MARRAKCHI, Habib Mehrez. Architecture level optimization of 3-dimensional tree-based FPGA. *Microelectronics Journal*, 2014, 45 (4), pp.355-366. 10.1016/j.mejo.2013.12.011 . hal-00944759

**HAL Id: hal-00944759**

**<https://hal.science/hal-00944759v1>**

Submitted on 17 Feb 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Architecture Level Optimization of 3-Dimensional Tree-based FPGA

Vinod Pangracious<sup>a,1,\*</sup>, Emna Amouri<sup>a,2</sup>, Zied Marakchi<sup>b,3</sup>, Habib Mehrez<sup>a,4</sup>

<sup>a</sup>*LIP6/ University of Pierre et Marie Curie*

<sup>b</sup>*FlexRas Technologies Paris France*

---

## Abstract

We describe a methodology to design and optimize Three-dimensional (3D) Tree-based FPGA by introducing a break-point at particular tree level interconnect to optimize the speed, area, and power consumption. The ability of the design flow to decide a horizontal or vertical network break-point based on design specifications is a defining feature of our design methodology. The vertical partitioning is organized in such a way to balance the placement of logic blocks and switch blocks into multiple tiers while the horizontal partitioning optimizes the interconnect delay by segregating the logic blocks and programmable interconnect resources into multiple tiers to build a 3D stacked Tree-based FPGA. We finally evaluate the effect of Look-Up-Table (LUT) size, cluster size, speed, area and power consumption of the proposed 3D Tree-based FPGA using our home grown experimental flow and show the horizontal partitioned 3D stacked Tree-based FPGA with LUT and cluster size 4 has the best area-delay product to design and manufacture 3D Tree-based FPGA.

*Keywords:* 3D Integration, Tree-based FPGA, Placement, Partitioning, Routing, Butterfly-fat-tree

---

---

\*Corresponding Author

<sup>1</sup>PhD Student at Laboratoire d'Informatique de Paris VI

<sup>2</sup>Post Doctoral Fellow at Laboratoire d'Informatique de Paris VI

<sup>3</sup>Chief Technology Officer at FlexRas Technologies Paris France

<sup>4</sup>Professor at Laboratoire d'Informatique de Paris VI

## 1. Introduction

The modern Field Programmable Gate Arrays (FPGAs) have become a viable alternative to cell-based design technology by providing re-configurable computing platforms with improved performance and higher density. While the re-configurability provides flexibility, FPGA also leads to area and performance overhead in comparison to cell-based application specific integrated circuits (ASICs). With the development of sub-100-nm CMOS technologies, the design and manufacturing cost of cell-based implementation have become exorbitant for most ASICs, making FPGA increasingly popular for prototype designs. However current FPGA architectures cannot meet the speed and area requirements of many ASIC due to their high programming overhead.

To provide the required reconfigurable functionality, FPGA provide a large amount of programmable interconnect resources and it consumes 90% of the total FPGA area (A. Rahman et al., 1990; M. Lin et al., 2006). Since die area is one of main factors that determine the manufacturing costs, reducing the silicon footprint of the programmable routing resources can lead to significant improvement in speed, area, power consumption and manufacturing cost to an interconnect dominated FPGAs. Three-dimensional integration (3D) is a promising technology for reducing interconnect length (R. Reif et al , 2002). It involves stacking of multiple silicon dies or wafers interconnected using Through Silicon Vias (TSV). The 3D technology using vertical interconnects (TSVs) (V. Pavlidis et al., 2006) has the potential to reduce the programmable interconnects length by bringing the logic components close together, which leads to significant improvement in functionality, scale of integration, silicon area and speed of integrated circuits provided that the devices are efficiently packed, placed and wired. There are many different 3D integration technologies presented in literature, but the most appealing techniques to date are those involving either low-temperature silicon epitaxy or wafer bonding. In an interconnect dominated FPGA, 3D integration can address problems pertaining to routing congestion, limited I/O connections, low resource utilization, and long wire delays. Recently Xilinx developed a 65nm passive silicon interposer based 2.5D high density 28nm heterogeneous FPGAs (R. Chaware et al , 2012). The passive silicon interposer provide large wiring density interconnection, minimize coefficient of thermal expansion (CTE) mismatch between the Cu/low-k die and copper filled TSV interposer, and improve chip performance due to shorter interconnection from chip to the substrate. However this type of design and manufacturing

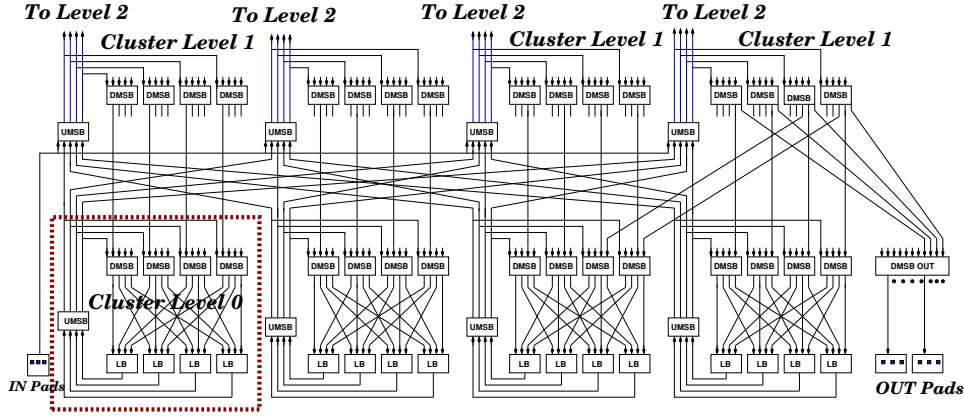


Figure 1: A two-level Tree-based Multilevel FPGA interconnect with Upward and Downward BFT based interconnection network topology

methods fails to achieve true 3D chip performance in terms of speed, power consumption and silicon area reduction.

A true 3D integration technology can lead to significant reduction in wire length and interconnect delay by using TSVs (R. Reif et al, 2002). A number of recent publication proposed novel 3D architectures and design methodologies that lead FPGA with better performance than existing planner FPGAs (A. Rahman et al., 1990; M. Lin et al., 2006; C. Ababei et al., 2006; K. Siozios et al., 2011). There are two major types of 3D FPGA architectures found in the literature. The first one is developed by monolithic stacking, whereby the active devices are lithographically built in between metal layers (M. Lin et al., 2006) and the second type is evolved from original 2D structure by extending the 2D switch boxes (SBs) to 3D ones (K. Siozios et al., 2011; C. Ababei et al., 2006). So far, there are two design and exploration frameworks targeting 3D FPGA architectures: the three-dimensional place and route (TPR) (C. Ababei et al., 2006) and 3D MEANDER (K. Siozios et al., 2011). In TPR, all SBs are assumed to be 3D-SBs and the number of TSVs is assumed to be unlimited, which is an impractical assumption as far as design and manufacturing of 3D chips is concerned. Meanwhile 3D MEANDER is a fully-fledged design framework for 3D FPGAs and it provides the capability to analyze the impact of different deployment strategy for 3D-SBs in multi-tier FPGAs. It proposes various 3D FPGA architectures and design styles in which 2D-SBs and 3D-SBs are intermittently used in certain regular spatial patterns. Nonetheless the number of available TSVs within

3D-SBs is assumed to be fixed and that means the methodology does not investigate the impact of different numbers of TSVs in a 3D-SB. A dynamically re-configurable 3D FPGA is presented in (S. Chiricescu et al , 2001), which consisted of three physical layers: logic blocks along with local interconnects, programmable interconnects layer and memory layer. The performance analysis of a monolithically stacked 3D FPGA using three physical layers presented in (M. Lin et al., 2006).

## 2. Motivation And Problem Formulation

According to (K. Siozios et al., 2011; C. Ababei et al., 2006) the SBs has been the most area-consuming unit compared to other design elements in 2D FPGAs and this situation is becoming even worse in 3D FPGAs because the TSVs are located on 3D-SBs. Although the design and manufacturing engineers are trying to reduce TSV dimensions, the minimum feature size on the die is also shrinking. Therefore, the TSVs are expected to remain larger than wire dimensions in metal layers within the die (S Gupta et al , 2005). Moreover it has been reported in (Cha-I Chen et al , 2011) that the TSV utilization is actually quite low if the 3D-SBs are with full vertical connectivity in use. The experiments carried out in our laboratory and recent publications point out that the utilization of TSVs is actually very low in 3D Mesh-based FPGAs (Cha-I Chen et al , 2011) with full vertical connectivity, which motivates us to explore new architectures that can be better optimized to achieve higher speed, reduced power consumption, area and to increase logic density. In this paper, we prefer to use a Tree-based multilevel FPGA architecture, because from our experimental and design experience, we believe, due to the multilevel Butterfly Fat-Tree (BFT) based interconnect topology, Tree-based FPGA is a better architecture style to build high density 3D re-configurable systems compared to Mesh-based industrial FPGAs. In a Tree-based FPGA architecture (Z. Marrakchi et al., 2009, 2005, 2006), the programmable interconnects are arranged in a multilevel network with the switch blocks placed at different tree levels and the Logic Blocks (LBs) are grouped into clusters located at different levels. Due to the multilevel network arrangement, we do not have to deal with 3D SBs in the case of Tree-based FPGA, rather all switch blocks remain as 2D and only the interconnects are partitioned between multi-tiers and interconnected using TSVs.

