

Analysis of Impact of FPGA Routing Architecture Parameters on Area and Delay

Christopher Schäfer, Mirjana Stojilović, and Lazar Saranovac

Abstract — FPGAs are increasingly replacing more expensive microprocessors and ASICs in a wide variety of applications. Their performance is limited mainly by the characteristics of the routing network. This paper presents a detailed study of the influence of the parameters defining an FPGA routing network on the routing area, the critical path delay, and the channel width required to place and route benchmark circuits. The goal was to find the set of parameter values providing maximal performance of the routing network. The results have shown that the trade-offs are inevitable.

Keywords — FPGA, Routing architecture, VPR

I. INTRODUCTION

FPGAs consist of configurable logic blocks (CLBs) to implement logic functions, programmable routing to interconnect CLB outputs and inputs, and I/O pads to enable off-chip connections. Nowadays, some FPGAs also integrate digital signal processing (DSP) blocks, memory blocks, and even some application-specific blocks. This way, in addition to being a low-cost alternative to microprocessors and ASICs, FPGAs are becoming high-performance alternatives too.

The main performance bottleneck of FPGAs is the routing network. The largest portion of the area of an FPGA is consumed by programmable routing [1] and the critical path delay is mostly due to routing delays, rather than logic block delays. Furthermore, as FPGAs move into increasingly deep-submicron technologies, the fraction of the total delay due to routing is increasing with each process generation [2]. Consequently, FPGAs are unable to fully exploit the potential of deep-submicron technologies, unless their routing networks are made both fast and area-efficient.

In most cases, FPGA vendors implement the same routing network architecture, but with different values of parameters defining a network. In the literature on FPGAs, it is not explained how these values were devised and if any of them provide the best possible performance. Therefore, this paper addresses the impact of each of these

parameters on the overall area consumed by routing and on the critical path delay, with the purpose of proposing the optimal set of parameter values (optimal routing network).

II. FPGA ROUTING ARCHITECTURE

Based on the arrangement and distribution of logic elements and routing resources, i.e., routing architecture, FPGAs can be classified into: symmetrical, hierarchical, row-based and Sea-of-Gates [3]. The majority of FPGAs available at the global market are symmetrical or hierarchical structures.

A. Routing network parameters

A routing channel is a set of wiring tracks between two consecutive rows or two consecutive columns in the CLB array, as shown in Fig. 1. The tracks between the CLB array and I/O pads also compose a routing channel. Channels are defined by the channel width W , equal to the number of routing tracks per channel. Usually, W is constant throughout the CLB array. However, in the case of directionally-biased FPGAs [4], vertical channels have more routing tracks than horizontal channels. Additionally, in the case of non-uniform FPGAs [4], the center region has higher channel width than outer regions. There exist other layouts, depending on where the manufacturer expects the greatest routing congestion.

The segment length L specifies how many CLBs the track is spanning. In commercial FPGAs routing channels most often contain tracks of different segment length. Different configurations of segment lengths per channel affect the routing area and the critical path delay, as will be explored in the experimental part of the paper.

To specify the connections between the tracks and CLBs, three additional parameters are used. The input connection block flexibility $F_{c,in}$ indicates how many tracks per channel are connected to each CLB input. The output connection block flexibility $F_{c,out}$ indicates how many tracks per channel are connected to each CLB output. The switch block flexibility F_s is the number of possible connections a wire segment can make to other wire segments inside a switch block [5].

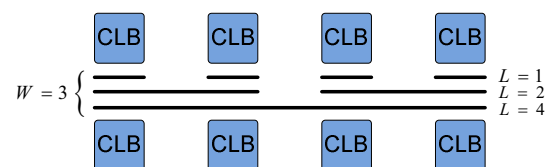


Fig. 1. Channel width and segment length.

This work has been supported in part by the Ministry of Education and Science of the Republic of Serbia (Project TR 32043).

C. Schäfer (e-mail: christopher.schaefer@tum.de) is with the Technical University Munich, Arcisstrasse 21, Munich, Germany. He is currently a trainee at the Institute Mihajlo Pupin, by IEAST exchange programme.

M. Stojilović is with the Institute Mihajlo Pupin, University of Belgrade, Volgina 15, Belgrade, Serbia.

L. Saranovac is with the School of Electrical Engineering, University of Belgrade, P.O. Box 35-54, Belgrade, Serbia.

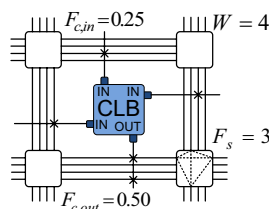


Fig. 2. Input, output, and switch block flexibility.

