Design Optimization with Automated Cell Generation

Alexis Landrault, Nadine Azemard, Philippe Maurine, Michel Robert, Daniel Auvergne

Design Optimization with Automated Cell Generation

A. Landrault[1], N. Azémard[2], P. Maurine[2], M. Robert[2], D. Auvergne[2]

[1] LEAT, UMR 6071 CNRS, 254 rue A. Einstein, 06560 Sophia Antipolis, France.
[2] LIRMM, UMR 9928 CNRS, 161, rue Ada F-34392 Montpellier cedex 5 France.

Abstract. It is well recognized that designs based on automated standard cell flow have been found slower and larger in area than comparable designs manually generated or optimized. On the other hand it becomes necessary for designers to quickly prototype IP blocks in newly available processes. This paper describes an approach combining a performance optimization by path classification (POPS) tool with a transistor level layout synthesis tool (I^2P^2) dedicated to CMOS synchronous design fast generation. Validations are given on a 0.18 μm CMOS process by comparing standard cell approach to the proposed approach.

1 Introduction

Standard cell libraries have been successfully used for years as very efficient alternative to design very large circuits without transistor level verification. However with the increasing complexity of designs, this concept becomes less and less attractive and results in not so high quality designs. This is particularly true for what concerns the circuit speed and area [1]. It appears that most of the time standard cells are too generic and not well suited to the block being created. As a result the final design is not well optimized in terms of timing, power and area. As a solution to overcome the limitations of standard cell based design a library free technology mapping has been proposed [2]. It allows generating on the fly any digital cell at the size imposed by the timing constraint. The step to be solved is then the timing estimation and optimization of the design.

In this paper, we describe an investigation towards the possibility of combining a transistor level layout synthesis (I2P2 [3]) and a performance optimization by path classification (POPS [4]). The main contribution of this paper is to link automatic layout generation of digital circuits and path performance estimation and optimization to generate high quality designs, by directly evaluating the interconnection load from the generated layout. This is an alternative to solve the timing closure problem.

The paper is organized as follows. In section 2, we describe the standard cell based approaches. In section 3, we present the new flow based on the combination of a transistor level layout synthesis (and the associated software engine integrated in the I^2P^2 tool) with a delay/power performance optimization tool. In section 4, we describe the implementation of the interface between both tools. In section 5, we analyze the results obtained with this new "transistor level layout synthesis" that enables transistor level optimization before to conclude in section 6.

2   State of the Art: Design Methods Using Standard Cell Based Approach

The overall goal of the "Standard Cell" flows [5] is to generate the layout of the design from a behavioral description. The Boolean equations obtained after logical optimization are mapped onto a target pre-defined standard library. Afterwards the final layout is obtained after placement and routing of the footprint of the individual pre-characterized cells. All these standard cell based approaches involve a well-understood trade-off, and each cell of the library is speed and power characterized. Furthermore, the existing Industrial EDA tools are relatively well adapted to this approach.

   Meanwhile some negative aspects emerge.

- With new UDSM technologies where interconnects are of significant importance it is more difficult to predict the cell drive and to satisfy the timing constraints. Timing closure is one of the most critical steps to satisfy in circuit design. Various iterations are needed between each point-tool of the flow. This can seriously damage the flow convergence.

- In fact, it is well known that the quality of designs highly depend on the library to be used [6] and of the variety of functionalities and sizes for each primitive gate [7]. Most of the designers recognize that standard cell libraries contain "too generic" cells [8], and that significant improvements can be achieved just by resizing some existing cells (drive continuity) or by adding a few customized cells (design dependency) in the initial library.

   These drawbacks persist for new emerging approaches (such as "Liquid Library" or tools built around a unified data model) because the effectiveness of these alternative methodologies is highly dependent on the standard library development and process migration, which are costly.

3   Optimization at Transistor Level Using Layout Synthesis

We propose a standard cell independent based approach, working at transistor level using performance optimization by path classification tool. The idea beyond this approach is to realize the performance optimized final layout of a circuit, starting from its structural HDL description.