In a Tree-based FPGA architecture (Z. Marrakchi et al., 2009), the Logic Blocks (LBs) are grouped into clusters located at different levels. Each clus-

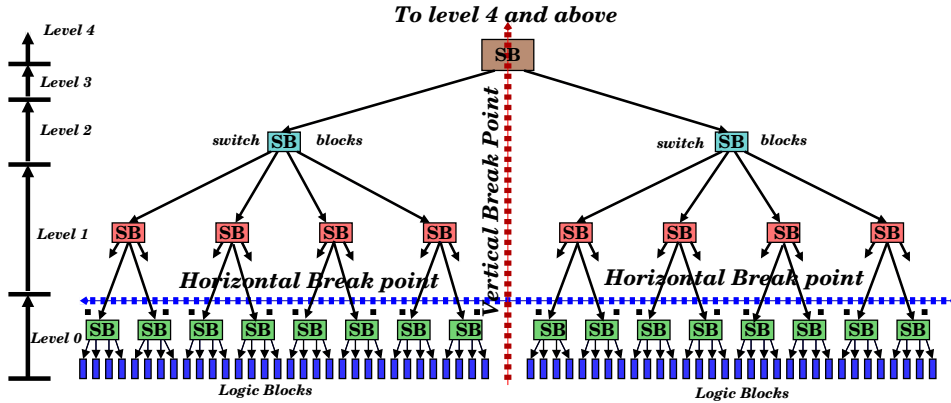


Figure 2: A three-level Tree-based FPGA interconnect network break point representation: Horizontal break-point: blue dotted line, Vertical break-point: red dotted line

ter contains a switch block to connect local LBs. Figure 1 illustrates a 2 level arity 4 Tree-based FPGA architecture. The switch blocks are divided into Mini Switch Blocks (MSBs). The Tree-based FPGA architecture unifies two unidirectional upward and downward interconnection networks using a BFT based network topology to connect Downward MSBs (DMSBs) and Upward MSBs (UMSBs) to LBs inputs and outputs. Design and implementation of two-dimensional layout for Tree-based FPGA is a challenging task, since the interconnect delay increases exponentially as the tree grows to higher levels (Z. Marrakchi et al., 2009). As illustrated in Figure 2, we propose two innovative 3D stacking methodologies using vertical or horizontal network partitioning to improve density and network delay of 3D Tree-based FPGA. Figure 2 shows a 3 level, arity 4 Tree-based FPGA architecture with horizontal and vertical break-point. In the case of *horizontal partitioning* the tree-based programmable interconnect network is horizontally partitioned at a particular tree level called the *break-point* and interconnected using TSVs to optimize network delay. In this case the logic density and interconnects below the break-point will be placed in active layer 1 and the interconnect networks above the break point will be placed at active layer 0 of the 3D stacked chip. On the other hand, the *vertical partitioning*, as illustrated in Figure 2, the hardware positions are fixed. The logic units and interconnect networks are placed equally on multiple active layers of the 3D stacked chip. Thus the silicon area and power consumption of the active layers are balanced and design complexity is reduced. The horizontal partitioning method

provide higher speed and additional design flexibility to optimize the programmable network delay and inter-layer heat dissipation of the 3D chip.

### 3. Summary of Results and Outline of the Paper

In this article we focus on performance optimization of programmable interconnects networks that are placed in multiple active layers of the horizontal or vertically partitioned design methodology to design and manufacture a high-performance 3D Tree-based FPGA. The main contribution of the article as follows. We propose innovative design and exploration methodologies to improve the speed and density of 3D Tree-based FPGA using vertical and horizontal break-points of tree-based programmable interconnect networks. Using Rent-based analytical wire length distribution models, we propose a methodology to optimize total count and area of TSVs and programmable routing resources. Using an extensive sets of benchmarks, we analyze the speed, area, power consumption and the effect of LUT and cluster size of the 3D stacked Tree-based FPGA. Using a comprehensive experimental setup we show that the 3D homogeneous Tree-based FPGA provides 65.13% improvement in speed and reduces 36% interconnect network area compared to 2D Mesh-based planer FPGA. This article is organized as follows. Section 4 describes the 3D Tree-based FPGA experimental and design methodology. Section 5 describes the experimental results. Section 6 presents the impact of LUT and cluster size of Tree-based FPGA architecture on performance. Section 7 explains power optimization methodology of 3D Tree-based FPGA. Section 8 describe 3D Thermal modeling and analysis of Tree-based FPGA architecture and finally section 9 concludes the article.

### 4. Experimental Flow

The proposed experimental flow for design and exploration of 3D Tree-based FPGA architecture is illustrated in Figure 3. The HDL code generator is designed to generate VHDL code based on a hierarchical design approach that partitions the design into smaller sections, implements them separately and assembles them together at the final design phase. The physical design experiments are performed using the layout generated using ST Micro’s 130nm technology node (V. Pangracious et al., 2013). Mentor’s circuit simulator *Eldo* is used to estimate the wire delay and power consumption of switches and interconnection networks at different tree levels.

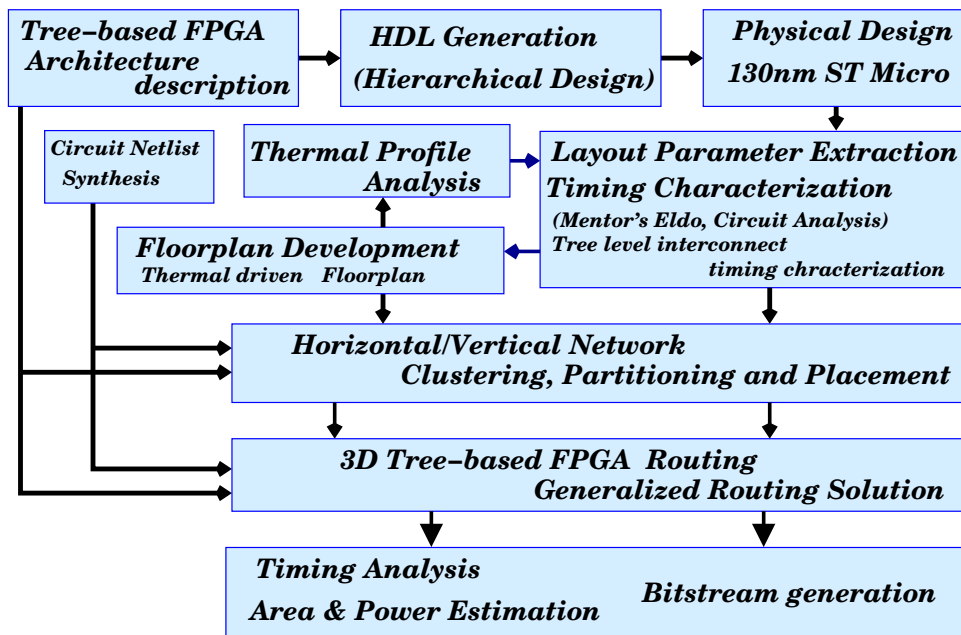


Figure 3: An integrated design and evaluation software flow for 3D Tree-based FPGA: VHDL generation, physical design, thermal model along with placement and routing

#### 4.1. 3D Physical Design Methodology

The physical design process begins with the RTL description of Tree-based FPGA generated using VHDL code generator as illustrated in section 4. Figure 4 presents the 3D physical design flow used in the design of 3D Tree-based FPGA. Based on the type of partitioning being used, the design is partitioned into two independent designs (tier 0 and tier 1). In the case of horizontal partitioning, tier 1 contains LUTs and local programmable interconnects from levels 0 to 3 (design2) and tier 0 contains programmable interconnect above the break-point along with IOs (design1) and for vertical partitioning total logic blocks and interconnect are partitioned equally into two designs. We then used cadence design compiler to compile VHDL into structural Verilog for each die. The compiled Verilog is then input into Cadence Encounter to perform semi-automated physical design steps. The design tool augmented to test different 3D stacking methodologies. We used both Face-2-Face (F2F) and Face-2-Back (F2B) stacking methodology using via first TSV process. The insulation material between TSV and silicon is oxide with 1000 Å thickness. The I/O signals of the F2F stacked chip are



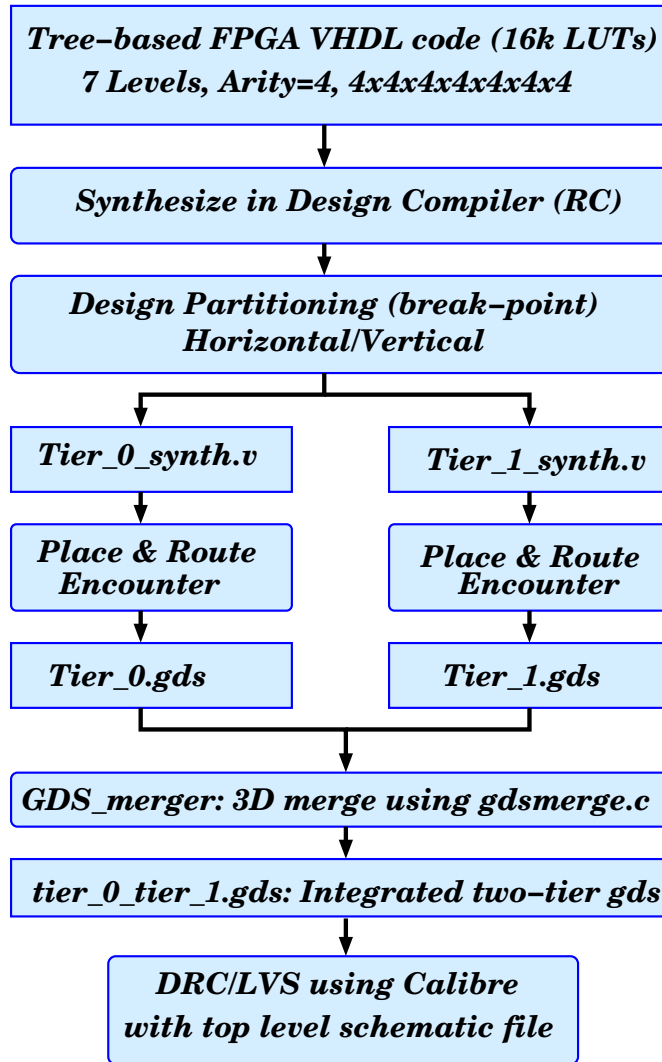


Figure 4: 3D physical design methodology developed to implement multi-tier 3D Tree-based FPGA using 2D CAD tools

routed through TSVs to the back surface of tier 0 and from there, they will be fanned out past the edge of the device to connect to I/O pads on the surface of the 3D FPGA chip, while in F2B stacking, the tier 0 via-first TSVs have their landing pads on Metal 1 and Metal 6. The connection between via-first TSVs are made using local interconnection and vias in between adjacent dies. In the case of F2F stacking wafer thinning is done after bonding, while in

F2B, the tier 0 die is thinned down to TSVs first and bonded using the TSV landing pads. These landing pads includes *keep-out-zones* uniformly located around them to reduce coupling effects on active devices located around it. We used Encounter and Caliber-LVS to perform early analysis on the design before sign-off analysis is undertaken. To perform the DRC/LVS of the two-tier 3D FPGA layout, we used a GDS-merger (c program) tool to merge two independent layout into an integrated chip layout and compare it with the top level schematic by using Calibre-LVS as illustrated in Figure 4. The merger tools interconnects those pins with same names in design1(tier0) and design2 (tier1) and no major change required in the top level schematic files to perform Calibre-LVS.

#### 4.2. Floorplanning And Thermal Analysis

The goal is to distribute the BFT based programmable interconnect levels into two active layers in order to minimize the interconnect delay and balance the temperature uniformly across the active layers of the 3D Tree-based FPGA. The multilevel BFT-based programmable interconnect network is divided at a particular level called the *break-point level* and interface nets are interconnected using TSVs to optimize the network delay at the break-point level and above. The 2-dimensional Tree-based FPGA design is partitioned based on design specification (horizontal or vertical) to form a two-tier 3D Tree-based FPGA. To generate the two-tier Tree-based FPGA floorplan, we used a thermal driven floorplanning tool (K Sankaranarayanan et al., 2005) configured with ST micro’s 130nm technology node. This tool is configured to optimize wire length and temperature of the block level floorplan of the two-tier Tree-based FPGA chip. The floorplan tool takes a list of functional blocks, areas, aspect ratios, connectivity between the blocks and power consumption of each functional blocks as inputs. For example, in the case of horizontal partitioning, we have created two floorplans: the first floorplan consists of the logic units and local interconnections up to *level 3* of the Tree-based FPGA and the second floor plan consists of programmable interconnect levels *levels 4, 5, 6*. The floorplan tool generates thermal estimations and interconnection wire delay of local and global metal layer.

