On Exploiting Partitioning-based Placement Approach For Performances Improvement of 3D FPGA

Sonda Chtourou, Mohamed Abid
CES Research Laboratory,
University of Sfax, Tunisia
Email: sonda.chtourou@ceslab.org.

Zied Marrakchi
Mentor Graphics,
Tunis
Email:zied_marrakchi@mentor.com

Emna Amouri and Habib Mehrez
LIP6 Research Laboratory
UPMC, Paris, France
Email: Habib.Mehrez@lip6.fr

***Abstract*— Three-dimensional Field Programmable Gate Arrays (3D FPGAs) represent a viable alternative to overcome challenges of integration complexity in modern embedded systems. Mapping applications into 3D FPGAs requires a set of accompanying suite of Computer-Aided Design (CAD) tools. One of critical issue of a 3D FPGA-based implementation is the quality and efficiency of associated CAD algorithms. In this paper, we are interested to investigate placement algorithms aspect to optimize proposed 3D FPGA performances. In fact, the way we distribute clusters between 3D FPGA layers has an important impact on performances. We present partitioning-based placement algorithm for 3D FPGA. The circuit is first divided into two layers with limited number of inter-layer interconnections, and then placed on individual layers. Placement solution of each layer is then gradually improved using adapted simulated annealing algorithm. We conduct experiments using exploration platform to compare partitioning-based and simulated annealing based placement approaches for proposed 3D FPGA architecture. Exploration results show that using partitioning-based placement algorithm achieves a saving in terms of power consumption, area and performance by an average of 15%, 18% and 10% Unlike DFPGA, MS-FPGA can deal with complex circuits.**

Keywords- Reconfigurable architecture, 3D FPGA, CAD tools, placement algorithms, partitioning-based approach.

I. INTRODUCTION

Since their introduction in the mid eighties, Field Programmable Gate Arrays (FPGAs) have seen a rapid growth and have become widely used in the computational devices domain which was originally dominated by microprocessors and Application Specific Integrated Circuits (ASICs). The distinguishing characteristic of FPGAs is their flexibility to implement any type of digital circuit after the manufacturing process. FPGAs enable a fast emulation of design alternatives and then lead to a faster design cycle. Moreover, they offer low Non-Recurring Engineering costs and fast time-to-market. All mentioned features make FPGAs an attractive choice for small and medium volume designs. However, compared to full custom ASIC design, FPGAs consume more power and energy and operate at lower frequency [1]. In fact, homogeneous FPGAs provide a large amount of programmable interconnect resources which occupy 80% of the total FPGA area [2]. Interconnect architectures represent the major factor behind power dissipation, area and delay. The 3D Integrated Circuit (IC) technology has emerged as one of the most promising solutions for overcoming the challenges in interconnection and integration complexity of modern circuit designs [3]. The 3D technology can be used effectively to reduce global interconnect wire-length and improve the system performance.

II. MOTIVATION AND PROBLEM FORMULATION

The quality of 3D FPGA device is controlled by three factors which are: quality of the 3D FPGA architecture, quality of the CAD tools used to map circuits into the 3D FPGA and electrical (i.e. transistor-level) design of the FPGA. Defining a 3D FPGA architecture is a challenge of fixing logic and routing resources so that these algorithms produce the most efficient possible results. In previous works [4,5], we explored new interconnect topologies and new 3D cluster-based Mesh FPGA architecture to improve the system overall performances. The proposed FPGA is constructed by stacking 2 identical 2D functional layers, while providing the required communication interlayer using Through-Silicon Vias (TSVs). The architecture of each layer is a cluster-based Mesh architecture which has a depopulated interconnect and presents a new hierarchical topology for the Switch Box (SB) which unifies a downward and an upward unidirectional networks. Results show that compared to 2D cluster-based Mesh FPGA, 3D cluster-based FPGA gains in terms of frequency, energy and area by an average of 23%, 37% and 47% respectively [5].

Mapping applications into FPGAs requires a set of accompanying suite of Computer-Aided Design (CAD) tools to perform circuit implementation stages which are: synthesis, technology mapping, clustering, placement and routing. One of critical issue of a FPGA-based implementation is the quality and efficiency of associated CAD tools. In fact, benefits of a well designed and rich FPGA architecture might be impaired if CAD tools cannot take advantage of the architectural advancement provided by the FPGA. Moreover, efficient CAD algorithms are essential to reduce the performance gaps between FPGAs and other computational devices like ASICs. Recent works investigated improvements of CAD algorithms used in each stage of logic circuit mapping into FPGA. For example, improving the logic synthesis algorithm step can

978-1-5386-3250-5/17 $31.00 © 2017 IEEE
DOI 10.1109/HPCS.2017.91

decrease the amount and depth of needed logic and power consumption [6]. In [7], authors propose to revisit partitioning, placement and clustering steps. They show that proposed algorithms can reduce the use of routing interconnect and improve the performance by shortening connections. They also show that improving the routing step can reduce switches number in the critical path and achieve better performance. Other works like for example [8] and [9] propose power-aware algorithms to reduce power consumption in different steps of FPGA CAD flow.

In this work, we are interested to investigate placement algorithms aspect to optimize proposed 3D FPGA performances. In fact, clusters placement distribution between 3D FPGA layers has an important impact on performances. In this context, partitioning-based placement algorithms have been investigated in the case of 3D FPGA to improve performances [10, 11, 12]. Most of them mainly focus on a min-cut approach that minimizes total interconnect length and the maximum cut between the two adjacent layers (called cutsize). In this paper, we aim to explore and investigate the effect of using partitioning-based placement with proposed two-tier 3D FPGA architecture in order to optimize clusters placement between 3D FPGA layers. The rest of the paper is organized as follow: Section III gives a brief overview of target 3D FPGA architecture. Section IV presents the experimentation platform and used metrics models to estimate area, power consumption and delay. Section V details the partitioning-based placement approach and basic simulated annealing based placement approach. Exploration and comparison results between partitioning-based and simulated annealing based placement approaches are discussed in section VI.

III. MODELING OF 3D FPGA ARCHITECTURE

This section gives a brief introduction of target cluster-based 3D FPGA architecture. Further details about proposed 3D FPGA architecture can be found in [5]. Target 3D cluster-based FPGA is constructed by stacking identical 2D functional layers connected with a flexible number of vertical connections (see Figure 1). Each layer is an improved 2D mesh of clusters architecture that achieves a saving of 32% of area and 30% of power consumption compared to reference VPR cluster-based mesh FPGA [13]. The main characteristics of this architecture are summarized in the following points:

- Intra-cluster interconnect: We optimize the intra-cluster interconnect topology by depopulating the intra-cluster full crossbar, using the approach of the tree architecture [14]. In fact, the interconnect depopulation allows increasing silicon utilization through efficient use of the interconnect structure. As illustrated in Figure 2, each cluster contains Local Blocs (LBs) connected with a depopulated intra-cluster interconnect. The depopulated intra-cluster interconnect is divided into Mini Switch Blocks (MSBs) and it is composed of a downward network and an upward network. The downward network is based on the Butterfly-Fat-Tree (BFT) topology which connects cluster inputs to LBs inputs using Downward MSBs (DMSBs). The upward network connects LB outputs to cluster outputs using Upward MSB (UMSB).
- Inter-cluster interconnect: We remove Connection Block (CB) level which is used in common mesh architectures and we propose a new hierarchical Switch Block (SB) interconnect to assure inter-cluster routing interconnect (see Figure 3). Many studies showed that using hierarchical routing interconnect leads to better density [13]. Proposed hierarchical SB interconnect unifies a downward and an upward networks based on the BFT topology.
- Single driver interconnect: We use only single-driver interconnect based on unidirectional wires. In fact, it was shown in [15] that single driver interconnect has a good impact on power consumption, area and performance improvement. These enhancements come from the elimination of switches (buffers and tri-states) used to turn interconnect wires for a specific configuration.

To connect the 2 FPGA layers, each SB have a number of vertical connection (TSVs) is reach its adjacent vertical SB. The number of TSV is flexible and can be adjusted with associated CAD tools. The number of TSVs included in routing channel is equal to $a \times W/2$ where a is percentage of TSVs and W is the channel width. Theses architecture parameters are flexible and can be adjusted based on the design requirements.

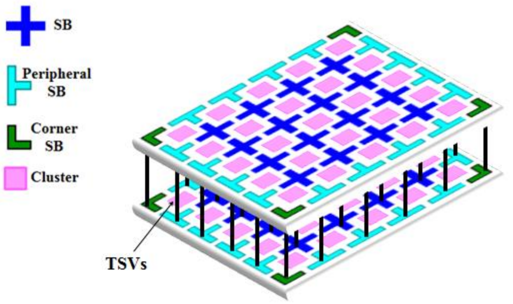

Fig. 1. 3D cluster-based FPGA architecture with 2 layers.

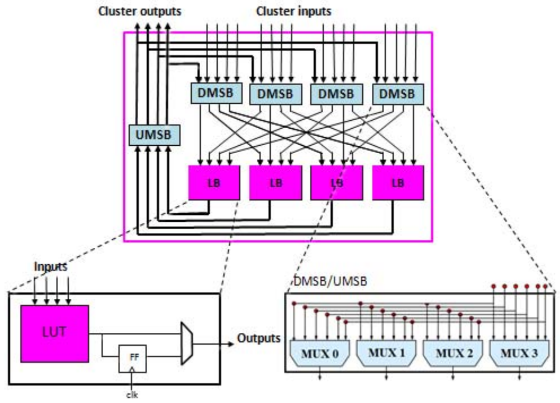

Fig. 2. Cluster architecture.

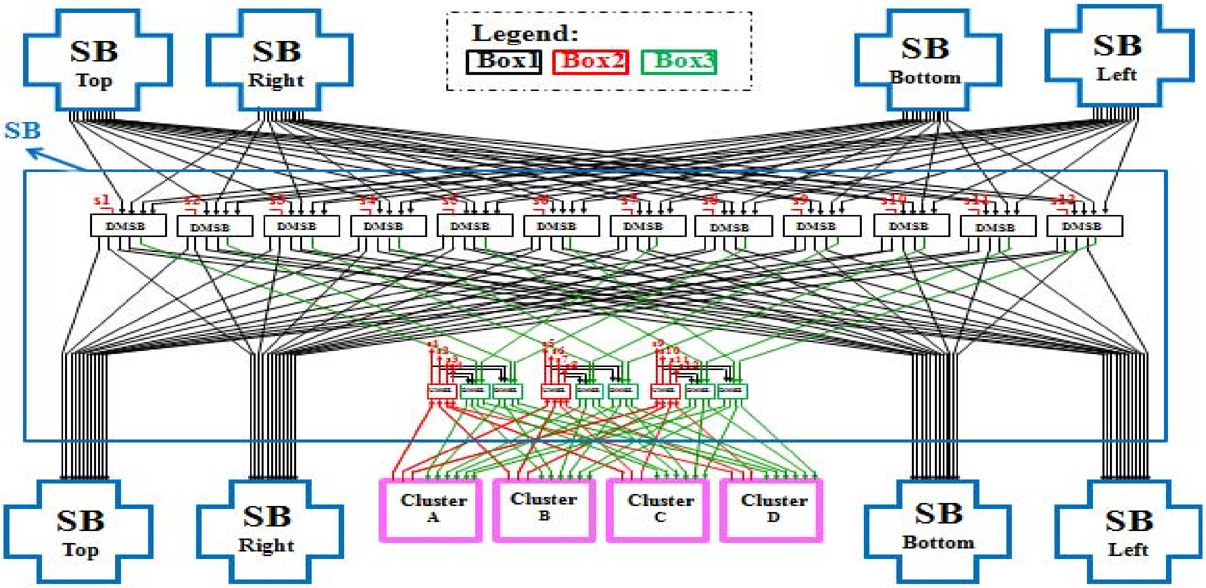

Fig. 3. Hierarchical SB interconnect.

IV. EXPLORATION PLATFORM AND METRIC MODELS

To investigate the quality of new FPGA architecture, one needs exploration platform capable of implementing benchmark circuits automatically in the FPGA of interest and evaluate performances. The exploration platform includes a set of CAD tools and metrics models. CAD tools are used to convert the high-level circuit described in VHDL or Verilog into configuration bitstream which can be loaded into FPGA. Converting a circuit description into configuration bitstream can be divided into 5 stages: synthesis, technology mapping, clustering, placement and routing (as illustrated in Figure 4). Once a circuit has been implemented in an FPGA architecture, one needs next accurate power, area and delay models to evaluate the quality of the circuit implementation in the FPGA architecture under test. The first phase of proposed CAD flow is the synthesis phase which consists in translating a circuit description (e.g. VHDL or Verilog) into a gate-level representation. The gate-level representation is network of Boolean logic gates and FFs. In proposed flow, we use ODINII synthesis tool [17]. For technology mapping stage, we use ABC tool [18] to transform the Boolean network to a network of k-input LUTs and FFs. Then, the circuit passes though T-VPack tool [7] to achieve the packing phase which consists in grouping N LBs together to form CLBs according to cluster inputs and cluster size constraints. After packing phase, placement tool is used to place the CLBs and IOs instances of the circuit on the CLBs and IOs blocks of FPGA. In our methodology, we propose to explore two approaches to accomplish placement phase which are simulated annealing-based placement [4] and partitioning-based placement (see Section V). Once placement is done, routing tools are used to interconnect resources. The routing algorithm we use is the PathFinder [20] which employs an iterative, negotiation-based approach to successfully route all nets in a circuit. The router uses Dijikstra algorithm to route every net with the shortest path. At the end of an iteration, resources can be congested because they are used by multiple nets. During subsequent iterations, the cost of overused routing resources is gradually increased. To avoid congestion, nets are forced to negotiate for shared routing resources based on the lowest congestion alternatives. To determine the minimum number of the channel width *Wmin* that successfully route the circuit, we continuously decrease the tracks number per channel and route the circuit until it fails to route. Once the *Wmin* is determined and routing phase is achieved, we estimate resulting power/energy consumption, area and delay with developed models.

A. Power Model

To better understand and explore the power behavior of the proposed 3D FPGA architecture, we need a flexible and detailed power model capable of estimating power consumption of target architecture. The proposed power model incorporates three components: Activity estimation, Architecture generator and Low-Level Power Estimation. The first module is the activity estimation (ACE2) [21] which employs a transition density model to determine the switching density of all nets. The Architecture generator decomposes the entire FGPA into low-level components which are inverters, multiplexers and wires. The Low-Level power estimation component is used after the decomposition of the FPGA circuit into inverters, multiplexers and wires to estimate power consumption at transistor level. Total power dissipation is the sum of dynamic and static powers. The dynamic and static powers of each component are calculated as defined by equations used in VersaPower model [22].

B. Area Model

This section describes the area model used to compute the density of 3D FPGA architecture under investigation. Discussions with FPGA vendors have revealed that routing area is transistor-dominant and not wiring-dominant [7]. According to [23], we have to care about the number of switches since the switches area is larger than wires area. For example, if the wire pitch is 5 to 8λ (λ is equal to the half of the minimum distance between source and drain of transistor), the area of a wire crossing is 25 to 64λ^2 whereas the switch transistor area is 2500 λ^2 and the area of SRAM used to configure a switch is roughly 1200 λ^2. Therefore for a simple pass gate, the switches area is 40 larger times than wires area. This difference may be larger if we use more than a pass gate for the switch and then we need to take much care about switch count in the interconnect.

The area model proposed in [14] is based on symbolic standard cells library to estimate the FPGA required area. However, it does not use accurate model to determine different buffers sizes across the FPGA. In our work, we implement a new area estimating model based on transistor-counting algorithm consistent with the methodology used in [22] to compute components area. We use routing graph resource to parse all FPGA components and compute accurately the total number of transistors of all entities within the FPGA using same assumptions of low-level decomposition detailed in [22].

C. Delay Model

Timing analysis allows evaluating performance of the final placed and routed circuit implemented on the FPGA. The performance is estimated by calculating the delay of circuit elements along its slowest path (critical path). The delay through the routing network may be easily dominant in a programmable technology. Care is required to minimize interconnect delays. The 2 following factors are significant in this respect:

- Wires delay: Delay on a wire is proportional to distance and capacitive loading (fanout). This makes interconnect delay roughly proportional to distance run. Consequently, short signals runs are faster than long signals runs.
- Switches delay: Each programmable switches in a path (crossbar, multiplexer) adds delay. This delay is generally much larger than the propagation or fanout delay. Consequently, one generally wants to minimize the number of switch elements in a path, even if this means using some longer signals runs.

Wire length and switches delays depend respectively on physical layout and cells library characteristics. The delay of these elements can be pre-characterized since they are independent of placement and routing phases. Figure 5 shows used methodology for timing characterization of wires and switches within target FPGA architectures. The process begins with the RTL description of target FPGA generated using HDL generator. The HDL generator is designed to generate VHDL code based on architecture description file. Then, we use cadence design compiler to compile VHDL into Verilog. The compiled Verilog is used as input into Cadence Encouter to perform physical design layout generation and to extract layout parameters (wires length, capacitances ...). The physical design experiments are performed using the layout generated using ST Micro's 130nm technology node. Finally, the ELDO circuit simulator is used to obtain highly accurate wires and switches delays estimation based on extracted layout parameters. Proposed model considers also the effect of the resulting LUT delay as a function of the LUT size to be consistent with experimentation done in [24].

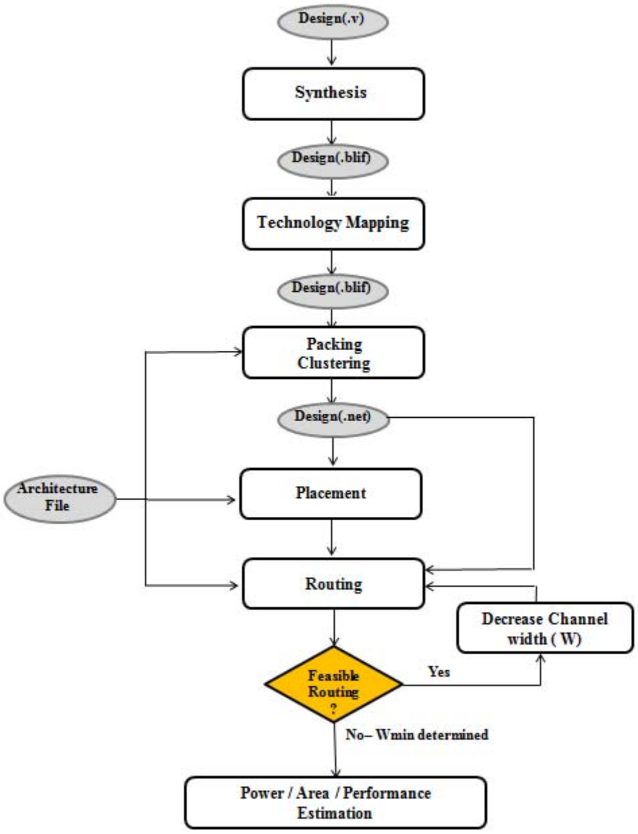

Fig. 4. FPGA Configuration Flow.

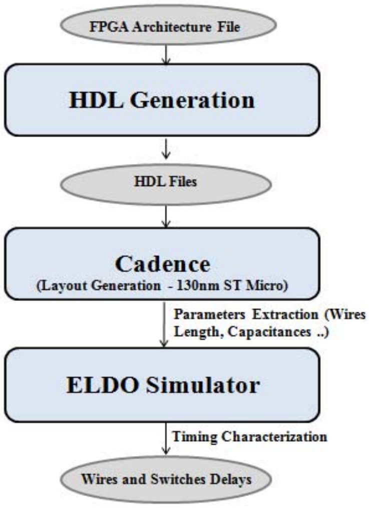
FPGA Architecture File
HDL Generation
HDL Files
Cadence
(Layout Generation - 130nm ST Micro)
Parameters Extraction (Wires Length, Capacitances ..)
ELDO Simulator
Timing Characterization
Wires and Switches Delays

Fig. 5. Timing characterization methodology.

V. PARTITIONING-BASED PLACEMENT VS. SIMULATED ANNEALING-BASED PLACEMENT

As mentioned in Section II, in this paper we are interested to investigate CAD algorithms aspect to optimize proposed 3D FPGA performances. We mainly focus on placement phase which is the fourth stage of the FPGA CAD flow (see Figure 4). Placement consists in determining the physical locations of clusters (instance) required by the netlist in the FPGA clusters (see Figure 6). Placer optimization goals may change from one FPGA architecture to another. The placement optimizations consist in placing connected blocks close together to minimize the required wiring (wire length-driven placement) and sometimes to place LBs to balance the wiring density across the FPGA (routability driven placement) or to optimize circuit performance (timing-driven placement). There are 3 major classes of placement approaches which are min-cut (Partitioning-based) [25, 26], analytic [27, 28] and simulated annealing based placers [19]. Partitioning and simulated annealing approaches are the most common and used in FPGA CAD tools. Simulated Annealing-based algorithm are the common and basic placement algorithm used in many commercial and academic FPGA CAD tools [29]. Partitioning-based placement algorithms have been investigated in the case of 3D FPGA to improve performances [10, 11, 12]. Most of them mainly focus on a min-cut approach that minimizes total interconnect length and the maximum cut between the two adjacent layers (called cutsize). In this paper, we aim to explore and investigate the effect of using partitioning-based placement with proposed two-tier 3D FPGA architecture. In this section, we present used partitioning-based placement and basic simulated annealing based placement approaches.

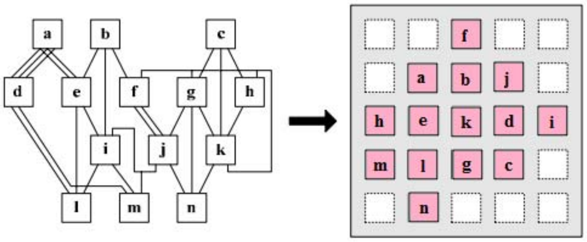

Fig. 6. Example of placement.

A. Simulated annealing-based placement: SA

Simulated annealing mimics the annealing process used to cool gradually molten metal to produce high-quality metal objects [19]. The method of simulated annealing is a wirelength-driven placement. It tends to optimize the total wirelength used to route the circuit. Placement algorithm starts with a random initial assignment of circuit's LBs in available locations in the FPGA. To improve gradually the placement, a number of LB swaps are made. A pair of LBs is randomly selected and iteratively swapped. A cost function is used to evaluate if the swap should be kept or not. If the cost function decreases, the swap is always kept. But, if the cost increases, the swap may or may not be kept. In fact, there is still a chance to accept the swap even if it makes the placement worse. This probability of acceptance is given by $e^{-\frac{\Delta C}{T}}$, where ΔC denotes the change in cost function and T represents temperature parameter that controls probability of accepting moves that worsen the placement. Initially, T is high enough and then almost swaps are accepted. Then, as the algorithm executes, the placement improves and the T decreases gradually, as follows the rate of acceptance decreases and the probability of accepting a seemingly-bad swap becomes very low. This ability to accept hill-climbing moves that make a placement worse allows simulated annealing to escape local minima of the cost function.

The simulated annealing objective cost function, is function of the total wirelength of the current placement. The total wirelength is an estimate of the routing resources required to successfully route all nets of the design. Reducing wirelength signifies that fewer routing resources are needed to route nets. Wirelength reduction is an important point since routing resources in a FPGA are limited and their overuse impedes the performance of the design. The total wirelength can be estimated as the sum of the bounding box dimensions of all nets by using Equation 1:

$$WireCost = \sum_{i=1}^{N} q(i) \times (bb_x(i) + bb_y(i)) \qquad (1)$$

where *N* is the total number of nets in the netlist, *bbx(i)* and *bby(i)* are respectively the width and length of the bounding box of the net *i* and *q(i)* is a correction factor. Figure 7 presents the calculation of *bbx(i)* and *bbx(i)* for a net that has 6 terminals.

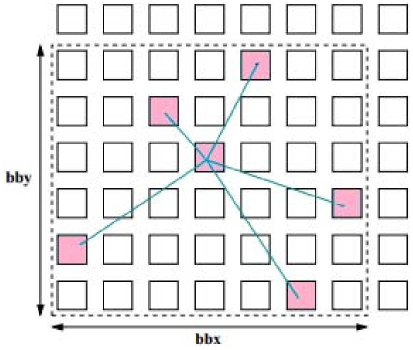

Fig. 7. Bounding Box of a hypothetical 6-terminals net [7].

B. Partitioning-based placement: SA-Parti

There are several considerations that must be taken into account while designing 3D FPGA architectures. Designers must find a trade-off between TSVs high fabrication cost, routability, area, power and performance. The way we distribute clusters between 3D FPGA layers has an important impact on routing congestion and TSVs utilization. In fact, the number of different paths to connect a source to a destination depends on their enclosing clusters positions. If source and destination clusters are not placed in the same FPGA layer, the source have to use TSVs connections to reach its destination (as illustrated in Figure 8). Reducing the maximum cut between the two adjacent layers (called cutsize) using min-cut partitioning algorithm can play an important role to optimize the use of interlayer interconnect.

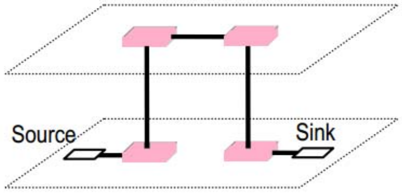

Fig. 8. Illustration of bad placement of clusters into 3D FPGA layers.

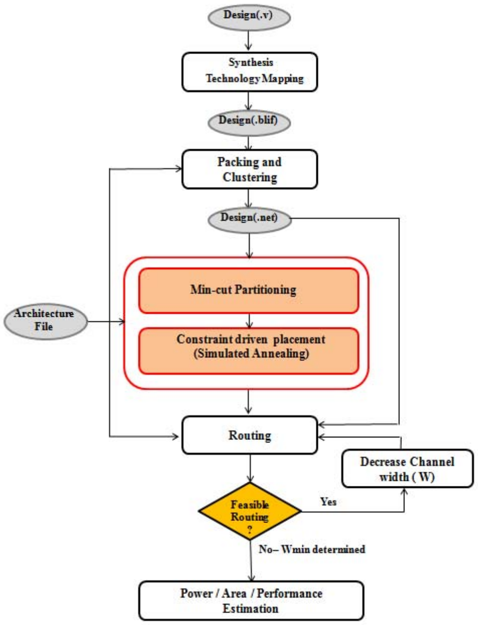

Fig. 9. New configuration flow including partitioning-based placement tool.

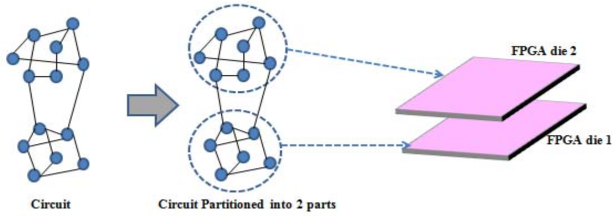

Fig. 10. Partitioning of the circuit into 2 parts.

The partitioning-based placement approach first employs a partitioning phase followed by a simulated-annealing placement. The new FPGA configuration flow including the partitioning-based placement is illustrated in Figure 9. Compared to the configuration flow presented in Figure 4, we add a new step that performs partitioning of the circuit. As illustrated in Figure 10, a circuit partitioning algorithm is applied to divide the circuit into 2 balanced partitions (equal to the number of tiers of proposed 3D FPGA). The partitioning step is performed using the min-cut hMetis partitioning algorithm [30]. The goal of this algorithm is to minimize the maximum cut between the two adjacent layers. The partitioning algorithm determines the adequate part where a given logic block must be placed to minimize the number of cuts in the nets that connect the blocks between partitions, while leaving highly-connected blocks in one partition. By partitioning the problem into 2 sub-parts, a drastic reduction in search space can be achieved in the placement phase. In fact, the placement engine is allowed to move blocks only inside the FPGA layer decided by the partitioning phase (see Figure 9). We adapted simulated annealing algorithm to perform the constrained-placement. It starts to place clusters with a random initial assignment in the regions which they should belong to (see Figure 10). To improve gradually the placement, a pair of clusters, from the same region, is randomly selected and iteratively swapped. The same cost function, described in Equation 1, is used to evaluate if the swap should be kept or not.

VI. EXPERIMENTATION AND COMPARISON RESULTS

The aim of this section is to compare partitioning-based (Parti-SA) and simulated annealing (SA) based approaches with proposed 3D FPGA architecture. Our comparison metrics are power consumption, area and performance. Results correspond to the total power consumption, area and delay using the smallest FPGA array and W_{min} implementing some of the MCNC benchmark circuits with various logic sizes. For both approaches, we consider two-tier 3D FPGA architecture with cluster size 8, LUT size 4, percentage of TSVs per channel 40% according to [5].

Table I shows comparison results in term of maximal cutsize using SA and Parti-SA placement algorithms. Figures 11 and 12 illustrate respectively the total number of TSVs and TSVs utilization for both of SA and Parti-SA. Results show that clusters placement distribution between two-tier 3D FPGA layers has an impact on routing congestion and TSVs utilization. In fact, Parti-SA reduces the total number of TSVs in the FPGA by an average of 20% compared to SA algorithm (see Figure 11). Parti-SA tries to place highly connected clusters in the same FPGA layer which reduce the cutsize by an average of 12% (see Table I) and hence it optimizes interlayer interconnects and reduces required TSVs compared to SA placement algorithm. The reduction of the total TSVs leads to a better utilization of TSVs. As illustrated in Figure 12, the percentage of TSVs utilization increases for all benchmarks by an average of 12%.

Figures 13 to 14 illustrate respectively the total power consumption, area and critical path delay for both of SA and Parti-SA. We note that Parti-SA offers a saving of 15% in term of power consumption compared to SA. To understand this power variation, we need to compare the total number of buffers of 3D FPGA for the 2 placement approaches since the power consumption depends especially on buffers [13]. As detailed, using Parti-SA reduce the total number of TSVs. In proposed architecture TSVs number is given by Equation 2 where *Wmin* is the minimal channel width, *a* is the percentage of TSVs and *N* is the FPGA size. *N* and *a* parameters are equals for both placement approaches. Therefore, the *Wmin* is the architecture parameter that controls the number of TSVs in the 3D FPGA. The *Wmin* is reduced with Parti-SA compared to

SA and hence the total number of buffers used to drive global routing wires in the channel width decreases.

$$Nb\_TSVs = (\frac{W_{\min}}{2}) \times a \times (N+1)^2 \qquad (2)$$

The second key metric is area which depends especially on the number of multiplexers. We note that Parti-SA offers a saving of 18% in term of area compared to SA (see Figure 14). As previously detailed, using Parti-SA reduce *Wmin* which affects the number of inputs and outputs of DMSBs located at Box 1 in each SB. Since DMSBs of Box 1 are full crossbar, the number of 2-to-1 multiplexers in Box1 can be analytically modeled with Equation 2 and then it has linear dependency on *Wmin* and quadratic dependency on *a*. *a* architecture parameter is the same for both placement approaches. However, Parti-SA allows to route circuits with lower *Wmin* is compared to SA and hence the number and size of 2-to-1 multiplexers used to implement DMSBs of Box 1 is reduced. From multiplexers reduction, we get the area improvement.

$$Nb\_DMSBs\_multiplexers(Box\_1) = (\frac{W_{\min}}{2}) \times (5 + 4 \times a) \qquad (3)$$

We note also an improvement in critical path delay. As shown in Figure 15, compared to SA, using Parti-SA ameliorates the performance of the 3D FPGA by an average of 10%. In fact, Parti-SA allows connecting clusters which are not neighbors with lower switches. Thereby, it can knock out a lot of switches from the critical path and reduce the overall delay. From simulation experiments, we can conclude that the circuit cutsize reduction is an important factor for routability and performances improvement.

TABLE I. TOTAL TSVS: SA VS. PARTI-SA.

Bench	Cut Number	
	SA	Parti-SA
clma	1022	857
elliptic	483	465
ex1010	527	409
ex5p	318	302
frisk	573	494
pdc	815	720
s38417	396	340
S38584	549	487

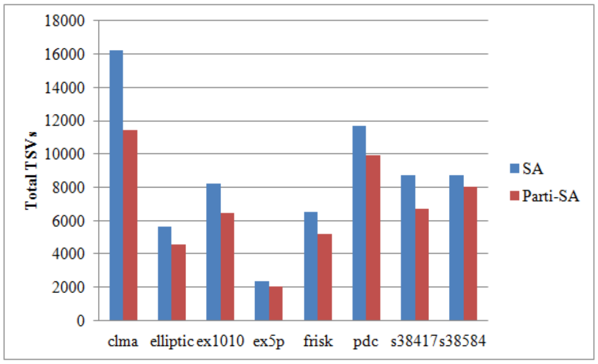

Fig. 11. Total TSVs: SA vs. Parti-SA.

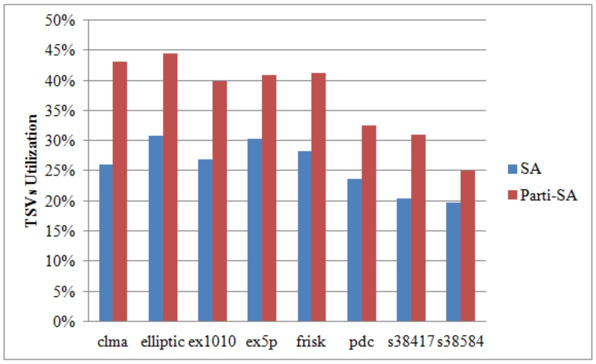

Fig. 12. TSVs Utilization: SA vs. Parti-SA

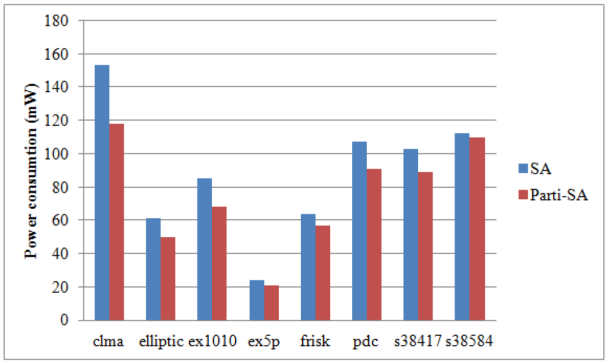

Fig. 13. Power consumption: SA vs. Parti-SA

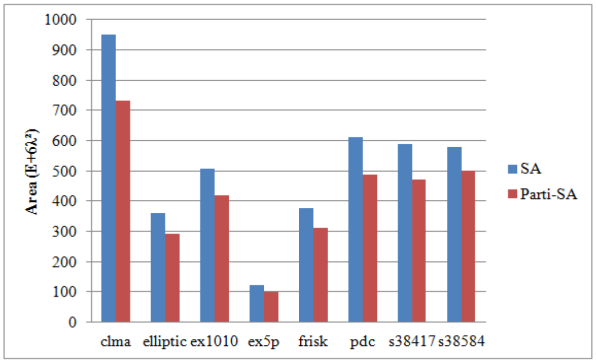

Fig. 14. Area: SA vs. Parti-SA.

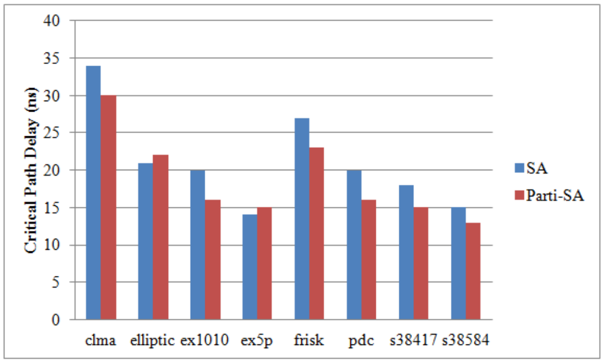

Fig. 15. Critical path delay: SA vs. Parti-SA.

VII. CONCLUSION

In this paper, we proposed to investigate CAD algorithms aspect to optimize proposed 3D FPGA performances. Benefits that partitioning-based placement approach can offer for proposed 3D FPGA were analyzed and compared to basic simulated annealing based placement approach. Results showed a decrease of 15% for power consumption, 18% for area and 10% for delay using the partitioning-based algorithm compared to basic simulated annealing based placement algorithm. There is still room to investigate and optimize the mapping of application on the 3D FPGA architecture to take more advantage of the 3D integration. As perspectives, we propose to explore power-aware algorithms in order to reduce power consumption at different steps of FPGA CAD flow.

REFERENCES

[1] I. Kuon and J. Rose, "Measuring the Gap Between FPGAs and ASICs", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2007, vol. 26, no. 2, pp 203-215.

[2] P. E. Gaillardon, X. Tang, G. Kim and G. De Micheli, "A Novel FPGA Architecture Based on Ultrafine Grain Reconfigurable Logic Cells", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2015, vol. 23, no.10, pp 2187-2197.

[3] M. Lin, A. Gamal, Y. Lu and S.Wong, "Performance Benefits of Monolithically Stacked 3D FPGA", International symposium on Field Programmable Array FPGA (FPGA), 22-24 February 2006, Monterey, California, pp 113-122.

[4] S. Chtourou, M. Abid, V. Pangracious, E. Amouri, Z. Marrakchi, H. Mehrez, "Three-dimensional Mesh of Clusters: An Alternative Unified High Performance Interconnect Architecture for 3D-FPGA Implementation", IEEE 3D System Integration Conference (3D IC), 1-3 December 2014, Kinsale, Ireland, pp 1-7.

[5] S. Chtourou, Z. Marrakchi, E. Amouri, V. Pangracious, H. Mehrez, M. Abid, "Exploration of Mesh-based FPGA Architecture: Comparison of 2D and 3D Technologies in terms of Power, Area and Performance", Conference on Parallel, Distributed and Network Based Processing (PDP), 17-19 February 2016, Heraklion, Greece, pp 104-111.

[6] D. Chen, J. Cong and Y. Fan, "Low-Power High-Level Synthesis for FPGA Architectures", International Symposium on Low Power Electronics and Design, 25-27 August 2003, Seoul, South Korea, pp 134–139.

[7] V. Betz, J. Rose and A. Marquart, "Architecture and CAD for Deep Sub-micron FPGAs", The Springer International Series in Engineering and Computer Science, 1999.

[8] J. Lamoureux and Steven J. E. Wilton, "On the Interaction between Power-Aware Computer-Aided Design Algorithms for Field-Programmable Gate Arrays", Journal of Low Power Electronics, 2005, vol.1, no. 2, pp 119–132.

[9] K. Siozios and D. Soudris, "A Power-Aware Placement and Routing Algorithm Targeting 3D FPGAs", 2008, vol. 4, no. 3, pp 275-289.

[10] C. Ababei, Y. Feng, B. Goplen, M. Hushrav, Z. Tianpei, K. Bazargan and S. Sapatnekar, "Placement and Routing in 3D Integrated Circuits", IEEE Design and Test, 2005, vol. 22, no. 6, pp. 520-531.

[11] 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", International Symposium on Physical Design, April 2004, Phoenix, Ariz, USA, pp. 108–115.

[12] K. Siozios, A. Bartzas and D. Soudris, "Architecture Level Exploration of Alternative schmes Targeting 3D FPGAs: A Software Supported Methodology", International Journal of Reconfigurable Computing, 2011, vol. 2011.

[13] S. Chtourou et al., "Improvement of cluster-based Mesh FPGA architecture using novel hierarchical interconnect topology and long routing wires", Microprocessors and Microsystems Journal, 2016, vol. 40, pp 16 - 26.

[14] Z. Marrakchi, H. Mrabet, U. Farooq, H. Mehrez, "FPGA Interconnect Topologies Exploration", International Journal of Reconfigurable Computing, 2009, vol. 2009.

[15] S. Chtourou, Z. Marrakchi, E. Amouri, H. Mehrez, M. Abid, "The Effect of Interconnect Depopulation on FPGA Performances in Terms of Power, Area and Delay", International Conference on High Performance Computing & Simulation (HPCS), 18-22 July 2016, Innsbruck, Austria, pp 104-111.

[16] G. Lemieux and D. Lewis, "Directional and Single-Driver Wires in FPGA Interconnect" , Field Programmable Technology (FPT), 6-8 December 2004, Brisbane, Australia, pp 41 - 48.

[17] P. Jamieson, K. Kent, F. Gharibian and L. Shannon, "Odin II - An Open-Source Verilog HDL Synthesis Tool for CAD Research", in FCCM, 2-4 May 2010, Charlotte, NC, pp 149-156.

[18] A. Mishchenko, "ABC: A System for Sequential Synthesis and verification", 2012, Available at: http://www.eecs.berkeley.edu/alanmi/abc/abc.htm.

[19] S.Kirkpatrick, C.D.Gelatt, and M.P.Vecchi, "Optimization by Simulated Annealing", Science Magazine , 1983, vol. 220, no. 4598, pp. 671-680.

[20] L. McMurchie and C. Ebeling, "Pathfinder: A NegotiationBased Performance-Driven Router for FPGAs", International symposium on Field Programmable Array FPGA (FPGA), 12-14 February 1995, Monterey, California, pp 111-117.

[21] J. Lamoureux and S. Wilton, "Activity Estimation for Field Programmable Gate Arrays", Field Programmable Technology (FPT), 13-15 December 2006, Bangkok, Thailand, pp 1-8.

[22] J. Goeders and S. Wilton, "Power Aware Architecture Exploration for Field Programmable Gate Arrays", Journal of Low Power Electronics, 2014, vol. 10, no.3, pp 297-312.

[23] A. DeHon, "Rent's Rule Based Switching Requirements", International workshop on System-level interconnect prediction (SLIP), 31 March - 01 April 2001, pp 197-204.

[24] E. Ahmed, J. Rose, "The effect of LUT and cluster size on deep submicron FPGA performance and density", Very Large Scale Integration (VLSI) Systems, 2004, vol. 12, no. 3, pp 288-298.

[25] A. Dunlop and B.Kernighan, "A Procedure for Placement of Standard-cell VLSI Circuits", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1985, vol. 4, no. 1, pp 92–98.

[26] D. Huang and A. Kahng, "Partitioning-based Standard-cell Global Placement with an Exact Objective", ACM Symposium on Physical Design, 14-16 April 1997, New York, NY, pp 18–25.

[27] G. Sigl, K. Doll, and F. Johannes, "Analytical Placement: A Linear or a Quadratic Objective Function", Design Automation Conference, 17-21 June 1991, San-Francisco, California, pp 427–432.

[28] C. Alpert, L. Hagen and A. Kahng, "Multilevel Circuit Partitioning", Design Automation Conference, 9-13 June 1997, Anaheim, CA, pp. 530–533.

[29] Y. Liu, "Toward More Efficient Annealing-Based Placement for Heterogeneous FPGAs", University of Toronto 2014

[30] G. Karypis and V. Kumar, "Multilevel k-way Hypergraph Partitioning", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 1999, Volume: 11, Issue: 3, pp 285-300.