Developing A Concurrent Methodology For Standard-Cell Library Generation

Donald G. Baltus Thomas Varga Robert C. Armstrong
LSI Logic Corporation

John Duh T.G. Matheson
Mentor Graphics Corporation

Abstract - This paper describes the development of a concurrent methodology for standard cell library generation. Use of a novel physical design automation method enables a high degree of concurrency among process, circuit, and layout development. In addition to reducing overall time-to-market, the new method allows optimization to occur simultaneously across the circuit, layout, and process design spaces. The result is libraries with improved density, circuit performance, and process yield.

Introduction

With each generation of IC fabrication technology comes the task of developing ASIC standard-cell libraries that are compatible with each new process. Optimizing a fabrication process and a library together is a complex undertaking, because changes in one affect the other. Process parameters determine the electrical and geometric constraints affecting the layout in the library, while layout architecture and performance requirements constrain the process variables. A variety of techniques have been developed to create optimized layout given a set of process parameters [1][2], and recently operations-research techniques have been applied to optimizing processes and libraries together. [3]

The optimization problem is often constrained by EDA trade-offs faced during creation of the layout. The central trade-off is the one between the level of automation and layout density. Rising time-to-market pressures and shortening fabrication process lifetimes push in one direction by demanding that layout generation be automated as much as possible. Automation shortens design times and also enables concurrent development of the fabrication process and the cell library by allowing layout libraries to automatically track the process when design rule changes require layout modification or shifting electrical models require transistor resizing. Concurrent development of the process and the library is essential if the two are to be optimized together. The high value of silicon real estate pulls in the opposite direction by demanding that the automation not sacrifice any layout density. Although automation brings many benefits, manual layout sets the standard for layout density. Manual layout is still widely employed in standard cell design because cell density remains highly leveraged; cells often appear in hundreds of different designs and their density is directly reflected in the density of the resulting designs.

This struggle between the need for higher levels of automation and the uncompromising need for layout density has shaped library development methodologies for some time. This paper describes two phases in the development of an automated methodology for standard cell library generation at LSI Logic Corporation.

The first phase attempted to automate the creation of cell libraries as much as possible without sacrificing any layout density as compared to hand-crafted cell layout. It was also required that the generated cell layouts be compatible with other (existing) LSI Logic tools and layout objects. The required layout density and controllability was achieved by creating an application-specific symbolic layout and compaction system. The developed approach allows a designer to specify the basic cell topology while automating the details of the layout process. Because the methodology is designed specifically to create standard cell topologies, the created layouts can incorporate many context specific layout optimizations and can conform to all required library compatibility constraints.

The second phase explored the benefits of concurrency, allowing the layout automation to shape the overall methodology of both process and library development. The available concurrency allows optimization to occur simultaneously across the circuit, layout, and process design spaces. The result is libraries with improved layout density, circuit performance, and process yield.

We report here the methodology of both phases, the results of each, and the benefits that have resulted from the interaction of automation and methodology.

Evaluating the Trade-Offs

LSI Logic has historically relied on manual layout techniques to produce cell libraries, investing whatever effort was required to produce high quality cells in a timely fashion. Because it is time consuming to manually modify physical cell layout once it has been created, it has been necessary to decouple as much as possible the tasks of process and library development. The two tasks have thus been accomplished almost in series, as is shown in Figure 1. The sequential ordering of these tasks increases the time between process freeze and library availability and limits the amount of joint exploration that can be done across the process and the layout architecture design spaces. The use of a manual cell layout methodology (in order to achieve required layout densities) thus has the effect of shaping the entire library development methodology.

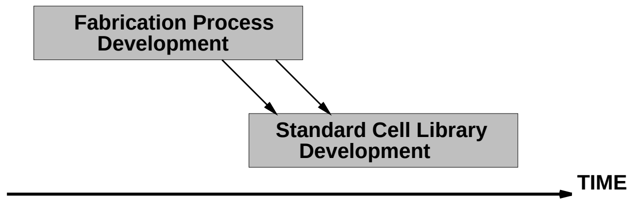

Fabrication Process Development
Standard Cell Library Development
TIME

Figure. 1. Library Development Flow With Manual Layout

The first goal in developing a concurrent methodology for standard cell library generation was to develop an automated method for cell layout generation. In order to allow process development and library development to occur simultaneously, it was necessary to eliminate the long delays and high costs associated with manually updating physical cell layouts each time process and circuit design changes occur. A number of different layout automation techniques were considered. What follows is a brief summary of the basic methodologies that have been developed in the area of physical design automation for standard cell layout generation.

