GDS2trim: Physical Layout Manipulation Utility for continuous transistor sizing

Panagiotis Giannakou* ‡, Charalampos Antoniadis* ‡, Christos Kalonakis * , Dimos Dioudis*‡,
Georgios Stamoulis*, Panagiota Tsompanopoulou*‡, Nestor Evmorfopoulos*, John Moondanos*‡, Georgios Dimitriu*‡
*Department of Electrical and Computer Engineering
University of Thessaly
{panagian, haadonia, hrkalona, ntioudis, georges, yota, nestevmo, jmoondan, dimitriu}@inf.uth.gr
‡Institute for Research & Technology Thessaly

***Abstract*—Layout resizing has proven to be one of the trickiest parts of the continuous transistor sizing procedure, as it involves layout manipulation which is constrained by a large number of layout design rules that must be taken into consideration and many standard cell libraries are hand-drawn. In this paper, GDS2trim a layout manipulation tool for continuous transistor sizing is presented. GDS2trim, by making use of key features such as minimized input/output operations and parallelism, implements an automated layout processing methodology both for design and cell layouts allowing globalized circuit optimization.**

I. Introduction

Many efforts have been made in the field of continuous transistor sizing optimization problem as it closes the performance gap between a full-custom design circuit implementation and a standard cell based one [8]. A serious problem that rises during such optimization efforts is the layout resizing procedure. After defining the optimal transistor size for each cell used in the design, new standard cells must be implemented that incorporate the computed sizes. This is a rather complicated task, as each process design methodology is constrained by a large number of layout design rules. On top of that, many cells that belong to the cell libraries are custom-designed in order to meet the desired performance metrics.

The existing work on this matter, did not feature a fully automated manipulation [4] [6] [5] [7], as the already implemented approaches allowed only a few cells to be continuously resized, usually on the critical path of the circuit and required manual insertion of the resized standard cells into the cell library [1] [3] [2], making a large scale, fully automated circuit optimization not applicable.

By constraining transistor sizes to vary only between values that fit within the original standard cell footprint, GDS2trim provides a fully automated layout manipulation solution that can be implemented in a circuit-wide scale and provide fully implementable optimized layouts without human intervention.

In the rest of the paper, section 2 describes the architecture of the tool with description of each part that it consists of. In section 3, the layout resizing mechanism is explained and in section 4 the design layout handling is explained. The main implementation features of the tool are presented in section 5 and finally section 6 contains the conclusion of the presented work.

II. Architecture

In order the described utility to be fully automatic both the design layout and the standard cell layouts that compose the design must be analyzed and processed. In the following sub-sections the standard cell layout resizing module and the design layout handling module architecture are described.

A. Standard cell layout resizing module

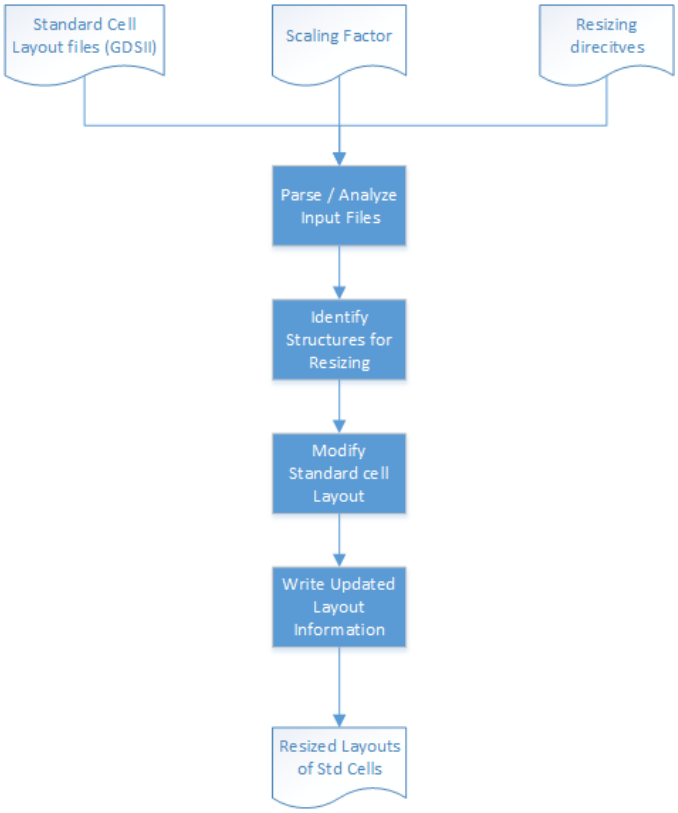

Fig. 1: Cell resizing module architecture

The cell layout resizing module is responsible for the layout resizing of every standard cell used by the design. As inputs to the module, a GDSII type layout file, a pre-computed scaling

factor for the specific cell and a resizing directives set for this cell are required. The module performs all the designated by the directives modifications on the polygon database of the cell and a new cell layout is produced that features the new transistor sizes.

