Topological Characteristics of Logic Networks Generated by a Graph-Based Methodology

Maicon Schneider Cardoso, Regis Zanandrea, Renato Souza de Souza, João Júnior da Silva Machado,
Leomar Soares da Rosa Junior, Felipe de Souza Marques
Group of Architecture and Integrated Circuits
Federal University of Pelotas
Pelotas, Rio Grande do Sul
{mscardoso, rzanandrea, rsdsouza, jjdsmachado, leomarjr, felipem}@inf.ufpel.edu.br

Abstract— **Graph-based methodologies for supergate design have gained relevance recently. Due to the non-series-parallel arrangements and the transistor sharing technique, these methodologies can deliver a network with fewer transistors, leading to an efficient logic design. However, through its optimization processes, these methods introduces some topology particularities in the logic network, which impacts directly in the layout. This paper presents a methodology to identify these aspects in order to guide the cell layout generation. The results were performed over a set of intensively used benchmarks and pointed that 67.69% of the investigated networks presents a planar topology, while 21.85% shows a different number of transistors between its logic plans and 93.73% of the physical cells will contain at least one gap in its diffusion areas.**

Keywords—graph-based methodology; non-series-parallel; planarity; duality; Kernel Finder

I. Introduction

Considering the number of transistors in a logic network that are needed to implement a Boolean function, graph-based minimization methodologies have gained relevance recently [1,2]. Through the path sharing technique, these methods could reach several optimizations in the transistors count when compared to other methodologies [3,4]. On the other hand, this graph-based approach can produce non-series-parallel arrangements, logic cells that contain switches in bridge connection. Due to the occurrence of this configuration, a singular non-planar topology can be created, leading for a lack of duality and a difference in the logic plans size, factors which affect directly the physical layout and some of the methods for the automatic cell generation.

This paper presents a flow to identify these topological characteristics on logic networks produced by a graph-based methodology. Starting from a Boolean function, we generate the optimized network in terms of number of transistors via the state-of-art tool Kernel Finder [1]. This procedure is followed by the next topologic tests: the planarity and duality check, the plans sizes check (which consists in the verification of the difference between the number of transistors that each logic plan contains) and the diffusion gap check. This way, it provides useful information about all those topology impacts in widely used Boolean functions, allowing us to evaluate the number of networks that will suffer with this aspects concerning its layouts.

The reminder of this paper is organized as follows. Section II reviews some important concepts about logic networks and graph theory which are needed to the fully comprehension of this paper; section III shows the problem description to motivate our proposition; section IV presents the proposed methodology; section V shows the obtained results; finally, section VI presents the conclusions of this paper and the proposed future works.

II. Preliminaries

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

A. Logic Networks

1) Logic Plans

A logic CMOS network is composed by two logic plans: the pull-up (PU) and pull-down(PD). The PU plan is composed by PMOS transistors and provides the connection between the output and V_{DD}. In the opposite, the PD is composed by NMOS transistors and its function is to connect the output to V_{SS}.

A logic network and its plans can be represented by graphs structures, where each transistor is an edge and each node is a vertex.

2) Series-Parallel Networks

A series-parallel (SP) transistor network is an arrangement such that its component switches have either series or parallel connections. Therefore, there is no bridge connection [4].

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

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

3) Non-Series-Parallel Networks

A transistor network is non-series-parallel (NSP) if, and only if, there is at least one bridge connection between its switches [4].

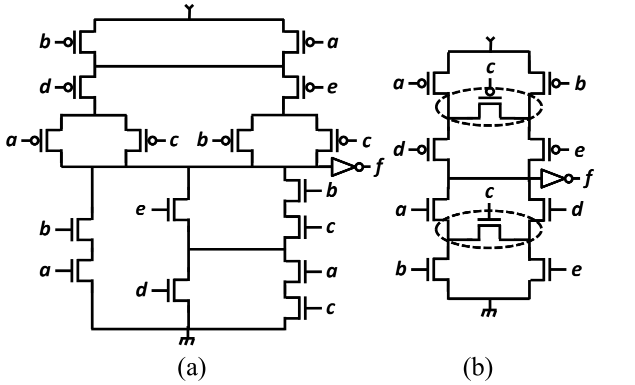

Fig. 1. Networks implementing Equation 1. (a) Series-parallel network. (b) Non-series-parallel network.

Fig. 1 (b) illustrates an NSP network that implements Equation 1. The two transistors pointed by dashed lines are in bridge connection. It is notable the optimization in terms of number of transistors to represent the Equation 1 obtained by this topology..