   The main motivations to link these two tools are twofold.

- Firstly to avoid to support the library and standard cell generation using this adaptive library concept based on layout synthesis. The dependency to a standard library is very costly in term of time (around 6 months to develop and characterize a library) and in term of human resources (around 5 people/month) while it only requires less than a day, using I^2P^2, to migrate to a new process by creating a new technology file.

- Secondly to optimize the performance using a deterministic method based on path classification at transistor level. With the combined use of POPS and I^2P^2, the transistors can be individually sized or re-sized continuously (no discrete size limitations) using a fast an accurate modeling of the performance of CMOS gates.

To realize the technology mapping, a logical synthesis commercial tool is supplied with a functionality set called the "virtual library". We associate to each function a virtual cell that can be considered as a set of connected transistors (only at symbolic level, no layout generated). The complete set of functionalities appearing in the virtual library is expected to imply less transistor count in the design. Actually, virtual cells appear as an interface between synthesis, place and route and the layout generation.

The proposed flow takes as input a behavioral description of the circuit function (VHDL or Verilog). It is composed of the two following steps.

- First a high-level logic synthesis and optimization, and a technology mapping are realized on a virtual library based approach.
- Followed by a transistor level layout synthesis step, at which the layout of the circuit is generated, using as inputs a set of constraints, a structural description of the design and the content of the virtual library. This includes placement, routing and physical layout generation. Timing analysis and optimization are realized using POPS.

The two developed tools (I^2P^2 and POPS) achieve this last step. As it is illustrated in Fig.1, the "transistor level layout synthesis" tool starts from a structural description of the circuit and generates the layout in CIF or GDSII format.

This is developed around three major steps:

- creation of the virtual cells from their functionality, resulting in an associated transistor network,
- placement and routing based on a physical estimation obtained from the targeted layout style,
- timing analysis and optimization realized with POPS.

These features enable more optimized and refined performance results regarding the initial design constraints. Information exchanged between these tools is made through the common data structure of I^2P^2 and the interface with POPS. The physical layout of each row is then automatically generated, with full respect of the technology rules.

The starting point of this flow is to provide a transistor net list with an optimal number of transistors corresponding to the required functionality, then to estimate and optimize the timing and power characteristics of the circuit before the final layout availability. The last point is to generate a dense layout of the net list, based on well-specified technology rules.

3.1   Layout Synthesis Tool

Layout Style
We have chosen, for the transistor-level generation, a layout style derived from the well-known "linear-matrix" [9] style, which is mainly characterized by the minimization of the space between the two N and P diffusion zones. The layout of the circuit is constructed by a sequence of NMOS and PMOS transistor rows and the routing is realized over the diffusion zones (gain in terms of area) and avoids the use of metal 2 to save the porosity of each row and to facilitate the routing step.

As described in Fig. 2, all the ports are placed at the center between N and P diffusions. This layout style has been chosen because it's perfectly adapted to software implementation (regular style with vertical placement of the poly grid) [10]. With
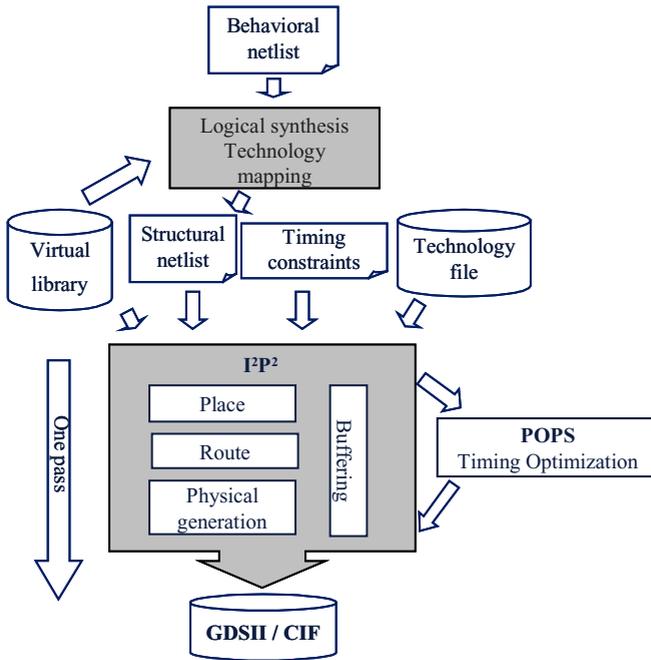
Fig.1. Flow description
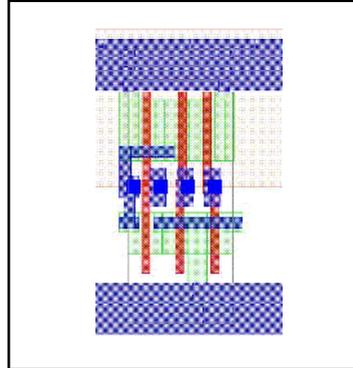
Fig. 2. Layout Style

