Improving Transistor Folding Technique in ASTRAN CAD Tool

Gustavo H. Smaniotto[1], Matheus T. Moreira[3], Adriel M. Ziesemer Jr.[2],
Felipe S. Marques[1], Leomar da Rosa Jr.[1]
Universidade Federal de Pelotas[1], Instituto Federal Sul-rio-grandense[2],
Pontifícia Universidade Católica do Rio Grande do Sul[3]
{ghsmaniotto, felipem, leomarjr}@inf.ufpel.edu.br[1],
adriel.ziesemer@canoas.ifrs.edu.br[2], matheus.moreira@acad.pucrs.br[3]

ABSTRACT
Traditional synthesis flow dedicated to design ASICs adopts the standard cell approach to generate VLSI circuits. As consequence, system-level layout is not fully optimized due to the restricted number of cells present in the library. To reduce this problem, ASTRAN, an open source automatic synthesis tool, was developed. This tool generates layouts with generic cell structures and obtains results with similar density compared to handcrafted cells. A key step on ASTRAN flow is the transistor folding, which consists in breaking the transistors that exceed the height limit defined in the project rules. However, this step is executed in ASTRAN only into single transistors. This paper optimizes this strategy introducing a new folding methodology that identifies all transistors stacks, applying the folding for each of these arrangements. The results obtained through this new folding technique show reductions in diffusion with contacts and cell area.

Categories and Subject Descriptors
B.7.2 [**Integrated Circuits**]: Design Aids – *Layout.*

General Terms
Design, Layout.

Keywords
Transistor Folding; Automatic Physical Synthesis; CAD Tool.

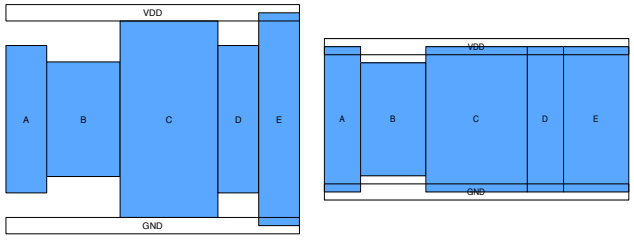
1. INTRODUCTION
The design of application specific integrated circuits (ASICs) is not a trivial task, even when considering some automation approach during the design process. To build an ASIC it is necessary to optimize the circuit from the logic to the physical synthesis step in order to achieve an efficient implementation. However, this is a time consuming approach, especially when designing large circuits.

Currently, one of the most adopted techniques to design digital circuits is based on the standard cell approach, where pre-characterized cells are used to build the final circuit. However, some researches point to the fact that the use of standard cells compromises fine grain optimization due to the reduced number of cells in the library. This leads to efficiency losses in terms of area, delay and power if comparing to equivalent circuits designed through full custom approach [8]. Also, the physical synthesis based on standard cells is more reliable since each cell in the library was tested and validated by experienced layout designers. However, for each technology it is needed to adapt the entire cell library to this new technology. This takes time, since it is necessary to characterize each cell from the library.

To address the problem caused of reduced cell number on library, an open source academic tool dedicated to automatic physical synthesis of VLSI circuits was proposed. This tool, called ASTRAN (Automatic Synthesis of Transistor Network) [1][9] was built to be used as an alternative to traditional synthesis flow based on standard cells libraries. ASTRAN generates cells on demand, making possible to optimize cells to specific problems as, for example, asynchronous circuits [6], leakage reduction, negative bias temperature instability (NBTI), among others.

During the physical synthesis, an important step is the folding transistor technique [2-5]. This technique is used to adjust the transistor width according to the maximum height of diffusion row, which is library specific. The idea is to break large transistors in small and equivalent ones respecting the technology rules and the defined height.

As illustrated in Figure 1(a), the tallest transistor in the network delimits the diffusion row height. To produce optimized cells it is necessary to apply the transistor folding in order to keep the cell height standardized. The idea is illustrated in Figure 1(b), where all transistors present different width, but same diffusion height. As a consequence, by applying the transistor folding, it is possible to reduce area overhead (white spaces in Figure 1(a)) paying a small price in terms of cell width.



(a) (b)

Figure 1. Cells height: (a) without transistor folding, (b) by applying transistor folding technique [10].

This paper presents a discussion regarding the implementation of transistor folding technique in the ASTRAN flow [10]. The new method addresses the folding issue regarding not only single transistors, but also series transistors arrangements in the cells.