B. Symmetrical FPGA

A simplified drawing of a symmetrical FPGA is shown in Fig. 3. This FPGA consists of an array of CLBs that are connected through horizontal and vertical routing channels. The number of segments in each channel determines the resources available for routing. The access of the CLB I/O pins to routing channels is established through connection blocks (CBs). These typically consist of pass gates and multiplexers. Programmable switch blocks (SWs) connect the vertical and horizontal routing channels. Usually, there are 2 I/O ports per each row and column [6].

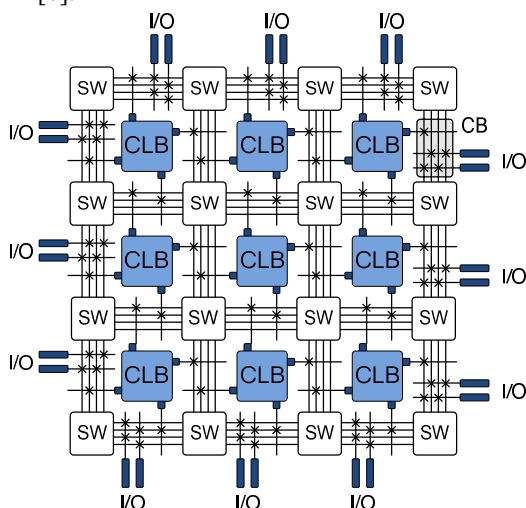


Fig. 3. Symmetrical FPGA Architecture.

III. PREVIOUS STUDIES

In the papers on analyzing effects of routing network parameters, the authors focused on the impact of segment length [2] or the usage of directionally-biased [7][4] and non-uniform [4] routing architectures.

The goal in [2] was to determine the best distribution of segment lengths and the best mixture of different routing switch types. The influence of L on the critical path delay and the routing area were analyzed in the first step. Afterwards a second segment length was added and the measurements were repeated to find the usage of switches in the connection blocks. Finally, they compared the results with the performance of a Xilinx 4000X-like architecture.

The work in [7] was focused on the Altera Stratix routing architecture. Special attention was paid to its directionally-biased architecture by looking for the optimal ratio of the vertical channel width to the horizontal channel width, because this significantly impacts the manufacturing costs.

In paper [4], the effect of routing track distribution on the area-efficiency was analyzed for both directionally-

biased and non-uniform routing. The main questions were if there was a density advantage of directionally-biased architectures over the one with channels having the same capacity, and whether it was favorable to have capacities varying from channel to channel.

Our studies are in part similar to [2]. We focused on the impact of the segment length when it varies from 1 to 8. Furthermore, we added a second segment length and analyzed whether the critical path delay improve. Unlike the mentioned papers, we also investigated the influence of the connection block flexibilities $F_{c,in}$ and $F_{c,out}$ on the routing network performance.

IV. SOFTWARE ENVIRONMENT

This chapter introduces the CAD environment, in particular the tool VPR (FPGA placement and routing tool designed to enable FPGA architecture exploration) [8]. VPR is the core of our experimental setup. After introducing the overall CAD flow, we provide detailed description of VPR and the tools provided by us.

A. CAD flow

The VPR tool suite consists of the programs ODIN (front-end synthesis tool), ABC (logic optimization and technology mapping tool), T-Vpack (tool for packing and clustering), and VPR placement and routing tool [9]. We extended the toolset with an architecture generator and test script modules. The extended CAD flow is shown in Fig.4.

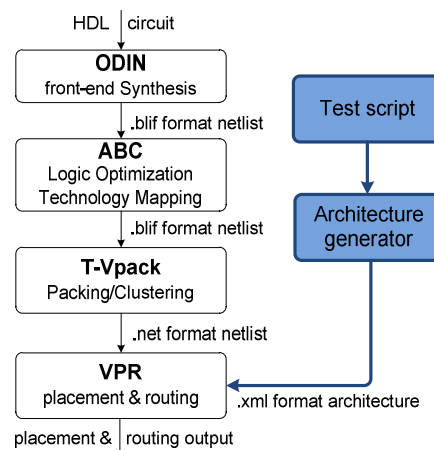


Fig. 4. Extended CAD flow.

