Modeling FPGA Logic Architecture

Petar Minev[1] and Valentina Kukenska[2]

***Abstract* – Designing FPGA architectures is a process related to numerous experiments involving different combinations of architectural parameters. The development of analytical models, indicative of the relation between architecture parameters and their impact on the working area, power consumption and FPGA speed, would narrow the range of researched architectures and make designing new FPGA IC families faster.**

***Keywords* – Field-Programmable Gate Arrays, Modelling, Logic Architecture, Logic Blocks, CAD.**

I. INTRODUCTION

A single empirical approach of research and comparison has dominated the practice of developing new or improving already existing FPGA architectures. With it, a number of standard schemes, corresponding to the target schemes that the FPGA family is designed for, are synthesized in different architectures with the help of CAD tools. They allow changing the given architecture parameters. This approach is similar to the one used in the development and research of computer architectures, wherein a standard myriad of software applications are compiled with different processor architectures in order to measure their output.

An alternative to this empirical approach is the use of theoretical methodic, wherein the applications are modeled through statistical and graphical theories, while redesigning and attaching an application to a given architecture is modeled using probability calculations and another theoretical apparatus. The creation of such a methodic has been the object of different studies, usually focusing on a narrow range of architectures and CAD tools. These studies mainly review the tracking architecture and only provide a small amount of information about the logic architecture. The issue for creating and implementing a fundamental and applicable theory in studying FPGA architectures still remains open.

The purely experimental methodic for FPGA architecture research requires a large number of experiments with many different combinations of architecture parameters. The development of analytical models, indicative of the relation between architecture parameters and their impact on the working area, power consumption and FPGA speed, would reduce the number of studied architectures. Once the count of possible solutions is reduced, we can use the traditional experimental methodic to pinpoint the architecture parameters. This would considerably accelerate the while process of FPGA design.

[1]Petar Minev, Department of Computer Systems and Technologies, Technical University of Gabrovo, Phone: +359 66 827 411, E-mail: pminev@tugab.bg.

[2]Valentina Kukenska, PhD, Department of Computer Systems and Technologies, Technical University of Gabrovo, Phone: +359 66 827 456, E-mail: vally@tugab.bg.

A. Architecture of FPGA devices

The architecture of modern FPGA devices is shown on fig.1. It consists of logic blocks, configured in a specific way as to provide for the links among the LBs. The very LBs are also configured in order to perform a given task. Configuration is done by storing configuration data in SRAM cells, which are part of the structure of FPGA IS.

The logic block structure, also called logic architecture is shown on fig.2. It consists of *N* number of basic logic elements with k number of inputs each.

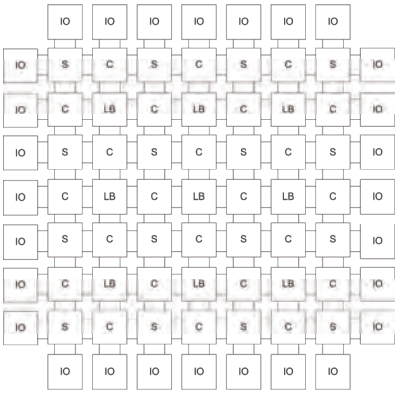

Fig.1. Overview of FPGA architecture

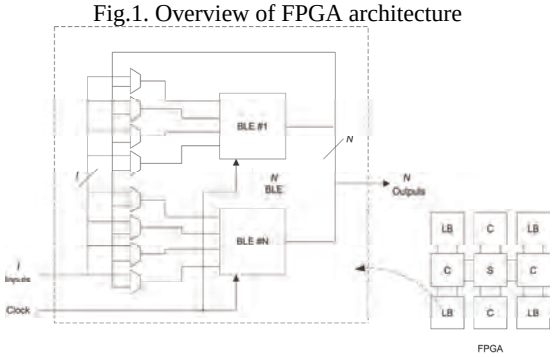

Fig.2. Logic architecture of FPGA

B. Specifying the number of I inputs for a logic block

The number of inputs I can be either limited, i.e. $I < k.N$ or unlimited - $I = k.N$. In most of the FPGA IC families, produced by Xilinx Inc., the number of LB inputs is unlimited and equal to the total number of the BLE elements' inputs in the LB. In the case of the other major manufacturer - Altera Inc., the FPGA devices feature a smaller number of inputs than the overall count of BLE elements' inputs in the LB. Table 1 shows some of the main FPGA IC families of Xilinx and Altera, and their logical architecture parameters.