respect to this style, a temporary layout can be very quickly generated for each virtual cell needed in the design. This layout used only for the prediction of the cell size and the port location will then be deleted. The final detailed layout of the design will be done only at the very end of the flow. Place, route and timing analysis tools will then use these accurate estimations of the width, the height and the port position during all the block generation steps.

Concept of "Virtual Library"

The proposed flow presented here is based on the possibility to work directly at transistor level. By using transistors generated (at symbolic level first to fill the data structure) on the fly instead of using a static set of pre-characterized cells, we can perform performance optimization by resizing transistor instead of using a set of pre-packaged functionalities. We have to keep in mind that, no layout is generated before the final generation. Layout will be generated only for cell size prediction (temporary and then deleted) and at the very end of the flow at row-level.

The set of cells constituting the "Virtual Library" is mostly composed of complex gates or flexible cells. As a consequence the number of available cells is virtually unlimited (in terms of logical functionality). The constraints are only fixed by the target technology and performance limitations (maximum number of serial transistors).

In addition, as the final layout is automatically generated, the driving capability of all these "virtual cells" can be continuously modified using the POPS tool (by updating the transistor net list), at the contrary of the discrete drive possibility offered in usual standard cell design.

Logic Optimization / Technology Mapping

"Virtual Libraries" are constituted of a large number of cells, compared to standard library, because the number of different logic functions available in CMOS technology are only limited by the maximum number of N and P transistors in series. For instance, for a maximum of four serial transistors (for each plan, N and P) we may dispose of a library with 3503 different functionalities with the added possibility of continuously sizing each logical function. This implies more flexibility to achieve the mapping step on a given circuit and results in less transistor numbers compared to the standard approach.

The virtual library also includes specific functionalities such as DFF or Muller cells. These functionalities are associated to Spice files, describing the transistor network.

3.2 Place and Route, Layout Generation

Place and route and layout generation tools implemented in I^2P^2 have been vastly presented in previous work [3]. In this section, we briefly summarize the strategy of these tools.

Virtual Cell Place and Route

Placement and routing steps are performed from the "Virtual Cell" structural net list representation using predictive performance analytical models.

- Firstly, a placement based on an iterative partitioning and global routing between each partition is performed [11]. Then cells are placed in each partition to allow rows creation. A global routing step is lastly done to generate virtual channels.
- Secondly, the symbolic view of each virtual cell (using the Euler's trail solution [12]) is created. Afterward the symbolic transistors rows are generated then optimized (flipping, merging) and routed ("inner Row" maze routing).

- Finally, all the remaining connections are routed by a virtual channel router using detail routing algorithms (constraint graphs and multi-layer maze router).

Layout Generation
The final layout generation consists in creating from the symbolic view of each row a constraint graph that enables to obtain a "compacted" layout in the "linear-matrix" style for any given technology rules. The main advantage of this procedure is to guarantee technology independence.