B. Versatile placement and routing – VPR

VPR is an open-source CAD tool provided by the University of Toronto. It is widely used for research and studies in the field of FPGA architectures. The tool targets island-style FPGAs with cluster-based logic blocks and I/O pads around the periphery [10]. In its current version VPR is able to process heterogeneous FPGAs with three different types of switching blocks (Disjoint, Universal and Wilton) and different segment types with various lengths within one circuit [9].

The two main steps of VPR are the placement and routing. For the placement the tool assigns the logic blocks to legal CLB locations. The description of the logic blocks and other architecture components is made in an XML input file. For initialization, a random placement is constructed, which is later improved by using the

simulated annealing metaheuristic. Optimization goals are minimized wiring and maximized circuit speed. Once the placement is done, the routing algorithm, either timing-driven or breadth-first, determines the required connections between the logic blocks. VPR outputs placement and routing information, along with a routing resource graph [11][9]. An important characteristic of VPR is its versatility; the routing architecture and channel width can be specified easily.

C. Architecture generator

The architecture generator tool outputs an XML-format architecture description file for VPR. This file builds upon the k4_n10 [12] architecture description provided by VPR team, to assure compatibility with the set of selected VPR benchmarks. However, the timing parameters, capacitances, and resistances are modified to match a 65nm CMOS architecture file available at the Intelligent FPGA Architecture Repository (iFAR) [13]. The segment length L , the input connection block flexibility $F_{c,in}$, the output connection block flexibility $F_{c,out}$, and the switch block flexibility F_s are prepared by the Test script, and passed to the Architecture generator tool.

D. Test script

A simple Test script was written in Perl to automate experimental evaluation. Depending on the type of the test, the script prepares values of L , $F_{c,in}$, $F_{c,out}$, and F_s for the Architecture generator tool. Additionally, the script calls VPR tool suite for each benchmark from the set and stores results in the corresponding report files.

V. EXPERIMENTAL RESULTS

In order to evaluate the effect of changing routing network parameters on the channel width, the critical path delay and the total routing area, we performed several experiments. Firstly, we selected 11 benchmark circuits given in Table 1, all from the benchmarking set provided by VPR team [14][15]. In each of the experiments we were changing only one parameter, while keeping the others on their default values ($F_{c,in} = 0.25$, $F_{c,out} = 1$, $L = 2$, $F_s = 3$). The measured values were obtained from VPR reports. The critical path delay was measured in s; the routing area was measured in the number of minimum-width transistors for the given technology.

A. Effects of varying $F_{c,in}$ and $F_{c,out}$

To analyze the effects of the input block connectivity ($F_{c,in}$) on the critical path delay, the channel width, and the total routing area, we performed three independent experiments. In each of them $F_{c,in}$ was increased from 0.1 to 1.0, in steps of 0.1. For each $F_{c,in}$ in the range, the script was calling VPR for all benchmark circuits and the default values of $F_{c,out}$ and L . The same experiment was performed to evaluate the effects of $F_{c,out}$. The routing area, the critical path delay, and the channel width were measured using VPR, averaged on all 11 benchmarks, and shown in Fig. 5.

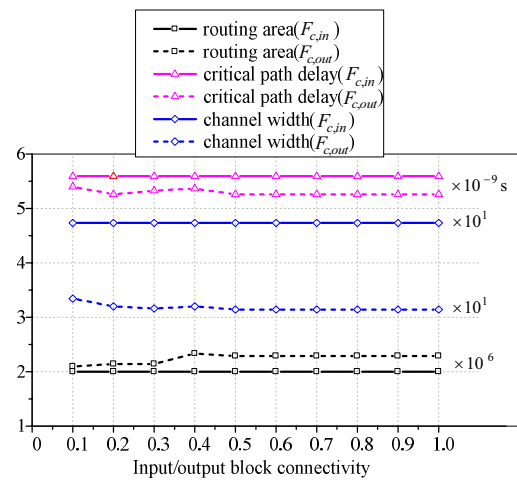


Fig. 5. The effects of $F_{c,in}$ and $F_{c,out}$ on routing area, critical path delay, and channel width.

The data should be multiplied by the appropriate factor written next to the lines. It can be seen that all measured values are insensitive to variations of $F_{c,in}$. This happens because for all $F_{c,in}$ the channel width needed to route all benchmarks is high enough not to depend on $F_{c,in}$. However, when changing $F_{c,out}$, these values become insensitive for $F_{c,out} > 0.5$. Otherwise we observe slight variations. This is due to the fact that one CLB output can drive more than one CLB input. Therefore, having larger number of tracks to which an output can connect has higher impact on the routing success than the number of tracks to which an input can connect.

