Hindawi Publishing Corporation
International Journal of Reconfigurable Computing
Volume 2008, Article ID 764942, 18 pages
doi:10.1155/2008/764942

Research Article

Architecture-Level Exploration of Alternative Interconnection Schemes Targeting 3D FPGAs: A Software-Supported Methodology

Kostas Siozios, Alexandros Bartzas, and Dimitrios Soudris

Department of Electrical and Computer Engineering, Democritus University of Thrace, 67100 Xanthi, Greece

Correspondence should be addressed to Kostas Siozios, ksiop@ee.duth.gr

Received 6 March 2008; Revised 24 July 2008; Accepted 11 September 2008

Recommended by Lionel Torres

In current reconfigurable architectures, the interconnection structures increasingly contribute more to the delay and power consumption. The demand for increased clock frequencies and logic density (smaller area footprint) makes the problem even more important. Three-dimensional (3D) architectures are able to alleviate this problem by accommodating a number of functional layers, each of which might be fabricated in different technology. However, the benefits of such integration technology have not been sufficiently explored yet. In this paper, we propose a software-supported methodology for exploring and evaluating alternative interconnection schemes for 3D FPGAs. In order to support the proposed methodology, three new CAD tools were developed (part of the 3D MEANDER Design Framework). During our exploration, we study the impact of vertical interconnection between functional layers in a number of design parameters. More specifically, the average gains in operation frequency, power consumption, and wirelength are 35%, 32%, and 13%, respectively, compared to existing 2D FPGAs with identical logic resources. Also, we achieve higher utilization ratio for the vertical interconnections compared to existing approaches by 8% for designing 3D FPGAs, leading to cheaper and more reliable devices.

1. Introduction

In the real-estate market, an often-stated truism is that as land becomes more expensive, there is a tendency to build upwards rather than outwards. This idea has some resonance in the domain of silicon-integrated circuits (ICs), where the size of the die is limited among others by yield and performance constraints. In the next few years, enormous changes are about to happen that will influence the future of LSI. The shift from horizontal to vertical stacking of circuits has the potential to rewrite the conventions of electronics design. Three-dimensional (3D) ICs, which contain multiple functional layers, mitigate many of the limitations introduced by the current process technologies, as they enhance dramatically among others the device performance, the functionality, and the packaging density, as compared to two-dimensional (2D) ones [1].

A qualitative comparison regarding the gains introduced by the 3D integration process, compared to existing system design approaches is summarized in Table 1. More specifically, 3D integration technology provides increased performance in numerous design criteria as compared to the existing 2D approaches, while the wide acceptance of such a fabrication process and the development of supporting CAD tools are still open issues.

Although 3D integration promises considerable benefits, several challenges need to be satisfied. An important challenge is the design space exploration, which is essential to build efficient devices (in terms of high-performance, low energy, electromagnetic interference (EMI), etc.), as well as the design of architectures that exploit all the advantages offered by 3D integration. In addition, CAD tools that facilitate the design of 3D circuits are required. Up to date, there are only a few academic approaches [2, 3] for mapping applications on 3D FPGAs, while there is no complete CAD flow in order to promote the commercialization of this new design paradigm. Furthermore, there is no commercial CAD tool targeting 3D devices, similar to the standalone tools and/or design flows provided by Cadence, Mentor Graphics, and Xilinx for 2D technologies. Consequently, there is

Table 1: Comparison between alternative design implementations.

Property	Single chips	System-on-chip (SoC)	3D integration
Modular flexibility	High	Low	Medium
System performance	Low	Medium-High	High
Physical dimension of products	Large	Medium	Small
Complexity of fabrication process	Low	Medium-High	Medium-High
Fabrication cost	Low	Medium	High
Design methodology, CAD tools	Available	Available	Not deployed yet

an absolute necessity to develop algorithms and software tools to exploit the advantages of the third dimension and solve time-consuming and complex tasks, such as partitioning, placement and routing (P&R) for 3D-reconfigurable architectures.

The benefits of using 3D architectures in logic chips will be especially great for designing field-programmable gate arrays (FPGAs), as these devices always exhibit limitations that occur due to increased wirelength. Compared to ASIC solutions, they consume more power and energy, while they operate in lower frequencies. Since 3D integration technology provides increased number of neighbors, each logic block can access a greater number of nearest neighbors, alleviating the requirement for lengthier connections. Due to this, it is likely that the reconfigurable architectures will drive rapid adoption of 3D IC technology faster than any other device (e.g., ASIC).

Many of the problems alleviated with the usage of 3D integration technology are tightly coupled to the total wirelength. It is common for architecture designers to estimate the longest interconnect equal to twice the length of the die edge. In order to show the potential gains of the new integration approach in this field, Figure 1 illustrates an example structure where the interconnection length is significantly reduced compared to conventional 2D architectures. More specifically, for a given total area equal to A (both for 2D and 3D devices), as the number of layers increases, the area of each layer as well as the longest interconnection is reduced. For instance, if we employ architecture with four layers, the corresponding interconnection length is almost the half (compared to 2D devices).

The 3D integration technology has impact both in physical level (i.e., wirelength, density) as well as product/system level (i.e., performance, power/energy, cost, functionality, and security). In particular, the main design parameters improved by the exploitation of 3D integration are as follows.

(1) *Wirelength*. The interconnection network of large-scale reconfigurable architectures exhibits increased resistance (R) and capacitance (C) values. However, in the 3D approach, the circuits are split up into smaller parts and stacked appropriately alleviating such problems [1, 4, 5].

(2) *Density*. 3D designs support the possibility of implementing more logic in the same footprint area, compared to existing well-established 2D technologies. In other words, the increased functionality extends Moore's law and enables a new generation of tiny but powerful devices.

(3) *Performance*. In current technologies, timing is interconnection-driven. As the propagation delay is proportional to the square of the wirelength, its significant reduction leads to overall performance gains. Furthermore, by shortening the distance, the electrical signals have to travel, 3D interconnect technology could deliver the performance gains promised by Moore's law [6].

(4) *Power/Energy*. Wirelength reduction has an impact on the cycle time and energy dissipation, as the interconnect structures increasingly consume more of the power budgets in modern designs [7].

(5) *Cost*. Three-dimensional reconfigurable architectures can potentially provide a reduction in manufacturing costs, as they might not require the integration of state-of-the-art production lines. The prices of manufacturing equipment have soared with each new generation, and using depreciated equipment they would have a major impact on total cost of development new systems. In addition to this, it is much easier to boost yield with older manufacturing processes, while this cost would drop even further by manufacturing high volume of such 3D FPGAs [6]. Finally, the cost of producing 3D FPGAs is tightly related to the 3D assembly procedure (i.e., die to wafer, wafer to wafer, etc.).

(6) *Functionality*. The design of reconfigurable architectures with three dimensions adds a higher order of connectivity, while it opens a world of new design possibilities/options. By integrating heterogeneous blocks, the derived 3D FPGAs exhibit higher efficiency compared to existing solutions [8]. This feature is even more interesting by allowing components with completely incompatible manufacturing technologies to be combined in the different functional layers of a single 3D device.

(7) *Security*. In addition to that, the 3D integration technology provides more advanced IP security, as the stacked structure makes almost impossible any attempt for reverse engineering. This task can be even more difficult by partitioning the target application in such way so that to obscure the function of each layer.

Recently, many research groups from academia [2–4, 9, 10], industry [11, 12], and research institutes [1] have spent significant effort on designing and manufacturing applications in 3D technologies. Beyne presented a survey of existing 3D fabrication technologies in [1]. This work focuses on available interconnection architectures among the layers of 3D ICs and emphasizes the open issues for current and upcoming 3D technologies. A few companies [11, 12] develop 3D ICs for commercial purposes by stacking wafers,

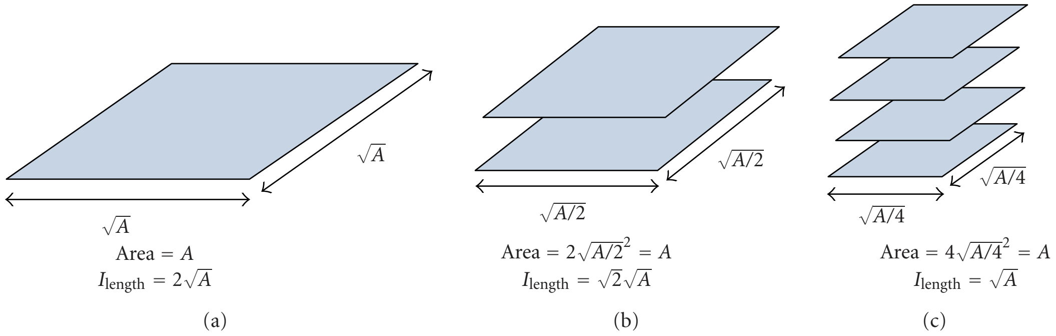

$\sqrt{A}$
$\sqrt{A}$
Area = A
$I_{\text{length}} = 2\sqrt{A}$
(a)
$\sqrt{A/2}$
$\sqrt{A/2}$
Area = $2\sqrt{A/2}^2 = A$
$I_{\text{length}} = \sqrt{2}\sqrt{A}$
(b)
$\sqrt{A/4}$
$\sqrt{A/4}$
Area = $4\sqrt{A/4}^2 = A$
$I_{\text{length}} = \sqrt{A}$
(c)

Figure 1: Variation on interconnection length for (a) a 2D device, (b) a 3D architecture with two layers, and (c) a 3D architecture with four layers.

where the distance between the layers is determined by the wafer thickness. Note that the existing industrial research primarily concerns the manufacturing and fabrication processes rather than the development of CAD tools to support the design of emerging 3D technologies.

In [2], the potential of a 3D FPGA technology, based on 3D switch boxes (SBs), is evaluated using analytic models. In order to support the implementation of an application on such a device, a software tool based on [13] is employed. Such an approach exhibits some main drawbacks. First, there is no restriction regarding the amount of vertical connections, which can lead to unacceptable numbers (in terms of fabrication technology) for the 3D device. Moreover, these connections can be formed in any SB, leading to waste of silicon area, while it also increases the fabrication cost of the 3D stack. Furthermore, the assumption that the vertical interconnections are electrical equivalent to routing wires with same length placed on a layer leads to nonrealistic results. Finally, the employed tool cannot estimate/calculate other important design parameters, such as, the energy/power consumption.

An integration process for 3D ICs is presented in [9]. This fabrication technology is based on low-temperature Cu-Cu wafer bonding, where device wafers are bonded in a face-to-back manner with short vertical interconnections. This approach invests effort to minimize either the total wirelength or the number of the inter-layer interconnections. However, other design objectives, such as power consumption or delay are not taken into consideration. Furthermore, the described approach does not permit any architecture level exploration of the target 3D device, as there is no option regarding the hardware resources modification.

A tool flow employed to implement applications on 3D ICs is presented in [3]. The placement algorithm is partitioning-based followed by a simulated-annealing refinement for minimizing the total interconnection length. However, this flow handles only the total wirelength as cost function, ignoring other critical design parameters such as power/energy consumption.