The three most dominant layout automation techniques have been schematic driven layout synthesis [4], symbolic layout with compaction [5][6], and procedural module generation [7][8].

Schematic driven layout synthesis starts with a sized circuit schematic and produces a complete cell layout, typically using a variety of graph-theoretical techniques and heuristics for transistor chaining and placement, followed by routing and possibly compaction. Because it is completely automatic, layout synthesis tracks circuit and process changes well. While it is an active area of research and commercial products have appeared, work still remains in order to develop techniques that consistently approach hand density. Designers can often optimize the placement and routing of cell contents better than the layout synthesis heuristics.

Symbolic layout with compaction starts with a topology created by a designer and then applies virtual-grid or constraint-graph compaction to minimize the cell area for a given set of design rules. Although compaction provides a high level of process portability, there is usually some sacrifice in layout density as compared to manual layout. Because the compaction process is applied to rigid layout objects, layout optimizations which involve changing the shape or orientation of layout objects are generally not possible.

Procedural module generation encodes the procedure for assembling a cell in an executable program that, when run, produces layout tailored to the design rules. Procedural layout can approach hand density and usually tracks small changes in design rules well. Because design rule constraints are explicitly embedded in the procedural code, however, large process changes can interact with the code in unpredictable ways, resulting in a loss of layout density.

Phase One: Automation With Manual Quality

The goal of the first phase was to automate the creation of the standard cell libraries so that they could be developed concurrently with the fabrication process, as shown in Figure 2.

The performance constraints of the project made it necessary to create layouts with densities comparable to those created by human designers. It was further required that the generated cell layouts be compatible with other LSI Logic tools and layout objects. Because existing (general purpose) layout automation techniques didn't satisfy these requirements, it was necessary to develop an automation methodology specifically matched to the requirements and opportunities of standard cell layout generation at LSI Logic.

The application-specific layout generation software was developed using ICGen, a process portable layout framework developed by Mentor Graphics Corporation. ICGen is a physical design automation development environment which allows users to create their own (problem-specific) application software. In addition, a project partnership was formed between LSI Logic and Mentor Graphics. The hope was that LSI could provide feedback in the ongoing development of the ICGen tool and that MGC could provide expertise in the area of layout automation. Since much of the tool and methodology development occurred at LSI Logic, however, it was ensured that the developed software was closely linked to the specifics of the LSI Logic library development process.

Design Approach

The development of a standard cell layout generation methodology for LSI Logic was based on three basic design goals:

1. Create layouts which meet or exceed manual layout density.
2. Create layouts that are compatible with other LSI Logic tools.
3. Automatically track process changes.

The first goal reflects the realities of the ASIC business and the overriding importance of layout density in creating a standard cell library. The second goal is necessary in order to make incremental methodology improvements possible. The standard cell library layout objects are used by numerous downstream tools (during characterization, placement, and routing for instance). It was not possible (under the constraints of a production schedule) to update all of these tools so as to accommodate "non-standard" layout objects. The third goal enables concurrency and its benefits.

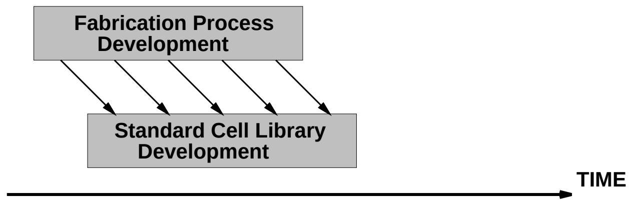

Fabrication Process Development
Standard Cell Library Development
TIME

Figure. 2. Library Development Flow With Phase One

The required layout density and controllability was achieved by creating an application-specific symbolic layout and compaction system on top of the ICGen framework. The automated placement and routing that is a part of a schematic-driven layout synthesis system can often "get in the way" of layout designers by preventing them from using their expertise in layout design. The selected symbolic layout based approach leverages designer expertise by allowing a user to describe the basic cell topology while still automating most of the details of the layout process. Because ICGen allowed the LSI Logic developers to define and use their own parameterized layout elements, it was possible to create a compaction-based system that could effect many LSI-specific layout optimizations. Such optimizations would not have been possible with a more general purpose symbolic layout system.