TABLE 1: BENCHMARK CIRCUIT DESCRIPTIONS.

Circuit	Number of 4-input CLBs	Number of Nets
iir	94	521
iir1	138	511
cf_cordic_v_8_8_8	162	685
cf_cordic_v_18_18_18	716	3723
cf_fir_3_8_8	90	296
cf_fir_24_16_16	871	4335
oc54_cpu	443	1989
des_area	347	1026
des_perf	845	4626
rs_decoder_1	190	986
crc33_d264	350	367

B. Effects of varying segment length

To analyze the effects of the segment length L , we were varying L from 1 to 8, and calling VPR with the default settings for $F_{c,in}$ and $F_{c,out}$. The routing area, the critical path delay, and the channel width were measured, averaged on all 11 benchmarks, and shown in Fig. 6. The data should be multiplied by the appropriate factor written next to the lines. The results show that short segment lengths lead to high critical path delay, high routing area and low channel width. This is due to the large number of switches inside SW blocks, because they are needed to make short tracks longer. However, they introduce additional delay affecting both the critical path and the area. Increasing L (longer tracks) leads to better delay, but affects the channel width, because feasible routes become constrained to use longer wires. Since the routing area is dominated by transistors, the higher channel width has no high impact on the total routing area. It can be seen that the delay and the area are almost constant for $2 \leq L \leq 6$.

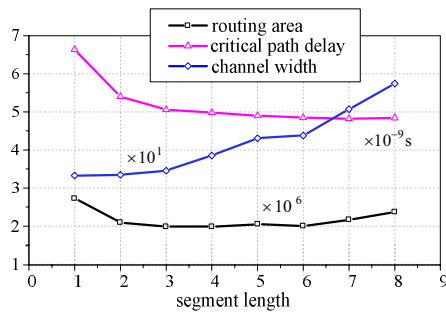


Fig. 6. The effects of segment length on routing area, critical path delay, and channel width.

Fig. 7 shows the values of the segment length for which the minimal delay, area, and channel width were obtained. Depending on the benchmark, the delay was minimized for $2 \leq L \leq 8$, the area was minimized for $2 \leq L \leq 6$, while the channel width was minimized for $1 \leq L \leq 2$. This indicated that wires spanning 2 CLBs assure good compromise for the routing network.

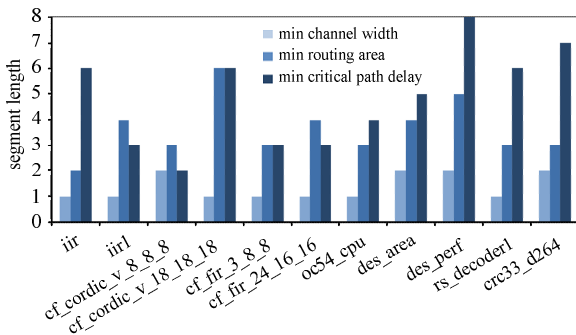


Fig. 7. Segment length leading to minimal channel width, routing area and critical path delay, for each benchmark.

Some routing architectures try to balance between delay and area by introducing two or more segment lengths within routing channels [16]. To find the best combination, we ran the test with the following pairs: (2,4), (2,6), (2,8), (4,6), and (6,8). Then, we compared the results with the minimal and the maximal delay from Fig. 6. The results for all benchmarks are shown in Fig. 8. It can be seen that two segment lengths indeed help to decrease the critical path delay, but for the majority of benchmarks this delay is not smaller than the minimal delay from Fig. 6.

VI. CONCLUSION

The experimental analysis has shown that having optimal routing network implies having a trade-off between the critical path delay and the routing area. This trade-off is achieved by varying parameters $F_{c,in}$, $F_{c,out}$, and L . If $F_{c,out}$ increases, the area increases, but the delay decreases. Increasing L helps decreasing the delay, but the area is minimized for limited L ($2 \leq L \leq 6$). Two different segment lengths within a routing channel help keeping the delay close to the minimal value, but do not bring significant improvements. Finally, $F_{c,in}$ exhibits no influence on the routing network performance. Therefore, for the selected set of benchmarks, we could say that the default values suggested by the VPR team ($F_{c,in} = 0.25$, $F_{c,out} = 1$, $L = 2$, $F_s = 3$) provide good FPGA routing network performance.