To summarize, in the literature there are two main approaches for designing 3D FPGAs. The first of them affects devices, where each layer can be thought as a "functional layer" [2, 3], while in the second approach each of the layers is specialized (i.e., memory, switches, logic, etc.) [14]. Even though throughout this paper we study a 3D-reconfigurable architecture where all of the layers can be thought to have identical logic resources (we are interested only on the interlayer communication scenario), however, this is not a prerequest, as our proposed methodology can also handle devices with irregular (i.e., heterogeneous) layers.

In this paper, a design methodology for architecture level exploration of alternative interconnection schemes targeting 3D FPGAs is discussed. This methodology is software-supported by three new CAD tools, namely, 3D partitioning (3DPart), 3D placement and routing optimizer (3DPRO), and 3DPower. More specifically, the first one is responsible for the application partitioning to device layers, the second one deals with the placement of each layer and the routing procedure on the 3D-reconfigurable architectures (with full-custom interconnection fabric), while the last one performs the power/energy estimations of these devices. All of them are part of the new Design Framework, named 3D MEANDER [5, 15].

The derived interconnection scheme is integrated into a 3D Virtex-based device for evaluation purposes. During our evaluation procedure, we quantify a number of cost factors. Mainly, we study the application's delay (or performance), the energy consumption, and total wirelength over a plethora of 3D FPGAs with different interconnection schemes. More specifically, the interconnection scenarios affect different number of vertical connections, as well as alternative spatial allocation of them over each functional layer. To the best of our knowledge, the proposed software-supported architecture methodology for exploring/evaluating 3D FPGAs with full-custom interconnections schemes is presented for the first time in the literature. During this evaluation, we prove that we can design 3D architectures with better utilization ratio of vertical interconnection, leading to lower fabrication costs and higher reliability for the 3D devices.

The rest of the paper is organized as follows. In Section 2, we describe the modeling approach of the 3D FPGA architecture, while the proposed architecture exploration methodology and the supporting CAD tools are presented in Sections 3 and 4, respectively. The evaluation results that demonstrate the efficiency of the proposed methodology under numerous design criteria are presented in Section 5, while the main points of the work are summarized in Section 6.

2. Modeling of the 3D FPGA Architecture

The proposed 3D FPGAs can be constructed by stacking a number of identical 2D functional layers, while we provide the required communication by interlayer through-silicon-vias (TSVs) among vertically adjacent SBs. The architecture of each layer, shown in Figure 2(a), is similar to Xilinx Virtex. In order to model such a device, some of the existing 2D Switches Boxes (SBs) has to be extended to employ connections to the other layers of the 3D FPGA. For our case study, the employed SBs are similar to the ones found in a Xilinx FPGA [16], while more advanced SBs patterns might be found in relevant references. The employed SBs have a permutation function, which defines the track that each routing channel is connected to inside an SB, described by the expression $f(t) = t$, where t is one of the routing tracks [13, 17]. More specifically, this permutation function is shown in Figure 2(b), where the routing tracks of the SB from left side can connect to routing tracks from the rest parts of the SB, marked with the same ID number.

As the implemented permutation function affects the routing efficiency of the target architecture, it also affects the utilization ratio of the 3D TSVs (i.e., 3D SBs). The employed architecture has two flavors of this SB pattern. The first of them affects a 2D SB (as shown in Figure 2(b)), where an incoming routing track can be connected to wires in the three other directions of the SB ($F_s = 3$), whereas the latter (i.e., 3D SB) supports also connections in the third dimension (Figure 2(c)). In the second approach, the incoming routing track is possible to be connected to tracks placed on one of the five other directions (three on the same layer, the upper, and lower layers) ($F_s = 5$). The 2D SB is formed by $6 \times W$ transistors, while the 3D approach requires $15 \times W$ transistors, where W denotes the width of routing channel that crossed in each SB. We have to mention that the interlayer connections occupy much more silicon area, when they are compared to 2D interconnection resources. Due to this, careful selection of the total number of 3D SBs that exist in each of the functional layers, as well as their spatial distribution over the layers, is one of the upmost parameters for steering the optimal selection procedure of the appropriate connectivity across the layers of the 3D device in order to achieve a high-performance and low-power implementation of 3D FPGAs at the minimal fabrication cost

For all of the simulation/evaluation experiments presented in this work, we use a multisegment routing architecture similar to the one that appears in the Xilinx Virtex for the tracks in each layer (composed from routing segments of lengths $L1$, $L2$, $L6$, and long lines, while the distribution of the segments in each channel is 8%, 20%, 60%, and 12%, resp.). An abstract of this multisegment interconnection architecture consisted of wires with lengths $L1$, $L2$, $L6$, and long lines is depicted in Figure 2(d). In order to model the interconnection fabric of each layer, we employ the RC model proposed in [13].

Another critical parameter of the 3D-reconfigurable architecture affects the vertical interconnection that provides the required connectivity among layers. For our study, this communication is realized with through-silicon-vias (TSVs). As this integration technology has not been explored sufficiently yet, careful design of systems that employ such interconnection is required. Also, due to the large variation of the TSV parameters among alternative process technologies, such as diameter, length, dielectric thickness, and fill material, a wide range of measured resistances, capacitances, and inductances have been reported in the literature [11, 12, 18–21].

Electrical characterization of these structures is a crucial requirement since electrical models are necessary to accurately describe the interconnect power and speed of a 3D circuit. The employed values of electrical equivalent circuit, shown in (1), for each TSV, are based on existing approach from [21]:

$$L_{\mathrm{TSV}} = \frac{L_0}{1 + \log\left(f/10^8\right)^{0.26}},$$
$$R_{\mathrm{TSV}} = R_0 \times \sqrt{1 + \frac{f}{10^8}}. \tag{1}$$

In order to model the impact of TSVs on the 3D FPGA (i.e., ground-signal-ground TSV configuration), we employ a high-frequency equivalent circuit, shown in Figure 3. More specifically, the electrical model of each TSV is expressed as a resistor (R_{TSV}) and an inductor (L_{TSV}), while the capacitive coupling between the TSVs is modeled as coupling capacitors (C_{ox}, C_{si} and C_{TSV}). Regarding the parameter C_{TSV}, it denotes the capacitance of the thin oxide layer surrounding the TSV barrel, while the C_{ox} corresponds to the capacitance of the oxide layer on the silicon surface and the fringing field between the TSVs. The capacitance of the silicon substrate is denoted by C_{si}, and the loss property of the silicon substrate between the signal TSV and the ground TSV is denoted by G_{si}. For this setup, the values of these parameters are $C_{\mathrm{TSV}} = 910\,\mathrm{fF}$, $G_{\mathrm{si}} = 1.69\,\mathrm{m}/\Omega$, $C_{\mathrm{si}} = 9\,\mathrm{fF}$, and $C_{\mathrm{ox}} = 3\,\mathrm{fF}$. In these equations L_0 and R_0 correspond to the iduntance and the resistance of a TSV, respectively. For a frequency of 0.1 GHz, considering skin effect of the via barrel, the values of these parameters are $L_0 = 35\,\mathrm{pH}$ and $R_0 = 12\,\mathrm{m}\Omega$, respectively. Based on exploration results shown in [21], it is proven that such a TSV modeling exhibits negligible error for operation frequencies up to 20 GHz.

3. Exploration Methodology for Building 3D FPGAs

The proposed methodology for exploring alternative interconnection schemes for 3D-reconfigurable architectures is composed by three steps, as they are depicted in Figure 4, each of which is implemented as a CAD tool.

Figure 5 shows a detailed description of each methodology step. The input to this methodology is the application graph that describes the functionality of the digital system. The first step of the methodology deals with the application's

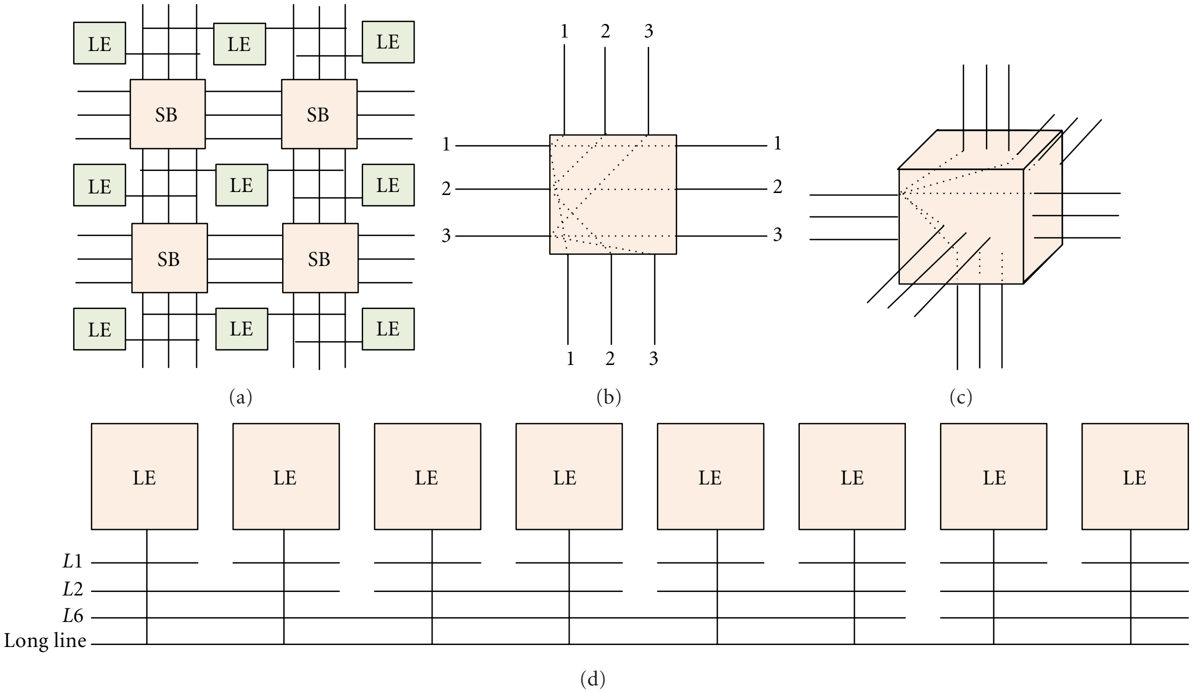


Figure 2: An abstract view of different parts from our architecture: (a) part from the device layer, (b) a 2D SB, (c) a 3D SB, and (d) a multisegment interconnection architecture.