In Ahmed work, [1] has derived the following analytical

model, specifying the right number of inputs (I) for a single LB, given k and N.

$$I = \frac{k}{2}(N+1) \quad (1)$$

The model has been derived from an attempt to find out the minimal number of inputs (I), necessary to provide for a 98% of LB usage in the FPGA architecture. The average error percentage, as reported by the authors of the model, is 10.1% against a standard deviation of 7.6%.

TABLE I
LOGIC ARCHITECTURE OF FPGA IC FAMILIES, MANUFACTURED BY XILINX AND ALTERA

	Xilinx FPGA Families					Altera FPGA Families					
Parameter	Spartan 3	Virtex	Virtex II	Virtex 4	Virtex 5	Cyclone	Cyclone II	Cyclone III	Stratix	Stratix II	Stratix III
N	8	4	8	8	8	10	16	16	10	8	10
k	4	4	4	4	6	4	4	4	4	6	6
I	32	16	32	32	48	23	36	36	26	38	44

In Betz & Rose works [3] and [4] derived the following dependency between the necessary number of inputs (I) and N if k = 4:

$$I = 2 \cdot N + 2 \quad (2)$$

The authors show this is the right number of inputs I to get a 98% usage of the logic resources in the logic block given that k=4. What's interesting in this case is that you can get a 100% usage when I ranges from 50% to 60% of the overall number of BLE elements' inputs. Similar results are reported in the work of Fang & Rose [7], who explore the dependency between the average number of used inputs per LB (λ) out of the overall number of BLEs (N) in a single LB. They derived the following dependency:

$$\lambda = 0.88N + 3{,}2 \quad (3)$$

This dependency is derived from a regression analysis with the implementation of a 20x20 multiplier in a logic architecture, wherein N varies from 1 to 17 and k = 4. The square of the deduced correlation coefficient for regression dependency (3) is 0,994, which confirms its linear nature. The authors believe this survey is representative and covers the needs of a wide circle of schemes implemented in FPGA logic architecture.

The three models presented above are based on experimental research with different extent of representation. Most representative is the study of Ahmed [1], wherein k is included, apart from N. In this particular research N varies from 1 to 10, and k varies from 2 to 7, which makes a total of 60 different architectures. 28 standard test schemes are implemented in each of these 60 architectures. The parameter I is also configured (changes) in the range from 1 to $k \times N$. The purpose is to find what I can get you 98% of usage of the resources in the logic block.

The other two studies do not aim at deriving a representative model for specifying the necessary number of inputs I per LB and the proposed models cannot be used as universally applicable (for all logic architectures and all schemes implemented in them).

In the study of Lam [9], a theoretical approach is used to deduce an analytical model, which defines the necessary average number of inputs per LB given certain parameters of the logic infrastructure and the FPGA-implemented schemes.

$$i = \frac{(k+1-\gamma)N^p}{1+\frac{1}{f}}, \quad (4)$$

Where γ is the average number of unused BLE inputs and is defined in a table with regard to k, and f is the average number of pins connected to a given output of the entire scheme. The value p is known as Rent's constant. The parameter f is a function of i, N, k and p, which makes model (4) recursive and hard to calculate manually. The model is validated with two algorithms for grouping TV-Pack and iRAC. The results from the model are similar to those experimentally derived from iRAC and are quite different to the ones deduced with TV-PACK. This is understandable given the fact that TV-PACK tries to minimize the number of used BLE elements, whereas iRAC – the number of used LB pins.

C. Rent's rule

Rent's rule was empirically derived for digital IC with average extent of integration that IBM produced back in 1960 [5]. It represents the relation between the number of pins (external connections) P for an area of B number of logic blocks, where each logic block has C number of pins – equation (5).

$$P = CB^p, \quad (5)$$

where p is Rent's constant. For the different types of IC, p is derived experimentally. This rule is successfully applied for specifying the necessary number of pins for different types of IC. Thus, for instance, it has been used to find the right count of pins for the all generations of the Intel's Pentium processors [6].

Though Rent's rule was not derived for FPGA schemes, it represents an interconnection among parameters, which are compatible with those in the FPGA logic architecture. At that stage it looks a proper base in the search of dependency among the parameters in the FPGA logic architecture and of a model that describes it in the right way.

The analysis of the existing models for FPGA logic architecture gives us grounds to separate them in two categories: linear and nonlinear.

The linear models are easy to apply and use, but they do not indicate the extent of complexity of the schemes implemented in the architecture. This is no issue for the nonlinear models, but they are recursive and hard for manual calculation. Further research is needed to find a proper, easy to apply model to indicate the impact of the complexity of the schemes implemented in the logic architecture. A possible solution could be a model, based on Rent's rule, which can be applied to the logic architecture and its parameters. Since the functional dependency in Rent's rule is exponential for proving the hypothesis that the rule is appropriate to apply to the FPGA architecture, it is necessary to check whether the

interconnection among the parameters in the FPGA logic architecture is linear or nonlinear (exponential), and then continue with building a model and its verification.

II. Modeling The Relation Among The Parameters In The FPGA Logic Architecture

A. Making a hypothesis about the relation among the parameters in the FPGA logic architecture

If the relation among the FPGA architecture parameters proves nonlinear, we could model their dependency using Rent's rule (6):

$$P = CB^{p}, \tag{6}$$

where $P=I_{au}+N_{au,}$ $C=k_{au}+1$ и $B=N_{au}$, and p is Rent's contant.

Once we apply logarithm to equation (6), we derive a linear equation of the type:

$$\log(P) = \log(C) + p\log(B) \tag{7}$$

If we explore the relation among parameters *I*, *N*, *k* and it turns out it is of the type (7), we could then continue with deducing a model and verifying it.

Exploring the relation among the FPGA architecture parameters

The methodic used to explore the relation among the parameters in FPGA logic architectures is shown on fig. 3. It is based on the methodic for designing FPGA devices, also used for testing the qualities of new FPGA architectures.

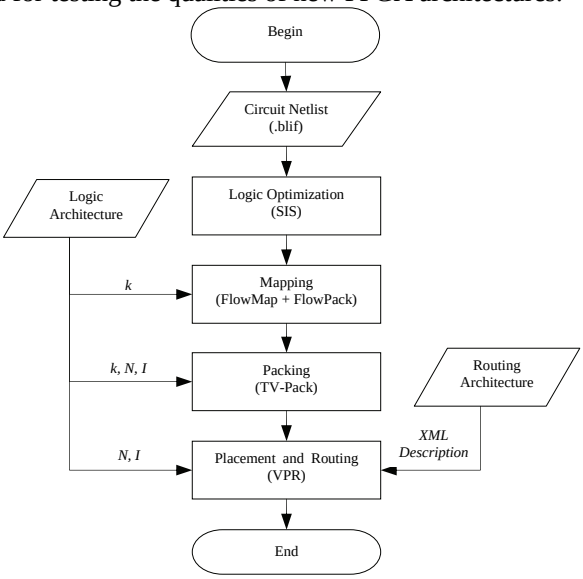

Fig. 3.Methodic for design and testing of FPGA devices

Firstly, the standard test scheme in the shape of a netlist with logic elements and triggers is synthesized and optimized with the help of SIS algorithm [10] in order to derive an optimized netlist with logic element and triggers, too. Once optimized the netlist undergoes decomposition and grouping using the FlowMap and FlowPack algorithms, thus obtaining a number of sub-schemes, each featuring a BLE element from the FPGA architecture. During the packing stage, the TV-Pack algorithm groups the BLE elements in clusters of *N* elements up until the netlist runs out of BLE elements for grouping.