The required process portability was achieved in large part simply by building the standard cell layout engine on top of ICGen, a process-portable layout framework. The ICGen system provides data structures and procedural interfaces that allow physical design data to be represented in a process-portable format. Additional portability was achieved by structuring the layout engine so as to take transistor sizes and architecture parameters as inputs (in addition to process design rule information). Transistor sizings and architecture parameters generally change along with design rules. In order to be truly process portable, a layout generation system must track these inputs in addition to process design rules. Portability across transistor sizes is illustrated in Figure. 3 which shows mask layouts generated using inputs that are identical in all aspects except for transistor sizing. All layouts shown in this paper were generated using the MOSIS scalable CMOS design rules.

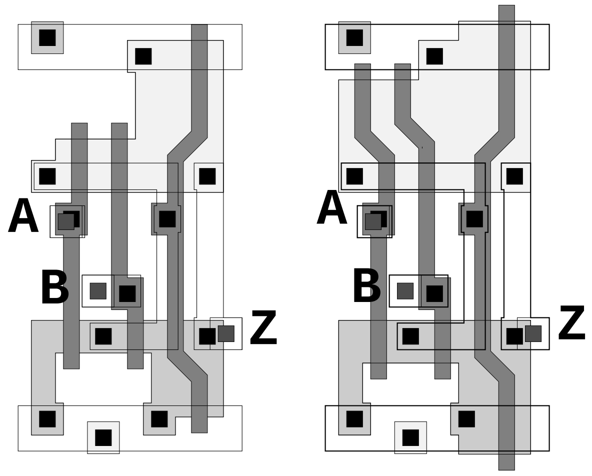


Figure. 3. Mask Level Layouts Illustrating Transistor Resizing

The details of the layout engine implementation have been reported elsewhere [9], and are described briefly here only to show how they affect the methodology and the results. The steps involved in the standard cell layout generation process correspond roughly to those of any symbolic layout and compaction system.

- A designer first creates a symbolic layout representation which specifies the basic placement and routing of cell layout objects.
- Given a technology database which describes the required sizings and shapes of mask level layout objects, each object in the symbolic layout representation is translated to a corresponding mask level layout object. Transistor sizings, derived from a circuit schematic, are introduced during this step.
- The symbolic layout topology is next used to create a constraint graph defining valid locations for each mask level layout object both relative to cell boundaries and relative to other layout objects in the cell. Architecture parameters are introduced during this phase as application-specific layout constraints.
- Finally, a compaction phase is used to position the layout objects within the bounds defined by the constraint graph. The compaction process uses the ICGen “As Close As Possible” (ACAP) tool kit to position objects in the x dimension [10] and a more general purpose graph-solving algorithm to position objects along the (more constrained) y dimension.
- During the compaction phase, the ICGen “extensible element” capability is used to create transistors with optimally placed bends. Because it is possible to reshape elements as needed during this step, the effect is similar to compacting the internals of the layout elements in addition to compacting around them. The result is a significant improvement in layout density.
- The system completes the layout using the ICGen boolean capabilities to effect a number of mask manipulation steps. This phase adds derived layers and performs other kinds of mask processing steps, such as notch and gap filling.

The mask layout generated for a five-input NAND gate is shown in Figure 4. The shown layout clearly illustrates the improved layout density that results when each individual layout object can be optimized to fit the specific constraints of the local layout context.