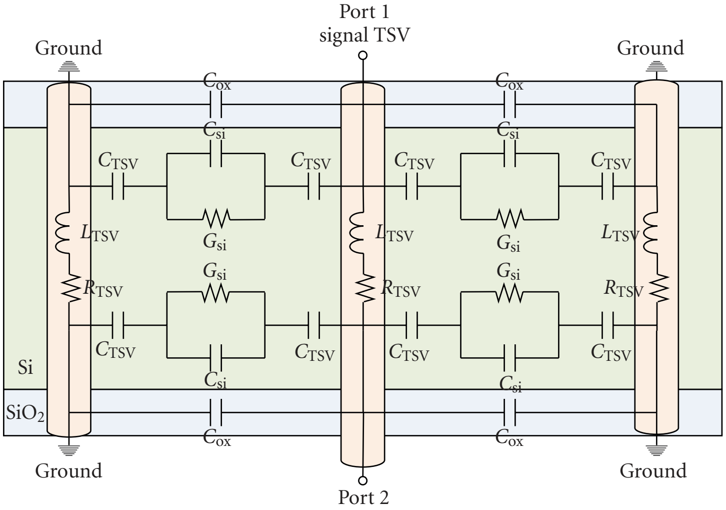


Figure 3: The electrical equivalent circuit for modeling a TSV [21].

graph partitioning, assignment to 3D device layers, and layer ordering. The procedure of splitting an application to a number of parts, which essentially assigns these parts to the available functional layers and orders them to build the 3D FPGA with reasonable execution times, is implemented in the *3DPart* tool by incorporating Pareto-based methods [22]. Such an approach utilizes in a better way the available hardware resources of each layer.

In contrast to existing solutions for application partitioning on 2D [23] or 3D FPGAs [2, 3], which mainly focus on a *min-cut* approach [23], our partitioning algorithm exhibits higher flexibility (as it takes into consideration additional constraints, such as area balance or power/temperature distribution), leading to more accurate partitions. The employed cost functions, which steer the algorithms of the partitioning step of our proposed methodology, provide a tradeoff between the required number of TSVs and the application metrics (such as delay, power/energy consumption). We have to mention that in contrast to a conventional min-cut approach, our proposed algorithm is aware about the number of interlayer connections between successive layers, while it also pays effort to balance

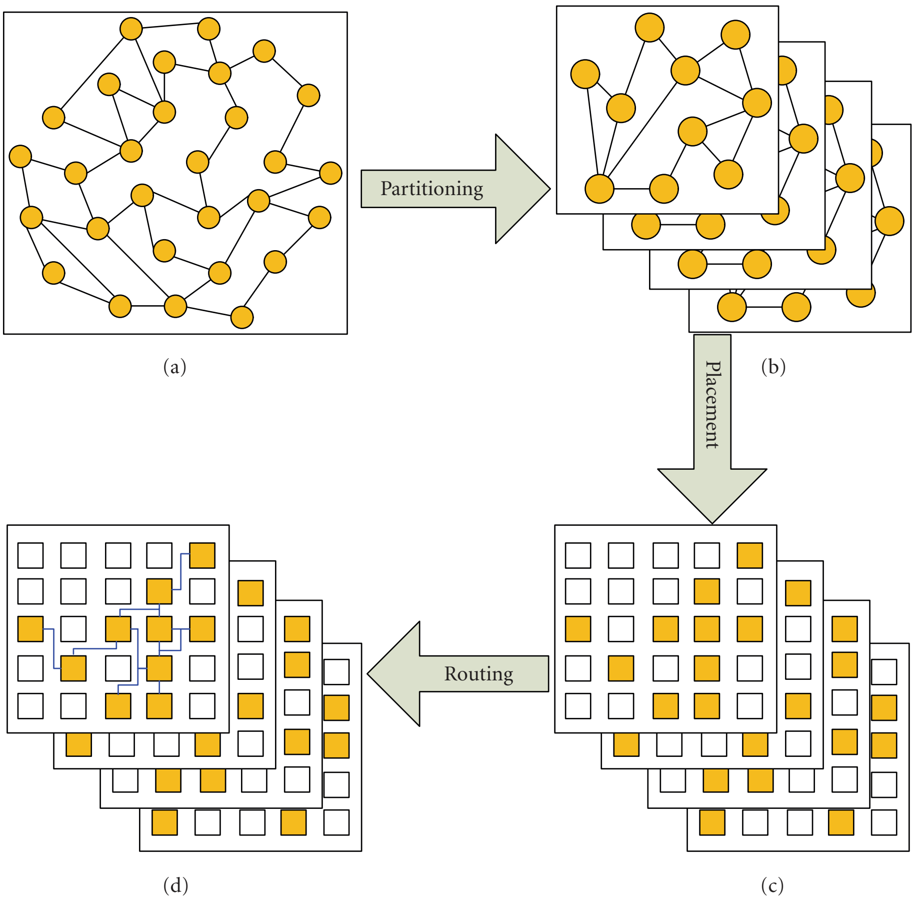


Figure 4: The design procedure for application mapping on 3D FPGAs: (a) initial application graph, (b) application partitioned to layers, (c) application placement, and (d) application routing to target 3D device.

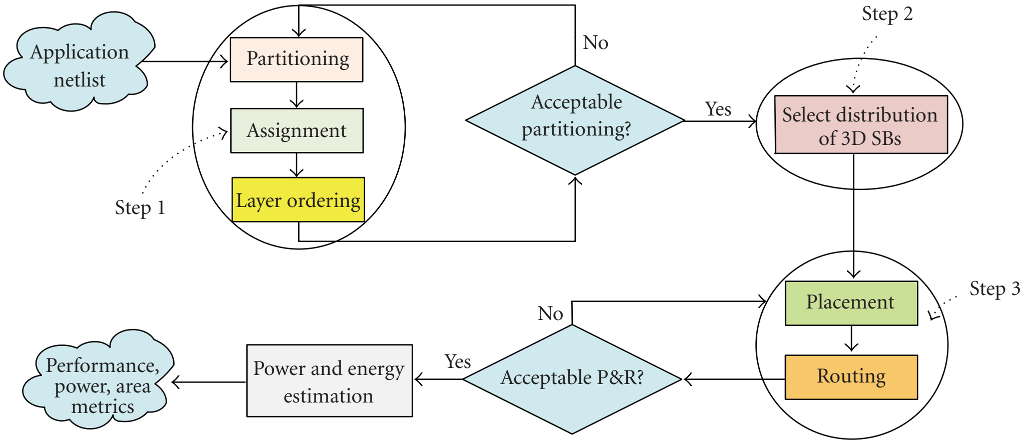


Figure 5: The proposed methodology for exploring alternative 3D-reconfigurable architectures.

them over the 3D device. Furthermore, the modularity of the exploration framework gives the opportunity to the designer to develop and employ more advanced cost functions.

By the end of the first step, we have assigned the logic functionality of the application to the available hardware resources placed on each layer of the 3D FPGA. In order to evaluate the derived result, we quantify the efficiency of the partitioning, in terms of numerous design parameters. More specifically, the resulted decision about the derived partitioning is based on the interlayer connectivity demand (i.e., number of required TSVs), the estimation about power/delay, as well as the power distribution among layers. Based on the design goals for the derived 3D-reconfigurable architecture, the output of the first step is either accepted or not. Whether an acceptable solution is derived, we proceed to the second step of our methodology. On the other hand (when the partition does not meet the design constraints), the application is fed back to the first step. Employing the 3DPart tool performs this step.

The second step of the proposed methodology deals with the selection and distribution of 3D SBs. As we design FPGAs, it is well worth to distribute them uniformly over the layer's area. Even though, more advanced distributions might be found in relevant approaches [5], they lead to increase design cost due to the higher complexity. Consequently, for our study, we select to distribute the available number of 3D SBs, uniformly over the layer's area.

The third step of the proposed methodology deals with the application's placement and routing (P&R) on a 3D-reconfigurable architecture, while it is software supported by the 3DPRO tool. During this step, the logic functions of each layer are assigned to hardware blocks placed on specific spatial locations (x_i, y_i, z_i), while the appropriate interconnections among them are formed by the routing resources. Different cost functions might be employed during the P&R steps. More specifically, based on the application constraints, it is possible to use either a connectivity-aware or a power-aware approach. In the first case, the logic functions are placed and routed by having as goal to achieve as high as possible operation frequencies, while in the latter approach alternative design parameters (such as power dissipation) are the primary design goals.

By the end of this step, we have the complete P&R of the application on the 3D FPGA device. In order to prove the effectiveness of the proposed methodology, we study the maximum operation frequency, the power/energy consumption, as well as the hardware resources utilization (in terms of routing fabric). The last parameter is very crucial to determine the percentage of utilized vertical interconnections (i.e., TSVs), since they exhibit increased fabrication cost. When the P&R does not meet the designer's criteria, there is a feedback to the third step of the proposed methodology for additional improvements.

In order to determine the power/energy consumption of the application implemented onto the derived 3D-reconfigurable architecture, we use a new CAD tool, named 3DPower. This tool incorporates existing power models proposed in [24]. These models were extended to handle sufficiently the extra hardware parameters (such as multiple layers, TSV connections, etc.) introduced by the integration on the third dimension.

4. Meander Framework for 3D FPGAs

The proposed exploration methodology for building sufficient 3D-reconfigurable architectures is software supported by three new CAD tools, named 3DPart, 3DPRO, and 3DPower. These tools are part from the 3D MEANDER design framework [5, 15, 16], depicted in Figure 6. This flow utilizes existing CAD tools from the 2D toolset, which do not need to be aware of the three-dimensional FPGA topology. More specifically, new tools replaced the P&R and power consumption estimation, as these tasks consider the particular features of the 3D FPGAs. We have replaced the current version of P&R tool of the 2D flow (i.e., EX-VPR [25]) with the proposed P&R tool, named 3DPRO. We have also replaced the existing PowerModel tool, with the new 3DPower for modeling and calculating power/energy consumption in 3D architectures, while we have added an additional tool, named 3DPart, which deals with the application partitioning to 3D stack. To the best of our knowledge, this toolset is the first complete framework in academia for exploring alternative 3D-reconfigurable architectures starting from a hardware description language (HDL) and ending up to configuration file generation. Next, we describe in more detail the employed algorithms, as well as their software implementation, regarding the three new CAD tools.

4.1. Partitioning

The first of the new CAD tool deals with three tasks: (i) the application's partitioning, (ii) the partitioning to layer assignment, and (iii) the layer ordering. All of them are crucial for efficient implementation onto 3D architectures. Up to now, many years of research have been spent to develop fast and accurate algorithms just for supporting the first task (i.e., the application's partitioning). As the three-dimensional integration technologies are not studied efficiently yet, we are not aware about any other existing tool either for assigning partitions to devices layers of a 3D FPGA or order these layers.

An efficient partitioning algorithm can alleviate a number of design problems. Among others, by assigning closely layers that exhibit high data transfers, it is feasible to achieve higher operation frequencies. In addition to that, clusters of functions with high bandwidth requirements should be assigned to logic blocks that belong to the same layer, as there is plethora of routing resources compared to the reduced resources available for interlayer connectivity. Moreover, the appropriate selection of layer ordering, based on the existing power sources on them, might prevent failures related to heat dissipation. Finally, the layer ordering might alleviate congestion problems, as the interlayer communication resources are limited compared to routing wires of each layer.

