A Design Flow for Physical Synthesis of Digital Cells with ASTRAN

Adriel Ziesemer Jr. and Ricardo Reis
Universidade Federal do Rio Grande do Sul (UFRGS), PGMicro
Av. Bento Gonçalves 9500
Porto Alegre, RS, Brazil
{amziesemerj,reis}@inf.ufrgs.br

Matheus T. Moreira, Michel E. Arendt and Ney L. V. Calazans
PUCRS, PPGCC
Av. Ipiranga 6681
Porto Alegre, RS, Brazil
{matheus.moreira,ney.calazans}@pucrs.br

ABSTRACT

As the foundries update their advanced processes with new complex design rules and cell libraries grow in size and complexity, the cost of library development become increasingly higher. In this work we present the methodology used in ASTRAN to allow automatic layout generation of cell libraries for technologies down to 45nm from its transistor level netlist description in SPICE format. It supports non-complementary logic cells, allowing generation of any kind of transistor networks, and continuous transistor sizing. We describe our new generation flow which is currently being used to generate a library with more than 500 asynchronous cells in a 65nm process.

Categories and Subject Descriptors

B.7.2 [**INTEGRATED CIRCUITS**]: Design Aids-*Layout*

Keywords

EDA; CAD; cell synthesis; layout generation; standard-cell

1. INTRODUCTION

Cell library-based synthesis flows for ASICs is one of the most used methodologies in both industry and academia for design of VLSI circuits. It is known to be very reliable and predictable since the same cell library can be characterized and used in several different designs. However, the number of cells available in the library can limit the quality of the circuit, specially facing specific problems like: asynchronous circuits design, leakage reduction, SEU, NBTI, etc. Moreover, the layout of these cells is usually designed by hand, which also limits the adoption and development of promising technologies.

Layout Design Automation (LDA) is a challenge since the beginning of the silicon era. In technology nodes below 130nm, the number of new design rules increased considerably due to lithography issues. As a consequence, the complexity of layout generation is increasing, which demands new techniques for the generation of dense layouts. Previous works with layout migration [3,9] and cell synthesis [4,5,10] do not support layout generation for newer technology nodes bellow 130nm. Celltk [6] supports 90nm but presents a large area overhead compared to standard-cells. Nangate [2] has a cell synthesis tool capable of producing dense layouts but there is not detailed information regarding their flow.

In this work we present the cell synthesis flow used in ASTRAN, which is capable of generating dense cell layouts using simultaneous two-dimensional (2-D) layout compaction and support most of the conditional design rules that applies to recent technology nodes. We detail aspects of the layout generation flow, highlighting improvements over our previous work [11].

GLSVLSI'14, May 21–23, 2014, Houston, Texas, USA.
ACM 978-1-4503-2816-6/14/05.
http://dx.doi.org/10.1145/2591513.2591577.

2. OVERVIEW

ASTRAN [1] is an academic netlist-to-layout synthesis tool. It has a cell library creator module called CELLGEN [10] which is capable of synthesizing cell layouts in technologies down to 65nm (commercial) or 45nm (freePDK45). The tool generates layouts under a linear (1-D) layout style and supports unrestricted circuit structures, continuous transistor sizing, folding, poly and over-the-cell metal 1 routing, as well as rule relaxations for DFM. It is capable of generating dense layouts when compared to high quality standard-cells and improve productivity allowing generation and characterization of around 25 cells per day on average considering a single designer work.

The input of ASTRAN is a transistor level description of a circuit in SPICE format. Each circuit can be synthesized into a standard cell-level layout. The technology rules are set according to the values defined by the foundry. The cell topology (height, routing grid, wells/power rails position and other library specific aspects) are defined according to the target library and can be configured. The layout style used by ASTRAN is detailed in [11]. We managed to make it as flexible as possible in order to be able to generate hard to route cells and produce dense layouts.

3. GENERATION FLOW

The flow employed by ASTRAN is illustrated in Figure 1 and is discussed bellow:

Cell Area Estimation

ASTRAN first estimates the maximum transistor width and the routing resources in order to create a graph that can be

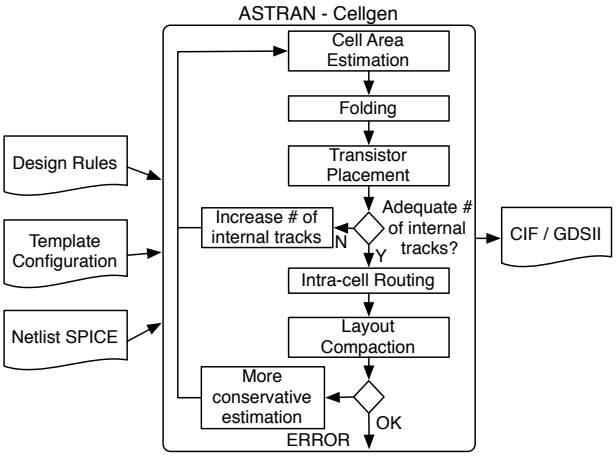

Figure 1: Cell generation flow employed by ASTRAN

compacted to the layout level afterwards. Because the tool can under or overestimate the number of horizontal tracks that fits in the cell height, we usually start with an optimistic approach (maximizing both the PMOS/NMOS diffusion regions and number of tracks) and then become more conservative (reducing the number of tracks and the size of the PMOS/NMOS regions) if the tool is not able to compact the layout. Such choices are parameterizable for each layout.