2. ASTRAN
ASTRAN [9] is a tool that was developed to automate the physical synthesis of VLSI circuits. This tool automatically implements cells-level layout for several fabrication processes. It supports cells with different characteristics, such as: wide range of transistor network arrangement, unrestricted circuits structure (including non-complementary topologies), conditional design rules, transistor folding, redundant contact insertion, cells generation on demand (after transistor sizing).

The input file of ASTRAN is a transistor network description in a SPICE file format. Each input file contains the transistor list, the nets that connect them, the width and length of gates and the circuit in/out pins. The layout is designed according to the technology that is provided by the user. The output files of ASTRAN can be written in CIF and GDSII format and exported to LEF format. These files are used to describe 1-D layout style, which consist in two rows of transistors (PMOS and NMOS) with vertical gates. The layout style is detailed in [9] and it is shown in Figure 7. More information about the ASTRAN tool can be found in [10].

3. TRANSISTOR FOLDING

In cell-level layout design the transistor sizing is one of the most influential characteristics that impacts the cell area. Transistor folding is used to control the transistors width according to P and N diffusion sizes defined in project rules. This technique breaks large transistors (that exceed the maximum diffusion size) in many small transistors called legs. The number of legs is determined by dividing the transistor width by the diffusion size. The legs are chaining in parallel and the sum of theirs widths is equal to the original broken transistor width as shown in Figure 2. The folding issue is discussed by many authors [2-5]. These related works address different aspects as, for example, faster and efficient algorithms, optimal area cell, and 2-D cell layout with transistor folding.

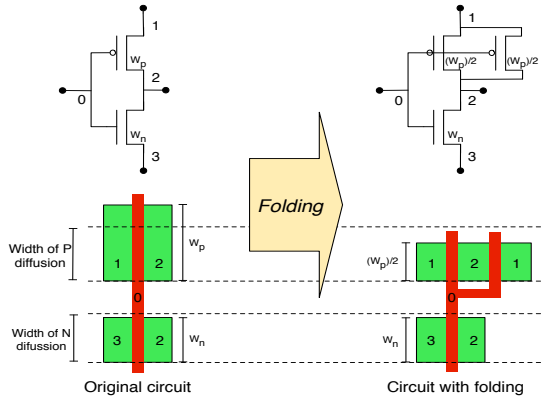


Figure 2. Example of transistor folding [10].

4. TRANSISTOR FOLDING IN ASTRAN

According [4], the folding problem can be classificated in four different forms. ASTRAN implements the static folding with dynamic placement [10]. In static folding the transistor is fold to respect the diffusion size (folding limit) and this occur before the tranistor placement. On the other hand, in the dynamic folding, for example, it is possible to break transistors that respect the folding limit, in many legs (respecting the minimum size of diffusion). In this way, the dynamic placement gives freedom to place transistors in any position of the cell layout in order to minimize cell area. This strategy was chosen because the cell height and the diffusion size are determined by the project rules, and because the transistor folding potentially affects the placement quality. The folding technique is applied after estimating the cell area and before the transistors placement, as shown in Figure 3.

The folding technique implemented in ASTRAN is applied only in single transistors. In other words, it is applied if the single transistor width exceeds the maximum diffusion sizes (P or N diffusion if the transistor is PMOS or NMOS). However, the layout can be even more optimized if the folding is applied in a series transistor arrangement. Figure 4(b) illustrates that. To make it possible, it is necessary to modify the previous approach, enabling the remove of internal/intermediate connections and reducing the number of contacts between gates. This also benefits the routing step, as less connections and contacts should be considered in that posterior stage.

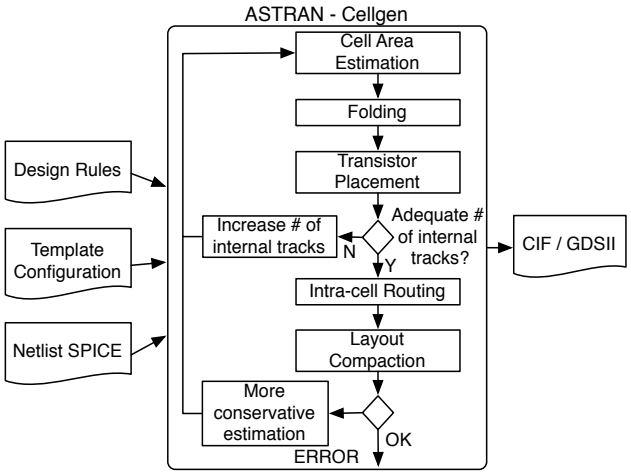
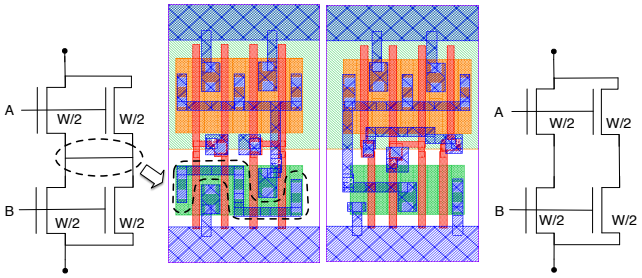


Figure 3. Layout synthesis flow [10].