3.3   Timing Estimation and Optimization: POPS

Once the circuit transistor net list has been generated the essential steps of transistor sizing and timing analysis is performed. The efficiency of the proposed layout synthesis flow presented in Fig.1, and the ability to obtain an optimal circuit is conditioned by the following assumptions.
- The initial phase needs to produce a layout as optimal as possible.
- The modifications performed during the incremental processes need to be efficient and constructive, while impacting the minimal number of transistors, to allow a quick convergence.

Consequently, two new problems need to be addressed. Firstly, as there is no characterization involved in the Layout Synthesis flow, an accurate predictive model for speed and power is required.

For that we use a tool developed (POPS: Performance Optimization by Path Selection) for analyzing and optimizing the paths of a combinatorial circuit in submicronic processes. The following properties are targeted:
- path analysis and enumeration in a speed or power performance order, or in a speed/power trade off order
- accurate performance modeling (delay and power) considering submicronic effects [13],
- definition of local optimization protocols in order to reduce the problem complexity [14] by applying simple optimization criteria defined from explicit performance modeling.

The objective of this tool is to allow nearly ideal profiling of the speed/power distribution on the paths of a circuit. This ideally would impose that all the paths be at the targeted delay constraint. Speeding up or down the longest or shortest paths is obtained by operating on a restricted number of gates [14]. The selection on critical paths of poor drive capability or lightly loaded gates reduces the global optimization problem [15,16] to local optimizations allowing effective management of the circuit delay and power. Moreover minimizing the size of gates belonging to non-critical paths (shortest paths for example) is the natural alternative of power saving implementation techniques [17]. In this way, in a reasonable CPU time, it appears possible to manage speed or power on significant circuits.

Realistic delay evaluation based path evaluation and ordering gives facilities in circuit verification and optimization. This can be used for path classification in increasing or decreasing order of delays as well than for implementing power saving techniques or in trading speed for power. The application given in [4] shows clearly the

interest in controlling the path number for circuit optimization. It has been clearly demonstrated that if delay constraints can be satisfied with regular transistor sizing, local sizing on selected gates results in power minimized implementation. Considering the longest and shortest paths it has been clearly shown that circuit optimization can be obtained by sizing few paths of a circuit.

4   Implementation and Tools Interfacing

As previously discussed, in order to converge most efficiently during the optimization phase, the architecture of I^2P^2 is based on a unique data-structure done in C++. This prototype integrates plug and play facilities to ease the exchange of the different "engines" under development such as POPS that has also been developed in C++.

The interfacing between POPS and I^2P^2 tool is realized using the Spice format. Once the layout of a given circuit have been placed, routed and generated, I^2P^2 produces the Spice format back-annotation of the circuit.

This Spice description file is then used as input for the external performance optimization tool. After the optimization step, POPS gives as output a net list (Spice format) containing the updated sizes of the transistors as shown in Fig.3
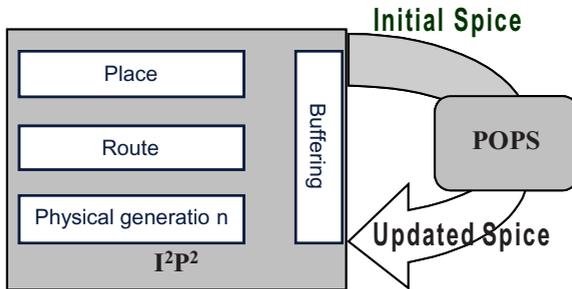

Fig. 3. POPS/ I^2P^2 interface

As illustrated, POPS modifies the initial Spice circuit net list, generated by I^2P^2 in updating the transistor sizes of the critical path. This new sizes of transistors are then modified in the data structure of the I^2P^2 software to enable a new final layout generation in CIF or GDSII format.