The development of research in partitioning in the past two decades can be found in a comprehensive survey [26]. As the performance variation regarding numerous design parameters is tightly firmed to the employed interlayer communication fabric, a good partitioning among others have to limit the number of signals travelling through layers. This constraint is also known as a min-cut partitioning approach. However, apart from the min-cut, which is thought to be the objective for relevant approaches [2, 3, 23], the proposed one also takes into consideration additional design parameters (i.e., spatial distribution of

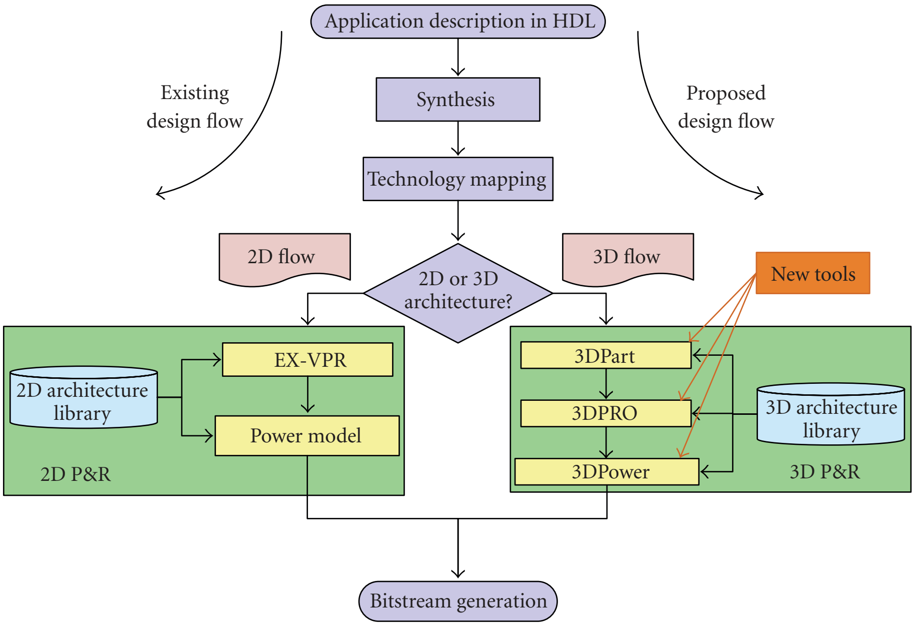


Figure 6: The MEANDER framework: (a) the left branch concerns the design of 2D conventional FPGAs and (b) the right branch concerns the design of 3D FPGAs.

TSVs among layers, area per layer, operation frequency, etc.).

The proposed algorithm that realizes these tasks is shown in Algorithm 1. Initially, the algorithm calculates the partitioning of application's hypergraph (i.e., application netlist) to a number (equals at least to the total device layers). Then, two iterative loops are applied in order to find the layer to which each of the logic functions has to be assigned. More specifically, firstly we are interest to derive an acceptable assignment of partitions to the device layers (in terms of the design constraints), then we try to determine the optimal ordering of these layers. Whether the number of partitions is higher compared to the device layers, then during the first task, more than one partition might be combined and assigned to the same layer of the 3D FPGA.

4.2. Placement Algorithm

After splitting the application to device layers, the placement algorithm assigns the application's ith logic function to the available hardware logic block, placed on physical location (x_i, y_i, z_i). The placement algorithm is based on simulated annealing. By analogy with this physical process, each step of the simulated annealing algorithm replaces the current solution by a random "nearby" solution, chosen with a probability that depends on the difference between the corresponding function values and on a global parameter called temperature, which is gradually decreased during the process. More specifically, during the execution of the placement algorithm, pairs of logic blocks are selected and swapped randomly, until either the resulted placement is good enough or the maximum number of iterations is reached. The efficiency of a placement is characterized by calculating the cost function, shown in (2) and (3):

$$\begin{aligned}\Delta\text{Cost} = \alpha \times \frac{\Delta\text{Wire\_cost}}{\text{Previous Wire\_cost}} \\ + (1-\alpha)\frac{\Delta\text{Time}_{\text{cost}}}{\text{Previous Time}_{\text{cost}}},\end{aligned} \tag{2}$$

where

$$\text{Timing}_{\text{cost}} = \sum_{\forall i,j \in \text{application}} \{\text{Delay}(i,j) \times \text{criticality}(i,j)^{\text{const}}\},$$

$$\begin{aligned}\text{Wiring\_cost} = \sum_{i=1}^{\text{Total\_Nets}} \left\{ q(i) \times \left[\left(\frac{\text{bb}_x(i)}{C_{\text{av},x}^{\beta}(i)} + \frac{\text{bb}_y(i)}{C_{\text{av},y}^{\beta}(i)} \right) \right.\right. \\ \left.\left. + \left(\varepsilon \times \frac{\text{bb}_z(i)}{C_{\text{av},z}^{\gamma}(i)} \right) \right] \right\}.\end{aligned} \tag{3}$$

Whenever the value of this cost function is reduced, the swap is kept. However, if the cost value increases, then the probability of keeping the swap is reduced with the execution time.

In this cost function, the factor α balances the effort of placement algorithm to optimize either the wirelength or the application's delay. The delay(i, j) denotes the delay between the logic elements i and j (a source-sink path of a network), the factor *const* is a constant, while the criticallity(i, j) gives the importance in terms of how close to the critical path is the network i. Similar to [13, 25], its mathematical expression is defined as Criticallity$(i, j) = 1 - \text{Slack}(i, j)/\text{Delay}_{\text{max}}$, where delay$_{\text{max}}$ is the delay of the circuit critical path, and slack(i, j) is the amount of delay

```
while (accept partition ← True) do
  {
  subgraphs ← split (netlist, number of layers);
  while (accept partition to layer assignment ← True) do
    {
    while (accept layer ordering ← True) do
      {
      connections ← calculate (interconnections among subgraphs);
      estimate variation among partitions (power, area, delay);
      goal ← evaluate retrieved partitioning;
      if (goal not optimal) then
        {
        try to repartition the netlist ();
        }
      else
        {
        accept partition ← True;
        }
      }
    }
  }
```

Algorithm 1: Algorithm for application partitioning, partitions to layer assignment and layer ordering tasks.

```
$P$ ← Initial_Random_{Placement}();
$T$ ← Initial_Temperature();
$R_{limit}$ ← Initial_$R_{limit}$ ();
while (Exit_Criterion() not TRUE) // outer loop
  {
  while (loop_criterion() not TRUE) // inner loop
    {
    $P_{new}$ ← Random_swap_Placements($P, R_{limit}$);
    $\Delta Cost$ ← Cost($P_{new}$) − Cost($P$);
    $r$ ← Random_value(0, 1);
    if ($r < e^{-\Delta CT}$) // accept the movement
      {
      $P$ ← $P_{new}$;
      }
    }
  $R_{limit}$ ← Update ($R_{limit}$);
  $T$ ← Update (Temperature);
  }
```

Algorithm 2: The proposed simulating annealing-based algorithms for placement on 3D FPGAs.

that could be added to this connection before it affected the application's critical path. The factors $\mathrm{bb}_x(i)$, $\mathrm{bb}_y(i)$, and $\mathrm{bb}_z(i)$ denote the dimensions of the 3D bounding box for network i, while the $q(i)$ is a scaling factor for this bounding box, used to make more accurate estimations about the wire-length for nets with more than 3 terminals [13]. The $C_{\mathrm{av},x}(i)$, $C_{\mathrm{av},y}(i)$, and $C_{\mathrm{av},z}(i)$ parameters correspond to the average width of routing tracks on x, y, and z axis of the bounding box for network i, while they are used in order to force placement algorithm to take into consideration the available (fabricated) routing resources. The value of these parameters depends solely on the fabricated interconnection resources, while it is constant during the placement. The values of β and γ control the relative cost of employing narrower and wider routing channels. More specifically, when their values are 0, then the cost function results in the conventional bounding box approach. Otherwise, as higher the values of these parameters are, then more and more tracks from narrowest routing channels have increased cost value, compared to the wider channels. We employ a different relative cost (γ) for the TSVs, as the placement algorithm has to pay effort to not waste this kind of connections. Finally, by using an additional factor, denoted as ε, we discourage the placer to put functions in different layers.

Algorithm 2 shows the proposed 3D placement algorithm which is realized as part of the 3DPRO CAD tool. As it was already mentioned, the functionality of this approach is based on simulated annealing. In order to obtain high-quality solutions in a reasonable computation time with such an approach, a good annealing schedule is essential. The proposed schedule incorporates some of the features provided in relevant references [13, 25, 27–29], while we propose a new temperature update scheme, as well as an exit criterion. The total moves per temperature are equal to $N = m \times (\mathrm{LE})^{4/3}$, where m denotes a default number of moves (usually 10), while the LE represents the number of logic elements that build the FPGA. This approach is similar to the one found in existing approaches [13, 25, 28].

In this algorithm, the value of P denotes each of the derived placements; R_{limit} determines the maximum horizontal and vertical distance between two logic blocks that are swapped during the annealing procedure, while its value is reduced linearly during the algorithm's execution.

For our specific architecture, the mathematic expression that describes this reduction is shown in (4):

$$R_{\text{limit}} = d \times \text{Previous\_}R_{\text{limit}}. \tag{4}$$

Whenever the value of R_{limit} is small enough, then the swaps of logic blocks occur for relative closely placed blocks. Such local swaps tend to result in relatively small changes in the placement cost, increasing their probability of acceptance. Initially, the value of this parameter is set to span of the entire layer, while whenever the temperature is updated, then this value is recalculated.

The temperature update is defined by (5):

$$\begin{gathered} T_i = T_0 \times e^{-Ai}, \\ \text{where } A = \frac{1}{N} \times \ln\left(\frac{T_0}{T_N}\right). \end{gathered} \tag{5}$$

In this equation, T_i is the temperature for iteration i, where i increases from 0 to N. Regarding the T_0 and T_N parameters, they correspond to initial and final temperatures, respectively. The employed temperature update scheme guarantees that we will result very closely to the optimal placement. Employing an annealing procedure that spends more time at the most productive temperatures (those that a significant fraction of moves is being accepted), compared to the case where temperature is high (almost any swap is kept), leads to significant improvement in placement's cost. We have to mention that, in practice, the ideal cooling rate cannot be determined beforehand, and should be empirically adjusted for each problem. The procedure of swapping the spatial location of logic blocks (i.e., annealing) is continued as far as the temperature is higher than a small fraction of the average cost of a net. After that point, any movement that results in increase of cost is unlikely to be accepted.

4.3. Routing Algorithm

By defining the placement of logic functions on the 3D FPGA, the routing algorithm forms the appropriate connections among the utilized hardware blocks through the available interconnection fabric. The proposed routing algorithm is an extended version of the Pathfinder negotiated congestion [30]. During the first iterations, a number of networks are allowed to share the same routing fabric. However, as the number of iterations increases, this is gradually prohibited, until the final routing, where each network uses dedicated routing fabric. The proposed routing algorithm can find the narrowest horizontal and vertical channel widths for which the application is fully routable.