For this study the communication is realized with Through Silicon Via (TSV) and electrical characterization of TSV is performed using the approach presented in (D M Jang et al., 2007). One important aspect of thermal-aware floorplanner is the trade-off between temperature and performance. We used the wire delay model associated with floorplanner to optimize the wire length.

However the floorplan solution is always a trade-off between temperature and wire delay of the blocks used in simulation. To manage this trade-off, we have taken steps during design phase to make sure the placement of high power blocks do not lead to hotspots without compromising on design performance. The floorplan tool is augmented to include the flexibility of creating horizontal or vertical break-points in the BFT based interconnect network according to the 3D Tree-based FPGA design specifications.

One of the main concerns in the design and manufacturing of 3D-ICs is heat dissipation (A. Gayasen et al , 2008). By stacking multiple active layers and increasing logic density, it become more difficult to remove the inter-tier heat. Hotspot power dissipation results in significantly higher temperatures in 3D stacked chips compared to the same power dissipation in single 2D chips. The reason for the increase in temperature is due to the reduced thermal spreading in the thinned dies on the one hand, and to the use of low thermal conductivity adhesives on the other hand. Therefor a detailed thermal analysis at the design stage is required. The floorplan tool uses 3D resistance mesh based thermal model presented in (J. Ayala et al., 2009) to extract the thermal profile of the floorplans of the two-tier 3D Tree-based FPGA. The 3D Thermal resistance mesh based multi-layer thermal model for Tree-based FPGA consider the spatial distribution of signal TSVs to control the heat transfer among different module in the multi-tier chip. The thermal model also consider the impact of TSVs material (Cu, Tungsten or doped Poly-silicon ) while estimating the temperature profile. The effective thermal conductivity of active and passive layers in 3D stacked chip is calculated by equation 1. The  $k_{cu}$  and  $K_{th}$  are the thermal conductivity of copper and silicon active layer. The heat transfer take place on those locations where Cu TSVs are placed. Using this module, the inter-layer heat transfer and thermal profile of 3D FPGA is modeled and analyzed.

$$k_{eff} = k_{cu} \cdot (TSV_{Area}) + K_{th} \cdot (Level_{BP_{Area}} - TSV_{Area}) \quad (1)$$

#### 4.3. Partitioning, Placement And Routing

Synthesis consists of translating a circuit description into gate-level representation. As presented in Figure 5, the operation is independent of the architecture. In our flow we use SIS (E. M. Sentovich et al , 1992) synthesis tool. SIS requires architecture parameters like  $k$ , the LUT input number. In our flow we use FlowMap algorithm (J. Cong and Y. Ding et al , 2000), which is included in SIS package. As presented in Figure 5, this tool depends only on LUT size and can target any interconnect topology. We use

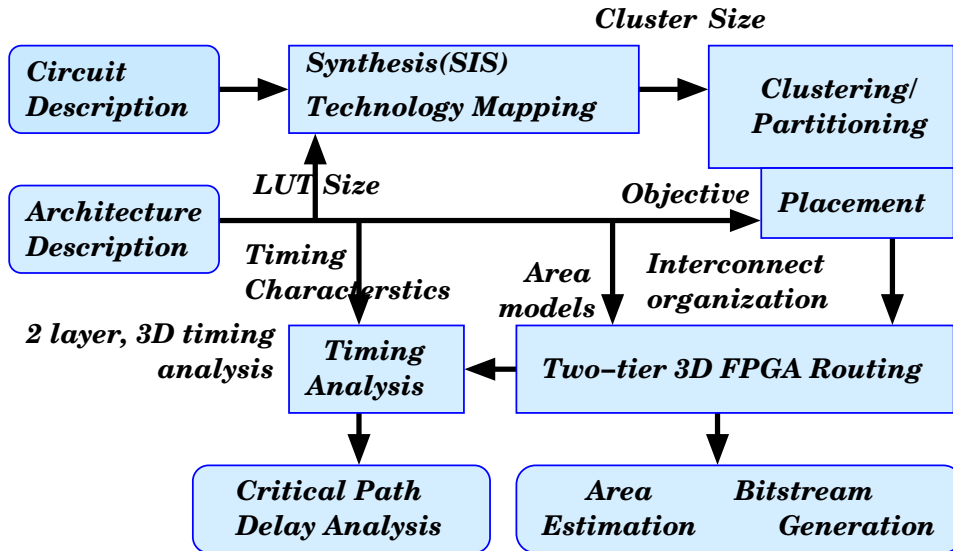


Figure 5: Architecture exploration software flow developed to evaluate the performance of 3D Tree-based FPGA

a top-down recursive partitioning and clustering approach. The aim is to reduce external communications and to collect highly connected cells into the same cluster. First, we construct the top level clusters, then each cluster is partitioned into sub-clusters, until the bottom level of the architecture is reached. Then during the placement phase, each cluster is assigned to a random position inside its owner cluster. The partitioning in each level consists of three phases. First we run a multilevel coarsening phase where the size of hypergraph is successively decreased using the first choice algorithm (N. Selvakumaran et al , 2006). Then k-way partitioning of the smaller hypergraph is computed such that the balancing constraint is satisfied. After that we run the un-coarsening phase where the partitioning is successively refined using FM algorithm (C. M. Fiduccia et al , 1982), as it is projected in the larger hypergraphs. The objective of the refinement is to minimize the hyperedge-cut, which is the total number of hyperedges that span multiple partitions. Since the structure of Tree is maintained in our two-tier 3D FPGA, the break-point will not play any role in application partitioning and placement process. However it is used during architecture optimization process. Figure 5 presents the block level representation of Tree-based FPGA architecture exploration platform.

For Tree-based architecture, the netlist obtained in .NET format first partitions the LUTs and I/Os into different clusters in such a way the inter-cluster communication is minimized. Once the netlist is partitioned into a tree of nested clusters, we attribute randomly to each cluster a position inside its owner. Since the two-tier 3D Tree-based FPGA is stacked with programmable routing resources on top of the logic blocks and interconnected using TSVs, no detailed placement is required. After partitioning and placement is done, placement file is generated, which contains positions of different blocks on the two-tier 3D stacked Tree-based FPGA architecture. This placement file along with the netlist file is then passed to 3D router, which is responsible for routing the netlist. The routing problem consists in assigning the nets that connect placed logic blocks (tier1) to routing resources in the interconnect structure (tier0). The upward interconnect adds extra paths to connect a LB to a destination but eliminates the predictability property. Hence we model the routing resources as a directed graph abstraction  $G(V;E)$ . The set of vertices  $V$  represents the in/out pins of logic blocks and the routing wires in the interconnect structure. An edge between two vertices represents a potential connection between the two vertices. The routing algorithm we implemented is *PathFinder* (L. McMurchie et al , 1995; Z. Marakchi et al., 2005, 2006), which uses an iterative, negotiation-based approach to successfully route all nets in a netlist. During the first routing iteration, nets are freely routed without paying attention to resource sharing. Two terminal nets are routed using Dijkstra’s shortest path algorithm (T.Cormen et al , 1990), and multi-terminal nets are decomposed into terminal pairs by the Prim’s minimum-spanning tree algorithm (T.Cormen et al , 1990). At the end of an iteration, resources can be congested because multiple nets use them. During subsequent iterations, the cost of using a resource is increased, taking into account the number of nets that share the resource, and the history of congestion on that resource. Thus, nets are made to negotiate for routing resources including those interconnections at the break-point.

With the help of the routing result, the different sub-paths are identified and each edge is annotated with delay of corresponding sub-path. The edges interconnect active layers of the 3D stacked Tree-based FPGA annotate corresponding TSV delay to the pins which the circuit specifies as a connection between inter-tier layers. Through this process a new direct 3D acyclic timing graph of the routed circuit is generated to evaluate the performance 3D Tree-based FPGA. In order to optimize the TSV count and routing resources, a Rent-based wire-length optimization methodology de-

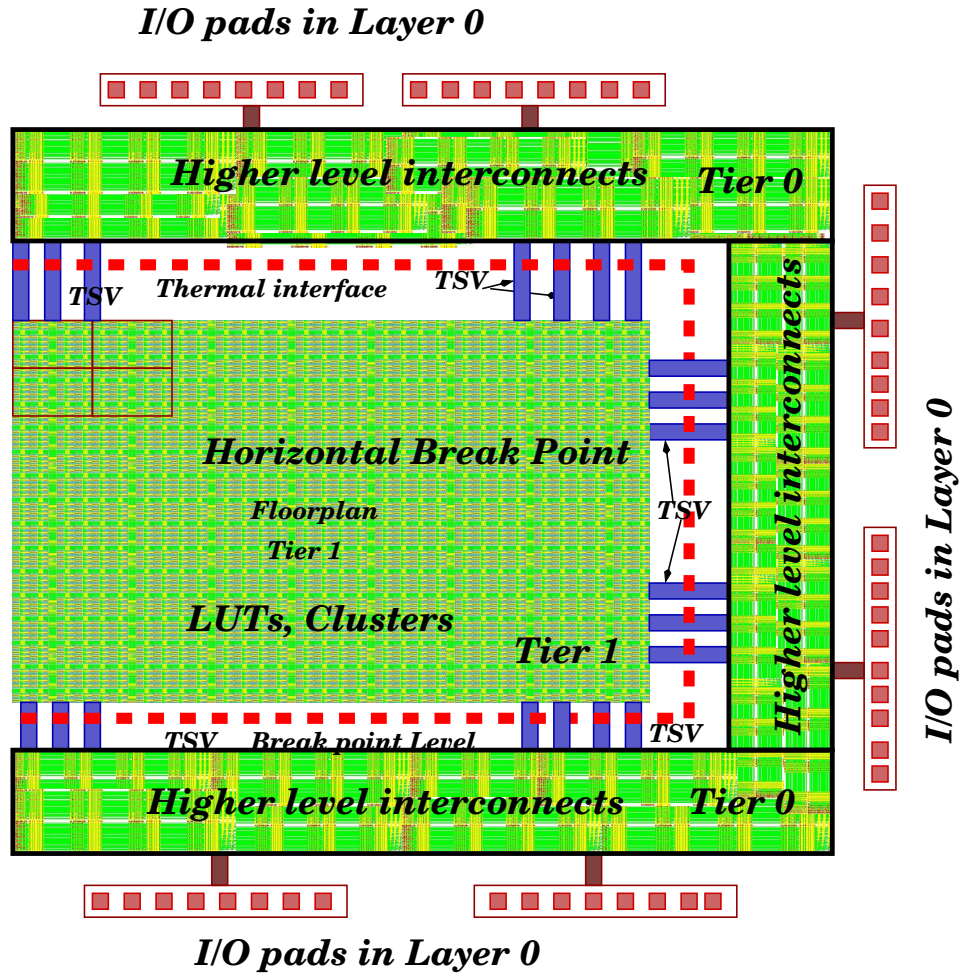


Figure 6: 3D stacked Tree-based FPGA with horizontal break point between *level 3* and *4* of the 7 levels FPGA architecture. The red dotted line indicate the horizontal break-point.

veloped using 3D router program. The optimizer first select the break-point level to optimize the TSV count and afterwards randomly chooses other tree levels to optimize routing architecture. Once the optimization is complete, the 3D router will estimate the area and static power consumption of the optimized 3D stacked Tree-based FPGA chip.

#### 4.4. Horizontal Partitioning

The location of the horizontal break-point is decided based on optimization of programmable interconnect network delay. The interconnect delay

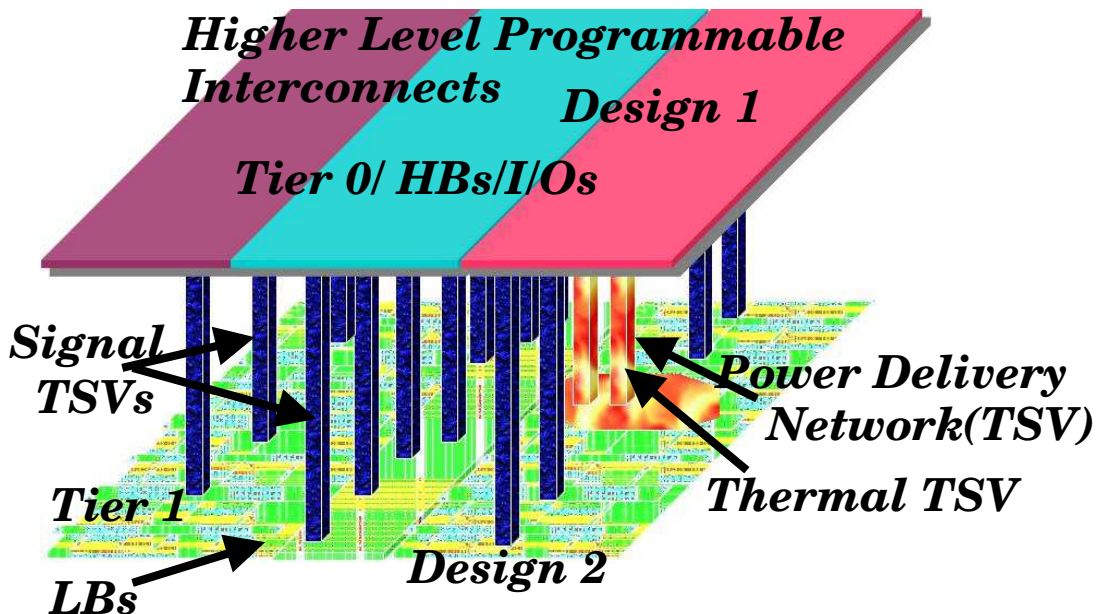


Figure 7: 3D representation two-tier 3D Tree-based FPGA with TSVs: Thermal model has the capability to include thermal TSVs or TTSVs in the simulation, but this a limited process and used only when it is necessary in multi-tier 3D designs.

of Tree-based programmable interconnects increases exponentially (Z. Marakchi et al., 2009; V. Pangracious et al., 2013) as the tree grows to higher levels. Figure 6 shows the 3D layout representation of Tree-based FPGA (V. Pangracious et al., 2013). In the case of horizontal partitioning method the LBs and local interconnects belong to levels below the break-point are placed in tier 1 and programmable interconnect resources at tree levels above break point are placed in tier 0 of the 3D stacked two-tier chip as illustrated in Figure 7. This will enable us to increase the logic density of the chip, since the logic density is completely segregated and placed in tier 1 and this design model provides additional flexibility in optimizing the interconnect delay and modeling inter-layer heat dissipation.

The setup used for wire length estimation and delay measurement using Mentor’s circuit simulator *Eldo* is reported in (V. Pangracious et al., 2013). Figure 8 shows the interconnect delays measured using 2D and 3D layouts. We used six metal 130nm process provided by ST Microelectronics that is modified to include TSVs specification. The delay measurement experiments used TSV size of  $4\mu\text{m}$  diameter and a minimum pitch of  $8\mu\text{m}$  (ITRS.,

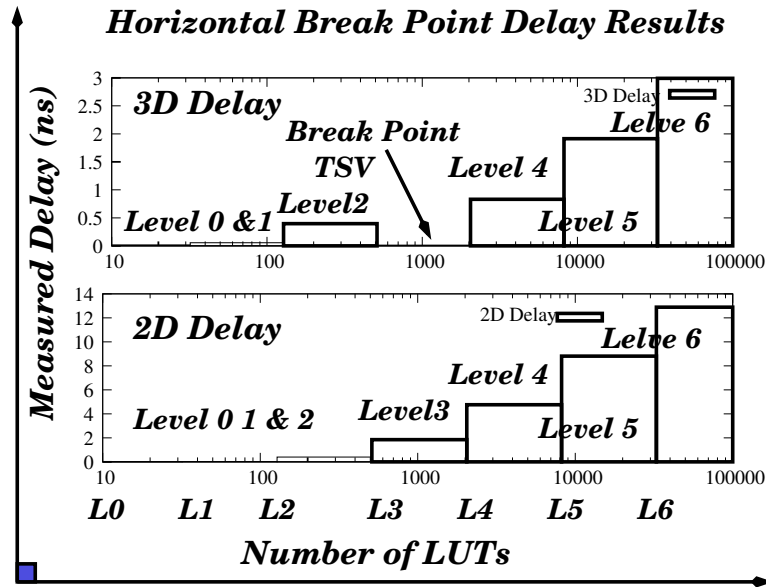


Figure 8: Horizontal break-point interconnect delay estimation of 7 level Tree-based FPGA architecture

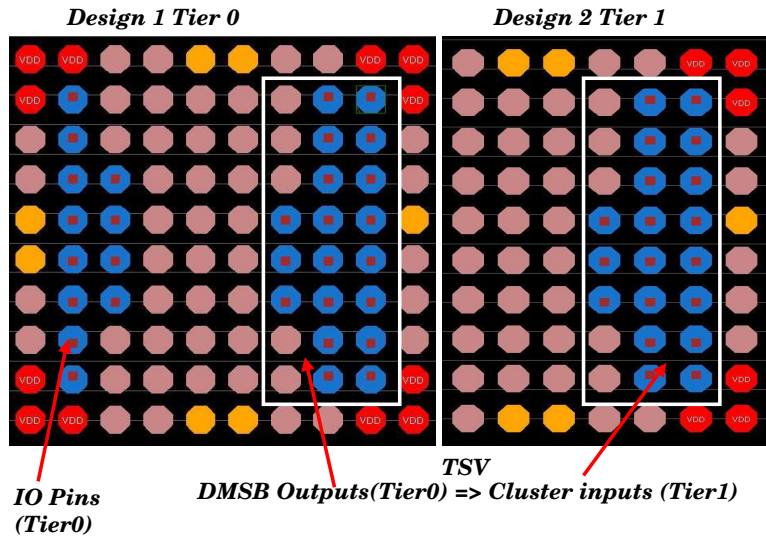


Figure 9: TSV and Direct-bond interface connection of two-tier Tree-based FPGA, placement of IOs and internal signals for tier 0 and tier 1. The integrated layout consider pins with same name on both layout as single net.



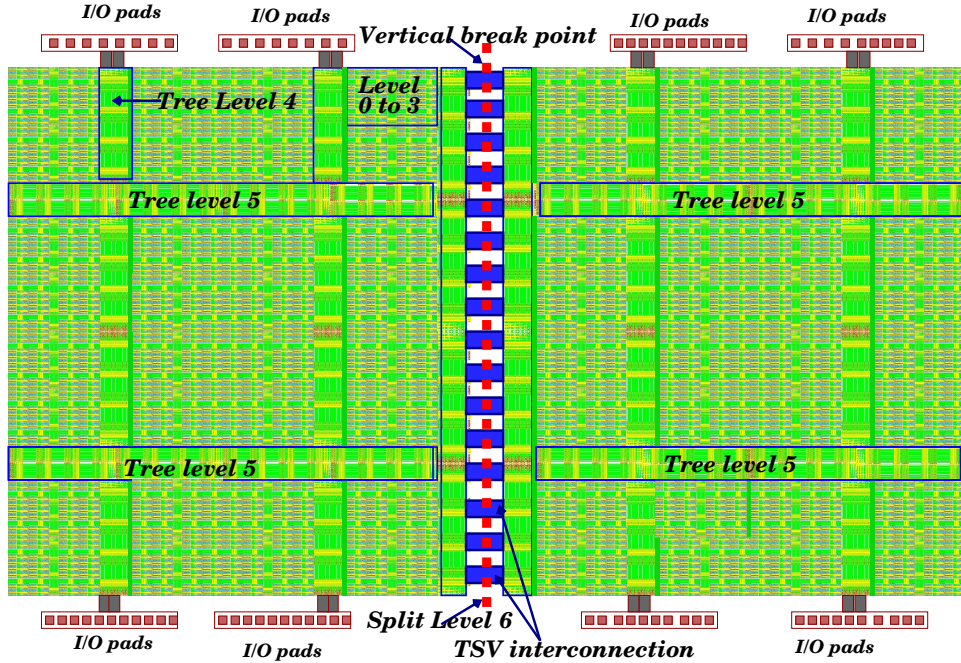


Figure 10: Representation of 3D stacked Tree-based FPGA with vertical break point

2012). The area around the TSV has been expanded to include *keep out zones* (ITRS., 2012; M. Pathak et al., 2010) to make TSVs fit within 8 standard cell area, which is essential to maintain the performance of active devices placed close to TSVs. The measured values of TSV resistance  $R_{TSV}$  is  $\approx 20m\Omega$  and capacitance  $C_{TSV}$  is  $\approx 94fF$ . The wire delay estimation of tree levels for the 3D stacked Tree-based FPGA is extracted from the floorplan using the thermally driven floorplanner (K Sankaranarayanan et al., 2005) and two-tier physical design. The break point interconnect delay is optimized using the TSV model from (D M Jang et al., 2007; K. Siozios et al., 2011). In tier 0, the locations of programmable interconnects levels are rearranged in order to optimize the wire delay at higher levels. Figure 9 shows the metal 6 TSV contact and landing pads on tier 0 and tier 1 dies.

#### 4.5. Vertical Partitioning

The main focus of vertical break-point method is to balance the total silicon area and power consumption of the Tree-based FPGA equally into the active layers of the 3D stacked chip. The total logic density and programmable routing resources are equally partitioned into multiple stacked

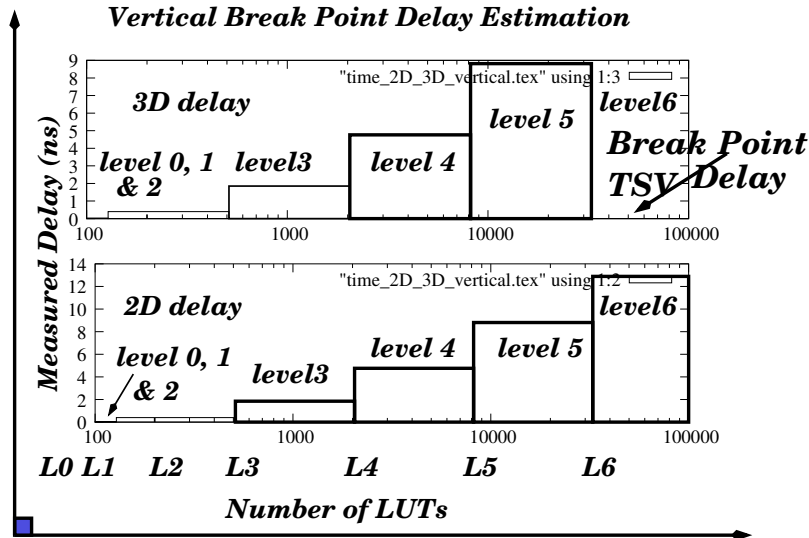


Figure 11: Vertical break-point interconnect delay estimation of 7 level Tree-based FPGA architecture

active layers. The highest level of the programmable Tree network is split vertically and interconnected using TSVs as illustrated in Figure 10. The advantages of vertical partitioning methodology compared to horizontal are balanced power consumption and silicon area in all layers of the 3D stacked chip and at same time design complexity is reduced. For the vertical partitioning method, the interconnect delay up to break-point levels is same as the 2D layout, but the largest wire length in Tree-based FPGA, which is the break point level in interconnected using TSVs and the delay has been reduced to TSV delay as illustrated in Figure 11. If we consider speed is most important design constraint, horizontal partitioning methodology is better.

## 5. Experimental Methodology

Evaluation of vertical and horizontal partitioning methodology of 3D Tree-based FPGA architecture is performed using the experimental flow described in section 4. To evaluate the performance of the proposed 3D Tree-based FPGA architecture, we place and route the largest set of 20 MCNC<sup>5</sup>

<sup>5</sup><http://er.cs.ucla.edu/benchmarks/ibm-place>.

benchmark circuits, and compare this with the 3D Mesh-based FPGA architecture (K. Siozios et al., 2011, 2012). In order to have a detailed critical path delay analysis and architecture optimization, we used both generalized and individual architecture experimentation methodologies.

### 5.1. Generalized Experimental Methodology and Result Analysis

In order to validate the performance of 3D Tree-based FPGA architecture, we have used a generalized fully connected (Rent set to 1) two-tier Tree-based FPGA architecture with 7 levels and arity 4 for each benchmarks circuits. Once the partitioning is over, the individual netlist are placed and routed using the experimental flow presented in Figure 5. The performance analysis of vertical and horizontal break point 3D Tree-based FPGA is reported in Table 1. The respective average speed improvements measured for horizontal and vertically partitioned stacking methodology are 65.13% and 43.52%. The horizontally partitioned 3D stacking methodology provides 1.5 times speed improvement compared to vertical partitioning method. The speed improvement in horizontal partitioning method is due to design optimization and minimization of interconnect wire length at the higher levels tree networks that are placed in tier 0 of the 3D stacked chip as illustrated in Figure 7. In tier 0 we have additional design flexibility to re-order programmable routing resources to optimize wire length. However in the vertical break-point method, the highest tree interconnect wire length is optimized using TSV interconnects and the rest of tree levels only limited optimization possible as illustrated in Figure 11.

The improvement in critical path delay between 3D Tree-based compared to Mesh-based FPGA is presented in figure 12. The multi-layer 3D Tree-based FPGA interconnect using TSVs shows an average of 65.13% speed improvement compared to the 2D counterpart. The 3D Mesh-based FPGA reported in (K. Siozios et al., 2011, 2012) with heterogeneous interconnect fabric using intermittent 2D and 3D switch blocks distribution with the same layout area measured an average speed improvement of 43%. In conclusion the comparison results presented in figure 12 shows horizontally partitioned 3D Tree-based FPGA is 1.5 times faster than 3D Mesh-based FPGA. The design and manufacturing solution presented in (K. Siozios et al., 2012) by using same silicon area for both 2D and 3D SBs is not piratical for high density FPGAs. This design style will increase silicon footprint of high density FPGAs, but the 3D multi-tier Tree-based FPGA with horizontal or vertical

Table 1: 3D Tree-based FPGA Performance Analysis

Tree Levels=7, Arity=4, Arch=4x4x4x4x4x4x4					
Critical Path Performance (nS)			Performance Gain(%)		
Name	Tree-based	Vertical	Horizontal	2D Vs 3D	2D Vs 3D
MCNC	2D (ns)	3D (ns)	3D (ns)	Verti (%)	Hori (%)
alu4	59.91	41.73	25.81	30.33	56.91
apex2	80.41	45.18	30.92	43.81	65.54
apex4	76.42	46.61	31.83	38.99	58.34
bigkey	79.1	27.60	20.19	65.11	74.48
clma	198.6	90.33	59.48	54.38	69.96
des	90.8	40.36	28.83	55.55	68.25
diffeq	62.6	48.46	26.66	22.59	57.41
dsip	61.9	28.55	19.78	53.88	68.05
elliptic	107.1	83.73	42.76	21.75	60.02
ex1010	143.1	74.85	45.42	47.69	68.26
ex5p	168.2	64.71	41.43	61.53	75.37
frisc	129.6	82.28	42.82	36.51	66.96
misex3	67.4	41.38	24.94	38.61	63.00
pdc	143.9	69.04	45.86	52.02	68.13
s298	130.81	81.54	45.81	37.67	64.98
s38417	75.46	43.38	30.69	42.78	59.33
s38584	118	69.54	40.51	41.07	65.67
seq	64.58	42.91	24.59	33.56	61.92
spla	109.54	58.57	38.29	46.26	65.04
tseng	131.1	70.47	45.51	46.25	65.07
Average	104.88	57.37	35.47	43.52	65.13

partitioning is more efficient as well as economical design and manufacturing methodology because in our design we have only 2D switch blocks.

### 5.2. Architecture Optimization and Result Analysis

The main objective of individual experiments is optimize TSV count and programmable routing resources in 3D Tree-based FPGA. Experiments are performed individually for each netlist using the optimization flow presented

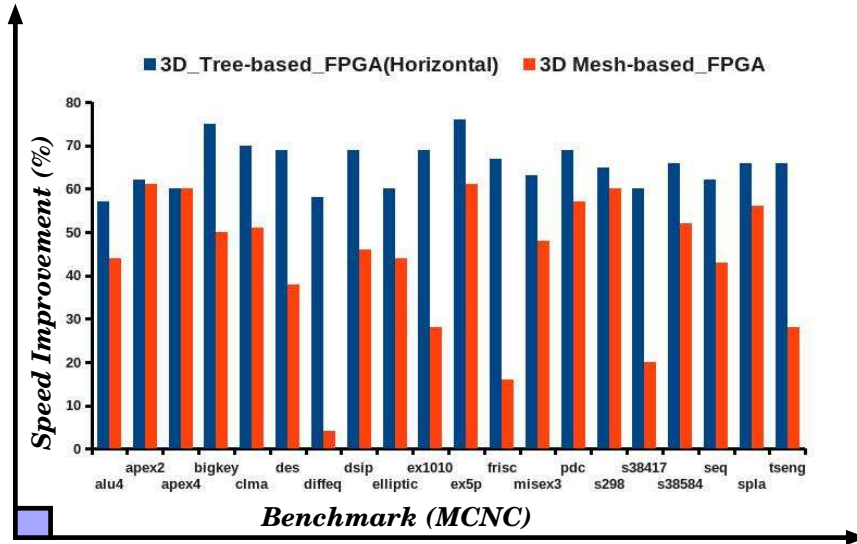


Figure 12: Comparison between 3D Tree-based FPGA and 3D Mesh-based FPGA(K. Siozios et al., 2011)

in Figure 13. The architecture optimizer designed as an add-on utility using the router program implemented using the *PathFinder* algorithm (L. McMurchie et al , 1995; Z. Marrakchi et al., 2005, 2006), which uses an iterative, negotiation-based approach to successfully route all nets in an application netlist. The router program in association with a binary search algorithm, considers the same architecture with different  $p$  values at each levels of the two-tier 3D Tree-based FPGA to determine the smallest number of input and output signals at each Tree levels by allowing to route the benchmark circuits. At first, the optimization program considers architecture break point level with different Rent ( $p$ ) values. The purpose is to find, for all benchmark circuits, the architecture with the fewest necessary TSVs between the break point levels while keeping the programmable interconnect resources placed in tier 0 and 1 intact. The solution provides the spatial distribution and minimum number of vertical interconnects required to route each benchmark in the two-tier Tree-based FPGA. From this solution we extract the minimum possible number and location of TSVs that can be removed from the architecture without compromising the performance of the 3D chip. The decision to remove TSVs is taken based on the spatial distribution and  $p$  values of all benchmark used in the optimization process. The highest  $p$

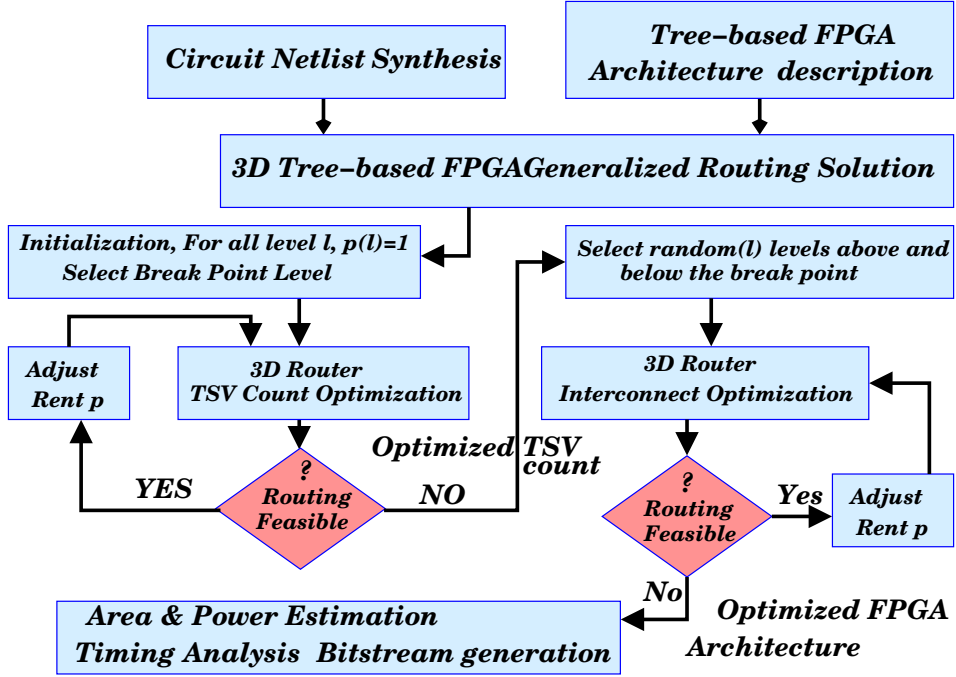


Figure 13: TSV count and interconnect optimization flow

value obtained from all benchmarks at each levels will be set as the architecture Rent. To make 3D Tree-based FPGA more efficient in terms of design and manufacturing, it is essential to minimize the TSV count because TSV consumes more silicon area than horizontal interconnects (M. Pathak et al., 2010). After completing the break-point optimization, we use the Rent's parameter (Z. Marrakchi et al., 2009) to optimize the programmable routing resources that are placed in tier 0 and 1 using random approach, in which the interconnect levels are selected randomly and modify its inputs and outputs signals depending on the previous result obtained at the same level. The Rent's parameter  $p$  defined for a Tree-based architecture is illustrated in equation 2. The Tree level is represented as  $\ell$  and  $k$  is the cluster arity,  $c$  is the number of in/out pins of an LB and IO is the number of in/out pins of a cluster located at level  $\ell$ . The optimization of upward and downward networks based on Rent's parameter is done as follows.

$$IO(\ell) = c.k^{\ell-p} \quad (2)$$

### 5.3. The Downward Network Model

As described in Figure 1, the Tree-based FPGA architecture unifies two unidirectional upward and downward interconnection networks using a BFT based network topology to connect Downward MSBs (DMSBs) and Upward MSBs (UMSBs) to LBs inputs and outputs. A cluster situated at level  $\ell$  contains  $N_{in}(\ell - 1)$  DMSBs, where  $N_{in}(\ell)$  is the number of inputs of cluster located at level  $\ell$  with  $k$  outputs and  $\frac{N_{in}(\ell) + kN_{out}(\ell - 1)}{N_{in}(\ell - 1)}$  inputs, whereas  $k$  is also the cluster arity size. Since DMSBs are full crossbar devices, the total number of switches at level  $\ell$  cluster is  $k(N_{in}(\ell) + kN_{out}(\ell - 1))$ . At each level  $\ell$ ,  $\frac{N}{k^\ell}$  clusters, whereas  $N$  is total number Logic Blocks and the total number of interconnects in the downward network is

$$\sum_{\ell=1}^{\log_k(N)} k \times N \times \frac{N_{in}(\ell) + kN_{out}(\ell - 1)}{k^\ell} \quad (3)$$

Following equation 2, we can simplify the number of outputs of a Logic Block is  $N_{out}(0) = c_{out}$  and the number of inputs equal  $N_{in}(\ell) = c_{in} \cdot k^{\ell \cdot p}$  and  $N_{in}(\ell - 1) = c_{out} \cdot k^{(\ell - 1)p}$  and so on. The total interconnects used at each level  $\ell$  can be calculated by equation 4.

$$N_{interconnects}(down) = N \times (k^p c_{in} + k c_{out}) \times \sum_{\ell=1}^{\log_k(N)} k^{(p-1)(\ell-1)} \quad (4)$$

### 5.4. The Upward Network Model

Similar to the downward interconnect network. The upward interconnect network also built using a *Butterfly-Fat-Tree* network topology. In level  $\ell$  every cluster contains  $N_{out}(\ell - 1)$  USBs with  $k$  inputs and outputs. USBs are also full crossbar devices with  $k^2 \times N_{out}(\ell - 1)$  switches at a level  $\ell$  cluster. There are  $\frac{N}{k^\ell}$  clusters at each level  $\ell$ , and the total number of upward interconnection block is

$$\sum_{\ell=1}^{\log_k(N)} \frac{k^2 \times N}{k^\ell} \times N_{out}(\ell - 1) \quad (5)$$

$N_{out}(0) = c_{out}$  is the outputs of Logic Block and using equation 2,  $N_{out}(\ell - 1) = c_{out} \cdot k^{(\ell - 1)p}$ . The total number of interconnect required for the upward

interconnect network is calculated using equation 6

$$N_{interconnects}(up) = N \times k \times c_{out} \times \sum_{\ell=1}^{\log_k(N)} k^{(p-1)(\ell-1)} \quad (6)$$

The total number interconnects in Tree-based FPGA architecture is

$$N_{interconnects}(Tree) = N_{interconnects}(down) + N_{interconnects}(up)$$

$$N_{interconnects}(Tree) = N \cdot (k^p c_{in} + 2k c_{out}) \sum_{\ell=1}^{\log_k(N)} k^{(p-1)(\ell-1)} \quad (7)$$

The total number of interconnects at different levels of the Tree is calculated by substituting  $p=1$  in the equation 7, where  $N$  is the total number of logic blocks,  $c_{in}$  and  $c_{out}$  are the number of inputs and outputs of logic blocks,  $k$  is the arity, and  $p$  and  $\ell$  are the Rent's parameter and tree interconnect level. However in normal cases the value of  $p$  ranges from  $0.3$  to  $0.8$ .

At first, the optimization program considers architecture break point level with different Rent ( $p$ ) values. The purpose is to find, for all benchmark circuits, the architecture with the fewest necessary TSVs between the break point levels. As described in (Z. Marrakchi et al., 2009), in a Tree-based FPGA the reduction in number of interconnects at level  $\ell$  impacts the number of interconnects at level  $\ell+1$ , since the number of DMSBs/UMSBs at level  $\ell+1$  is equal to the number of inputs/outputs at level  $\ell$ . Using equation 2 and 7, the Rent's value and optimized TSV count and interconnect requirements are calculated for each iteration to optimize break point levels. Once the break-point optimization is completed, the optimizer randomly chooses other tree levels above or below the break-point to optimize the routing resources. Table 2 presents the TSV count optimization results of horizontal partitioning method. A minimum possible reduction of 35% TSVs and an average speed degradation of 4.44% are recorded in these experiments. A similar experiment with 3D Mesh-based FPGA (K. Siozios et al., 2011) with 40% reduction of TSV resulted in speed degradation of 11.5% as illustrated in Table 2, which indicates the impact of TSV and routing resources optimization on speed is minimized in 3D Tree-based FPGA compared to 3D Mash-based FPGA.

Table 3 presents the results from TSV and architecture optimization experiments on each interconnect level of the Tree-based 3D FPGA. A minimum reduction of 35% and 38% TSVs are recorded for horizontal and vertical



Table 2: 3D Tree-based FPGA with 7 level and Arity 4: TSV count Optimization Results

Tree Levels=7, Arity=4, Arch=4x4x4x4x4x4x4				
	Tree-based FPGA		Tree	Mesh
Circuits	Optimized	$3D_{TSV}$	35% TSV Reduction	40% TSV Reduction
MCNC	Rent's "p"	Gain(%)	Speed degradation%	Speed degrade%
alu4	0.47	53	4.3	2.34
apex2	0.51	49	5.8	11
apex4	0.61	39	1.1	10
bigkey	0.60	40	2.8	4.1
clma	0.58	42	4.8	25
des	0.56	44	4.1	8
diffeq	0.64	36	4.5	-14
dsip	0.65	35	4.1	4
elliptic	0.62	38	3.4	34
ex1010	0.55	45	3.5	5
ex5p	0.58	42	5.1	12
frisc	0.62	38	5.4	28
misex3	0.64	36	5.2	-8
pdc	0.59	41	3.8	10
s298	0.55	45	5.8	19
s38417	0.64	36	5.1	8
s38584	0.62	38	4.5	9
seq	0.61	39	5.5	8
spla	0.58	42	5.2	6
tseng	0.63	37	4.8	7
average			4.44	11.5
Maximum Interconnect Requirement, p=0.65				
Minimum possible TSV reduction=35%				

break-point. An average speed degradation of 4.44% and 3.2% is recorded in horizontal and vertical break-point. The optimized silicon area for individual interconnect levels are reported in Table 3. Using our optimization flow, overall interconnect area of the 3D Tree-based FPGA is reduced by 36%,

Table 3: Architecture Optimization Results

Tree Levels=7	Arity=4, Arch=4x4x4x4x4x4x4		
Tree-based Architecture Levels	3D Chip Active Layer	Optimized Rent 'p'	Optimized Area $\mu m^2$
Logic Blocks	Layer 1	–	93635273
Switch Level 0	Layer 1	0.67	2412
Switch Level 1	Layer 1	0.54	10800
Switch Level 2	Layer 1	0.66	37496
Switch Level 3	Layer 1	0.65	232128
<i>BreakPoint<sub>Horizontal</sub></i>	Horizontal Break Point Level 3 $p_{Vertical}=0.66$		
Level 3 to 4	TSV Area= $40192\mu m^2$		
Switch-blocks Tree-Level 4	Layer 2	0.67	6072770
Switch Level 5	Layer 2	0.66	45553499
Switch Level 6	Layer 2	0.62	42139683
<i>BreakPoint<sub>Vertical</sub></i>	Vertical Break Point Level 6 $p_{Horizontal}=0.65$		
Level 6	TSV Area= $61091\mu m^2$		
Speed Degradation	Vertical=3.2%, Horizontal=4.7%		

which makes 3D stacked Tree-based FPGA a cost effective solution.