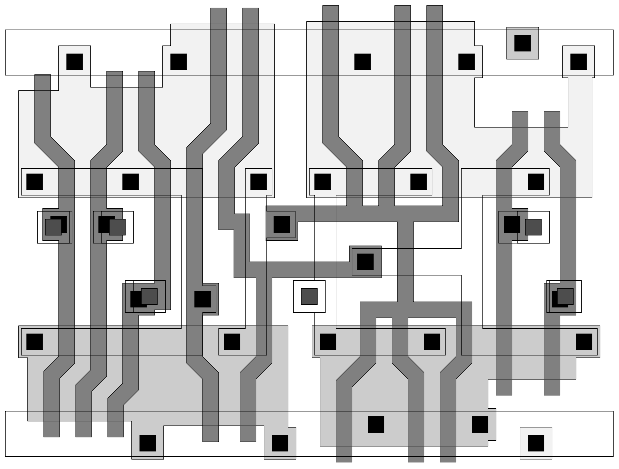


Figure. 4. Mask Level Layout of a Five-Input NAND Gate

First Phase Experimental Results

The first phase development effort resulted in the creation of a production library including over 200 standard cells for the LSI Logic LCBG10P process. All of the design goals were met, most notably the goal of equaling manual layout densities. The density of the LCBG10P library was compared with a previous generation of manually created cells, normalizing the comparison in terms of the routing grid size for each generation. As shown in Figure 5., the automated cells had virtually the same density as the hand-crafted ones. In addition, because the automated cells tracked process changes, significant concurrency was enabled. Manually created libraries had in the past lagged the process freeze by up to three months (Figure 1). All the LCBG10P standard cell layout objects were available less than one week after process freeze.

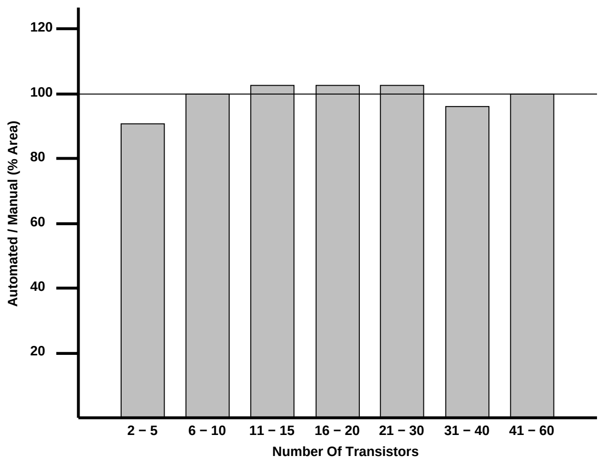
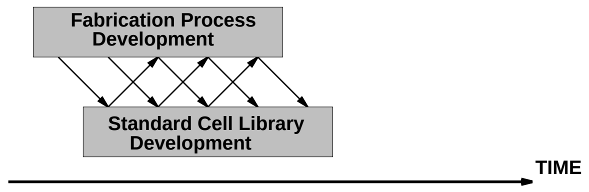


Figure. 5. First Phase Layout Density, Compared To Manual

Phase Two: Global Optimization

The success of the first-phase layout automation encouraged LSI Logic to explore how the automation capabilities that had been created might improve the overall design methodology. The first phase only allowed single-direction information flow from the process to the layout. In the second phase, an attempt was made to allow bi-directional feedback between layout generation and process development, as indicated by the arrows shown in Figure 6.



Fabrication Process Development
Standard Cell Library Development
TIME

Figure. 6. Library Development Flow With Phase Two

The goal was to determine how the overall development methodology (of creating both a process and a library) could be improved now that it was no longer constrained by manual layout techniques. While the first phase development was implemented by both LSI Logic and Mentor Graphics, the second phase development was implemented by a team from LSI Logic working independently.

Design Reuse and Concurrency

The most immediate benefit in the second phase was that the entire set of symbolic layout specifications developed during the first phase for the LCBG10P process could be reused (with only minor changes) to get immediate feedback on the standard cell library implications of the new fabrication process. This direct reuse represented a significant saving of design time in itself. Perhaps more significantly, the availability of immediate library feedback enabled an unprecedented level of concurrent exploration of cell architecture and process variation. Because the cell specifications developed in the first phase took both process rules and transistor sizes as inputs, concurrency was enabled among numerous tasks:

- Library Architecture Optimization
- Circuit Optimization (transistor sizing)
- Process Optimization (design rules)

The available concurrency not only reduced time-to-market, but also made it possible to explore the implications of library architecture and process design variations across entire libraries of cells.