As the vertical interconnections among layers are limited, the routing algorithm sets the weights of TSVs to a higher value (compared to routing wires of each plane) in order to discourage the router to form unnecessary bends between horizontal and vertical wires. Also, this forces the router not to connect logic blocks placed on one layer by using interconnection fabric from different layers.

The employed cost function that guides the proposed routing algorithm follows:

$$\begin{aligned} \Delta\text{Cost}(n) = {} & [\text{Criticallity}(i,j) \times \text{Delay}(n)] \\ & + (1 - \text{Criticallity}(i,j)) \times [b(n) \times h(n) \times p(n)], \end{aligned} \tag{6}$$

where Delay(n) is the delay of hardware component n, while the parameters $b(n)$, $h(n)$, and $p(n)$ represent the base cost, the historical congestion cost, and the present congestion cost for the hardware component n, respectively. In order to come to acceptable solutions without overusing the routing resources, the value of $p(n)$ increases with the execution time.

4.4. Power Estimation

The third developed tool, named *3DPower*, is responsible for the modeling and calculation of energy/power consumption for applications implemented onto 3D FPGAs. This tool adopts some principles from existing work proposed in [24] regarding conventional (2D) reconfigurable architectures. However, its models are refined extensively in order to be aware about a number of (heterogeneous) functional layers, as well as the multiple fabrication technologies of 3D stacked ICs. The pseudocode of this algorithm is shown in Algorithm 3.

In this algorithm, the transition density is an efficient measure of the switching activity of each signal within the circuit. Such a model has two parameters [24]:

(1) transition density (7) that denotes the average number of transitions per unit time, where $n_x(T)$ represents the number of transitions within time T:

$$D(x) = \lim_{T \to \infty} \frac{n_x(T)}{T}, \tag{7}$$

(2) static probability (8) that corresponds to the probability of the signal being high for a certain time period:

$$P(x) = \lim_{T \to \infty} \frac{1}{T} \int_{-T/2}^{T/2} x(t)\,dt. \tag{8}$$

5. Exploration and Comparison Results

This section provides both qualitative and quantitative comparisons among the proposed methodology for implementing digital applications on 3D-reconfigurable architectures, compared to alternative solutions that can be found in relevant literature. Since the efficiency of application implementation on 3D FPGAs depends mainly on the employed P&R algorithms, we perform a qualitative comparison among our proposed tool (3DPRO), the PR3D [3], and the TPR [2], which are the only available tools for P&R on 3D FPGAs. The results are summarized in Table 2. Given a 3D topology, the proposed methodology, and hence the CAD tool, can explore a plethora of parameters such as delay, energy/power

```
for i = 0 to Total Networks
    {
    for each Logic Block that form Network i
        {
        calculate static probability();
        calculate transition density; ()
        }
    calculate activity of net i();
    net_power = 0;
    for each segment used to route this net
        {
        calculate capacitance of segment();
        net power = net power + switching power for this network();
        }
    total power = total power + net power;
    write power file;
```

Algorithm 3: Algorithm of the *3DPower* tool for 3D FPGAs.

Table 2: Qualitative comparison between TPR and our proposed solution.

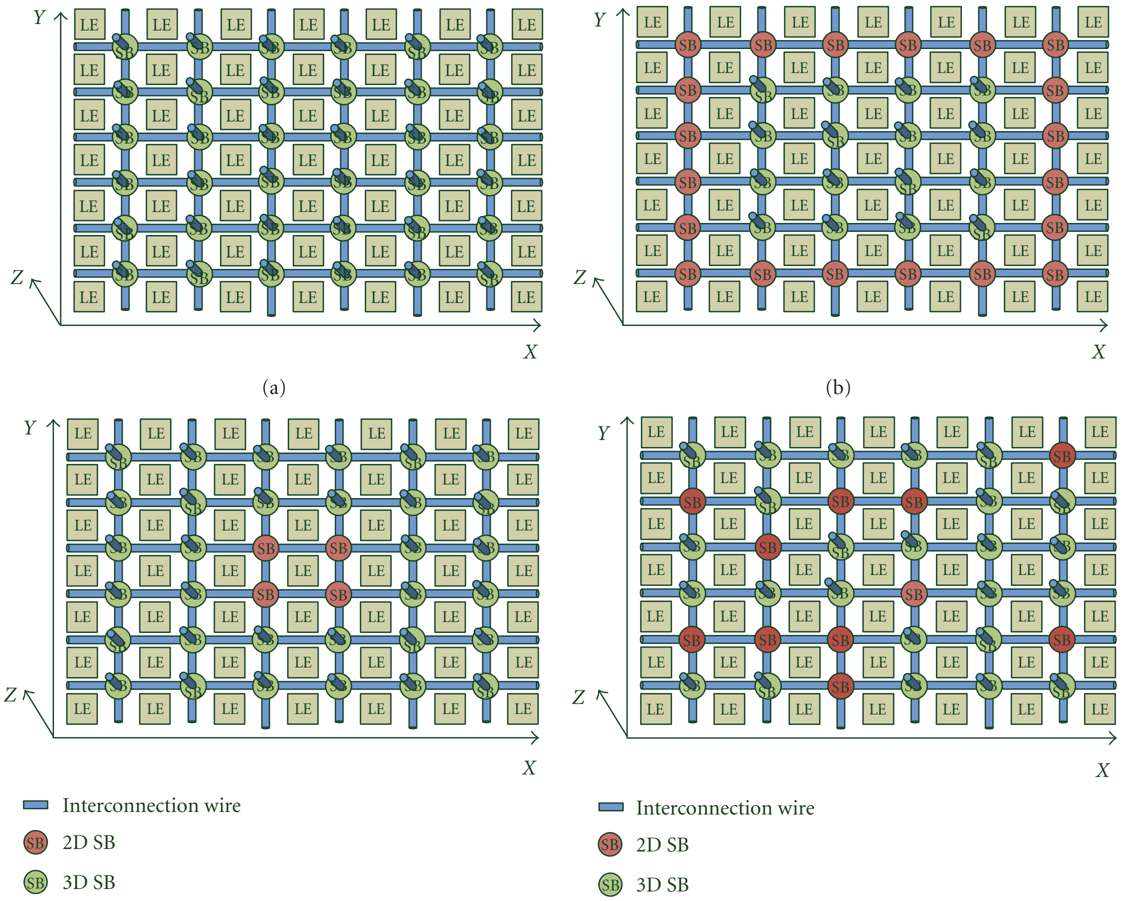
Feature	TPR [2]	PR3D [3]	3DPRO (Proposed)
Architecture exploration	Yes	No	Yes
Measure delay	Yes	Yes	Yes
Measure wirelength	Yes	Yes	Yes
Measure power	No	Yes	Yes
Supported switch boxes	Subset Wilton Universal	ASIC devices	Designer specified
Heterogeneous interconnect (simultaneously 2D/3D SBs)	No	Yes	Yes
Vias exploration	No	No	Yes
Part of complete framework	No	No	Yes

consumption, leakage power, and silicon area. Furthermore, it supports the evaluation of alternative architectures, in terms of fabricated TSVs, while the TPR [2] employs a full-connectivity scenario, where each SB can form connections on the adjacent stacked functional layers. However, this scenario does not correspond to a realistic approach for the 3D P&R problem, as the TSVs are prefabricated before the design implementation. Also, it is not possible to integrate so high number (or density) of TSVs per layer, as they occupy significant area, increasing among others the yield cost. Consequently, the 3DPRO provides more flexibility to the designer in order to perform architecture level exploration.

The effectiveness of the proposed methodology is evaluated with the usage of the 20 largest benchmarks from the Microelectronics Center of North Carolina (MCNC) benchmark suite [31]. For each of them, we perform an exploration regarding various design parameters.

During our exploration methodology we study the impact of alternative distribution of 3D SBs into 3D FPGAs. More specifically, we quantify the potential gains of employing the available interconnections schemes shown in Figure 7, each of which has advantages and disadvantages. Among others, the approach, where all the SBs of the layer form connections to the rest layers (shown in Figure 7(a)), provides the maximum connectivity improvement, in a penalty of the increased silicon area occupied from so high amount of TSVs. The second and third approaches affect distribution scenarios, where the 3D SBs are assigned either to the center (Figure 7(b)) or the periphery (Figure 7(c)) of each layer, respectively. Both of these implementations exhibit a piece-wise regular interconnection architecture, which leads to retrieve potential gains from the employed supporting CAD tools (i.e., P&R). On the other hand, such approaches might increase either the wirelength or the spatial distribution of other crucial design parameters (i.e., distribution of power sources). For instance, regarding Figure 7(b), the center of the layer exhibits significant higher connectivity demands, which might result to increased power dissipation, or on-chip temperature values. Finally, the last approach (shown in Figure 7(d)) affects a full-custom assignment for 3D SBs, where these connections are assigned based on the connectivity demands for interlayer connectivity, introduced either from a specific application (i.e., MPEG4, GSM) or from an application domain (i.e., multimedia, communications, etc).

Based on the previous conclusions, we might derive that none of these assignments of 3D SBs is efficient in terms of performance, power consumption, and silicon area. Due to this, throughout this paper, we employ a new scheme for assigning 3D SBs, which is depicted in Figure 8.



(a)

(b)

(c)

(d)

Figure 7: Alternative distribution scenarios for 3D SBs: (a) all the SBs are 3D, (b) the 3D SBs are assigned into the device center, (c) the 3D SBs are assigned into the device periphery, and (d) a full-custom assignment of 3D SBs.

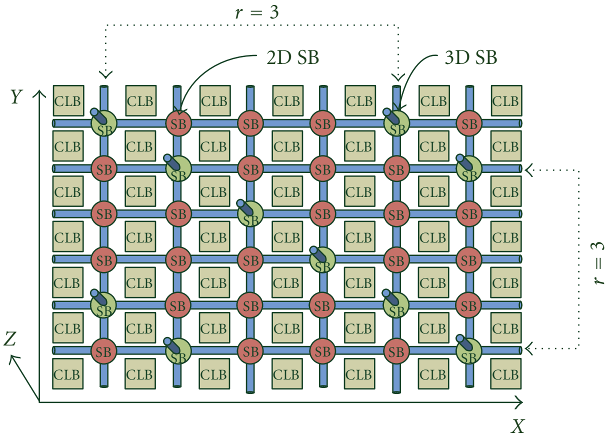
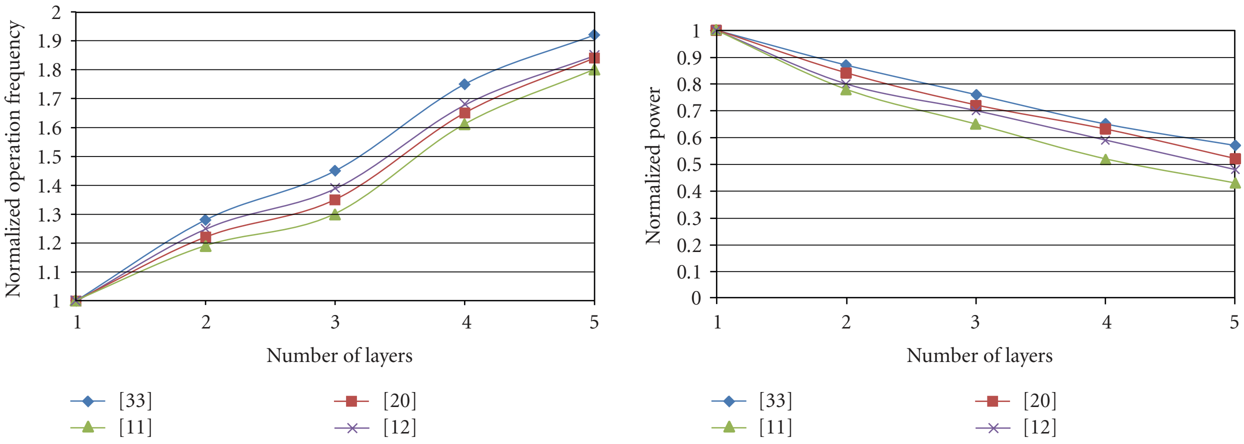


Figure 8: A layer from a 3D FPGA architecture with $r = 3$.