## 6. LUT And Cluster size Effect on Performance

In this section we evaluate the impact of LUT and cluster size on performance and power consumption of two-tier 3D Tree-based FPGA. Figure 14 presents the effect of increasing LUT (lookup table) size from 3 to 7 with cluster size fixed to 4 using horizontal and vertical break-point stacking on critical path delay of 3D Tree-based FPGA. As the LUT size increases, the area of chip and switch delay increases. The critical path delay analysis experiments consider the impact of increased switch delay, number of interconnects and TSVs as LUT size increases. The results shows that, LUT size equal 4 has the best area-delay product as illustrated in Figure 14. Even though the critical path delay improves as LUTs size increases as shown in Figure 14, the speed improvement measured for 3D Tree-based FPGA de-

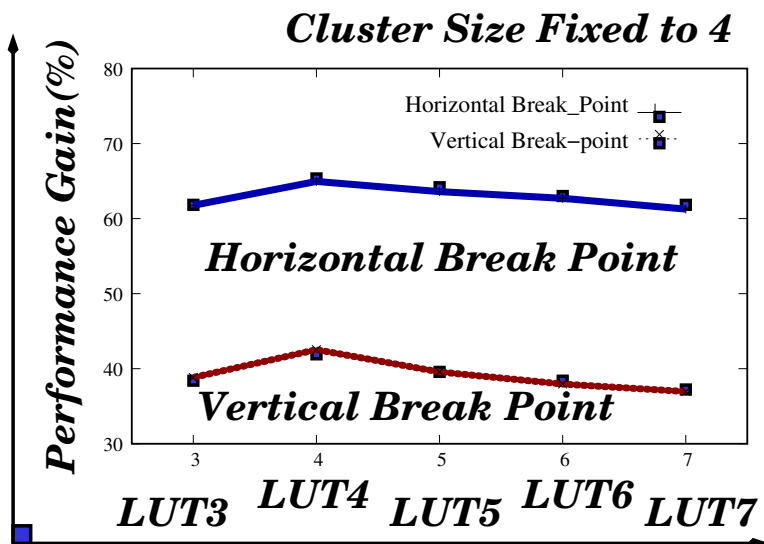


Figure 14: Impact of LUT size on performance with cluster size fixed to 4

creases due to localization of routing resources and increased switch delay. Figure 15 presents the effect of increasing cluster size from 4 to 7 with LUT size fixed to 4. As cluster size increases the logic density and switch size increases, which forces the mapped application to use more local routing resources in the tree levels close to logic blocks than routing resources at higher tree levels in a timing driven routing procedure. This makes the critical delay shorter as cluster size increases. By varying the break-point location, the critical path delay of 3D Tree-based FPGA can be optimized for the horizontal partitioning method, however this process makes the architecture more application-specific. Our area and critical path delay analysis against various LUT and cluster size analysis reveals cluster and LUT size equal to 4 is better in terms of speed, power and silicon area to design and manufacture a genera-purpose high density and high speed 3D Tree-based FPGA systems.

## 7. Power Optimization

The power optimization of two-tier 3D stacked Tree-based FPGA is achieved through the minimization of TSV count and programmable routing resources. The optimized routing resources and TSV count are listed in Table 3. In Mesh-based industrial 3D FPGA, the same power is used for individual blocks

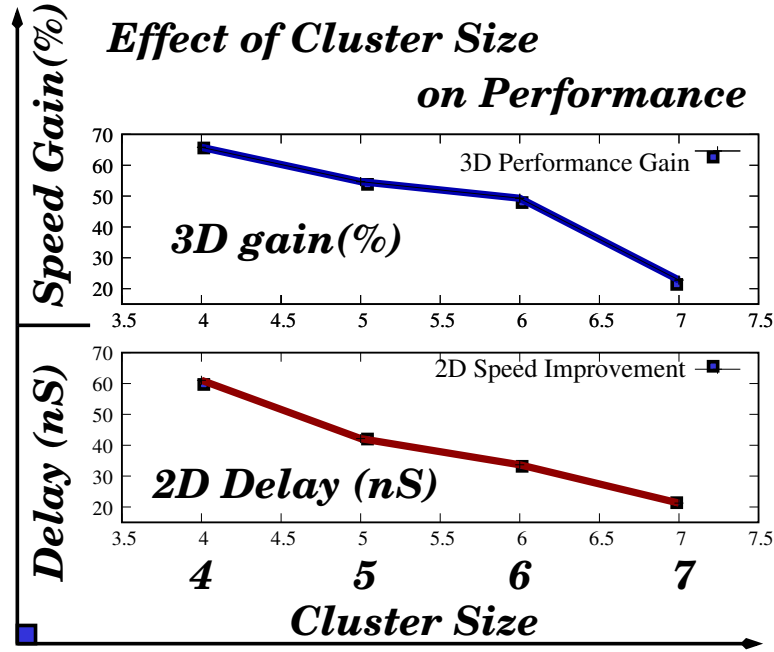


Figure 15: Impact of Cluster size on performance with LUT size fixed to 4

in multiple tiers of 3D chip. This doubles the total FPGA power for two-tier Mesh-based FPGA and this leads to pessimistic prediction of inter-layer temperature. While for Tree-based 3D FPGA, the power consumption of the dies in each tier is balanced through the optimization process of routing resources and TSV count. Figure 16 shows the interconnect power at different levels of the 3D Tree-based FPGA. The Rent parameter based architecture optimization shows 35.13% reduction in total power consumption of 7 level Tree-based 3D interconnect network. This is very promising for FPGA architecture in terms of silicon area, since FPGA is an interconnect-dominated architecture and it is impossible to manufacture it with huge number of TSV and switches. Figure 17 presents the effect of LUT and cluster size on estimation of power consumption. The power consumption increased exponentially as LUT and cluster size increase due to exponential growth of switch size as the tree grows to higher levels. Considering the power consumption and performance results, LUT and cluster size equal 4 is the best architecture for manufacturing 3D FPGA. Nonetheless higher LUT and cluster size can be used where performance is considered to be the major design criterion.

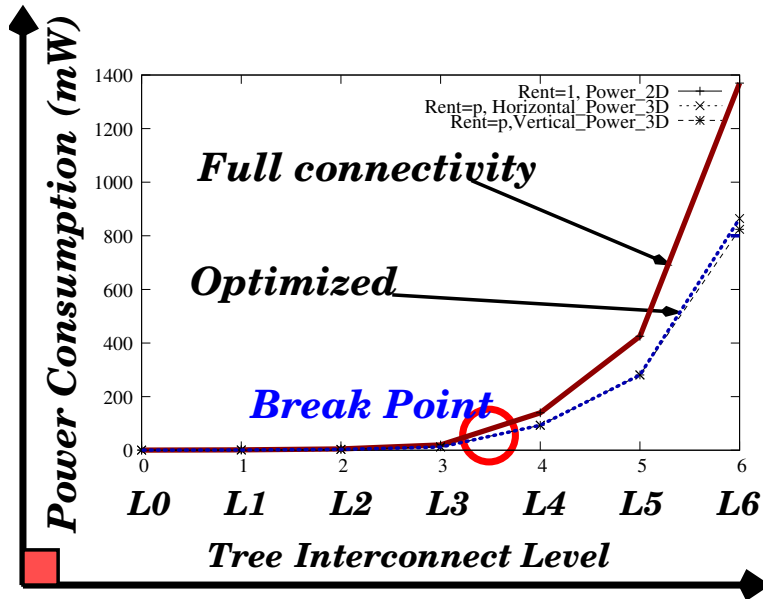


Figure 16: Power consumption analysis of 3D Tree-based programmable interconnect network

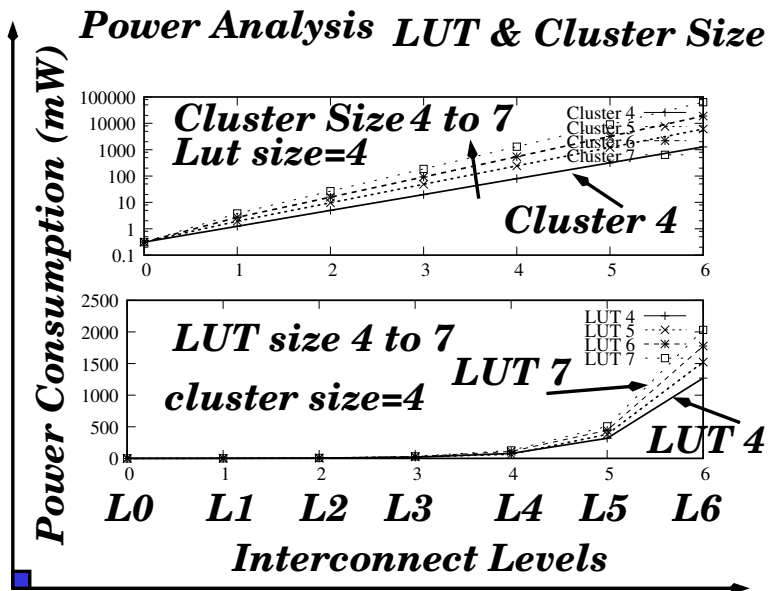


Figure 17: Impact of cluster and LUT size on power consumption

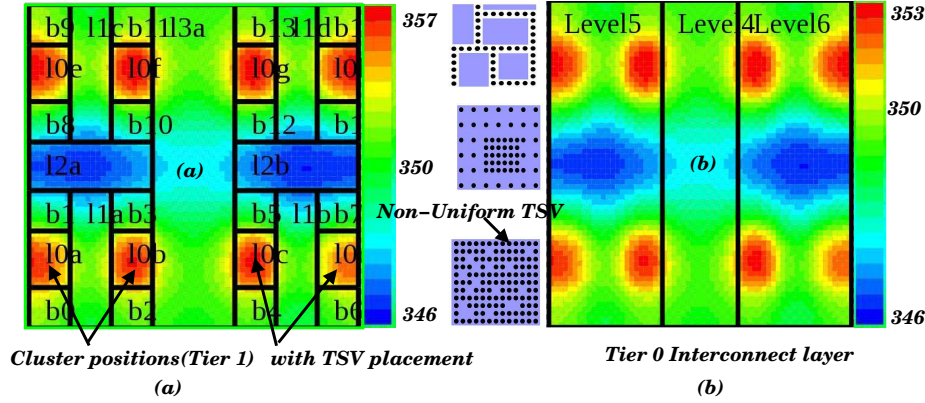


Figure 18: Two-tier floorplan with thermal profile of 3D Tree-based FPGA along TSV distribution

## 8. 3D Thermal Optimization

One of the major issues to mainstream acceptance of 3D ICs is the thermal problem. The heat coupling among high power devices in the 3D stack creates several hotspots and increases the background temperature significantly. Thermal issues in FPGAs are relatively unexplored. Some researchers have proposed the use of distributed sensors for monitoring temperatures in FPGAs (S. Velusamy et al, 2005; S. Lopez-Buedo et al, 2002). The management of inter-layer heat is growing in FPGAs. Recent articles on thermal management in 2.5D and 3D FPGAs from leading manufactures clearly indicate the importance of thermal issues in FPGA design (A. Rehman et al , 2006, 2012). Our 3D thermal model consider the impact of spatial distribution of signal TSV and power delivery network TSVs to compute the thermal profile of the 3D Tree-based FPGA chip (J. Ayala et al., 2009). Figure 18 presents the two-tier floorplan and TSV distribution styles used in the design and simulation 3D Tree-based FPGA. The floorplan (a) shows tier 1 design with clusters placed along with local interconnects. The high temperature spots are the locations where more than one cluster connects with interconnect level 3, which connect the inputs and outputs to tier 1 layout design. The heat transfer take place through copper TSVs (assumed in 3D thermal model) from tier 0 to tier 1.