B. *Graph Theory*

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 [5]. For logic networks, only NSP arrangements can present a lack of planarity. Fig. 2 (a) shows the well-known graph K_5 and its non-planar connection represented by the dashed edge *b*.

There are several algorithms to perform a planarity check in a graph, differing mostly in time complexity. One of these methodologies is the Boyer-Myrvold algorithm [6], a state-of-art way to verify planarity through the edge addition technique.

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 [5]. Fig. 2 (b) illustrates a graph *G* (edges represented by a continuous line and black vertices) and its dual graph *G** (dashed edges and white vertices).

3) Euler Path and Euler Circuit

An Euler path is defined as the path that travels the graph through every edge exactly once. The Euler's theorem [7] describes the necessary conditions for a graph to be Eulerian, i.e. whether it admits Euler paths. 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.

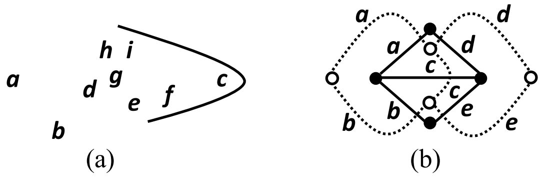

Fig. 2. (a) Non-planar graph. (b) A graph and its dual.

In physical synthesis context, 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 will be no gap in the diffusion area [7].

III. Problem Description

As discussed before, considering the number of switches in its logic networks, graph-based methodologies already proved themselves as a valuable alternative to the efficient logic design [1,2]. However, those methodologies produce arrangements with some particular aspects, which affects the layout and some of the automatic layout generation tools (its placement methods, specially).

The pivot of these particularities is the occurrence of non-series-parallel arrangements. Fig. 3 (a) shows an example of NSP cell in which it is notable its lack of planarity and duality, the difference in the number of transistors between its plans and the presence of Euler path only in the PD plan.

By definition, the lack of planarity is responsible for the occurrence of non-dual plans in the cell. Being this aspect a constraint for some placement methodologies [8,9], a non-planar cell can be a problem for an efficient automatic layout generation based on these methodologies, also increasing the routing procedure caused by the non-planar connection.

Considering the difference of the plans size (defined here as the difference in the number of switches between PU and PD plans), the impacts in the cell layout are illustrated in Fig. 3 (b). As we can see, the diffusion length of PU and PD plans are different. However, the cell length is geometrically limited by the plan with bigger diffusion length. Also, within this non-dual aspect, the complexity on the search for vertical matching of transistors, i.e. the maximum sharing of the transistors between the logic plans of the cell, is increased.

The last parameter observed in this paper is the presence of gaps in the layout. Gaps can be defined as areas in which two adjacent transistors cannot share their drain/sources with each other. It leads to an increase in the length of the cell and also in the routing procedure, since it is necessary to connect the disjoint areas via metal 1. As described before, Fig. 3 (a) shows a network that admits an Euler path only in its PD plan. In Fig. 3 (b) we can see the impacts in the layout, since only the PU diffusion presents a gap in its diffusion area (dashed zone).

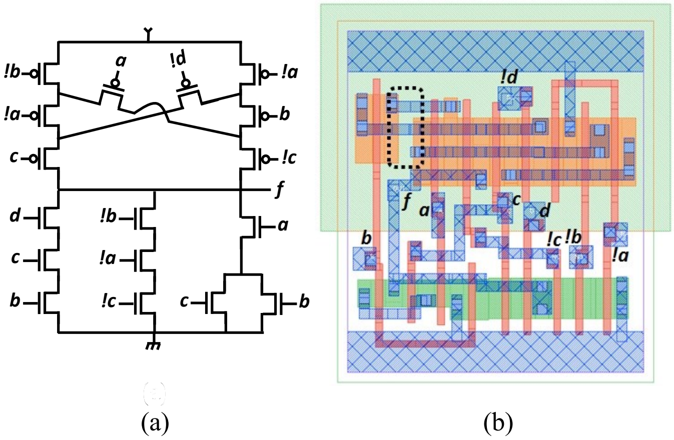

Fig. 3. (a) NSP non-dual network. (b) Resultant layout generated via [10].

This paper aims to investigate the occurrence of these aspects in networks created by a state-of-art methodology. We evaluate the percentage of the cells in a set of widely used benchmarks in which these topological characteristics can be observed. It allows us to take a deeper view in the impacts of a graph-based methodology for logic network optimization concerning the layout of this kind of arrangements.