Such an interconnection architecture is based on a uniform distribution of 3D SBs across the device layers, which leads to significant improvement of routability problems.

In order to show the way that the 3D SBs are assigned, let us assume a layer of size $X \times Y$, where the percentage of 3D SBs are K% (over the total number of SBs exist in the layer). Then, the pattern of assigning these 3D SBs in each row and column of the layer is derived as follows: place a 3D SB to a spatial location (x, y) of a certain layer, then the neighboring 3D SBs across horizontal and vertical axis are assigned to the locations $(x + r, y, z)$ and $(x, y + r, z)$, respectively. In this expression, r indicates Manhattan distance between successive assigned 3D SBs (i.e., the number of 2D SBs between two neighboring 3D SBs). Due to the uniform distribution of vertical connections, the router of 3DPRO tool is assisted significantly in order to employ the vertical connections in a more efficient manner.

We have to stress that the combination of 2D and 3D SBs may result in a number of fabrication issues. More specifically, as the 3D SBs are larger compared to 2D SBs (due to the increased number of transistors), it is not obvious the way that these two types of SBs are combined in the same layout. For this purpose, our proposed methodology can support two solutions: (i) each 2D SB occupies area equal to the one required by a 3D SB, and (ii) to suppose that all the SBs are 3D but with a variation on the number of vertical interconnections (i.e., instead of changing the number of 2D/3D SBs, we change the number of TSVs per 3D SB). Both



(a)

(b)

Figure 9: Average variation of application's delay and power consumption for a number of layers and TSVs with different electric characteristics.

of them is valid and might prevent the problem of nontrivial layouts of the regular structure shown in Figure 8.

Figures 9(a) and 9(b) plot the variation of maximum operation frequency and power consumption, respectively, over a number of different fabrication technologies for TSVs [11, 12, 20, 32], found in relevant references. We have to mention that among alternative process technologies, we employ the same number and distribution of TSVs. Even though more advanced TSVs might be found, the aim of Figure 9 is to illustrate that alternative fabrication processes for interlayer communication lead to significant improvement of critical design parameters (i.e., operation frequency and power consumption). More specifically, regarding the device with three functional layers, the performance improvement against the 2D FPGA balances between 20% and 40%, based on the selected electrical characteristics of the vertical connections. Similarly, such architecture reduces the power consumption compared to 2D FPGA from 22% up to 36%. These graphs concern the 3D architectures where all the SBs can form connections to the rest of the layers (i.e., 3D SBs).

Throughout this paper, the employed experimental setup for the targeted 3D-reconfigurable architectures can be summarized as follows:

(1) the 3D architectures consist of up to five functional layers;

(2) the hardware resources of each functional layer are identical. Based on this, both the amount of hardware resources (i.e., logic blocks, routing wires, TSVs, etc.) and their spatial location among layers are irrelative;

(3) the percentage of vertical interconnects (i.e., TSVs) per functional layer ranges from 10% up to 100%, with a step of 10%;

(4) each 3D SB realizes four vertical connections. In other words, each 3D SB placed on functional layer i has 4 TSVs to the layer $i-1$ and 4 other TSVs for the layer $i+1$. An exception to this occurs for the bottom and top layers of the 3D stack;

(5) the electrical parameters for each TSV correspond to fabrication technologies for 3D ICs found in relevant references.

The next figures show the average variation of some design parameters for 3D FPGAs with alternative interconnection scenarios regarding the TSVs distribution. For these graphs, the TSV's resistance is 350 mΩ, while its capacitance is 2.5 fF [12]. The horizontal axis corresponds to the percentage of fabricated TSVs on each layer, while the vertical one shows the normalized value of each design parameter (i.e., delay, power, Energy × Delay Product, etc.) in relation to a 2D FPGA. The percentage of TSVs for each layer corresponds to the number of 3D SBs placed on this layer, over the total number of SBs for this layer. However, (9) provides the mathematical expression for calculating this percentage. The architecture that corresponds to 100% fabricated TSVs per layer correspond to a 3D FPGA where every SB can form connections to the third dimension (similar to the TPR [2]):

$$\text{Percentage of 3D SBs} = \frac{\text{Number of 3D SBs per layer}}{\text{Total number of SBs } (2\text{D} + 3\text{D}) \text{ per layer}} \times 100\%. \tag{9}$$

Figure 10 plots the average variation of Energy × Delay product (EDP) for the MCNC benchmarks [31], benchmarks for alternative 3D FPGAs. The normalization was performed over the EDP value of a conventional (i.e., 2D) FPGA. It can be seen that the increase of the number of layers results in more efficient realizations of the applications in 3D FPGAs. Also, we can claim that the proposed partitioning and P&R algorithms provide promising results for 3D architectures, where only a percentage of SBs forms 3D connections.

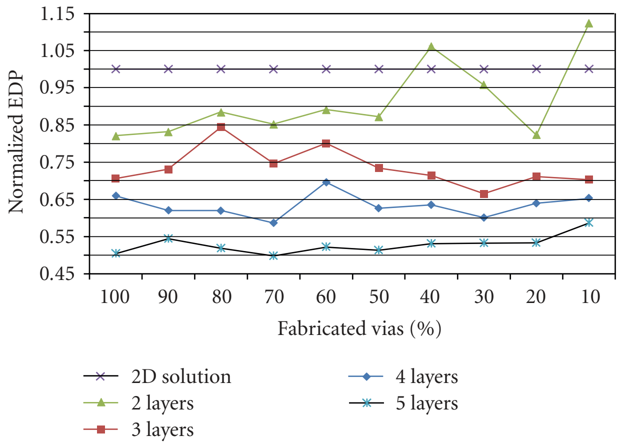


Figure 10: Average Energy×Delay Product (EDP) for different number of functional layers and percentage of fabricated TSVs.

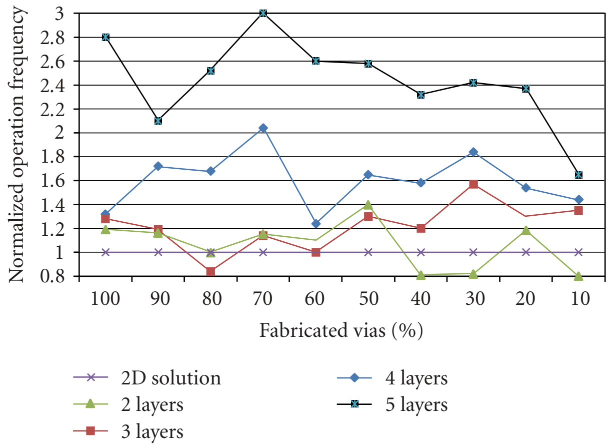


Figure 11: Average operation frequency over the MCNC benchmarks for different number of layers and percentages of fabricated TSVs.

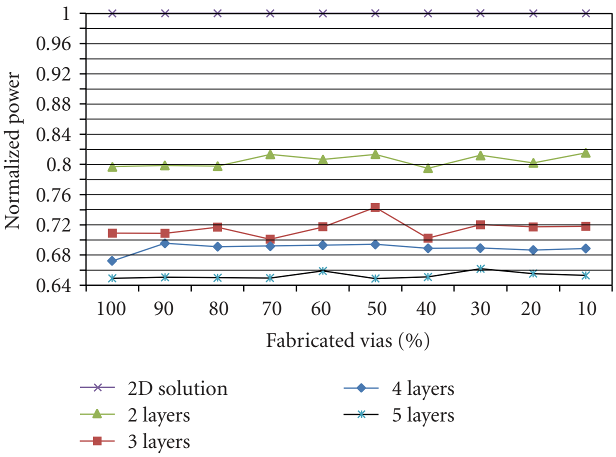


Figure 12: Average power consumption over the MCNC benchmarks for different number of functional layers and percentage of fabricated TSVs.

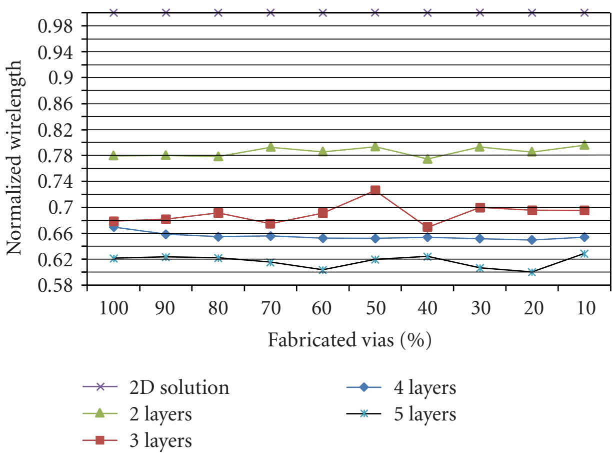


Figure 13: Average wirelength over the MCNC benchmarks for different number of functional layers and percentage of fabricated TSVs.

One of the main advantages of 3D integration is the increased operation frequency. Figure 11 plots the average variation of this parameter over the MCNC benchmarks for different number of layers and percentage of fabricated TSVs. The values in the vertical axis are normalized over the operation frequency that exhibits a 2D FPGA. Based on this graph, it is evident that the solution with five layers outperforms all the other implementations, achieving to increase the device operation frequency up to 30% (for percentage of fabricated vias 70%), as compared to 2D architectures. Moreover, by increasing the number of functional layers, the applications exhibit higher operation frequencies. However, it should be mentioned that for some architectures (i.e., those consisting of two layers), the operation frequency is low, as compared to the 2D solution, as the limited vertical interconnections stress the routing algorithm.

Due to the fact that reconfigurable devices exhibit high power dissipation, which is mainly occurred due to increased resistance/capacitance values exhibited by the interconnection network, we also study the total power requirements of the alternative 3D architectures. The results are summarized on Figure 12. Based on this graph, as the number of functional layers increases, the total power consumption reduced. Also, we can conclude that the 3D FPGA with five layers achieves to reduce the power consumption up to about 35%, as compared to 2D FPGA.