5   Results and Validation

Validations have been done in a 0.18μm CMOS technology. The comparison has been done with respect to the standard flow using an In-House library. We have run our prototype on ISCAS85 circuits. We proceed to various comparisons on transistor number (logic synthesis efficiency), transistor density (area) and especially timing/power performances (transistor width) to evaluate the proposed approach.
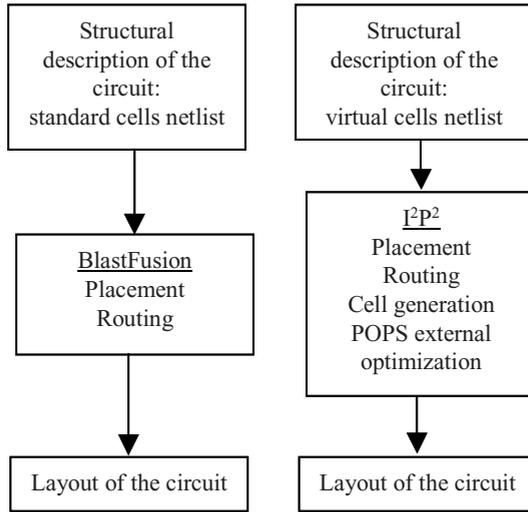
Fig. 4. Standard flow versus I^2P^2/POPS flow.

We validate the complete flow (POPS combined with I^2P^2) with respect to the standard approach, on a set of benchmarks of circuits of various complexities (up to fifteen thousand transistors). We target, for each approach, the best achievable timing constraint. We reduce on purpose the number of functionalities in the standard library as the extension from simple gates (such as the Inv-Nand-Nor) to complex combinatorial gates delay representation is obtained using serial transistor array reduction technique. The benchmark metric is the sum of the transistors width, the timing and the final area. The two design flows are compared in Fig.4.

The standard flow is realized with BlastFusion from Magma.

Transistor Number and Area Results

First, we compare the transistor number required to realize our benchmarks circuits and the associated area of the final layout. Results are summarized in Table 1.

Table 1. Transistor number and area comparison.

Circuit	Standard		I^2P^2/POPS	
	Transistor number	Area (μm^2)	Transistor number	Area (μm^2)
C17	26	242	26	216
C432	1270	51500	1258	45000
C880	1448	1147000	1438	120000

As shown, we obtain the same number of transistors, which is not surprising as we use the same set of functionalities to perform the mapping. The values obtained with the proposed flow (I2P2/POPS) results in a smaller area /power implementation.

Timing and Power Validation

By connecting the optimization (POPS) and the layout synthesis tools (I2P2), we target a performance improvement with respect to designs based on automated standard cell flow that have been found slower than comparable designs manually generated.

In Table 2 we compare the circuit performance in terms of timing and average transistor width. As illustrated the standard cell approach (Blastfusion) always result in a larger area implementation and a nearly equivalent delay. Next step, under development will be to compare the implementation area for an identical timing constraint. But considering the results given in Table 2 we can expect a much greater decrease in area using our automated synthesis tool.

Table 2. Timing and average transistor width comparison.

Circuit	Standard		I^2P^2/POPS	
	Sum of Transistor width (µm)	Timing (ns)	Average Transistor width(µm)	Timing (ns)
C17	40.4	0.178	30.4	0.176
C432	1950	1.18	1522	1.21
C880	3200	0.99	2450	0.98

This demonstrates that working at transistor level with the I^2P^2/POPS flow, give the possibility to resize in a deterministic way the transistors belonging to the cells to be sped-up (specific and continuous transistor resizing). This implies significant reduction of the global width of the transistors. As a direct effect, we can expect to obtain a substantial power dissipation reduction compared to standard cell approach.

Run Time Analysis and Migration Facilities

These results concerning area, timing and power are quite encouraging mostly if we consider the facility obtained in generating and migrating macro-blocks in very short time (3minutes for the c880 circuits). Moreover, apart from the time necessary to update the technology file of I2P2 and to update the calibration of POPS (few hours), the circuits have been migrated from one process to another in a very short time (few minutes). This reinforces one of the main advantages offered by this approach: the possibility to quickly prototype IP blocks.

