Topological Aspects of Non-Series-Parallel Transistors Networks

Maicon Schneider Cardoso, Regis Zanandrea, Renato Souza de Souza,
Leomar Soares da Rosa Jr., Felipe de Souza Marques

Group of Architecture and Integrated Circuits
Federal University of Pelotas
Pelotas, Rio Grande do Sul
{mscardoso, rzanandrea, rsdsouza, leomarjr, felipem}@inf.ufpel.edu.br

ABSTRACT

Considering the number of transistors necessary to implement Boolean functions, graph-based approaches have demonstrated relevant results lately. On the other hand, by decreasing the transistor count via non-series-parallel networks, some new aspects are introduced, impacting not only on the classical placement algorithms, but also on the layout itself. This paper proposes a flow to verify topological aspects of non-series-parallel networks in order to guide the cell layout generation. Results performed over intensively used benchmarks demonstrate that a relevant part of the non-series-parallel set has singular new topological characteristics: non-planarity and different number of transistors between the logic plans.

Categories and Subject Descriptors

B.7.2 [**Integrated Circuits**]: Design Aids – *Layout*.

General Terms

Algorithms, Design.

Keywords

Non-series-parallel networks; logic topology; planarity; duality; placement

1. INTRODUCTION

Graph-based logic minimization procedures have gained relevance recently [1,2]. Through the path sharing in the logic network, these methods could reach several optimizations in terms of number of transistors for a large set of functions when compared to other Boolean factorization methodologies [3,4]. However, these graph-based methodologies can produce arrangements with some new aspects, such non-planarity and non-duality, affecting some of the classical algorithms that have been used for the physical synthesis of a cell.

The placement procedure is one of the most important methods on the VLSI design flow. It has directly impacts on cell sizing and in routing complexity. For that reason, many algorithms try to find an optimized and quickly solution since these two aspects are fundamental to any Electronic Design Automation (EDA) tool.

This paper introduces a flow to verify topological aspects of non-series-parallel networks, since it has directly impacts on placement, as described before. Starting from a Boolean function, we generate the two optimized logic plans by using a state-of-art graph-based procedure [1], which is followed by the arrangement topology test. This way, it provides useful information about those impacts in widely used Boolean functions. So, allows us to evaluate what can be done to overcome these peculiar aspects inherent to non-series-parallel cells.

The reminder of this paper is organized as follows. Section II review some important concepts about logic networks and graph theory necessarily to the fully comprehension of this paper; section III introduces the motivation for the investigation of topological aspects in transistor networks and the classical placement methodologies; section IV presents the methodology proposed; section V shows the obtained results; finally, section VI presents the conclusions of this paper and the proposed future works.

2. PRELIMINARIES

This section reviews and presents some basic concepts and algorithms necessary to the fully understanding of this paper.

2.1 Logic Networks

2.1.1 Logic Plans

A CMOS gate is composed by two logic plans: the pull-up network (PUP) and pull-down-network (PDW). The PUP is composed by PMOS transistors and its function is to provide a connection between the output and V_{DD}. Similarly to that, the PDW is composed by NMOS transistors and its function is to connect the output to V_{SS}.

Logic plans can be represented by graphs where each transistor is an edge and each node is a vertex on the structure. It allows to apply several algorithms to optimize the logic network [1,2] or the physical layout [5].

2.1.2 Series-Parallel Networks

A SP transistor network is a kind of arrangement where all switches have either series or parallel connections. Therefore, there is no bridge connection [4].

Figure 1 (a) shows a SP network that implements the following Equation 1.

$$f = a.b + a.c.e + d.e + b.c.d \tag{1}$$

2.1.3 Non-Series-Parallel Networks

A transistor network is NSP if, and only if, there is at least one bridge connection between its components [4].

Figure 1 (b) illustrates a NSP network that implements Equation 1. The transistors pointed by dashed lines are in bridge configuration. It is notable the optimization in terms of number of transistors to represent the Equation 1 obtained by this arrangement.