The gains in the performance and energy consumption depend on the intrinsic feature of 3D integration, regarding the minimization of wirelength. Among others, shorter wires lead to smaller resistance/capacitance, and hence to reduced delay, power/energy consumption, as well as silicon area footprint. Figure 13 shows the average wirelength requirements for different number of functional layers and percentages of fabricated TSVs over the MCNC benchmarks. As the number of functional layers increases, the total wirelength is reduced due to the increased number of neighbors. Also, the wirelength reduction from 2D FPGAs to a 3D stack with two function layers is about

Table 3: Comparison results between MCNC benchmarks: implementation in 2D and 3D FPGA architecture with three functional layers as well as $K = 30\%$ and $K = 100\%$ TSVs.

Benchmark	Wirelength ($\times 10^3$)			Delay ($\times 10^{-9}$ sec)			Power ($\times 10^{-3}$ Watt)		
	2D	3D	3D	2D	3D	3D	2D	3D	3D
		$K = 30\%$	$K = 100\%$		$K = 30\%$	$K = 100\%$		$K = 30\%$	$K = 100\%$
alu4	37.81	35.65	37.09	73.7	45.9	47.2	115.93	67.22	73.52
apex2	58.99	56.06	53.16	105	53.4	50.5	106.99	52.91	54.69
apex4	37.76	38.16	37.92	79.2	43.7	41.1	068.38	38.91	34.77
bigkey	32.71	48.57	43.68	38.1	23.0	22.2	317.79	180.60	172.70
clma	430.44	294.60	280.50	170	114	103	456.00	323.02	283.89
des	52.70	45.44	43.82	63.4	43.3	39.2	231.04	142.05	158.61
diffeq	34.88	36.31	31.08	58.8	60.3	60.4	104.14	97.05	116.54
dsip	29.90	35.46	33.69	31.9	22.0	20.2	349.29	235.90	194.56
elliptic	102.24	92.15	94.91	89.1	92.6	83.7	272.58	280.38	266.23
ex1010	36.90	33.78	31.47	54.0	47.7	46.7	103.95	83.44	90.77
ex5p	129.65	167.00	146.42	163	73.5	73.8	117.65	46.71	46.37
frisc	174.05	99.83	91.26	100	110	105	152.35	160.13	163.65
misex3	39.31	38.26	37.88	86.9	40.3	46.4	86.93	41.12	41.80
pdc	238.78	173.32	160.37	153	79.7	78.3	179.82	86.11	82.73
s298	55.85	44.65	42.30	187	92.5	87.1	78.68	36.53	41.72
s38417	172.01	169.52	155.97	90.2	64.1	74.8	284.25	185.84	25.92
s38584	129.14	152.62	136.39	97.3	60.8	55.6	264.54	183.45	129.86
Seq	52.80	54.49	48.74	66.6	47.6	44.8	132.37	105.93	89.05
Spla	148.19	113.47	125.03	132	64.8	66.4	125.79	64.58	61.77
Tseng	25.89	23.85	21.07	63.9	54.8	55.5	93.76	71.52	71.60
Average	101.00	87.659	82.64	95.2	61.7	60.1	182.00	124.00	122.00
Ratio	1.00	0.87	0.82	1.00	0.65	0.63	1.00	0.68	0.67

22%, while if we increase the number of layers to five, then the additional reduction to total wirelength is only 18%.

Several points can be made from the previous graphs/plots. Among others, as we increase the number of layers, the applications are realized more efficiently in 3D FPGAs compared to 2D-reconfigurable architectures. Secondly, we can claim that the proposed algorithms provide promising results for 3D architectures, where only a percentage of SBs forms connections to the third dimension. More specifically, we can conclude that as we vary the number of fabricated TSVs on each layer, significant reduction on design parameters may be achieved, leading to more efficient 3D architectures.

It is worth mentioning that in contrast to Figure 9, where the increase of number of layers leads to monotonous gains in performance and power consumption, this is not valid for Figures 10–13. In these graphs, we also study the impact of different amount and distribution of 3D SBs over the device layers. More specifically, as we modify the percentage of 3D SBs over the total number of SBs placed onto a layer, we alter the connectivity resource graph (i.e., the graph that describes the routing resources of the target device). Due to this, the routing algorithm has to pay effort in order to find new paths to connect the logic blocks. As these paths do not exhibit same lengths, they have significant variations on the RC parameters. The employed models both for estimating delay [33] and power [24] are related to three main parameters of wires, namely, their length, resistance, and capacitance. The nonmonotonous form of these curves is due to these variations of the resistance/capacitance and wirelength for the routing paths.

The nonmonotonous behavior of these curves (for given the number of layers) can be explained as follows: each of these curves shows the impact of alternative 3D architectures, which occupy different percentage of TSVs. More specifically, as we modify the percentage of 3D SBs over the total number of SBs placed onto a layer, we alter the connectivity resource graph (i.e., the graph that describes the routing resources of the target device). Due to this, the routing algorithm has to pay effort in order to find new paths to connect the logic blocks. As these paths do not exhibit same lengths, they have significant variations on the RC parameters. The employed models both for estimating delay [33] and power [24] are related to three main parameters of wires, namely, their length, resistance, and capacitance. Consequently, the variations of these parameters results in the nonmonotonous behavior of these architecture solutions. Also, we have to mention that for each number of layers, these graphs do not show curves, but distinct architecture solutions regarding devices with specific percentages of TSVs. For demonstration purposes, we have just connected dots with lines in order to show the variation on 3D efficiency as we alter the amount of interlayer connections.

Table 4: Comparison results between 20 biggest MCNC benchmarks: via utilization in 3D FPGA architecture (with 30% and 100% via links, 3 layers, and maxima operation frequency).

Benchmark	30% 3D SBs			100% 3D SBs		
	Total vias (fabricated)	Actually utilized vias	(%)	Total vias (fabricated)	Actually utilized vias	(%)
alu4	2799	1148	41%	10109	3639	36%
apex2	3456	1140	33%	9600	4512	47%
apex4	2705	1190	44%	6242	2185	35%
Bigkey	3379	1385	41%	7798	3119	40%
Clma	17781	7290	41%	46570	19559	42%
Des	2540	813	32%	11642	5123	44%
Diffeq	2289	847	37%	7630	2365	31%
Dsip	3266	1143	35%	10886	3484	32%
Elliptic	6823	2661	39%	19246	8468	44%
ex1010	6919	2491	36%	23064	10609	46%
ex5p	2705	1353	50%	9710	4370	45%
frisk	5841	1752	30%	16224	6003	37%
misex3	4032	1976	49%	9600	3840	40%
Pdc	5774	2021	35%	22745	10918	48%
s298	2580	903	35%	5530	2488	45%
s38417	7776	3577	46%	25920	12182	47%
s38584	8995	3418	38%	29983	13492	45%
Seq	3639	1674	46%	7798	3197	41%
Spla	5391	1941	36%	16589	6636	40%
Tseng	1903	564	30%	6344	3900	61%
Average	5030	1964	39%	15161	6504	43%
Ratio	1.00	0.39		1.00	0.43	

In order to evaluate the 3D FPGAs with reduced number of vertical connections, we provide some experimental results regarding the P&R on devices consisting of 3 functional layers with different percentage of fabricated TSVs (shown in Table 3). More specifically, we evaluate a 3D FPGA with $K = 30\%$ of the TSVs fabricated against a device with identical functional layers in each layer but with TSVs in every SBs (i.e., $K = 100\%$). For the sake of completeness, we provide also the metrics regarding the 2D FPGA. The percentage selection of 30% is retrieved from Algorithm 2 due to the minimum of EDP curve for a 3D device with three functional layers. We evaluate the alternative architectures in terms of wirelength, delay, and power consumption. Comparing with 2D devices, the above-mentioned results prove that the 3D architectures provide significant reduction in wirelength, delay, and power consumption.

Considering the percentage of fabricated vias equal to 30%, the average reduction in the wirelength, the delay, and the power consumption is 13%, 35%, and 33%, respectively. Similarly, the corresponding values for 100% vias are 18%, 27%, and 33%, respectively. Indeed, the wirelength reduction (i.e., resistance and capacitance reduction), due to 3D integration, results in remarkable improvements in delay and energy consumption. However, these savings seem to be independent of the number of fabricated TSVs, as the 3D device with $K = 100\%$ achieves almost similar gains compared to the one with $K = 30\%$. The proposed methodology for eliminating the number of vertical connections has a penalty in the number of routing tracks of each layer. The average increase of them is about 8%. However, the solution with less fabricated TSVs is more reliable (due to technology parameters) and cost-efficient. Also, with the current process technologies, it is almost impossible to fabricate such a high amount of vertical interconnections. Due to this, the proposed approach derives the gains of the 3D integration with a more feasible technology approach. The extra penalty in layer's area (due to the increased amount of routing wires) cannot outperform the benefits of 3D FPGAs, as compared to conventional 2D solutions. To the best to our knowledge, it is the first time in the literature where the efficiency of a 3D FPGA architecture remains unchanged with less hardware resources (i.e., fewer vias).

More details about the TSVs utilization on the 20 biggest MCNC benchmarks can be found on Table 4. As we can concluded, the percentage of utilized vias for three-layer FPGA architectures does not depend on the percentage of fabricated vertical links (vias) between layers. More specifically, the average utilization ratio of vias for FPGAs composed by 30% and 100% 3D SBs (vias) are 39% and 43%, respectively. Consequently, we proved that the design of efficient 3D FPGA architectures with smaller number of vias than 100% is feasible with reduced fabrication costs. In contrary, all the existing designs support "fully-populated" TSVs 3D FPGA designs only.

The last point is very important because we manage to achieve the same improvements employing less hardware resources (i.e., TSVs). More specifically, from 3D fabrication/manufacturing point of view, the smaller number of vertical connections means: (i) smaller fabrication costs and (ii) larger useful silicon area in each layer (a TSV contact occupies much more silicon area than a simple metal contact). The increased connectivity in the vertical axis means more silicon and eventually greater cost. Even though there are also two other known P&R tools, however, we could not provide sufficient comparison results against them. More specifically, the approach in [2] does not provide any estimation regarding the power consumption, while the one in [3] is not publicly available. Additionally, both of them assume "fully populated" TSVs 3D FPGA devices only (scenario $K = 100\%$).

6. Conclusions

A systematic software-supported methodology for exploring and evaluating alternative interconnection schemes for 3D FPGAs is presented. The methodology is supported by three new CAD tools (part of the 3D MEANDER Design Framework). The evaluation results prove that it is possible to design 3D FPGAs with limited number of vertical connections without any penalty in performance or power consumption. More specifically, for the 20 biggest MCNC benchmarks, the average gains in operation frequency, total wirelength, and energy consumption are 35%, 13%, and 32%, respectively, compared to existing 2D FPGAs with identical logic resources.

