Challenges for Optimization of Transistors Array-based Via-Configurable Regular Layout

[1]V. Dal Bem, [1]A. I. Reis, [1]R. P. Ribas
{vdbem, andreis, rpribas}@inf.ufrgs.br

[1]PGMICRO, Federal University of Rio Grande do Sul, Porto Alegre, Brazil

Abstract

This paper points several relevant aspects about a potential litho-friendly design method: the transistors array-based via-configurable regular layout approach. The technical background leads the idealization of an EDA environment for optimized choice and tuning of regular blocks. The preliminary results indicate the potential for the proposed design methodology.

1. Introduction

Technology scaling has been threatened by the reduction of manufacturing yield and increasing costs of lithography masks. The very large amount of intra-chip devices worsens this scenario by increasing EDA algorithms complexity and extending time-to-market.

Structured-ASICs arise as a solution for these challenges. In this approach, some masks present a very regular layout, usually repeating a pattern (base block) over the circuit, which makes the lithography more predictable and leads to higher manufacturing yields [1-3]. These masks may even be pre-fabricated, lowering their costs. Increased layout regularity and reduced number of customizable masks also lead to more precise and time-efficient EDA algorithms [4].

Via-configurable fabrics are a variety of structured-ASIC. This class has fixed transistors and metal wires. Therefore, only via and contacts masks are configured to implement different functionalities and connections. Via-configurable fabrics can be divided according to their base blocks, which may be LUT-based [5], PLA-based [6], simple gates-based [7] or transistors array-based [1-3]. The transistors array-based approach is similar to the classical mask-programmable gate arrays [8] in many aspects. However, there are different proposals about base-block resources and design [1-3].

This paper is organized as follows. Section 2 lists and discusses several relevant aspects about transistors array-based via-configurable approaches, highlighting how different proposals behave in each one of them. Section 3 presents an idealized proposal of a blocks tuning environment. Section 4 presents preliminary results on optimization over specific block architecture. Finally section 5 exposes conclusions.

2. Background

2.1. Lithography Robustness

Lithography robustness is not a binary aspect. It is necessary to define a method for quantifying how friendly a design is for the lithography process.

The via-configurable approaches based on regular blocks as [1-3, 5-7] have a great potential for being litho-friendly, due to their increased regularity and pattern repetition methodology. However, there is a lack of a refined analysis about this aspect in the referred proposals.

2.2. Logical Capability

The logical capability of a base block is related to the amount of functions that can be implemented within this block. A greater logical capability can lead to designs with less base- block instances.

2.2.1. Single Unit Functional Coverage

The number of realizable functions in a single unit (base block) depends on the amount of transistors in the base block, on whether the transistors gates are aligned in both planes and on resources available for local routing. While the works in [1, 3] have aligned gates, the proposal in [2] does not have this restriction.

The intra-cell routing may forbid configuring some functions which could be realizable in the transistors array. In n-VCC [1] the intra-routing scheme considers the worst case for necessary metal lines, avoiding this effect. The impact of intra-cell resources on the logical capability of a VCTA-like structure is shown in [9].

2.2.2. Packing

The fixed number of transistors per base block can lead to unused transistors. Thus, it is convenient to pack different functions in the same block when possible. In n-VCC [1], a maximum of two different functions can be implemented in a same block. In [3] an algorithm for packing two functions in one unit is also presented. The VCTA [2] approach is more flexible in this aspect, since the intra-cell routing may allow multiple outputs.

2.2.3. Multiple-blocks

When a function does not fit in a single unit, several base blocks can be used to implement the target function. There is a trade-off between how area efficient is the abutment of blocks and how much space remains available for inter-cell routing. The proposal [2] makes use of the available intra-cell metal lines for inter-cell routing. Additionally, the architecture proposed in [1] has a portion used exclusively for external routing.

Function splitting can be done at transistor or mapping level. The splitting at transistor level may be area efficient but can have significant impact on algorithm runtime. Splitting at mapping level consists of expressing the target function by multiple simple gates. Inter-cell routing is a major issue for these algorithms.

2.3. Electrical Performance

In via configurable approaches all transistors of the same type have the same (drawn) width [1, 3]. In [2] all transistors have the same width regardless of the type. The work in [9] and [10] show that transistors sizing has small impact on performance for VCTA-like but the proposals as [1, 3] may benefit from an adequate transistor width choice. Nevertheless, there is no discussion about this issue in the referred proposals.

Since the width of the transistors is fixed, only folding can be used to obtain greater current capabilities. Nevertheless, due to the possible area penalty this strategy is mostly used with cells that have few transistors.

A logical function can also be implemented using more than one stage. The main goal is to have better implementations of cells with large drive strengths. The most intuitive way is to add a buffer to the cell output. Thus, the cell drive strength is modified only by changing the output buffer [1, 7].