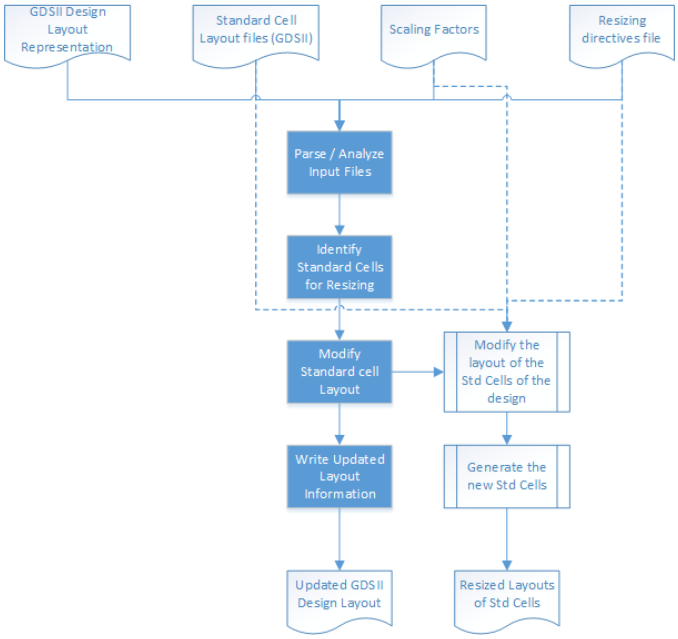

Fig. 2: Design layout handling module architecture

B. *Design layout handling module*

The design layout manipulation module requires as input the design layout GDSII file, the standard cell layout GDSII files, the scaling factor of each cell of the design and a resizing directives file. Then the module analyses the cells that the design is consisted and utilizes the cell layout resizing module to produce the resized cells of the design. Finally, both the resized design and the resized cell layouts are produced. In figure 2 the cell resizing module is depicted with the white boxes.

III. Standard Cell Layout Resizing

In order to succeed efficient, full-scale automation in layout resizing, only the shrinking of the transistor sizes is allowed. In that way all the performed transistor sizing modifications fit into the original standard cell footprint. Additionally, the resized standard cell layouts are generated by modifying the existing layouts, based on a set of resizing directives unique for every standard cell.

A. *Resizing Directives*

The resizing directives constitute a set of commands that layout resizing module executes in order to modify the cell layout. The set of directives is manually developed for every standard cell library, after studying the topology, the connectivity of each cell and taking into consideration the layout design rules of the library. The main features of the designed directives are described below:

- Each design directive is applied to a single object in the polygon database of the standard cell.
- Scaling of the p-type and n-type active areas is performed independently even though it is by the same scaling factor.
- Standard cell scaling is performed exclusively by shrinking the active areas and by the removal of contacts that may exist within the removed active area
- In cases where there are fingered transistors within the resized cell, scaling can occur either by shrinking the active area along the vertical axis or along the horizontal axis). Floating contacts are also removed.
- The incorporation of the layout design rules is implemented by the designer during design directive creation. Therefore, the layout manipulation utility does not perform any DRC check as it is assumed that the generated layout is correct by construction

B. *Layout resizing*

The cell characteristics that are mainly affected by the resizing procedure are the n-type and p-type diffusion areas and specific contacts that need to be removed. Both diffusion areas and contacts are represented as boundary objects in the cells layout database.

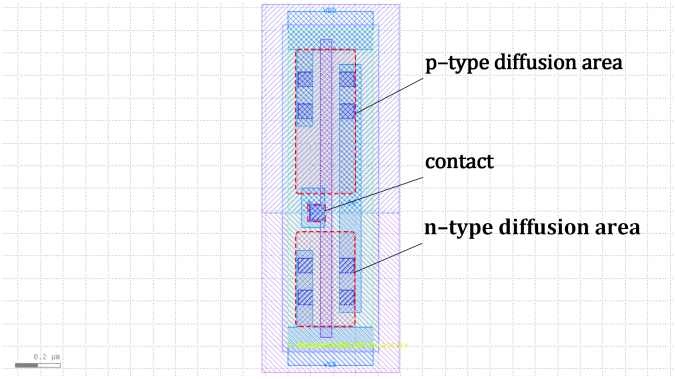

Fig. 3: Standard cell layout example

First the GDSII layout is analysed and the boundaries of the diffusion areas and contacts are isolated. Then the diffusion areas are scaled according to the corresponding directives and then contacts are removed where necessary. Every boundary representing a diffusion area is examined for possible resizing. The boundaries are resized via shrinking along the vertical axis in the possible ways: top-to-bottom, bottom to top and both as shown in Figure 4. Contact removal is performed by removing the corresponding boundary records from the layout file.