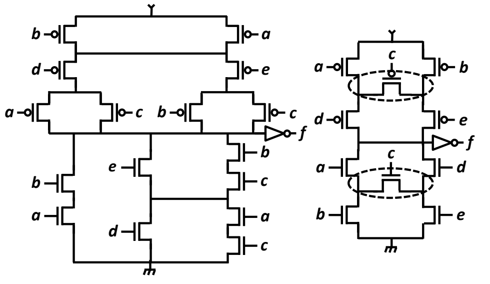

Figure 1. Networks implementing Equation 1. (a) Series-parallel network. (b) Non-series-parallel network.

2.2 Graph Theory

2.2.1 Planarity

A graph is said to be embeddable in the plane, or planar, if it can be drawn in the plane so that its edges intersect only at their ends. For logic networks, only NSP arrangements can present non-planarity topology.

There are several algorithms to perform a planarity check in a graph, but, basically, with different time complexity. One of these methodologies is the Boyer-Myrvold algorithm [7], a state-of-art way to verify planarity through the edge addition technique.

2.2.2 Duality

It is a property directly linked to the planarity. Considering a planar graph *G*, a dual graph *G** of it is constructed as follow: firstly, for every face of *G*, a vertex is created in *G**. Then, edges connecting the vertices in *G** are built by crossing each edge in *G*, in which the edge label is preserved. Figure 2 illustrates a graph *G* (edges represented by a continuous line and black vertices) and its dual graph *G** (dashed edges and white vertices).

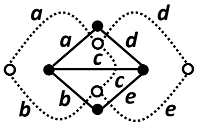

Figure 2. A graph and its dual.

2.3 Euler Path and Euler Circuit

An Euler path is defined as the path that travels the graph through every edge exactly once. In case of same starting and ending vertices, this path is defined as Euler circuit.

Euler's theorem [7] describes the conditions necessary to a graph be Eulerian, i.e. whether it admits Euler paths or Euler circuits. It can be summarized by the following corollary: if there are three or more odd degree vertices in the graph, then it is not Eulerian.

In physical synthesis scenario, Euler paths are extremely useful: if a plan of a logic network has an Eulerian path, then the diffusion area of this plan will be totally shared by its transistors, i.e. there is no gap in the diffusion line.

3. PROBLEM DESCRIPTION

Despite the notable optimizations concerning the number of transistors, NSP introduces a new challenge for most of placement algorithms. This section will review some classical methods for placement and identify some of the problematic cases occurred in this for NSP cells.

The first formulation of placement proposed was Uehara et. al. [5]. Introducing the concept of diffusion sharing between the transistors and the linear-matrix layout style, the algorithm consists to find an Euler path in a representative graph of the arrangement. This way, is possible to reduce the layout size and avoid diffusion gaps.

From that first paper, many others methodologies arises aiming to cover different aspects of the logic network (layout styles, technologies, topologies, etc.) and to decrease the time and complexity of the placement computing. Maziasz et. at. [8] introduces a methodology aiming the minimization of the height of the cell considering the routing channels necessary for routing procedure. The CLIP EAD tool [9] proposes a method to place transistors based on Integer Linear Programming (ILP), which provides a significant optimization for placement of 2D layout style and for networks with a larger number of transistors. Lib EAD tool [10] introduces the idea of clustering where the PMOS and NMOS transistors directly connected are enclosed in subsets of the network and each subset is placed individually. Another EAD tool called XPRESS [11] also uses this clustering technique to place transistors, aiming the decrease of diffusion gaps and height of a cell for linear-matrix layout style. Cellerity [12] was one of the first to use a Simulated Annealing technique to place transistors, allowing the placement for 2D layouts also. Finally, Iizuka et. al. proposes two algorithms of placement which are considered the state-of-art nowadays [13,14]. Both techniques use a SAT solving procedure to get the better placement of the transistors in the cell, providing optimized solutions in a short computing time compared to others classic methodologies and EDA tools mentioned before.

As we can see, the algorithms described above try to cover different aspects of the cell, and each one of them is concerned about a particular set of constraints. This paper discusses those that are related to the transistors arrangement topology, such as planarity, duality and the Eulerian properties. Table 1 summarizes these aspects for the algorithms mentioned before.

To comprehend the relation between these topological aspects and the NSP networks, let us consider the network generated by a state-of-art algorithm illustrated in Figure 3. As we can see, this NSP solution not only is non-planar but also the number of switches between the plans is different (10 and 8 switches for PUP and PDW, respectively). This way, methodologies that have constraints of planarity, duality, number of transistors or those that only accept series-parallel networks do not offer support for NSP arrangements. It is important to notice that most of EDA tools do not support this topology, as Table 1 shows.

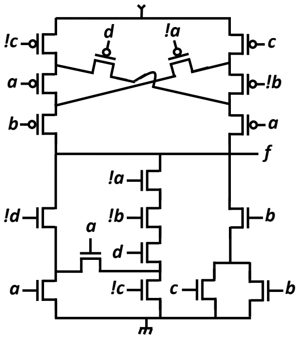

Figure 3. A non-series-parallel and non-planar network.

This paper aims to evaluate the impacts of these topological aspects considering common-used libraries, proposing a methodology to create NSP transistor arrangements based on a state-of-art algorithm to generate optimized logic networks.

Table 1. Constraints in algorithms of placement

Constraints	[5]	[8]	[9]	[10]	[11]	[12]	[13]	[14]
Series-parallel connections	Yes	Yes	Yes	No	No	No	Yes	No
Planarity	Yes	Yes	Yes	No	No	No	Yes	No
Duality	Yes	Yes	Yes	No	No	No	Yes	No
Equal number of switches PUP and PDW	Yes	Yes	Yes	No	No	No	Yes	No

4. PROPOSED METHODOLOGY

To evaluate topological aspects of logic networks, we have to generate several NSP arrangements for different Boolean functions. For the past decades, many alternatives to solve the minimization and factorization problems are proposed aiming to optimize logic networks [2,3,4]. Recently, graph-based approach algorithms have demonstrated that they can be an efficient way to build optimized arrangements, becoming a valuable alternative to the modern logic design. Kernel Finder (KF) [1] is a state-of-art methodology that uses NSP networks to perform this optimized transistor network generation. For this reason, we have used the KF tool to produce transistor networks to perform our analysis.

There are many ways to construct a logic network, even using a tool as KF. To generate the minimum arrangement produced by KF in terms of number of switches, we proposed the Algorithm 1.

Algorithm 1 Pseudocode for Network Construction via KF

```
1:  buildNetwork ( F )
2:     PUP ← kernelFinder ( F )
3:     PDW ← kernelFinder ( !F )
4:     if ( isPlanar ( PUP ) and isPlanar ( PDW ) ) then
5:         if ( PDW.n < PUP.n ) then
6:             PUP ← dual ( PDW )
7:         else then
8:             PDW ← dual ( PUP )
9:         end if
10:    end if
11:    return PUP ∪ PDW
12: end
```

The algorithm starts building the two logic plans *PDW* and *PUP* in lines 2 and 3 respectively. To perform this, KF computes the plans generated by *F* (direct Boolean function) and *!F* (complementary function). There are two possible situations: (1) *PDW* and *PUP* are both planar or (2) at least one of them is non-planar. Line 4 performs a planarity check for each plan through a Boyer-Myrvold method [6]. In case of a planar arrangement, lines 5 to 10 are executed. Line 5 investigates which plan has less transistors in the arrangement, considering that KF can generate plans with different number of switches for *F* and *!F*. In case of more transistors in *PDW*, then *PUP* receives the dual graph of *PDW* (line 6) and vice versa (line 8). As mentioned before, the other situation is when there is a non-planar plan. In this case, it is not possible to generate a dual graph. Finally, line 11 returns the full network composed by the *PUP* and *PDW* plans.