6   Conclusion

In this paper we have presented an original alternative to the classical standard cell based flow integrating a layout synthesis tool combined to a delay/power performance optimization tool. In this "virtual cell library" methodology the physical generation is obtained at the transistor level, integrating the different steps of physical layout generation and performance estimation. Optimization step is realized with the POPS tool that associates selective sizing technique to circuit path classification based on an incremental technique This approach may give great facilities in quickly evaluating performances (delay/power thanks to fast and accurate modeling) and prototyping different flavors of IP Blocks by using the latest available technology.

We have shown that by combining a performance optimization by path classification tool to the transistor level synthesis tool, it was possible to develop a prototype able to handle simple blocks (thousand of transistors). Optimization of hundred thousand transistor blocks is under development in a 0.13μm process.

References

[1] W. J. Dally, A. Chang, "The role of custom design in ASIC chips", proc. 37th design automation Conference, pp. 643-647,2000.

[2] Reis, R. Reis, D. Auvergne, M. Robert, "The Library Free Technology Mapping Problem", IWLS, Vol. 2, pp. 7.1.1-7.1.5, 1997.

[3] Landrault, L. Pellier, A. Richard, C. Jay, M. Robert, D. Auvergne, "An I.P. Migration and Prototyping Strategy Using Transistor Level Synthesis", DCIS'03, XVIII Design of Circuits and Integrated Systems Conference, Ciudad Real , Espagne, 19-21 November 2003, pp266-271.

[4] N. Azemard, D. Auvergne, "POPS : A tool for delay/power performance optimization ", Journal of Systems Architecture, Elsevier, n°47, pp375-382, 2001.

[5] D. McMillen, M. Butts, R. Composano, D. Hill, T.W. Williams, "An Industrial View of Electronic Design Automation", IEEE Transactions on Computer, Vol. 19, No 12, December 2000, pp. 1428-1448.

[6] K. Keutzer, K Scott, "Improving Cell Library for synthesis", Proc. Of the International Workshop on Logic Synthesis, 1993.

[7] K. Keutzer, K. Kolwicz, M. Lega, "Impact of Library Size on the Quality of Automated Synthesis", ICCAD 1987, pp. 120-123.

[8] P. de Dood, "Approach makes most of synthesis, place and route - Liquid Cell ease the flow", EETimes, September 10, 2001, Issue: 1183.

[9] A.D.Lopez, H.S.Law, "A Dense Gate-Matrix Layout for MOS VLSI", IEEE Transactions on Electron Devices, Vol. ED-27, No. 8, August 1980, pp. 1671-1675.

[10] F.Moraes, R.Reis, L.Torres, M.Robert, D.Auvergne, "Pre-Layout Performance Prediction For Automatic Macro-Cell Synthesis", IEEE-ISCAS'96, Atlanta (USA), Mai 1996, pp. 814-817.

[11] M. Fiduccia, R. M. Mattheyses, "A linear-time heuristics for improving network partitions.", Proceedings of the 19th Design Automation Conference, pages 175-181, 1982.

[12] M.A. Riepe, K.A. Sakallah, "Transistor level micro-placement and routing for two dimensional digital VLSI cell synthesis", University of Michigan, Ann Arbor ISPD '99 Monterey CA USA.

[13] M. Maxfield, "Delay effects Rule in Deep-Submicron Ics", Electronic Design, pp.109-121,1995

[14] S.W.Cheng, H.C.Chen, D.H.C.Du,A.Lim, "the Role of Long and Short Paths in Circuit Performance Optimization", IEEE trans. On CAD of I.C. and Systems, vol. 13, n°7, pp.857-864, July1994.

[15] R. Murgai, "On the Global Fanout Optimization Problem", In IWLS, Granlibakken, USA, 1999.

[16] C.L. Berman, J.L. Carter, K.F. Day, "The Fanout Problem: From Theory to Practice", In C.L. Seitz editor, Advanced Research in VLSI: Proceedings of the 1989 Decennial Caltech Conferences, pp. 69-99, MIT Press, March 1989.

[17] H.C.Chen, D.H.C.Du and L.R.Liu, "Critical Path Selection for Performance Optimization", IEEE trans. On CAD of Integrated Circuits and Systems, vol. 12, n°2, pp. 185-195, February 1995.