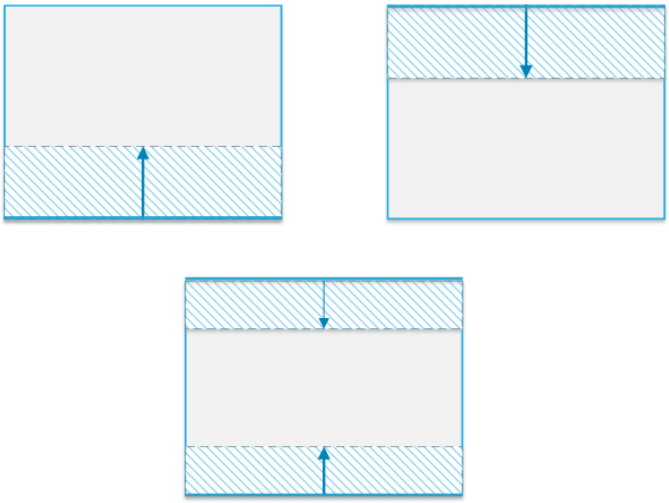

Fig. 4: Boundary resizing cases

IV. Design Layout Manipulation

The design layout manipulation module is the top level mechanism of the described tool. This module reads and analyses the main design layout alongside with other auxiliary files, namely the resizing directives file and the scale factors for every cell instance of the design. Afterwards the cell polygons and cell references are identified in order to be resized and the cell resizing module is utilized to perform the resizing of each cell instance. Every standard cell of the design is looked up for possible resizing according to the corresponding scale factor provided to the tool. The design layout manipulation module updates the standard cell references contained in the design layout in order to refer to the correct resized cell layout types and produces the final resized design layout.

Due to the fact that the original footprint of each cell remains unchanged, the polygon database of the design layout remains intact, a fact that is beneficial for the efficiency and execution time of the tool, as the original placement and metal layer routing remain unchanged after the cell resizing.

A general overview of the layout manipulation mechanism is shown in Figure 5

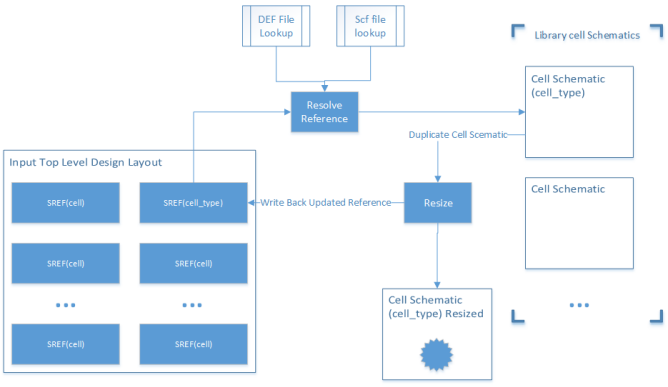

Fig. 5: Layout manipulation mechanism

V. Implementation Features

Alongside with the algorithmic key concepts that allow GDS2trim to carry out design-wide scale layout resizing optimizations and have a fully automated layout resizing mechanism, specific implementation features were utilized in order to further enhance the performance of the tool. These features are of crucial importance especially in the case of massive design layouts, which is a common phenomenon in today's ASIC large scale applications.

A. Minimized I/O

Reading and updating the design layout as well as producing new layouts may be costly in terms of input/output file operations, especially in cases of massive designs and standard cell libraries. This induces a significant overhead in the execution time of the tool. GDSII trim makes use of the file memory mapping technology in order to tackle that challenge. This technology allows the tool to interact minimally with storage units as it performs file operations only during the initial reading and the final writing of the file contents. All the processing and the necessary information updates are performed in the memory mapped file contents, which is faster by orders of magnitude compared to traditional I/O operations.

B. Parallelism

GDS2trim makes excellent use of task-level parallelism. Every cell resizing procedure of every cell instance is totally independent of every other. Thus, can be performed as a standalone task by a core of a modern multi-core architecture. This is of utmost importance, as large datasets can be processed simultaneously. The design layout manipulation module can be launched as a standalone task as well, as it has all the information necessary for the proper updating of the design layout at input time.

VI. Results

GDS2trim was tested on designs using the NANGATE 45nm OpenCell Library. This library contains a variety of widely used standard cell types in modern ASIC applications. The tool successfully analysed the input designs, identified the cell instances of each one and proceeded to resized cell layouts production. The produced cells are fully functional as the are produced in appliance with the library design rules and fully compatible with the state of the art schematic reading and editing tools.