As we can see, the planarity and duality tests are already executed in the implementation flow described for Algorithm 1. To complete the test stage, we need to verify the number of transistors in PUP and PDW, besides the Eulerian aspect of both plans.

Performing these network implementation and test flows we generate, via KF, an optimized (NSP) transistor arrangement derived from a Boolean expression

5. RESULTS

The experiments presented in this section were made under a set of well-known benchmarks presented in most of works referenced in this paper. The catalog are composed by the NSP handmade cells [15] (53 functions), Nimomya's catalog [16] (402), the 4-input P-class [17] (3982) and the eleven variable and ninety-nine literals expression discussed as study case in [1] (named from here as 11-input), which give us a total of 4438 Boolean functions.

5.1 Planarity and Duality

Table 2 summarizes the planarity and duality results. Notice that, if the network has at least one planar plan, then it is possible to construct the dual network from that plan.

Table 2. Planarity and duality

Benchmark	Both planar plans / Dual (#)	One planar plan / Dual (#)	Both non-planar / Non-dual (#)
[15]	53	0	0
11-input	1	0	0
[16]	340	14	49
[17]	3183	564	235

As we can see, for the functions [15] and for 11-input, we did not have any occurrence of non-planarity or non-duality. However, in [16] and [17], 12.43% and 20.07% of networks are not fully (both plans) planar, respectively. As discussed before, in those cases is not possible to generate dual plans, which increase the placement complexity and even impossibilities some methodologies to deal with this network.

5.2 Difference in the Number of Transistors

Considering the Algorithm 1, we aim to optimize the number of transistors in the design procedure of the full network. Table 3 enumerates how many functions (relatively to the entire benchmark) have different number of transistors (affecting the placement methods illustrated in Table 1) considering the minimum cell that can be created by KF.

Table 3. Functions with different number of transistors in pull-up and pull-down plans

Set	Func. (#)	Func. (%)	Sum of switches differences (#)	Average (#)	Std. dev. (#)
[15]	0	0	0	0	0
11-inp.	0	0	0	0	0
[16]	5	1.24	6	1.2	0.44
[17]	421	10.57	565	1.34	0.62

Table 3 shows that, in case of planarity in both plans, [15] and 11-input, the network with minimum number of switches generated via KF has the same number of transistor in PUP and PDW. For

[16] and [17], 1.24% and 10.57% of functions have difference in the transistor count between PUP and PDW, with an average of 1.2 and 1.34 more transistors per plan and 0.44 and 0.62 of standard deviation, respectively.

5.3 Eulerian Networks

The last aspect observed in this paper is the Eulerian characteristic. It describes how many networks will produce layouts with at least one diffusion gap, as reviewed in the section 2. Table 4 describes this characteristic.

Table 4. Eulerian networks

Set	Eulerian (#)	Eulerian (%)	Non-Eulerian (#)	Non-Eulerian (%)
[15]	38	71.70	15	28.30
11-input	1	100	0	0
[16]	211	52.49	191	47.51
[17]	1849	46.43	2133	53.57

Table 4 shows that 28.30%, 47.51% and 57.57% of the cells implementing the set of functions [15], [16] and [17], respectively, will have a gap, increasing the area. In addition, for these networks, the placement algorithms need to implement a diffusion break procedure, since this is not trivial.

6. CONCLUSIONS AND FUTURE WORKS

The paper presents some topological aspects of NSP networks. This was made by proposing an implementation and test methodology that creates optimized switch networks via a state-of-art algorithm, which allows the verification of how affected the placement algorithms with topological constraints will be dealing with this kind of arrangement.

The results show that, for the sets [15-17] adding the 11-input, in [16] and [17] catalogs, 12.43% and 20.07% of the networks, respectively, are fully non-planar. It demonstrates that a relevant set of NSP functions have singular topological aspects, which has directly impact of some classical placement methodologies and increase the complexity of this procedure. Also, is presented the number of functions with different switches count between PUP and PDW considering non-planar solutions, 1.24% and 10.57% for [16] and [17], respectively. Finally, our results demonstrate that a large number of cells will have gap in the diffusion area, 71.70%, 52.49% and 46.43% for [15], [16] and [17] catalogs, respectively.

As future works we intent to investigate the classical and state-of-art placement and routing methodologies applied to NSP networks aiming the development of a layout estimator and an EDA tool for arrangements without topological constraints.

7. ACKNOWLEDGMENTS

Research partially supported by Brazilian funding agencies CAPES, CNPq and FAPERGS.

8. REFERENCES

[1] Possani, V., Callegaro, V., Reis, A., Ribas, R., Marques, F., Da Rosa, L. . 2015. Graph-based transistor network generation method for supergate design. In *Very Large Scale Integration* (VLSI) *Systems*, IEEE Transactions on, 99 (Mar. 2015).

[2] Kagaris, D., et al. . 2007. A methodology for transistor-efficient supergate design. In *Very Large Scale Integration* (VLSI) Systems, IEEE Transactions on 15, 4 (Apr. 2007), 488-492.

[3] Martins, .M., et al. 2010. Boolean factoring with multi-objective goals. In *Computer Design* (ICCD), IEEE International Conference on, (Oct. 2010), 229-234.

[4] Da Rosa, L., et al. 2009. Switch level optimization of digital CMOS gate networks. In *Quality of Electronic Design* (ISQED), (Mar. 2009), 324-329.

[5] Uehara, T., et al. 2007. Optimal layout of CMOS functional arrays. In. *Computers*, IEEE Transactions on, 30, 5 (2007), 305-312. *Modeling and Simulation Design*. AK Peters Ltd., Natick, MA.

[6] Boyer, J., et al. 2004. On the cutting edge: simplified O(n) planarity by edge addition. In *Journal of Graph Algorithms and Applications*, 8, 3 (2004), 241-273.

[7] Roy, K. . 2007. Optimum Gate Ordering of CMOS Logic Gates Using Euler Path Approach: Some Insights and Explanations. In *Journal of Computing and Information Technology*, 15, 1 (2007), 85-92.

[8] Maziasz, R., et al . 1991. Exact Width And Height Minimization of CMOS Cells. In *28th Design Automation Conference* (Jun. 1991), 487-493.

[9] Grupta, A., et al. 2000. CLIP: integer-programming-based optimal layout synthesis of 2D CMOS cells. In *ACM Transactions on Design Automation of Electronic Systems*, 5, 3 (2000), 510-547.

[10] Hsieh, Y. . 1990. LiB: a cell layout generator. In *ACM/IEEE Conference on Design Automation* (1990), 474-479.

[11] Grupta, A., et al. 1996. XPRESS: A Cell Layout Generator with Integrated Transistor Folding. In *European Design and Test Conference* (Mar. 1996), 393-400.

[12] Guruswamy, M., et al. 1997. Cellerity: a fully automatic layout synthesis system for standard cell libraries. In *Design Automation Conference* (1997), 327-332.

[13] Iizuka, T., et al. 2004. High speed layout synthesis for minimum-width CMOS logic cells via Boolean satisfiability. In *Conference on Asia South Pacific Design Automation* (2004), 149-154.

[14] Iizuka, T., et al. . 2005. Exact Minimum-Width Transistor Placement for Dual and Non-dual CMOS cells. In *IEICE Transactions on Fundamentals of Electronics*, E88-A, 12 (2005), 3485-3491.

[15] Logics. Catalog of 53 Handmade Optimal Switch Networks. Logic Circuit Synthesis Labs. Retrieved March 25, 2015 from Federal University of Rio Grande do Sul: http://www.inf.ufrgs.br/logics/docman/53_NSP_Catalog.pdf.

[16] Harrison, M. . 1965. Introduction to Switching and Automata Theory. New York, NY, USA: McGraw-Hill (1965), 408-472.

[17] Correa, V., et al. . 2001. Classifying n-input Boolean Functions. In *7th Workshop IBERCHIP* (Mar. 2001), 58-66.