2.4. Dedicated Blocks for Specific Logic

Logic gates like multiplexers, full-adders, half-adders and sequential elements are cells of special interest. Thus, a base block that can efficiently implement those cells can have significant advantage in some designs. Although the studied proposals are aimed at random logic implementation, they show some analysis of such cells.

2.4.1. Muxes

The work in [3] depicts the implementation of a 2-to-1 Mux in a single base block. In [1] there is an extra block with an array of auxiliary inverters- for this cell.

2.4.2. Half-Adders and Full-Adders

Some approaches [1, 2] have dedicated blocks for arithmetic logic to reduce the area overhead. In [1] an ALU is designed with 2.4 times the area of a standard cell approach. The approach [2] uses 1.88 times the area of the standard cell approach in the layout of a CLA32.

2.4.3. Latches and Flip-Flops

The implementation of a flip-flop split in three base blocks is shown in [3]. The proposal in [1] indicates a good implementation of flip-flop if the auxiliary inverter-array blocks are efficiently used.

2.5. Circuit Level

All block-level characteristics analyzed in previous sections must have the impact confirmed on circuit level. The EDA flow for designing circuits using via-configurable approach is usually adapted from the standard-cell conventional flows [11, 1]. However the restrictions of regular layouts may allow using algorithms which reach exact solutions [4] leading to considerable gains in building specific EDA tools..

3. Ideal Blocks Tuning Environment

An alternative still unexplored approach is the optimization of the via-configurable base-block previously executing the conventional steps of the standard-cell flow.

This environment should take into consideration all the aspects mentioned in this paper for choosing the most convenient fabric proposal for the given circuit constraints. Moreover, it also should also be able to tune the physical resources of the chosen base block (like number of transistors and metal lines) for a set of target circuits.

This proposal would help minimizing the intrinsic penalties of the regular design method, and would carry the benefit of being independent to the posterior applied EDA flow. Such environment is currently under analysis and construction. The preliminary optimization results over a VCTA-like [2] architecture evidences the potential of a preliminary version of this flow, as described in next section.

4. Evaluation

The analysis and optimization performed in this work is based on the general array-based regular fabric named VCTA detailed in [2]. Fig. 1 illustrates block patterns in terms of metal lines and transistors. In this figure, three templates are shown, the first one with 8 (horizontal) transistors and 6 metal1 (vertical) lines, the second one with 12 transistors and 4 metal1 lines, and the third one with 12 transistors and 6 metal1 lines.

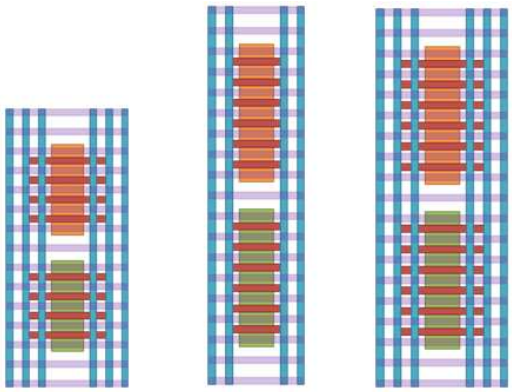

Fig. 1 – VCTA-like templates examples [2].

This evaluation aims to answer how much gain can be achieved in choosing the correct template for a set of benchmarks, in terms circuits' area, speed and power consumption. For that we have tested several template possibilities, and compared them to the template with 12 transistors and 6 metal1 lines, which is the closest one to the VCTA in terms of physical resources (thus labeled "original template").

The comparison is done at technology mapping level. Each tested template is used as basis for building a cells-library. The library of each template contains the combinational functions which can be expressed through via-configuration in that template, plus special interest functions like Flip-Flops and Adders. Each library is then characterized using the transistor netlist created for each function (thus excluding layout parasites) and used for mapping sets of benchmarks.

For power and area comparison a common timing constraint where applied. While for timing analysis, the constraint varied until exposing the best possible speed for mapping with each library.

Three sets of benchmarks were mapped and referred in Table 1 as combinational, sequential and arithmetic. Combinational is a set of 14 combinational ISCAS85 benchmarks [12], Sequential is a set of 30 sequential ISCAS95 benchmarks [13]. Arithmetic is a set of 5 arithmetic circuits (fpu100, divider, quadratic_func, adder_hcsa, antilog2) obtained from opencores [14].. Tab 1. shows the mapping results in area (μm^2) and power consumption (μw). Mainly due to the negligence of layout parasites in this flow, the achieved timing results presented no significant difference when using the different block templates, and thus are omitted in the table.

These results take in consideration the same post-processing step as described in [9], which increases the results accuracy due to the emulation of using multiple-output cells.