IV. Proposed Methodology

To evaluate topological aspects of logic networks, we have to generate transistor arrangements for different Boolean functions. For the past decades, many alternatives to solve the minimization and factorization problems were proposed [2-4]. Recently, graph-based approaches have demonstrated that they can be an efficient way to build optimized logic arrangements. Kernel Finder (KF) [1] is the state-of-art tool for the optimized transistor network generation. It is a graph-based methodology that can creates NSP networks via the path sharing technique. For this reason, in this paper we have used the KF tool to provide all of our transistor networks.

However, even using a tool as KF, there are many ways to construct a logic network. 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: networkGeneration ( F )
 2:    PU ← kernelFinder ( F )
 3:    PD ← kernelFinder ( !F )
 4:    if ( isPlanar ( PU ) and isPlanar ( PD ) ) then
 5:          if ( PD.n < PU.n ) then
 6:                PU ← dual ( PD )
 7:          else then
 8:                PD ← dual ( PU )
 9:          end if
10:    end if
11:    return PU ∪ PD
12: end
```

The algorithm starts building the two logic plans *PD* and *PU* 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) *PD* and *PU* 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, the code snippet in lines 5-10 is executed. Line 5 investigates which plan has fewer transistors in the arrangement, considering that KF can generate plans with different number of switches for *F* and *!F*. In case of most transistors in *PD*, then *PU* will receive the dual graph of *PD* (line 6) and vice versa (line 8). As mentioned before, the other situation is when the network is composed by a non-planar plan. In this case, it is not possible to generate a dual graph. Finally, line 11 returns the complete network, which is composed by the PU and PD plans.

As we can see, the planarity and duality check are already executed in the implementation flow described for Algorithm 1. To complete all the tests, we need to verify the size of the plans (number of transistors in PU and PD), besides the occurrences of diffusion gaps in the cell (testing the Euler's constraints of the network, as described in section II.B.3).

V. Results

The experiments presented in this section were made under a set of well-known benchmarks presented in the literature. The set of functions are composed by the NSP handmade cells [11] (53 functions), the Nimomya's catalog [12] (402), the 4-input P-class [13] (3982), the NPN-class up to 5-input (NPN5) catalog (616125 functions) and the eleven variables and ninety-nine literals expression discussed as study case in [1] (named from here as 11-input) (1 function), which give us a total of 620563 Boolean functions.

A. Planarity and Duality

Table I summarizes the planarity and duality results. Notice that, if the network has both planar plans, then, by Algorithm 1, the resultant arrangement will be dual.

TABLE I. Results of Planarity and Duality Checking

Benchmark	Planar/dual network (#)	One plan is non-planar (#)	Both plans are non-planar (#)
[11]	53	0	0
[12]	340	14	49
[13]	3183	564	235
NPN5	416359	155435	44331
11-input	1	0	0

As we can see, for the functions [11] and for 11-input, all the networks are planar and dual. For [12], [13] and NPN5 79.93%, 84.57% and 67.57% of the networks are planar and dual, respectively. In the opposite, considering both plans non-planar, the results for [12], [13] and NPN5 are 12.18%, 5.90% and 7.19%, respectively.

B. Difference in the Logic Plans Sizes

Table II enumerates how many functions (relative to the entire benchmark) have difference in its logic plans size, i.e. have difference in the number of transistors of PU and PD considering the cell created by KF through Algorithm 1.

TABLE II. Total of Functions with Differences in Plans Size

Benchmark	Functions (#)	Functions (%)
[11]	0	0
[12]	5	1.24
[13]	421	10.57
NPN5	135195	21.94
11-input	0	0

As expected, in case of planar/dual networks, [11] and 11-input functions, for instance, the generated network doesn't have any difference in the plans sizes. On the other hand, for [12], [13] and NPN5, 1.24%, 10.57% and 21.94% of the networks will present differences in the transistors count for PU and PD. Table III details this difference.

TABLE III. DETAILED PLANS SIZE DIFFERENCE

Benchmark	**Functions for each plan size difference (#)**				
	0	***1***	***2***	***3***	***4+***
[11]	53	0	0	0	0
[12]	397	4	1	0	0
[13]	3561	304	98	11	8