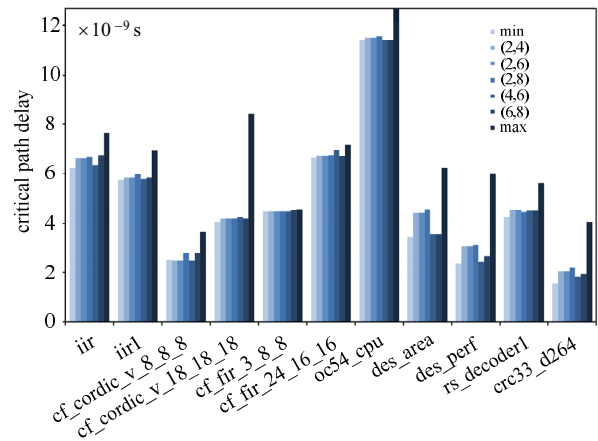


Fig. 8. Effects of a pair of segment lengths on the delay.

REFERENCES

- [1] M. H. Moaiyeri, A. Jahanian, K. Navi, "Comparative Performance Evaluation of Large FPGAs with CNFET- and CMOS-based Switches in Nanoscale," *Nano-Micro Lett.* 3 (3), pp. 178-188, 2011.
- [2] V. Betz, J. Rose, "FPGA Routing Architecture: Segmentation and Buffering to Optimize Speed and Density," presented at the 1999 Seventh ACM/SIGDA International Symposium on Field-Programmable GateArrays, Monterey, CA, USA.
- [3] I. Kuon, R. Tessier, J. Rose, *FPGA Architecture: Survey and Challenges*. Hanover, USA: now Publishers Inc, 2008, ch. 5.
- [4] V. Betz, J. Rose, "Directional Bias and Non-Uniformity in FPGA Global Routing Architecture," presented at the 1996 International Conference on Computer-Aided Design, San Jose, CA, USA.
- [5] J. Luu, I. Kuon, P. Jamieson, T. Campbell, A. Ye, W. M. Fang, J. Rose, "VPR 5.0: FPGA CAD and Architecture Exploration Tools with Single-Driver Routing, Heterogeneity and Process Scaling," presented at the 2009 Seventeenth ACM/SIGDA International Symposium on Field-Programmable GateArrays, Monterey, CA, USA.
- [6] D. Chen, J. Cong, P. Pan, *FPGA Design Automation: A Survey*. Hanover, MA, USA: now Publishers Inc, 2006, ch. 1.
- [7] D. Lewis, V. Betz, D. Jefferson, A. Lee, C. Lane, P. Leventis, S. Marquardt, C. McClintock, B. Pedersen, G. Powell, S. Reddy, C. Wysocki, R. Cliff, J. Rose, "The StratixTM Routing and Logic Architecture," presented at the 2003 Eleventh ACM/SIGDA International Symposium on Field-Programmable GateArrays, Monterey, CA, USA.
- [8] "Versatile Place and Route," <http://www.eecg.toronto.edu/vpr/>, University of Toronto.
- [9] V. Betz, "VPR and T-VPack User's Manual (Version 5.0)," 2008.
- [10] A. Sharma, C. Ebeling, S. Hauck, "Architecture Adaptive Routability-Driven Placement for FPGAs," presented at the 2005 Thirteenth ACM/SIGDA International Symposium on Field-Programmable GateArrays, Monterey, CA, USA.
- [11] N. Chan King Choy, "Activity-based Power Estimation and Characterization of DSP and Multiplier Blocks in FPGAs," M.A.Sc. thesis, Dept. Electrical and Computer Engineering, University of British Columbia, Vancouver, Canada, 2006.
- [12] http://www.eecg.toronto.edu/vpr/source/k4-n10_mult.xml/.
- [13] "N10K04L01.FC20FO10.AREA1DELAY1.CMOS65NM.BPTM," http://www.eecg.utoronto.ca/vpr/architectures/architecture_table.html/.
- [14] http://www.eecg.toronto.edu/vpr/source/blif_after_abc.tar.gz/.
- [15] S. Birk, G. Steffan, J. H. Anderson, "Parallelizing FPGA Placement Using Transactional Memory," presented at the 2010 International Conference on Field-Programmable Technology, Beijing, China.
- [16] G. Lemieux, D. Lewis, *Design of Interconnection Networks for Programmable Logic*. Norwell, MA, USA: Kluwer Academic Publishers, 2004, ch. 7.