Tab.1 – Comparison between the original and optimized templates

Set of benchmarks (average)	**Original template area**	**Best tested template area**	**Area gain**	**Original template consumption**	**Best tested template consumption**	**Consumption gain**
Combinational	1430,592	1430,592	0%	30.942	30.942	0%
Sequential	4524,096	4335,116	4,2%	16.407	16.036	2,26%
Arithmetic	17154,432	16797,216	2,8%	111.370	102.316	8,13%
Best-case circuit	**Original template area**	**Best tested template area**	**Area gain**	**Original template consumption**	**Best tested template consumption**	**Consumption gain**
Combinational	532,8	414,96	22,11%	6.274.921	5.358	14,6%
Sequential	72	63,84	11,33%	4.672.646	2.463.885	47,27%
Arithmetic	1062,72	994,08	6,4%	421.512	379.995	9,8%

5. Conclusions

Via-configurable transistors array approach has increasing potential with technology scaling. The regularity of such approach makes lithography processes more predictable, and its restrictive rules allow algorithms being more precise and time-efficient. However, there is a lot of uncovered space of analysis for a solid base-block choice, and it becomes convenient an EDA environment which properly chooses and tunes the base-block resources for a target circuit.

The potential of such idealized environment is evaluated through preliminary results over a VCTA-like architecture optimization. These results present a comparison analysis between the optimized blocks and the original one, and evidences how the appropriate tuning of the regular blocks resources can minimize the overhead of regular approaches. An average optimization is possible for entire sets of sequential and arithmetic circuits, while the individual analysis of each benchmark revealed cases of optimization up to 22,11% in area and 47,27% in power consumption, only by choosing a regular block more adequate and tuned to circuit needs.

6. Acknowledgements

Research partially funded by Nangate Inc. under a Nangate/UFRGS research agreement, by CAPES and CNPq Brazilian funding agencies, by FAPERGS under grant 11/2053-9 (Pronem)., and by the European Community's Seventh Framework Programme under grant 248538 – Synaptic.

7. References

[1] Y. Ran and M. Marek-Sadowska, “Designing via-configurable logic blocks for regular fabric,” *IEEE Trans. on VLSI*, vol.14, no.1, pp.1-14, January 2006.

[2] M. Pons, F. Moll, A. Rubio, J. Abella, X. Vera and A. González, “VCTA: A Via-Configurable Transistor Array regular fabric,” *VLSI-SoC'10*, pp.335-340, 2010.

[3] M. Li, H. Tung, C. Lai and R. Lin, “Standard Cell Like Via-Configurable Logic Block for Structured ASICs,” *ISVLSI'08*, pp.381-386, 7-9 April 2008.

[4] B. Taylor and L. Pileggi, “Exact Combinatorial Optimization Methods for Physical Design of Regular Logic Bricks,” *DAC'07*, pp.344-349, 4-8 June 2007.

[5] L. Pileggi, H. Schmit, A. J. Strojwas, P. Gopalakrishnan, V. Kheterpal, A. Koorapaty, C. Patel, V. Rovner and K.Y. Tong, “Exploring regular fabrics to optimize the performance-cost trade-off,” *DAC'03*, pp. 782- 787, 2-6 June 2003.

[6] N. Jayakumar and S.P. Khatri, “A metal and via maskset programmable VLSI design methodology using PLAs,” *ICCAD'04*, pp. 590- 594, 7-11 November 2004.

[7] B. Hu, H. Jiang, Q. Liu and M. Marek-Sadowska, “Synthesis and placement flow for gain-based programmable regular fabrics.” *ISPD'03*, pp. 197-203, 2003

[8] E. T. Lewis, “Optimization of device area and overall delay for CMOS VLSI designs,” *Proceedings of the IEEE*, vol.72, no.6, pp. 670- 689, 1984.

[9] V. Dal Bem, P. F. Butzen, F. S. Marranghello, A. I. Reis, R. P. Ribas, “Impact and optimization of lithography-aware regular layout in digital circuit design” *ICCD'12*, pp.279-284, 2011.

[10] F. S. Marranghello, V. Dal Bem, A. I. Reis, F. M. Echeto and R. P. Ribas,. “Transistor Sizing in Lithography-Aware Regular Fabrics”. *SBCCI'11*, pp. 97-102. 2011.

[11] N.V. Shenoy, J. Kawa and R. Camposano,”Design automation for mask programmable fabrics,” *DAC'04*, pp.192-197, 2004.

[12] F. Brglez and H. Fujiwara, “A neutral netlist of 10 combinational benchmark circuits,” *Proc. of Int’l Symp*. Circuits and Systems (ISCAS), 1985.

[13] Obtained as subset of IWLS2005 benchmarks. http://www.iwls.org/iwls2005/benchmarks.html.

[14] Community within open source hardware IP-cores.