Parameter *k* is input to algorithms FlowMap in order to show the size of the BLE elements and the size of the targeted subschemes. Parameters *k*, *N* and *I* are input to the packing algorithm so that the BLE elements from the decomposed netlist can form clusters of N elements, while observing the maximum number of inputs I in a cluster. An algorithm makes it possible for the factor k to vary in the range of 2 to 7, while N ranges from 1 to 20. This is the range of the current research, too.

The input factor *I* may vary from 1 to a maximum value of *k.N*. In this research no limits for *I* were set and all experiments featured its maximum value of *k.N*. Thus, it is not influencing the output parameters for the examined object. The packing ends with a netlist of LB (clusters), due to be positioned and tracked in the architecture of the FPGA device. The last two stages in the implementing process are performed by the VPR [2] program.

This study takes into account the statistics on the average number of used BLE (N_{au}) and the average number of used LB inputs (I_{au}). These are statistical reports, derived from the tasks of TV-Pack. The research used 10 standard schemes from the test pool MCNC [11], implemented in 120 different logic architectures.

Following the statistical analysis of the accumulated experimental data, as conducted by [8], several conclusions are drawn up:

The input factor *k* influences the output parameter k_{au}, but it does not affect the output parameters N_{au}.

The dispersion analysis also shows that the input factor *N* does not affect the output parameters k_{au}, but it seriously affects N_{au}. The interaction between *k* and *N* also impacts N_{au}. The two input parameters k and *N* also tangibly impact I_{au}. The same applies to the impact between them.

To find out whether the impact of the input factors on the output parameters is linear or nonlinear, [8] included further three output parameters in their research: $\log(N_{au})$, $\log(N_{au} + I_{au})$ и $\log(k_{au} + 1)$, which represent respectively the number of BLE in a single LB, the number of LB pins and the number of pins in a single BLE. The conclusions: *k* and *N* impact $\log(N_{au})$, while their relation is not influencing it; $\log(N_{au} + I_{au})$ is affected by both input factors and their relation; parameter $\log(k_{au} + 1)$ is only influenced by factor *k*.

The regression analysis that followed revealed that using the logarithmic form of output parameters means more tangible functional dependencies (larger R) and considerably reduced percentage of standard errors (fig. 4 ÷ 7). This allows the assumption that the dependency of the output parameters on the input factors in the FPGA logic architecture is nonlinear and their functional dependencies get linear in the process of applying logarithms.

As fig. 6 and fig. 7 show, the regression equations of the derived dependencies are of the type:

$$y = b_0 x + b_1 \tag{8}$$

and correspond exactly to equation (7). This gives us grounds to accept the hypothesis that the number of LB pins in FPGA can be defined using Rent's rule (5). Once accuracy

is improved and the percentage of errors drops, Rent's constant is defined with regard to the number of inputs k in a BLE.

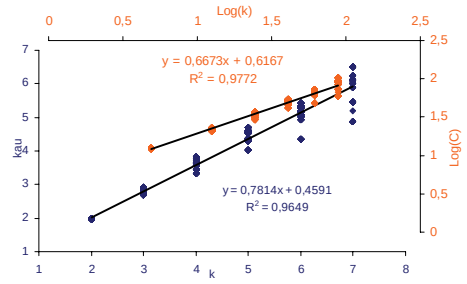

Fig.4. Comparing the dependencies of k_{au} on k and log(C) on log(k).

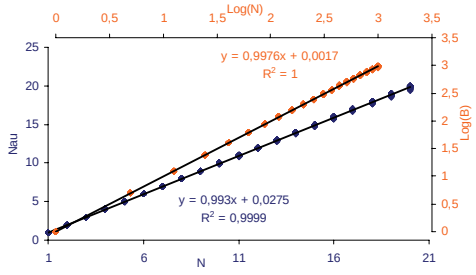

Fig.5. Comparing the dependencies of N_{au} on N and log(B) on log(N).

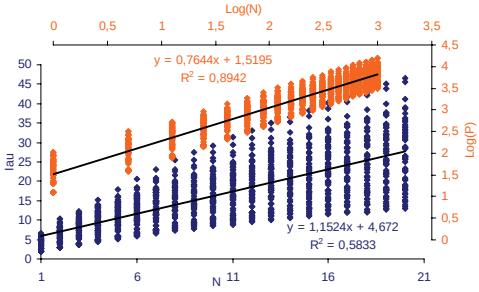

Fig. 6. Comparing the dependencies of I_{au} on N and log(P) on log(N).