In the example of Figure 6 the shrinking of both p- and n-type diffusion areas are visible. Moreover the floating contacts of the resized n-type diffusion area are removed.

A set of representative runtimes of the cell resizing procedure for various standard cells of the aforementioned library are presented in Table I. As it can be noticed, execution times defer among the same cell type resizing cases, as a different set of design directives is followed for certain scale factor ranges of each cell.

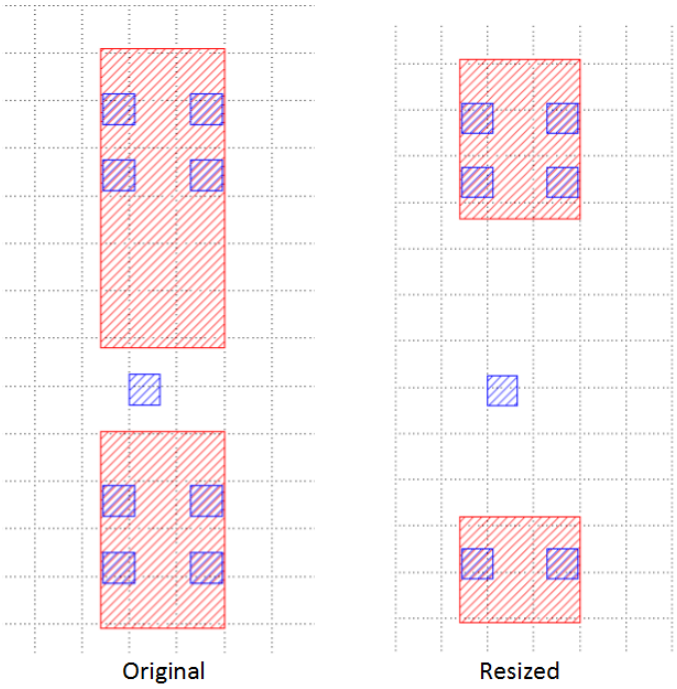

Fig. 6: Layout Resizing of cell INV_X1 to INV_X0.55 - Only diffusion and contact layers visible

TABLE I: Cell Generation Runtimes

Std Cell	Scale factor	Produced Cell	Execution Time (sec)
AND2_X1	0.79	AND2_X0.79	0.000166
AND2_X1	0.50	AND2_X0.50	0.000330
INV_X1	0.50	INV_X0.50	0.000293
INV_X1	0.35	INV_X0.35	0.000280
INV_X1	0.90	INV_X0.90	0.000309
OAI211_X1	0.55	OAI211_X0.55	0.000264
OAI211_X1	0.90	OAI211_X0.90	0.000465
AOI222_X1	0.37	AOI222_X0.37	0.000100
AOI222_X1	0.60	AOI222_X0.60	0.000213

VII. Conclusion

GDS2trim is a tool that successfully deals with the challenging problem of layout resizing in the field of continuous transistor sizing optimization.Through a minimal restriction of the possible transistor sizes the tool achieves to fully automate the layout resizing procedure, allowing globalized circuit optimization. Furthermore, by harnessing modern programming techniques and the power of parallel computation,*GDS2trim* can cope with massive input datasets and produce fast and correct results.

Acknowledgements

This work was supported by EU and the Greek State through ESPA 2007-2013, Action SYNERGASIA 2011, Project Code: 11SYN_5_719

References

[1] Cedabra. www.cadabra.com.

[2] Open-silicon. www.open-silicon.com.

[3] Sagantec. www.sagantec.com.

[4] D. Bhattacharya, V. Boppana, R. Murgai, and R. Roy. Process for automated generation of design-specific complex functional blocks to improve quality of synthesized digital integrated circuits in cmos using altering process, February 21 2006. US Patent 7,003,738.

[5] P. de Dood, B. Lee, and D. Albers. Optimization of circuit designs using a continuous spectrum of library cells, September 12 2006. US Patent 7,107,551.

[6] M. Guruswamy, D.W. Dulitz, R.L. Maziasz, S. Raman, V.K.R. Chiluvuri, and A. Berens. Automatic synthesis of standard cell layouts, November 16 1999. US Patent 5,984,510.

[7] R. Ogawa. Layout optimizing method for a semiconductor device, manufacturing method of a photomask, a manufacturing method for a semiconductor device, and computer program product, September 14 2006. US Patent App. 11/357,089.

[8] Etsuji Yoneno and Philippe Hurat. Power and performance optimization of cell-based designs with intelligent transistor sizing and cell creation. In *IEEE/DATC Electronic Design Processes Workshop*, 2001.