(a) (b)

Figure 4. (a) Obtained layout from original ASTRAN folding and (b) obtained layout after applying the proposed transistor folding technique.

5. PROPOSED METHODOLOGY FOR TRANSISTOR FOLDING IN ASTRAN

A graph is the data structure adopted to store transistor networks in ASTRAN. The transistors network described in a SPICE format is translated to a graph structure. To keep the association among them, each edge in the graph represents a transistor and its vertices represent the transistor sources and drains terminals.

To identify the series arrangements in transitors network it is used the edge compression method. This technique is applied over a graph to minimize the edges and the vertices. It is used for different purposes. For example, it can be applied to reduce the number of transistors in a network during the logic synthesis step [7]. But, in this work, the edge compression concept is used to find series transistor arrangements.

The idea behind the edge compression consists into reduce the Boolean expression represented by edges in the graph by searching series and parallel arrangements. Every time that an arrangement is found, edges are merged and vertices are removed. For our purpose, the edge compression is used only to identify series arrangements in the graph. Therefore, it is not necessary to remove and add merged edges in the graph during the search.

The proposed solution is divided in two well-defined steps:

A. *Identifying series arrangements*

At this step the edge compression concept is used over the graph to search series transistor arrangements. During the search it a bi-dimensional matrix is used to store each found arrangement. The matrix lines represent each series and the matrix columns each transistor of the current series. On each line, the transistors are ordered by the position in the series arrangement. In the sequence, step 2 is started.

B. *Apply series transistor folding*

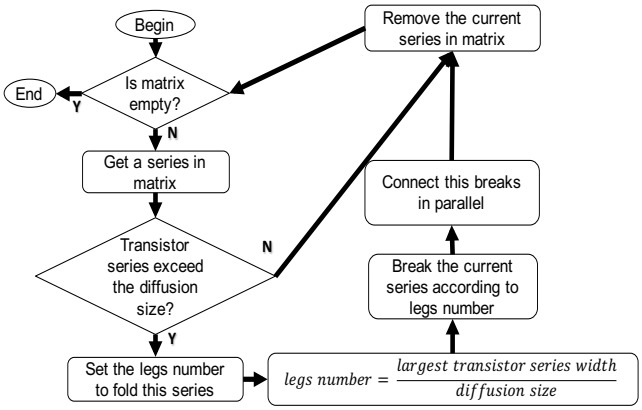


Figure 5. Proposed folding flow

The second step flow is illustrated in Figure 5. In this step the matrix is traversed in order to analyze each transistor series. While the matrix is not empty, it is selected a series in matrix, and it is analyzed if a transistor in this series exceeds the maximum diffusion size. If all transistors in this series exceeds the limit, then a series transistor folding needs to be applied, otherwise, remove the current series in the matrix. To apply the transistor folding it is necessary to set the legs number. This number is defined by the largest transistor width in the current series. To calculate the legs number, the largest transistor width is divided by the diffusion size.

In sequence, the current series are broken into the exact calculated number of legs. Basically, it will be created new series arrangements and these arrangements are connected in parallel with the original transistors series. However, the legs width sizes are calculated to each transistor on series. It is done by dividing the original transistor width by the number of legs previously obtained. Finally, the current series arrangement is removed of the matrix and the flow go to begin.

6. RESULTS

Experiments were performed in order to evaluate the proposed solution. During the experiments, some circuits used in [6] and [7] were considered. Firstly, the circuits were submitted to ASTRAN flow disregarding the proposed folding methodology. In the sequence, the same circuits were submitted to ASTRAN flow considering the proposed solution. Finally, a comparison between both results was done in terms of diffusion with contacts and cell area.

To perform the experiments it was adopted the 65nm technology node. The cell area was calculated using the software SightGL. For this, the layouts were generated in CIF format using ASTRAN and opened in the SightGL.

The number of diffusion with contacts was obtained considering both planes P and N. For example, Figure 7(b) presents 19 diffusions with contacts. On the other hand, Figure 7(d) presents 12.