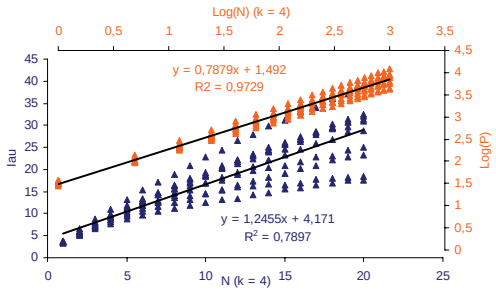

Fig.5. Comparing the dependencies of I_{au} on N and log(P) on log(N), if k = 4.

B. Modelling the relation among the parameters in the FPGA logic architecture

The confirmation of the hypothesis that the number of pins used in a LB of FPGA can be accurately defined with the help of Rent's rule, allows us to specify the necessary number of inputs in a logic block as a function of input factors k and N.

Once we apply algorithms to equation (7) and replace P, B and C with their equivalent parameters from the FPGA logic architecture, we can write down:

$$I_{au} = \left(k_{au} + \frac{N_{au}}{N}\right) N_{au}^{\,p_k} - N_{au}\,, \qquad (9)$$

where k_{au} = f(k); N_{au} = f(N) и p_k = f(k).

III. Conclusions And Future Work

The paper presents an analytical model, describing the FPGA logic architecture. Based on Rent's rule, this model specifies the necessary number of inputs per LB, given the number of BLE in the block and the number of inputs for BLE, as well as Rent's constant, which gives the scheme implemented in the logic architecture.

The model should undergo verification to confirm its reliability and the benefits from its usage.

Acknowledgement

This paper is financed by project: Creative Development Support of Doctoral Students, Post-Doctoral and Young Researches in the Field of Computer Science, BG 051PO001-3.3.04/13, EUROPEAN SOCIAL FUND 2007–2013г. OPERATIONAL PROGRAMME "HUMAN RESOURCES DEVELOPMENT"

References

[1] Ahmed E., "The Effect of LUT and Cluster Size on Deep-Submicron FPGA Performance and Density", University of Toronto, 2001

[2] Betz V. et All, VPR and T-VPack1 User's Manual, University of Toronto 2008.

[3] Betz V. and Rose J., "Cluster-Based Logic Blocks for FPGAs: Area-Efficiency vs. Input Sharing and Size", IEEE Custom Integrated Circuits Conference, Santa Clara, CA, 1997, pp. 551-554.

[4] Betz V. and Rose J., "How Much Logic Should Go in an FPGA Logic Block?", IEEE Design and Test Magazine, Spring 1998, pp. 10-15.

[5] Christie P. and D. Stroobandt, "The interpretation and application of Rent's rule," IEEE Trans. VLSI Syst. (Special Issue on System- Level Interconnect Prediction), vol. 8, no. 6, pp. 639–648, Dec. 2000.

[6] Davis J., V. K. De, and J. D. Meindl, "A stochastic wire-length distribution for gigascale integration (GSI)—Part I: Derivation and validation", IEEE Trans. Electron Dev., vol. 45, no. 3, pp. 580–589, 1998.

[7] Fang W., A Modeling Routing Demand for Early-Stage FPGA Architecture, Master Thesis, University of Toronto, 2007.

[8] Kukenska V., P. Minev, "A Study into the Interconnections of Parameters in the Logic Architecture of FPGA Devices", ETRAN2010, Serbia.

[9] Lam A. et All, "An Analitical Model Describing the Relationships between Logic Architecture and FPGA Density",

[10] Sentovich E.M. et All., "SIS: A System for Sequential Circuit Analysis", Tech. Report No. UCB/ERL M92/41, University of California, Berkeley, 1990.

[11] Yang S., "Logic Synthesis and Optimization Benchmarks, Version 3.0", Tech. Report, Microelectronics Centre of North Carolina, 1991.