NPN5	481994	19270	11193	4896	98772
11-input	1	0	0	0	0

C. *Diffusion Area Gaps*

The last aspect observed in this paper is relative to the Euler's paths presented in the networks generated by KF. This characteristic describes how many solutions will contain at least one diffusion gap in its physical layout.

Table IV shows a gap occurrence of 28.30%, 47.51%, 57.57% and 94.02% for [11], [12], [13] and NPN5, respectively, while the 11-input layout can be generated without any gap.

TABLE IV. FUNCTIONS WITH GAPS IN THE LAYOUT

Benchmark	**Functions (#)**	**Functions (%)**
[11]	15	28.30
[12]	191	47.51
[13]	2133	53.57
NPN5	579319	94.02
11-input	0	0

VI. CONCLUSIONS AND FUTURE WORKS

This paper proposes a methodology to implement logic networks through a graph-based state-of-art tool, which allow us to take a deeper view in some of topological aspects introduced by this new network generation methodology. The tests were performed over a set of well-known extensive benchmarks and investigates three important aspects in physical design: the planarity and duality of the arrangement, the difference in the logic plans sizes (considering the number of switches) and the presence of gaps in the diffusion area of the cell.

The results obtained shows that, for a total of 620563 transistor arrangements, 67.69% of the networks are planar and dual, while 25.14% has just one planar plan, i.e. has lack of duality and 7.17% has two non-planar plans. Considering the difference in the plans sizes, a total of 21.85% of the cells tested showed a difference in the number of transistors between PU and PD. Finally, regarding to the gaps on the diffusion area, 93.73% of the cells will contain at least one gap in its layout.

As future works we intent to keep investigating the impacts in the layout caused by graph-based approaches for logic design. We aim to look at geometrical aspects of the cells, its area and wirelength, as well as its electrical behavior, such as power dissipation and delay.

ACKNOWLEDGMENT

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

REFERENCES

[1] V. Possani, V. Callegaro, A. Reis, R. Ribas, F. Marques and L. da Rosa, "Graph-based transistor network generation method for supergate design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 2, pp.692-705, Mar. 2015.

[2] D. Kagaris and T. Haniotakis, "A methodology for transistor-efficient supergate design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 4, pp.488-492, Apr. 2007.

[3] A. Martins, L. da Rosa, A. Rasmussen, R. Ribas and A. Reis, "Boolean factoring with multi-objective goals," in *Proc. IEEE International Conference on Computer Design (ICCD)*, pp.229-234, Oct. 2010.

[4] L. da Rosa, R. Schneider, R. Ribas and A. Reis, "Switch level optimization of digital CMOS gate networks," in *Proc. International Symposium on Quality of Electronic Design (ISQED)*, pp.324-329, Mar. 2009.

[5] H. Lih-Hsing and L. Cheng-Kuan, "Planar Graphs," in *Graph Theory and Interconnection Networks*, 1th ed. Boca Raton, FL: CRC Press, 2009, ch. 10, pp. 161-167.

[6] J. Boyer and W. Myrvold, "On the cutting edge: simplified O(n) planarity by edge addition," *Journal of Graph Algorithms and Applications*, vol. 8, no. 3, pp. 241-273, 2004.

[7] K. Roy, "Optimun Gate Ordering of CMOS Logic Gates Using Euler Path Approach: Some Insights and Explanations," *Journal of Computing and Information Technology*, vol. 15, no. 1, pp. 85-92, 2007.

[8] A. Gupta, J. P. Hayes, "CLIP: integer-programming-based optimal layout synthesis of 2D CMOS cells," *ACM Transactions on Design Automation of Electronic Systems*, vol. 5, no. 3, pp. 510-547, 2000.

[9] T. Iizuka, M. Ikeda and K. Asada, "High speed layout synthesis for minimum-width CMOS logic cells via Boolean satisfiability," *Conference on Asia South Pacific Design Automation*, pp. 149-154, Jan. 2004.

[10] A. Ziesemer, R. Reis, M. Moreira, M. Arendt and N. Calazans, "Automatic layout synthesis with ASTRAN applied to asynchronous cells," in *Proc. IEEE Latin American Symposium on Circuits and Systems*, pp.1-4, Feb. 2014.

[11] Federal Univ. Rio Grande do Sul, Logics Lab. (2012, Oct.). *Catalog of 53 Handmade Optimum Switch Networks* [Online]. Available: http://www.inf.ufrgs.br/logics/docman/53_NSP_Catalog.pdf.

[12] M. Harrison, "Appendix 5: Table of Minimum Switching Circuits," in *Introduction to Switching and Automata Theory*, New York, NY: McGraw-Hill, 1965, pp. 408-472.

[13] V. P. Correia and A. Reis. "Classifying n-input Boolean Functions," *7th Workshop IBERCHIP*, pp. 58-66, Mar. 2001.