Table 1 shows the number of diffusions with contacts for every evaluated cell, considering the two techniques. It is possible to see that proposed method minimize the number of diffusions with contacts. These optimizations occur because our method reduces connections between transistors. The proposed folding was able to reduce this cost for all generated cell.

As the contacts between gates were reduced, it was possible reduced cell area. Figure 6 shows percentage of area difference between the proposed folding technique and the current transistor folding. As can be seen, the new transistor folding method produces cells with optimized (reduced) area. The arithmetic average of the area reduction is 4.318%.

Current version of ASTRAN produces layouts with 3.7% area overhead if compared to a high quality hand-made standard-cells layout [1]. With the proposed folding it is possible to generate more optimized layouts than current ASTRAN solutions, and, consequently, deliver designs closer to hand-made ones in terms of area and contacts.

Figure 7 illustrates three examples of layout circuits generated by ASTRAN with and without the proposed folding.

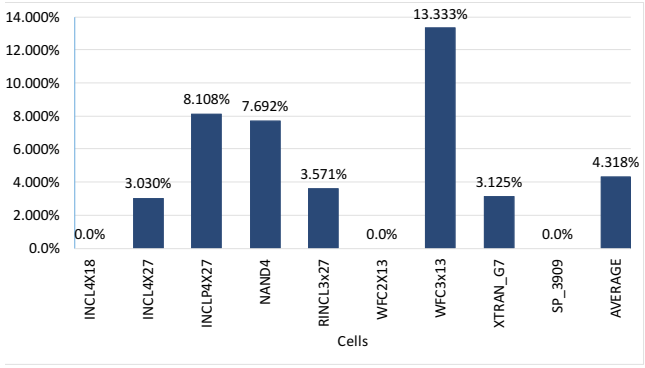


Figure 6. Percentage of area reduction when comparing the proposed folding technique to the old solution present in ASTRAN

7. CONCLUSION AND FUTURE WORKS

The preliminary results demonstrate that the proposed folding technique is able to deliver optimized results. The initial analyzes was performed over small cells. The gains in these cases are noticeable but somewhat modest. In future works, larger cells will be considered and larger gains are expected. The current ASTRAN version is not been distributed the proposed folding technique. Several tests will be performed before incorporating the solution to the tool. Also, it is intended to perform some electrical evaluations over the obtained circuits. These evaluations will consider different aspects as, for example, leakage, delay, total parasitics and energy per transistor.

8. ACKNOWLEDGMETNS

Research partially supported by Brazilian funding agencies CNPq and FAPERGS.

9. REFERENCES

[1] Astran – Automatic synthesis transistor networks. [Online]. Avaliable: https://github.com/aziesemer/astran

[2] Cheng, E. Y. C. and Sahni, S. 2001. A fast algorithm for transistor folding. In VLSI Design, vol. 12, 53-60.

Table 1. Number of diffusions with contacts when comparing proposed and current folding technique

Cells		INCL4x18	INCL4X27	INCLP4X27	NAND4	RINCL3X27	WFC2X13	WFC3X13	XTRAN_G7	SP_3909
Diffusion with Contacts	Current Folding	37	45	53	20	39	14	19	41	36
	Proposed Folding	29	31	35	13	24	12	12	36	34
Difference among methods		8	14	18	7	15	2	7	5	2