Acknowledgments

This paper is part of the 03ED593 research project, implemented within the framework of the Reinforcement Program of Human Research Manpower (PENED) and cofinanced by National and Community Funds (75% from EU.-European Social Fund and 25% from the Greek Ministry of Development: General Secretariat of Research and Technology).

References

[1] E. Beyne, "3D interconnection and packaging: impending reality or still a dream?" in *Proceedings of IEEE International Solid-State Circuits Conference (ISSCC '04)*, vol. 1, pp. 138–139, San Francisco, Calif, USA, February 2004.

[2] C. Ababei, Y. Feng, B. Goplen, et al., "Placement and routing in 3D integrated circuits," *IEEE Design & Test of Computers*, vol. 22, no. 6, pp. 520–531, 2005.

[3] S. Das, A. Fan, K.-N. Chen, C. S. Tan, N. Checka, and R. Reif, "Technology, performance, and computer-aided design of three-dimensional integrated circuits," in *Proceedings of International Symposium on Physical Design*, pp. 108–115, Phoenix, Ariz, USA, April 2004.

[4] A. Rahman, S. Das, A. P. Chandrakasan, and R. Reif, "Wiring requirement and three-dimensional integration technology for field programmable gate arrays," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 11, no. 1, pp. 44–54, 2003.

[5] K. Siozios, K. Sotiriadis, V. F. Pavlidis, and D. Soudris, "A software-supported methodology for designing high-performance 3D FPGA architectures," in *Proceedings of IFIP International Conference on Very Large Scale Integration (VLSI-SoC '07)*, pp. 54–59, Atlanta, Ga, USA, October 2007.

[6] C. H. Yu, "The 3rd dimension—more life for Moore's Law," in *Proceedings of International Microsystems, Packaging, Assembly Conference Taiwan (IMPACT '06)*, pp. 1–6, Taipei, Taiwan, October 2006.

[7] L. Shang, A. S. Kaviani, and K. Bathala, "Dynamic power consumption in Virtex$^{\text{TM}}$-II FPGA family," in *Proceedings of the 10th ACM/SIGDA International Symposium on Field Programmable Gate Arrays (FPGA '02)*, pp. 157–164, Monterey, Calif, USA, February 2002.

[8] D. Wang, Y. Xie, Y. Hu, H. Li, and X. Li, "Hierarchical fault tolerance memory architecture with 3-dimension interconnect," in *Proceedings of the 10th IEEE Region Annual International Conference (TENCON '07)*, pp. 1–4, Taipei, Taiwan, October-November 2007.

[9] R. Reif, A. Fan, K.-N. Chen, and S. Das, "Fabrication technologies for three-dimensional integrated circuits," in *Proceedings of the 3rd International Symposium on Quality Electronic Design (ISQED '02)*, pp. 33–37, San Jose, Calif, USA, March 2002.

[10] V. F. Pavlidis and E. G. Friedman, "Interconnect delay minimization through interlayer via placement in 3-D ICs," in *Proceedings of the 15th ACM Great Lakes Symposium on VLSI*, pp. 20–25, Chicago, Ill, USA, April 2005.

[11] A. W. Topol, D. C. La Tulipe Jr., L. Shi, et al., "Three-dimensional integrated circuits," *IBM Journal of Research and Development*, vol. 50, no. 4-5, pp. 491–506, 2006.

[12] S. Gupta, M. Hilbert, S. Hong, and R. Patti, "Techniques for producing 3D ICs with high-density interconnect," in *Proceedings of the 21st International VLSI Multilevel Interconnection Conference (VMIC '04)*, Waikoloa Beach, Hawaii, USA, September-October 2004.

[13] V. Betz, J. Rose, and A. Marquardt, *Architecture and CAD for Deep-Submicron FPGAs*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1999.

[14] M. Lin, A. El Gamal, Y.-C. Lu, and S. Wong, "Performance benefits of monolithically stacked 3D-FPGA," in *Proceedings of the 14th ACM/SIGDA International Symposium on Field Programmable Gate Arrays (FPGA '06)*, pp. 113–122, Monterey, Calif, USA, February 2006.

[15] http://vlsi.ee.duth.gr/amdrel.

[16] K. Siozios, G. Koutroumpezis, K. Tatas, et al., "A novel FPGA architecture and an integrated framework of CAD tools for implementing applications," *IEICE Transactions on Information and Systems*, vol. E88-D, no. 7, pp. 1369–1380, 2005.

[17] K. Siozios, K. Tatas, G. Koutroumpezis, D. Soudris, and A. Thanailakis, "An integrated framework for architecture level exploration of reconfigurable platform," in *Proceedings of the 15th International Conference on Field Programmable Logic and Applications (FPL '05)*, pp. 658–661, Tampere, Finland, August 2005.

[18] A. Rahman, J. Trezza, B. New, and S. Trimberger, "Die stacking technology for terabit chip-to-chip communications," in *Proceedings of IEEE Custom Integrated Circuits Conference*, pp. 587–590, San Jose, Calif, USA, September 2006.

[19] F. M. Finkbeiner, C. Adams, E. Apodaca, et al., "Development of ultra-low impedance through-wafer Micro-vias," *Nuclear Instruments and Methods in Physics Research Section A*, vol. 520, no. 1–3, pp. 463–465, 2004.

[20] S. M. Alam, R. E. Jones, S. Rauf, and R. Chatterjee, "Inter-strata connection characteristics and signal transmission in three-dimensional (3D) integration technology," in *Proceedings of the 8th International Symposium on Quality Electronic Design (ISQED '07)*, pp. 580–585, San Jose, Calif, USA, March 2007.

[21] D. M. Jang, C. Ryu, K. Y. Lee, et al., "Development and evaluation of 3-D SiP with vertically interconnected Through Silicon Vias (TSV)," in *Proceedings of the 57th Electronic Components and Technology Conference (ECTC '07)*, pp. 847–852, Reno, Nev, USA, May-June 2007.

[22] I. Das and J. E. Dennis, "Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems," *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 631–657, 1998.

[23] N. Selvakkumaran and G. Karypis, "Multiobjective hypergraph-partitioning algorithms for cut and maximum subdomain-degree minimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 3, pp. 504–517, 2006.

[24] K. K. W. Poon, S. J. E. Wilton, and A. Yan, "A detailed power model for field-programmable gate arrays," *ACM Transactions on Design Automation of Electronic Systems*, vol. 10, no. 2, pp. 279–302, 2005.

[25] K. Siozios, K. Tatas, G. Koutroumpezis, D. Soudris, and A. Thanailakis, "An integrated framework for architecture level exploration of reconfigurable platform," in *Proceedings of the 15th International Conference on Field Programmable Logic and Applications (FPL '05)*, pp. 658–661, Tampere, Finland, August 2005.

[26] C. J. Alpert and A. B. Kahng, "Recent directions in netlist partitioning: a survey," *Integration, the VLSI Journal*, vol. 19, no. 1-2, pp. 1–81, 1995.

[27] W. Swartz and C. Sechen, "New algorithms for the placement and routing of macro cells," in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD '90)*, pp. 336–339, Santa Clara, Calif, USA, November 1990.

[28] J. Lam and J.-M. Delosme, "Performance of a new annealing schedule," in *Proceedings of the 25th ACM/IEEE Design Automation Conference (DAC '88)*, pp. 306–311, Atlantic City, NJ, USA, June 1988.

[29] M. Huang, F. Romeo, and A. Sangiovanni-Vincentelli, "An efficient general cooling schedule for simulated annealing," in *Proceedings of International Conference on Computer-Aided Design (ICCAD '86)*, pp. 381–384, Santa Clara, Calif, USA, October 1986.

[30] L. McMurchie and C. Ebeling, "PathFinder: a negotiation-based performance-driven router for FPGAs," in *Proceedings of the 3rd International ACM Symposium on Field Programmable Gate Arrays (FPGA '95)*, pp. 111–117, Monterey, Calif, USA, February 1995.

[31] "Standard Cell Benchmark Circuits from the Microelectronics Center of North Carolina (MCNC)," http://vlsicad.cs.binghamton.edu/gz/PDWorkshop91.tgz.

[32] L. L. W. Leung and K. J. Chen, "Microwave characterization and modeling of high aspect ratio through-wafer interconnect vias in silicon substrates," *IEEE Transactions on Microwave Theory and Techniques*, vol. 53, no. 8, pp. 2472–2480, 2005.

[33] T. Okamoto and J. Cong, "Buffered Steiner tree construction with wire sizing for interconnect layout optimization," in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD '96)*, pp. 44–49, San Jose, Calif, USA, November 1996.

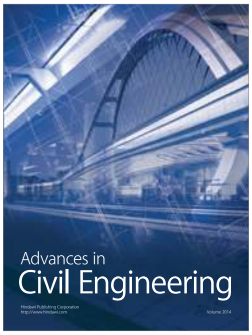
Journal of
Engineering

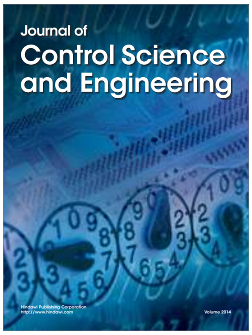
The Scientific
World Journal

International Journal of
Rotating
Machinery

Journal of
Sensors

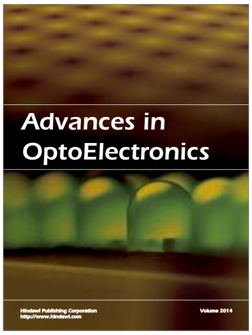
International Journal of
Distributed
Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science
and Engineering

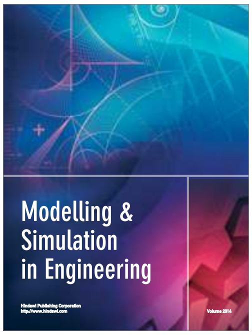
Journal of
Robotics

Journal of
Electrical and Computer
Engineering

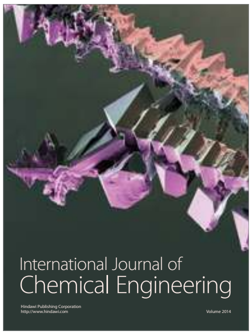

Advances in
OptoElectronics

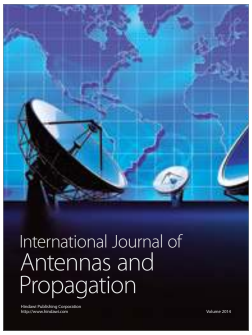
VLSI Design

International Journal of
Navigation and
Observation

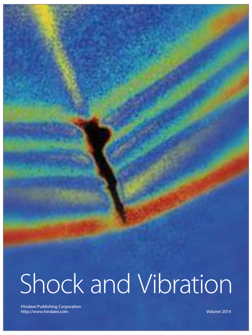

Modelling &
Simulation
in Engineering

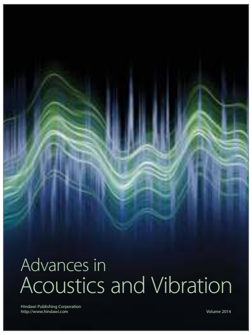
International Journal of
Aerospace
Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and
Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration