

Heterogeneous Architecture Exploration: Analysis vs. Parameter Sweep

Asma Kahoul, George A. Constantinides,
Alastair M. Smith, and Peter Y.K. Cheung

Department of Electrical & Electronic Engineering,
Imperial College London, Exhibition Road,
London SW7 2BT, United Kingdom.

{a.kahoul,g.constantinides,a.smith,p.cheung}@imperial.ac.uk

Abstract. *This paper argues the case for the use of analytical models in FPGA architecture layout exploration. We show that the problem when simplified, is amenable to formal optimization techniques such as integer linear programming. However, the simplification process may lead to inaccurate models. To test the overall methodology, we combine the resulting layouts with VPR 5.0. Our results show that the resulting architectures are better than those found using traditional parameter sweeping techniques.*

Key words: Floorplanning; Reconfigurable architectures; FPGA; integer linear programming (ILP)

1 Introduction

The advances in field programmable gate arrays (FPGAs) over the past decade have made it possible to place significantly large circuits on a single FPGA chip [1]. The need for estimation and optimization techniques has therefore become crucial to divide, conquer and explore the large design space of potential architectures. While there exists a significant amount of work on homogeneous architecture design, there is currently limited research on heterogeneous FPGA architectures consisting of a mix of coarse and fine grain components [2].

A commonly employed approach used to explore reconfigurable architecture layouts is to use tools such as Versatile Place and Route tool (VPR)[3]. This tool allows architects to have a baseline structure from which different architectures can be generated. These are in turn tested by placing and routing a set of designs using VPR's synthesis heuristic. The architecture that best suits one, or a combination of metrics such as area, delay, and power is thus selected [4]. While an exhaustive exploration of all possible layouts would lead to an optimal architecture, this would require excessive computational time. As a result, it is necessary to sample the design space, losing the scope for optimality.

This paper addresses the drawbacks of such parameter sweep techniques by using analytical modeling. Our framework targets column-based FPGA architectures consisting of different resource types such as CLBs, RAMs, and multipliers

arranged in columns. This framework explores the design space of architecture layouts efficiently by modeling heterogeneous architecture layouts using mathematical programming in the form of Integer Linear Programming (ILP).

An ILP model has been used in our previous work to achieve optimal solutions to problems such as architecture layout generation [2]. The model allows the elimination of the dependencies on heuristic synthesis algorithms and provides more efficient ways to explore the design space. While the results have shown provably optimal bounds on the relative computational speed for various benchmarks, their efficiency is limited by the assumptions made and the accuracy of the model itself. Hence, if the architecture models are poor compared with empirical flows such as VPR, then the results will be meaningless. Moreover, if the optimization process is not scalable in execution time, this approach loses its attractiveness compared to a typical parameter sweep approach. This paper shows that the results of analytical formulations can be fed back to VPR to verify the quality of the designs. We demonstrate that the resulting architectures are an improvement over using VPR and parameter sweep for a fixed time budget.

The remainder of the paper is organized as follows: Sections 2 and 3 discuss related work and our proposed enhancements of previous mathematical formulations. In section 4 and 5 we describe our contribution in testing the efficiency of numerical methods by comparing it to a typical layout parameter sweeping technique.

The main contributions of this paper can be summarized as follows:

1. An enhanced formulation of the heterogeneous FPGA layout problem, leveraging advances in facility layout models from the operational research community.
2. The quantification of analytical modeling efficiency over a parameter sweep approach in the design of new FPGA architecture layouts.

2 Related Work

The development of advanced architectures and more efficient Computer Aided Design (CAD) mapping algorithms have improved the speed, area and power consumption over the past decade [1]. The introduction of hard specific circuits in heterogeneous FPGAs, for example, made it possible to execute their functions more efficiently. A recent study shows that coarse grain components can reduce the area gap between FPGAs and ASICs from 40X to 20X while preserving the gap of speed and static power consumption [5]. The main disadvantage of heterogeneous devices is that these coarse grain components are beneficial only when they are used and are a waste in silicon, area and routing otherwise [6]. Consequently, the exploration of the mix of these coarse grain components and different architecture layouts has become an interesting research subject [4].

The design of modern FPGA architecture layouts in particular is a challenging task. Exploring new FPGA architectures aims at finding an optimal non-overlapped layout of the FPGA chip by positioning circuit blocks while mini-

mizing an objective function comprising selected performance metrics. There are many algorithms available in the literature for the floorplanning of Application-Specific Integrated Circuits (ASICs) [7]. Whereas these algorithms can be applied to homogeneous FPGAs consisting of only configurable logic blocks (CLBs), they must be modified significantly to be used with modern FPGAs.

In the absence of efficient floorplanning algorithms for heterogeneous architectures, methodologies based on parameter sweeping are currently used to design new FPGA architecture layouts. In comparison with such approaches, analytical tools offer the ability to explore a much wider design space within any given computational time budget. The analytical tool of [2], for instance, has been used in the past to produce optimal architectures within the accuracy of the formulation. However, the model suffers from its exponential time complexity with respect to the number of circuit blocks [2]. Enhanced formulations of the problem could result in reduced solution time and therefore be applied to larger circuits. As a result, we have taken advantage of the advances in the facility layout problem [8] in the aim of efficiently formulating the heterogeneous FPGA layout problem analytically.

Furthermore, ILP-based architecture exploration models suffer from their dependence on assumptions and simplifications which causes uncertainty about the efficiency of the results. On the other hand, empirical tools such VPR 5.0 offer much higher accuracy of the FPGA architectures models. In this paper, we propose a framework that combines the efficiency of the analytical tools with the accuracy of VPR 5.0. We present results showing the quality of architectures generated with our enhanced analytical model by feeding them back to VPR.

3 Heterogeneous Architecture Exploration using an Enhanced ILP Formulation

The design of efficient heterogeneous architecture layouts using analytical tools and accurate models such as VPR is the primary aim of this paper. To achieve this, we initially describe our ILP model and investigate different bounding procedures to improve the solution time. A generic floorplanning model is constructed to test the efficiency of our enhanced formulation. Based on this model, further restrictions are added to describe the column-based nature of modern FPGAs. The resulting architectures are used in Section 5 to illustrate the efficiency of analytical techniques in exploring the design space of architecture layouts in comparison with a parameter sweep approach.

3.1 Generic Formulation of the Floorplanning Problem

This section describes the generic linear programming formulation of the layout floorplanning problem and provides the key notations used in this paper. We denote a set of n rectangular circuit blocks as B . The width and length of each block $i \in B$ are represented by w_i, h_i respectively. The floorplanning problem described in this paper is a fixed-die (fixed outline) problem in which the FPGA

chip is modeled as a fixed rectangular shape of width W and height H . The locations of the blocks are determined by their centroid locations (x_i, y_i) in a two dimensional coordinate system aligned with the chip height, width, and its origin located at the south west corner of the chip.

Objective Function: Area minimization has been the main objective in traditional floorplanners [9]. However, due to the significant impact of interconnect on circuit delay caused by the rapidly increasing number of transistors and their switching speed, it has become necessary to design interconnect-based tools. Wire-length models are based on the fact that automatic routing tools use Manhattan geometry *i.e.* only vertical and horizontal wires to connect elements. The model initially optimizes the rectilinear distance between the centroid locations of connected blocks to obtain estimates of the total wire-length. The objective function is tuned with VPR 5.0 routing model in later sections to account for critical path. The wire-length optimization problem can be stated as follows:

<p style="margin: 0;"><i>Minimize</i> Wire-length</p> <p style="margin: 0;"><i>subject to</i> Fit in the chip constraints Non-Overlap constraints</p>

Fit in the Chip Constraints: To ensure that circuit blocks are contained within the die area, the origin (south west corner of the chip), the chip width, and its height are used as lower and upper bounds to the location of the blocks centroids as shown in inequalities (1).

$$\forall i \in B \begin{cases} \frac{1}{2}w_i \leq x_i \leq W - \frac{1}{2}w_i \\ \frac{1}{2}h_i \leq y_i \leq H - \frac{1}{2}h_i \end{cases} \quad (1)$$

Non-Overlap Constraints: In order to constrain the block placements and to prevent them from overlapping we apply a set of separation constraints. These constraints force the blocks to be separated either on the x -axis or the y -axis as shown in Fig. 1.

The non-overlapping constraints in either axes can be described using the following mathematical disjunctions:

$$\begin{aligned} & \left(x_i + \frac{1}{2}w_i \leq x_j - \frac{1}{2}w_j \right) \vee \left(y_i + \frac{1}{2}h_i \leq y_j - \frac{1}{2}h_j \right) \\ & \vee \left(x_j + \frac{1}{2}w_j \leq x_i - \frac{1}{2}w_i \right) \vee \left(y_j + \frac{1}{2}h_j \leq y_i - \frac{1}{2}h_i \right) \end{aligned}$$

These disjunctions ensure separation by setting **at least** one of the inequalities to true. The difficulty in formulating these separation constraints in ILP, is

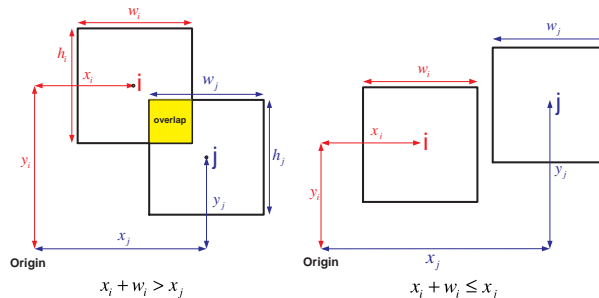


Fig. 1. Illustration of the separation constraints on the x -axis.

the result of introducing binary variables necessary to write the inequalities in a linear form. The most common approach to linearize a set of disjunctions is the so-called **Big-M** formulation illustrated in Equations (2).

$$Mz_{ij}^x + x_j - x_i - \frac{1}{2}w_i - \frac{1}{2}w_j \geq 0 \quad (2a)$$

$$Mz_{ji}^x + x_i - x_j - \frac{1}{2}w_j - \frac{1}{2}w_i \geq 0 \quad (2b)$$

$$Mz_{ij}^y + y_j - y_i - \frac{1}{2}h_i - \frac{1}{2}h_j \geq 0 \quad (2c)$$

$$Mz_{ji}^y + y_i - y_j - \frac{1}{2}h_j - \frac{1}{2}h_i \geq 0 \quad (2d)$$

$$z_{ij}^x + z_{ji}^x + z_{ij}^y + z_{ji}^y \leq 3 \quad (2e)$$

$$z_{ij}^x, z_{ij}^y \in \{1, 0\}, \forall i \neq j \quad (2f)$$

By forcing at least one of the binary variables to be zero using (2e) and (2f) we force the blocks to be separated in at least one direction. The **Big-M** formulation requires four binary variables for any pair of blocks which necessitates in total $4\binom{n}{2}$ variables. This explains the limitation of the **Big-M** formulation with regard to its exponential worst-case time complexity. Dropping the integrality constraints (LP relaxation) and solving the resulting LP problem is usually used to obtain global lower bounds on the optimal value of the problem. These are in turn used within a systematic solution technique such as the branch and bound scheme. However, this relaxation suffer tends to produce trivial bounds making the solution of the ILP problem longer to find. This motivates a tighter formulation of the floorplanning problem.

Consequently, we have taken advantage of the progress achieved in the area of disjunctive programming to obtain tighter lower bounds on the solution of the floorplanning problem which is in turn used to construct the architecture exploration framework. We aim to achieve this by adding a set of valid inequalities which capture the smallest set (convex hull) containing the feasible solutions

of the disjunctions. Such a formulation should also reduce the search space for valid architectures.

Existing convex hull representations of the disjunctive constraints in [8] are constructed for variable aspect ratio problems. Since our formulation targets fixed aspect ratio blocks, we derive the corresponding convex hull representation of the disjunctions using a set of continuous variables $\forall i < j: c_{ij}^x, c_{ij}^y, \Delta_{ij}^x, \Delta_{ij}^y$ as shown in (3).

$$\frac{1}{2}(w_i + w_j)z_{ij}^x \leq c_{ij}^x \leq (W - \frac{1}{2}w_i - \frac{1}{2}w_j)z_{ij}^x, \forall i \neq j \quad (3)$$

$$\begin{aligned} & -(W - \frac{1}{2}w_i - \frac{1}{2}w_j)(1 - z_{ij}^x - z_{ji}^x) \\ \leq \Delta_{ij}^x & \leq (W - \frac{1}{2}w_i - \frac{1}{2}w_j)(1 - z_{ij}^x - z_{ji}^x) \end{aligned} \quad (4)$$

$$\begin{aligned} \Delta_{ij}^x & = x_j - x_i - c_{ij}^x + c_{ji}^x \\ & -(H - \frac{1}{2}h_i - \frac{1}{2}h_j)(1 - z_{ij}^y - z_{ji}^y) \end{aligned} \quad (5)$$

$$\leq \Delta_{ij}^y \leq (H - \frac{1}{2}h_i - \frac{1}{2}h_j)(1 - z_{ij}^y - z_{ji}^y) \quad (6)$$

$$\Delta_{ij}^y = y_j - y_i - c_{ij}^y + c_{ji}^y \quad (7)$$

$$z_{ij}^x + z_{ji}^x + z_{ij}^y + z_{ji}^y = 1 \quad (8)$$

$$z_{ij}^x, z_{ij}^y, z_{ji}^x, z_{ji}^y \in \{1, 0\} \quad (9)$$

Contrary to the **big-M** formulation, the improved model enforces separation in one direction by setting one of the binary variables z_{ij} to one, utilizing the c_{ij} variables which represent the separation distance. These equations also provide better bounds as will be demonstrated by the experimental results.

3.2 Experimental Results

In order to demonstrate the validity of this formulation, we have conducted experiments on a set of MCNC benchmarks: *apte*, *xerox*, and *hp*. For comparison purposes, we use the model of [7], which is the only formulation in the literature that provides global lower bounds on the same problem. This formulation uses a branch of mathematical programming called semi-definite programming (SDP). The big-M relaxation produced trivial bounds for this set of benchmarks and is not included in our comparison.

The commercial ILP solver CPLEX running on a 3.5GHz, 2GB pentium 4 have been used to generate the convex hull relaxation bounds shown in Table 1. The gaps in the table represent the relative gap between the optimal solution and the lower bounds and it is calculated as $\frac{Optimal - Bounds}{Optimal}$.

The results show that the convex hull reformulation has successfully been used to obtain tighter global bounds in shorter time on the optimal floorplanning problem solution. By successfully formulating the floorplanning problem more accurately, the formulation can be used to build a heterogeneous architecture

Table 1. A Comparison of the lower bounds obtained using a convex hull relaxation and an SDP relaxation [7].

Benchmarks	Solution	SDP Relaxation [7]			Convex Hull relaxation		
		Bound	Gap(%)	time(sec)	Bound	Gap(%)	time(sec)
Apte	5205.4	2847.7	45.29	815	3586.03	31.11	0.29
Xerox	6538	1153.1	82.36	2721	5357.24	18.06	0.34
Hp	2101.2	773.23	63.20	7855	1384.37	34.12	0.4

model. This model along side with VPR 5.0 will be used in the following section to explore column-restricted FPGA architecture design.

3.3 Enhanced Heterogeneous FPGA Floorplanning Model Formulation

The floorplanning for column-restricted FPGAs requires the placement of circuit blocks of a particular resource type within the boundaries of the corresponding resource column. Constraints to map these nodes into their respective regions as well as setting the widths and locations of each column are added in this section. The convex hull relaxation based model discussed in the previous section is modified to include column restrictions. This will allow the exploration of different architecture floorplans of the FPGA chip. An example of the simplified FPGA architecture layout used in our formulation is shown in Fig. 2.

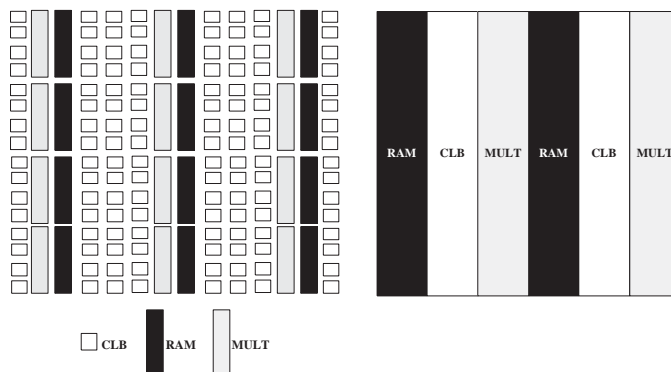


Fig. 2. A Simplified Heterogeneous Architecture Floorplan Consisting of Columns of CLBs, RAMs, and MULTs.

We introduce the following notations to model heterogenous FPGA architectures: In addition to their widths and heights, circuit blocks are constrained by their resource type denoted by $t_i \in T$ where $T = \{CLB, RAM, MULT\}$. We denote the set of resource columns available on the chip as R where each resource column $u \in R$ is a rectangular block of half-width w_u and half-height h_u which

equals half the chip height, centroid locations (x_u, y_u) , and resource type $t_u \in T$. The following summarizes the combined architecture and circuits floorplanning problems:

<p><i>Minimize</i> <i>Wire-length</i></p> <p><i>subject to</i> <i>Circuit Block Fit in the chip constraints</i> <i>Resource Columns Fit in the chip constraints</i> <i>Block Placement inside Resource Columns</i> <i>Block Non-Overlap constraints</i> <i>Resource Columns Non-Overlap constraints</i></p>

From the summary of the formulation we notice that both circuit blocks and resource columns are rectangular blocks placed within the boundaries of the FPGA chip while avoiding overlap. Consequently we use these similarities and propose a new formulation of the column-based floorplanning problem. In this formulation the non-overlap constraints are applied between circuit blocks and also resource columns. A circuit block i for instance, with resource type $t_i = CLB$, is allowed to overlap with any CLB column and is separated from the $MULTs$, $RAMs$ columns, and all other circuit blocks using the convex hull separation constraints.

Having successfully formulated the problem analytically we use this model in section 5 to generate heterogeneous architectures and compare it with a parameter sweeping approach. This latter, is described in the following section.

4 Architecture Exploration using a Parameter Sweeping Approach

In a typical design framework, FPGA architectures are constructed using an experimental methodology. This is conducted by mapping a set of benchmarks into potential architectures and comparing the results using selected performance metrics. We have created a tool that is based on this methodology and which uses layout parameters sweeping to generate a set of architectures. A test benchmark is then placed and routed on these architectures and the architecture that provides the lowest critical path is selected. which are in turn tested with VPR 5.0. The framework takes a test benchmarks and This tool is used to vary the layout of FPGA architectures by sweeping the positions and number of the resource columns within the chip area.

The parameter sweep framework consists of three main blocks and is interfaced with VPR 5.0 for placing and routing as shown in Fig. 3. The target FPGA architectures are initially auto-sized using VPR 5.0 layout options, which find the minimal dimensions of the chip that fits a given circuit. Given this fixed chip area, the sweeping procedure targets the number \mathbf{r} , and position \mathbf{p} of each resource type that could be placed on the architecture. These parameters are varied

using a structured approach in which the chip area is divided into subsets called repeating tiles. Each repeating tile comprises C resource columns as shown in Fig.4. All possible combinations of resources that fit in these repeating tiles are used to construct the set of potential architectures. In other words, instead of exploring all possible architecture layouts given a set of resource columns and a fixed chip area, we fully explore the layout of a smaller portion of the chip and duplicate it along the chip area. This procedure allows the exploration of architectures with significantly different layouts within a fixed time budget, resulting in a structured sampling approach of the design space.

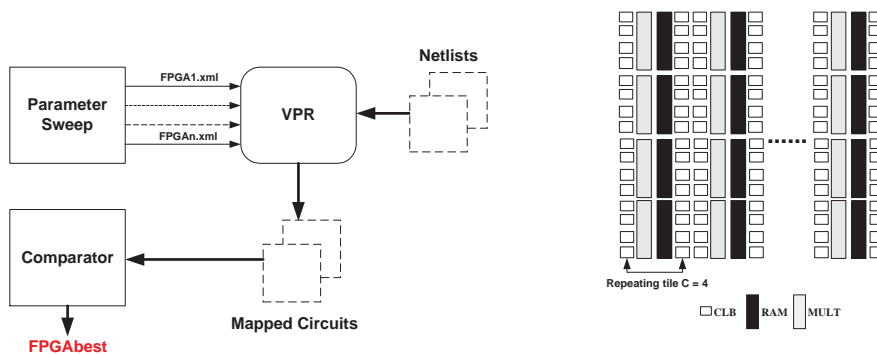


Fig. 3. Parameter-sweep approach.

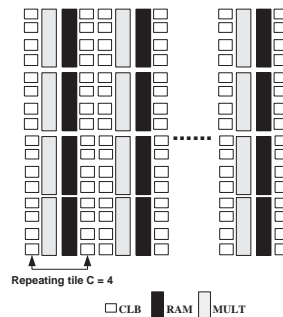


Fig. 4. An architecture constructed with a repeating tile of 4.

The size of the repeating tiles is chosen based on the time frame for the architecture exploration procedure. Increasing the size of the repeating tile results in a larger number of possible permutations and therefore a larger set of explored architectures. For example, given the choice from the set of resources $\{CLB, MULT, RAM\}$ a repeating tile of size $C=5$ results in $(3)^C$ combinations which is 243 potential architectures. The parameter sweep block output (set of combinations) are translated into VPR 5.0 architecture format, which in turn performs the placement and routing of the test circuit on the sample architectures.

The comparator block collects the results of the placement and routing of the test circuit on each architecture. The critical path is used as the comparison metric. Consequently, the architecture resulting in the lowest critical path is selected. The use of critical path as our performance metric provides information about the impact of architecture layout on circuit delays and more importantly inter-node delays. This is particularly important given the significant contribution of interconnect delays in the overall circuit delay.

This framework is used in the following section to compare the efficiency of the previously described ILP model in exploring the design space with a parameter sweep approach.

5 ILP-based analytical approach vs. Parameter-sweep approach

5.1 Experimental setup

The main focus of this paper is to illustrate that combining analytical models with more accurate tools such as VPR, performs better than a typical layout parameter sweep approach. We have therefore conducted a comparative experiment on a set of test benchmarks as shown in Fig. 5. The time budget for the parameter sweep framework is tuned accordingly with the time taken by the ILP model to obtain an optimal architecture. ASIC benchmarks were selected and modified to explore a more comprehensive design space. The resource usage of the benchmark blocks was allocated based on the resource distribution obtained from the technology mapped benchmarks from [2].

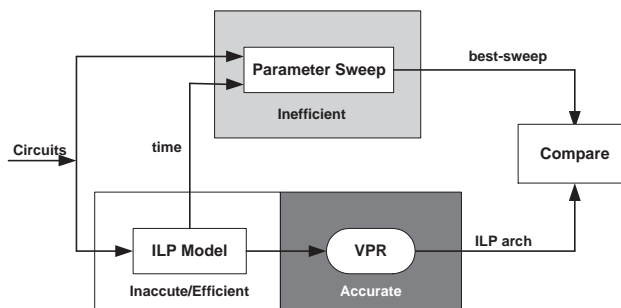


Fig. 5. Comparative experiment between ILP-based analytical model and parameter-sweep approach.

This experimental approach does not only compare the efficiency of both frameworks for the same time budget, but also combines the advantages of analytical techniques and empirical models such as VPR. This is achieved by taking the results generated by the simplified ILP model and feeding it back to VPR for a more accurate architecture model as shown in Fig. 5.

The objective function of the ILP model was tuned accordingly with the routing model of VPR 5.0. This has been achieved using an experimental approach where a best fit model has been applied to the Manhattan Distance between two circuit blocks and the corresponding routing delay.

5.2 Results

The experiment described in Fig. 5 was conducted and the results are described in this section. For the ILP approach optimal solutions were obtained for smaller benchmarks and the model has been left to run for 24 hours for larger benchmarks when only upper bounds were obtained. These solutions were translated

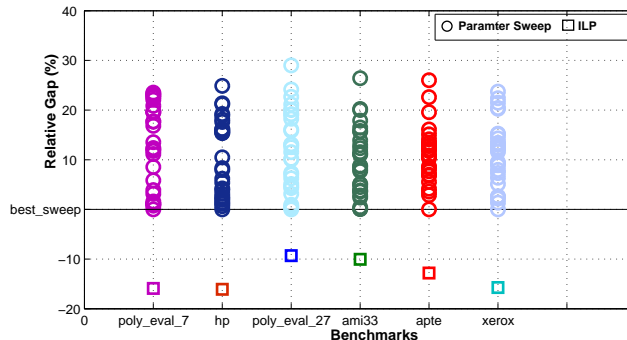


Fig. 6. Relative gap between parameter-swept architectures, ILP generated architecture, and the best parameter-swept architectures.

to VPR 5.0 architecture format and used for the placement and routing of the test benchmarks.

Fig. 6 shows the relative critical path gaps between the best architecture generated with the parameter sweep framework and a subset of other architectures explored with the same framework. These gaps present an important aspect of heterogeneous FPGA design, which is the significant impact of architecture layout on performance. This is clearly illustrated by up to 27% increase in the critical path induced by the layout of the architecture.

On the other hand, the figure shows the gaps between architectures generated by the ILP model relative to the best parameter sweep architecture obtained within the same time frame. These results illustrate the significant improvement of up to 15% on the critical path using our analytical framework over architectures designed with the parameter sweep approach. This is mainly caused by limitations of the parameter sweep approach in exploring a large design space within a restricted time budget. These limitations are induced by the size of the repeating tiles which restricts the potential architecture layouts explored. On the other hand, the results show that by simplifying the problem and applying formal optimization techniques in the form of ILP, better quality architectures are generated. In fact, whereas the ILP framework does not model heterogeneous architectures accurately, it improves on the parameter sweep technique by exploring a wider range of designs.

6 Conclusion and future work

This paper has presented the benefits of using an analytical framework in the design of heterogeneous FPGA architecture layouts over a typical parameter

sweep approach. The framework uses mathematical programming in the form of linear programming to model column-based architectures by targeting layout parameters. An enhanced formulation motivated by the advances in the facility layout problem has proved to successfully bound the design space and consequently reduce the solution time. Using this framework we have been able to simultaneously generate heterogeneous architecture layouts and reduce the critical path.

The efficiency of this framework has been tested using a comparative experiment. For this purpose, a parameter sweep tool has been developed to sample the design space of architecture layouts and test selected architectures on VPR 5.0. The experiments show an average improvement of up to 15% on the critical path induced by our analytical model in comparison with the parameter sweep approach. This shows that despite the assumptions that have been made to model the FPGA architectures in ILP, it still provides better architectures than a parameter sweep approach given the same time frame.

We conclude that FPGA architecture design can benefit from a combined framework which uses the efficiency of analytical tools and the accuracy of tools such VPR. Consequently, we propose as future work the extension of this framework to guide the search for an optimal architecture by feeding back VPR place and route results to the ILP model. This will require a learning mechanism in which information about the architectures explored is used to direct the search for optimal architectures.

References

1. K. Compton and S. Hauck, "Reconfigurable computing: a survey of systems and software," *ACM Comput. Surv.*, vol. 34, no. 2, pp. 171–210, 2002.
2. A. Smith, G. Constantinides, and P. Cheung, "Removed for blind review," in *Proceedings. International Conference on Field Programmable Logic and Applications*, 2005, pp. 341–346.
3. V. Betz and J. Rose, "VPR: A New Packing Placement and Routing Tool for FPGA Research," in *Workshop on Field-Programmable Logic and Applications*, vol. 2, no. 1, 1997, pp. 3–222.
4. M. Hutton, "FPGA Architecture Design Methodology," in *Proceedings. International Conference on Field Programmable Logic and Applications*, Aug. 2006, p. 1.
5. I. Kuon and J. Rose, "Measuring the gap between FPGAs and ASICs," in *Proceedings. 14th International Symposium on Field Programmable Gate Arrays*. ACM New York, NY, USA, 2006, pp. 21–30.
6. J. He and J. Rose, "Advantages of heterogeneous logic block architecture for FPGAs," in *Proceedings. Custom Integrated Circuits Conference*, 1993, pp. 7–4.
7. P. Takouda, M. Anjos, and A. Vannelli, "Global lower bounds for the VLSI macrocell floorplanning problem using semidefinite optimization," in *Proceedings. Fifth International Workshop on System-on-Chip for Real-Time Applications*, 2005, pp. 275–280.
8. H. Sherali, B. Fraticelli, and R. Meller, "Enhanced Model Formulations for Optimal Facility Layout," *Operations Research*, vol. 51, no. 4, p. 629, 2003.
9. Y. Feng and D. Mehta, "Heterogeneous Floorplanning for FPGAs," in *Proceedings. IEEE International Conference on VLSI Design*, 2006, pp. 257–262.