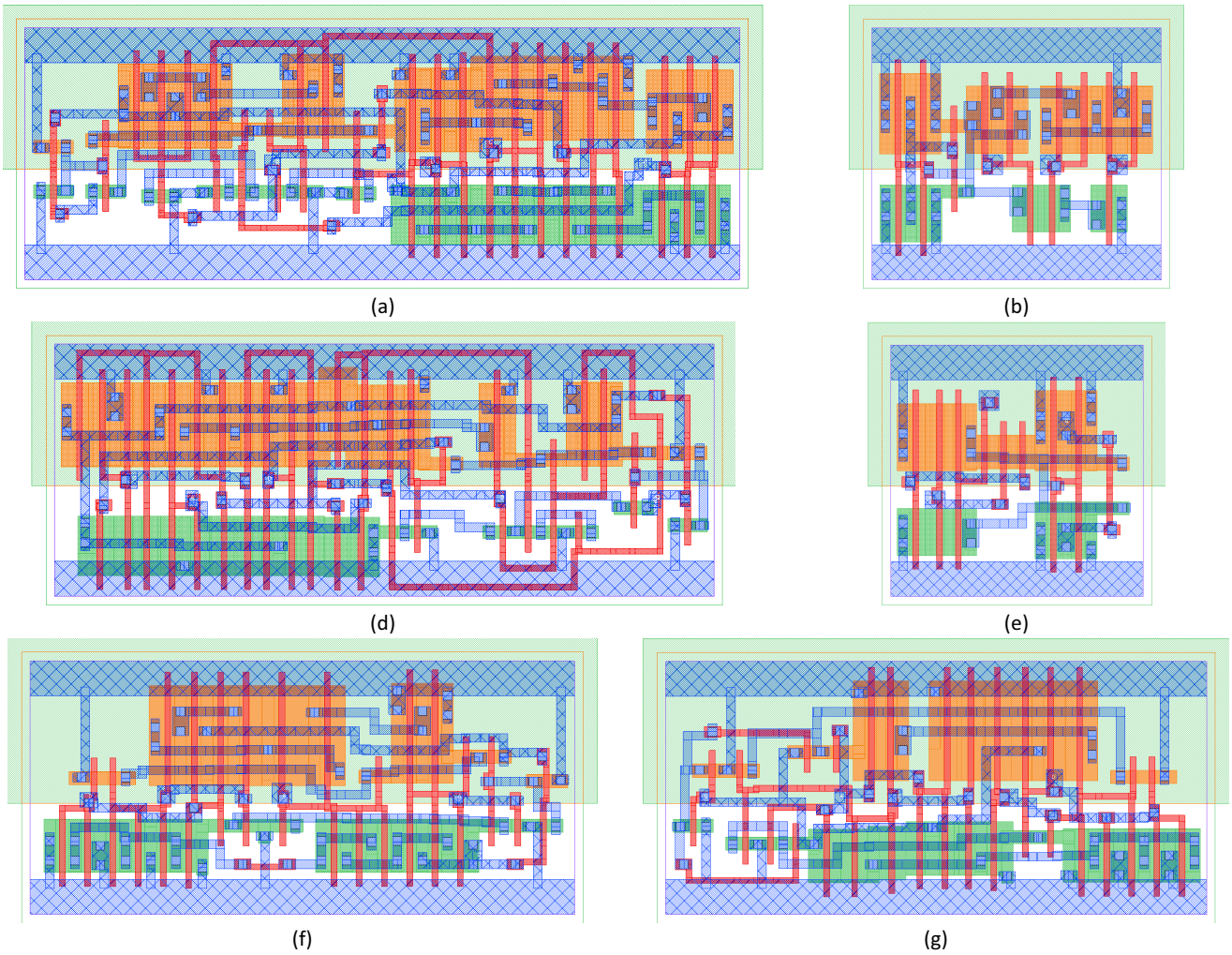


Figure 7. Cell layout differences of two circuits designed by ASTRAN. Circuit INCLP4X27 (a) with current folding and (c) with new folding. Circuit WFC3x13 (b) with current folding and (d) with new folding. Circuit INCL4X18 (e) with current folding and (f) with new folding.

[3] Cortadella, J. Area-Optimal Transistor Folding 1-D Gridded Cell Design. 2013. In IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems, vol. 32, (Nov. 2013).

[4] Gupta, A. and Hayes, J. P. 1998. Optimal 2-D cell layout with integrated transistor folding. In Proc. IEEE/ACM Int. Conf. Comput.-Aided Design, (Nov. 1998), 128-135.

[5] Kim, J. and Kang, S. M. An efficient transistor folding algorithm for row-based CMOS layout design. In Proc. ACM/IEEE Design Autom. Conf., (Jun. 1997), 456-459.

[6] Moreira, M. T., et al. 2014. Automated Synthesis of Cell Libraries for Asynchronous Circuits. In 27th Symposium on Integrated Circuits and Systems Design, 1-6.

[7] Possani, V., et al. 2013. Improving the methodology to build non-series-parallel transistor arrangements. In Symposium on Integrated Circuits and Systems Design, (Sep. 2013), 1-6.

[8] Roy, R., Bhattacharya, D. and Boppana, V. 2005. Transistor-Level optimization of digital designs with flex cells. In IEEE Trans. on Computers, vol. 38, (Feb. 2005), 53-61.

[9] Ziesemer A., et al. 2014. Automatic layout synthesis with ASTRAN applied to asynchronous cells. In Latin American Symposium on Circuits and System, (Feb. 2014), 1-4.

[10] Ziesemer Jr., A. M. 2014. Síntese Automática do Leioute de Redes de Trasistores. Doctoral Thesis. PPGMICRO/UFRGS, Porto Alegre, Brazil.