The inter-layer temperature is optimized by considering area and spatial distribution of TSVs and power delivery networks (PDNs). The TSVs and PDNs are effectively used as a 3D thermal net with help vias in metal layers to

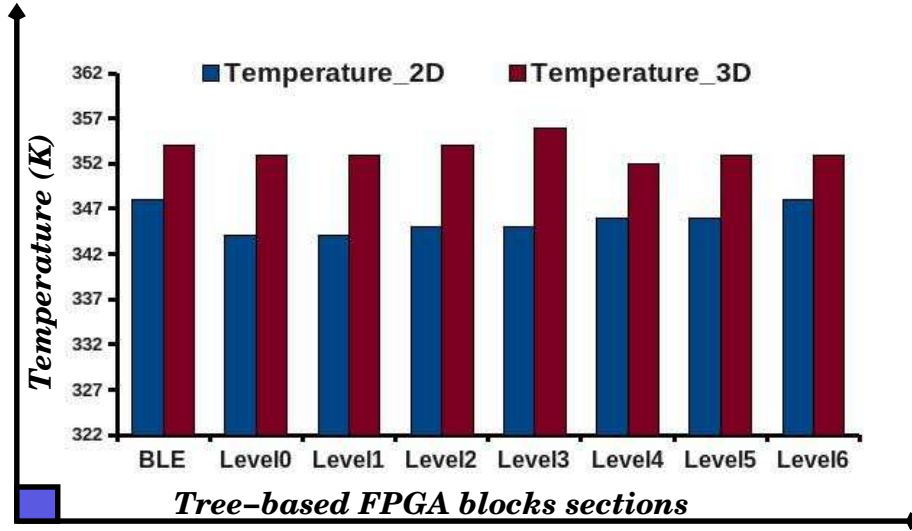


Figure 19: Measured inter-layer temperature results from 2 tier 3D Tree-based FPGA

transfer heat from tier 1 to tier 0 layer. The 3D thermal model considers the impact of via fill material based the type of technology used to manufacture TSVs, like via-first, via-middle or via-last process. While estimating the temperature profile, the 3D thermal model compute the effective thermal conductivity of active and passive layers based on TSV and silicon area in 3D stacked chip. Since the TSVs always pass through the silicon substrate, to calculated the effective thermal conductivity, we use equation 1. The via-first process use tungsten, while via-middle process use doped poly-silicon and via-last process use copper for via fill and  $SiO_2$  for isolation. Figure 19 shows the temperature at different Tree levels in 2-tier 3D Tree-based FPGA. The measured peak temperature of 2D Tree-based FPGA is 351K and average temperature is 346K. With our localized rearrangement of interconnects and switch blocks along with TSV area, the peak and average temperature are optimized at 355K and 351K respectively for 3D FPGA.

## 9. Conclusion and Future work

An efficient design and exploration methodology for 3D Tree-based FPGA presented. The horizontal and vertical break-point design methodology based on design specification is a defining feature of our design flow. A timely architecture and TSV count optimization methodology have been introduced

and a reduction of 36% in overall interconnect area observed. The maximum TSV count limited to 65% in horizontal and 62% in vertical break-point cases. The experimental analysis shows the horizontal break-point method is better for high speed applications. The impact of speed and power consumption on different LUT and cluster size is also presented. Therefore we believe that all the design and architecture styles presented in this paper can serve as a robust foundation for the design and manufacturing of even more practical 3D re-configurable systems based on Tree-based FPGA architectures.

One future direction we propose, is to implement two-tier 3D Tree-based FPGA using monolithic stacking. This approach will further reduce the wire length and thereby improve performance. Since the two-tier design is done in such a way to stack almost 80% of the programming overhead (tier 0) of Tree-based FPGA on top of logic blocks (tier 1) and interconnected using TSVs. In the case of monolithic stacking the interconnect layers between programming overhead and logic blocks will be implemented in a state-of-the-art CMOS technology. This design and implementation methodology provide additional flexibility to improve logic density, speed and reduce power consumption and silicon area. However the main challenge in this approach is to balance the density of TSVs to that of the via density in the CMOS technology used to implement Logic and interconnect layers.

## References

- A. Rahman, S. Das, A. Chandrakasan and R. Reif, *Wiring Requirements and Three-Dimensional Integration of Field Programmable Gate Arrays*, SLIP ACM, March, 2001.
- M. Lin, A. EL Gamal, Yi-Chang. Lu and S. Wong, *Performance Benefits of Monolithically Stacked 3D FPGA*, Proceedings of the ACM/SIGDA 14th ISFPGA NY USA, pp 113-122, 2006.
- V. Pavlidis and E. Friedman, *Interconnect-Based Design Methodologies for Three-Dimensional Integrated Circuits*, Proceedings of the IEEE, pp 123-140, Jan 2009.
- R. Reif, A. Fan, Kuan-Neng Chen, S. Das, *Fabrication Technologies for Three-Dimensional Integrated Circuits*, Proceedings of the IEEE International Symposium on Quality Electronic Design (ISEQED'02), 2002.



- C. Ababei, Y. Feng and B. Goplen, *Placement and Routing in 3D Integrated Circuits*, IEEE Design and Test of Computers, Vol 22, No 6, pp 520-531 2006.
- K. Siozios, A. Bartzas and D. Soudris, *Architecture Level Exploration of Alternative schmes Targeting 3D FPGAs: A Software Supported Methodology*, Intern' Journal of Reconfigurable Computing, 2011.
- Z. Marrakchi, H. Mrabet, U. Farooq and H. Mehrez, *FPGA Interconnect Topologies Exploration*, Inter' J' of Reconfigurable Comput, vol 15, No 7, pp 795-825, Nov 2009.
- V Pangracious Z Marrakchi E Amouri and H Meherez, " *Performance Analysis and Optimization of High Density Tree-based 3D Multilevel FPGA*", ARC 2013, Vol 7806, pp 197-209, 2013.
- J. Ayala, A. Sridhar, V. Pangracious, D. Atienza, Y. Leblebici: *Through Silicon Via-Based Grid for Thermal Control in 3D Chips*. NanoNet pp.90-98 2009.
- ITRS-2012, "International technology roadmap for semiconductors," [Online]. Available: <http://public.itrs.net>, March, 2012, pp:17-21
- M. Pathak, Y. Joon Lee, T. Moon and S. Kyu Lim *Through-silicon-via management during 3D physical design: When to add and how many?*, IEEE-ICCAD, 2010, pp 387-394.
- K. Sankaranarayanan, S. Velusamy, M. Stan, and K. Skadron, *A case for thermal-aware floorplanning at the microarchitectural level*, Journal of Instruction-Level Parallelism, vol. 7, Oct. 2005.
- D. M. Jang, C. Ryu and K. Y. Lee, *Development and Evaluation of 3D SiP with Vertically Interconnected Through Silicon Vias (TSV)*, Electronic Components and Technology Conference (ECTC'07), pp 847-852, June 2007.
- S. Lopez-Buedo, J. Garrido, and E. Boemo, *Dynamically inserting, operating, and eliminating thermal sensors of FPGA-based systems*, IEEE Transactions on Components Packaging Technology (CPM), vol. 25, no. 4, pp. 561566, Dec. 2002.

- S. Velusamy et al, *Monitoring temperature in FPGA based SoCs*, presented at the International Conference on Computer Aided Design (ICCAD), San Jose, CA, 2005.
- A. Rahman, J. Trezza, B. New and S. Trimberger, *Die Stacking Technology for Terabit Chip-to-Chip Communications*, IEEE Custom Integrated Circuit Conference (CICC), pp 587-590, 2006.
- A. Rahman, H. shi Li, D. Ibbostson and S. Ramaswami, *Design and Manufacturing Enablement for Three-Dimensional (3D) Integrated Circuits (ICs)*, IEEE Custom Integrated Circuit Conference (CICC), 2012.
- R. Chaware, K. Nagarajan, . Ng, S.Y. Pai, *Assembly Process Integration Challenges and Reliability Assessment of Multiple 28nm FPGAs Assembled on a Large 65nm Passive Interposer*, IEEE International Reliability Physics Symposium (IRPS), Anaheim, CA, pp 2B.2.1-2B.2.5 2012.
- S. Chiricescu, M. Leeser, and M. M. Vai, *Design and Analysis of A Dynamically Reconfigurable Three-Dimensional FPGA*, IEEE Transaction on Very Large Scale Integration (VLSI) Systems., Vol.9 No.1, pp.186-196, Feb 2001.
- Cha-I. Chen Bau-Cheng. LEE and Juinn-Dar. Huang, *Architectural Exploration of 3D FPGAs Towards A Better Balance Between Area and Delay*, DATE11, 2011.
- S. Gupta, M. Hilbert, S. Hong, R. Patti, *Techniques for Producing 3D ICs with High-Density Interconnect*, Tezzaron Semiconductor Naperville, IL 2005.
- C. M. Fiduccia and R. M. Mattheyses. *A Linear-time Heuristic for Improving Network Partitions*, Design Automation Conference, vol 7, pp 175-181, 1982.
- A. Gayasen, V. Narayanan, M. Kandemir and A. Rahman. *Designing a 3-D FPGA: Switch Box Architecture and Thermal Issues*, IEEE Transaction on VLSI Systems, Vol 16(7), pp 882-893, July 2008.
- L. McMurchie and C. Ebeling. *PathFinder: A Negotiation based Performance Driven Router for FPGAs. Conference of Field Programmable Gate Arrays FPGA*, Vol 12(2), pp 291-301, June 1995.

- N. Selvakumaran and G. Karypis, *Multi-Objective Hypergraph-Partitioning Algorithm for Cut and Maximum Subdomain-Degree Minimization*, IEEE Transactions on Computer Aided Design of Integrated Circuits, Vol 25(3), PP 504-517, 2006.
- T. Cormen, C. Leiserson, and R. Rivest. *Introduction to algorithms*. MIT Press, Cambridge, 1990.
- K. Siozios, A. Bartzas and D. Soudris, *A Novel Framework for Exploring 3-D FPGAs with Heterogeneous Interconnect Fabric*, ACM Transactions on Reconfigurable Technology and Systems, Vol 5, No 1, March 2012.
- Z. Marrakchi, H. Mrabet, H. Mehrez, *Hierarchical FPGA clustering based on multilevel partitioning approach to improve routability and reduce power dissipation* International Conference on Reconfigurable Computing and FPGAs, ReConFig 2005, Puebla City, Mexico, September, 2005.
- Z. Marrakchi, H. Mrabet, H. Mehrez, *A new Multilevel Hierarchical MFPGA and its suitable configuration tools*, pp 263-268, ISVLSI 2006.
- E. M. Sentovich, K. J. Singh, L. Lavango, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. Stephan, R. K. Brayton, and A. Sangiovanni-Vincentelli, *SIS, A System for Sequential Circuit Synthesis*, Technical report No. UCB/ERL, M92/41. University of California Berkeley, 1992.
- J. Cong and Y. Ding, *On Area/Depth Trade-off in LUT-Based FPGA Technology Mapping*, IEEE Transactions on VLSI Systems, Vol 2(2), pp 137-148, 2000.