Folding

Given the diffusion rows height limits calculated during the previous step, transistors can require to be folded. ASTRAN folds transistors by modifying the cell netlist, creating new transistors in parallel, before the execution of the placement step.

Transistor Placement

The purpose of the transistor placement step in the 1-D layout style is to find out a transistor ordering for the PMOS and NMOS networks that leads to a better design. We use a multidimensional cost function to evaluate the placement quality as described in [11].

Intra-cell Routing

The abstract cell layout representation obtained in the placement step is translated into a routing graph. The graph is generated according to the number and position of the tracks calculated in the cell area estimation step, the placement result and the width of the transistors.

Layout Compaction

Layout compaction is the process of translating the abstract cell representation produced by the previous steps into the cell layout. ASTRAN compacts the layout in 2-D simultaneously using Mixed-Integer Linear Programming (MILP). It was achieved using binary variables to model mutually exclusive constraints.

4. INCREMENTAL GENERATION FLOW

Recently we implemented a new feature that allows ASTRAN to automatically iterate with different parameters in case it fails to complete the cell generation process, as shown in Figure 1. Our incremental generation flow starts with an optimistic approach, attributing the maximal height for the PMOS and NMOS regions of the cell in order to minimize the number of transistor folding. After the placement step is completed, it evaluates the channel density of the current solution and increases the number or internal tracks if it exceeds the capacity of the cell, re-executing the flow from the beginning. At the end, the layout compaction step is called to generate the actual layout of the cell. Models which the MILP solver proves to be infeasible are aborted and re-started using a more conservative approach: reducing the height of the diffusion areas of the cell and the number of horizontal tracks.

5. CONCLUSIONS

This work details optimizations in a previously existing design flow for synthesizing the layout of digital cells. The addition of the incremental generation flow allowed a better automation of the generation process which is currently being used in ASCEnD [7,8] to automate the layout generation of over 500 asynchronous cells with the use of scripts.

6. REFERENCES

[1] Astran - automatic synthesis of transistor networks. https://code.google.com/p/gme-ufrgs/.

[2] Nangate library creator. http://www.nangate.com/.

[3] D.-S. Fu, Y.-Z. Chaung, Y.-H. Lin, and Y.-L. Li. Topology-driven cell layout migration with collinear constraints. In *IEEE ICCD*, pages 439–444, 2009.

[4] M. Guruswamy et al. CELLERITY: A fully automatic layout synthesis system for standard cell libraries. In *ACM DAC*, pages 327–332, Anaheim, California, United States, 1997. New York: ACM.

[5] T. Iizuka. *Optimal Layout Synthesis of Standard Cells in Large Scale Integration*. PhD thesis, Department of Electronic Engineering, Graduate School of Engineering, The University of Tokyo, Tokyo, Japan, 2007.

[6] R. Karmazin, C. Otero, and R. Manohar. celltk: Automated layout for asynchronous circuits with nonstandard cells. In *IEEE ASYNC*, 2013.

[7] M. Moreira, B. Oliveira, J. Pontes, F. Moraes, and N. Calazans. Adapting a c-element design flow for low power. In *Electronics, Circuits and Systems (ICECS), 2011 18th IEEE International Conference on*, pages 45–48, Dec 2011.

[8] M. T. Moreira, B. Oliveira, J. Pontes, and N. Calazans. A 65nm standard cell set and flow dedicated to automated asynchronous circuits design. In *International SoC*, pages 99–103. USA: IEEE, 2011.

[9] H. Said, M. Dessouky, R. El-Adawi, H. Abbas, and H. Shahein. Analog layout retargeting. In H. E. Graeb, editor, *Analog Layout Synthesis*, pages 205–242. Springer US, 2011.

[10] A. Ziesemer, C. Lazzari, and R. Reis. Transistor level automatic layout generator for non-complementary cmos cells. In *VLSI-SoC*, pages 116 –121. IEEE, 2007.

[11] A. Ziesemer, R. Reis, M. T. Moreira, M. E. Arendt, and N. L. V. Calazans. Automatic layout synthesis with astran applied to asynchronous cells. In *LASCAS*, 2014.