Architectural Exploration

The first phase goal was to meet or exceed manual layout density in the cells themselves. In the second phase it was possible to explore the effects of library architecture changes on the density of not just the cells, but on typical end-customer designs using the cells. More specifically, the available layout automation made it possible to explore trade-offs in:

- Cell/Transistor Sizing
- Cell Aspect Ratio
- Routing Porosity

Architecture parameters like routing porosity can only be evaluated in the context of real designs. For example, it is possible to create very dense cells that are hard to route due to the lack of routing channels in the cells, leading to global layout inefficiencies in end-user designs. The new automation capabilities allowed the effects of variations in such parameters to be quantitatively explored by building entire sample libraries and then measuring the performance and area results when each sample library was used to build a number of actual "lead vehicle" designs. This represented a methodological shift from focusing (locally) on the cells themselves to focusing (globally) on the effectiveness of the library of cells in actual designs. Furthermore, the availability of quantitative architecture data allowed more aggressive architectures to be explored (and ultimately selected). Because multiple libraries were developed in parallel, it was possible to select the architecture with the best combination of layout density and routability. It is clear that without the quantitative feedback, a more conservative architecture would have been selected, costing at least 5% in area as compared to the cell architecture that was ultimately selected.

Process Exploration

The available process-portable layout generation capabilities also allowed extensive exploration of the effects of potential process changes on downstream library size and performance. Rather than needing to estimate these effects by examining only a few pilot cells, it was possible to quickly and quantitatively measure the library-wide implications of proposed design-rule changes. The early availability of realistic standard cell libraries also increased the reliability of process test chips as predictors of future yield. In the past, only small prototype standard cell libraries could be available at the time that process test chips were being developed. The availability of extensive and reliable process data made it possible to make more informed decisions regarding trade-offs in:

- Aggressiveness of Design Rules
- Process Yield
- Library Electrical Performance

As with cell architecture development, this represented a methodological shift from local optimization to global optimization. The shift in methodology focus resulted directly in the use of more aggressive design rules. This came from two effects:

- The ability to explore process yield using realistic test chips.
- The ability to freeze the design rules late in the overall design cycle. (By contrast, freezing the design rules early leads to conservative choices for the rules.) Because the cost of changing rules late in the design cycle was low, the risk of being aggressive was reduced, and more aggressive design rules resulted.

The LSI Logic process R&D organization was able to investigate several levels of aggressiveness of key design rules in parallel with the architecture and cell topology development. The results of this investigation were used to select the architecture with best yield potential. The most aggressive rules turned out to be infeasible with current technology, but the next most aggressive rules proved workable. This provided increased confidence that the selected library design (both in terms of cell architecture and process design rules) was very near the point of minimum total product cost.

Other Design Flow Improvements

The new layout methodology also had the effect of smoothing the complete CAD flow for producing and characterizing libraries. The information available in the ICGen layout database allowed objects for downstream tools (such as abstracted routing and obstacle models) to be automatically created out of the same database as the layout objects themselves. This had the effect of streamlining the CAD flow and of improving the quality of the created objects.

Second Phase Experimental Results

The second phase resulted in the creation of a new production library including over 200 cells. The level of concurrency that was achieved allowed an unprecedented amount of global design exploration to take place. The result was a library created using quantitatively more aggressive design rules and cell architecture than would have been possible using a manual layout methodology.

Conclusions

The LSI Logic methodology for standard cell library development shifted in two phases from a serial development of the fabrication process and the library (dominated by the needs of manual layout) to a fully concurrent development methodology. The shift required the development of a new layout automation technique that produces manual quality layout using an application-specific symbolic layout and compaction method. The new methodology allows a dramatic reduction in the time-to-market for new libraries and enables global library optimization across the circuit, layout, and process design spaces. The result is libraries with improved layout density, circuit performance, and process yield.

References

1. V.K.R. Chiluvuri and I. Koren, "Layout-Synthesis Techniques For Yield Enhancement", IEEE Trans. on Semiconductor Manufacturing, v. 8:2, pp. 178-187, May 1995

2. C. Bamji and E. Malavasi, "Enhanced Network Flow Algorithm for Yield Optimization", in Proc. 33rd DAC, pp. 746-751, 1996.

3. A. N. Lokanathan, J. B. Brockman, J. E. Renaud, "A Methodology for Concurrent Fabrication Process/Cell Library Optimization", Proc. 33rd DAC, pp. 825-830, 1995.

4. R. L. Maziasz and John P. Hayes, Layout Minimization of CMOS Cells, Kluwer Academic Publishers, Boston, MA, 1992.

5. D. G. Boyer, "Symbolic Layout Compaction Review", Proc. 25th DAC, pp. 383-389, 1988.

6. J. Dao, N. Matsumoto, T. Hamai, C. Ogawa, and S. Mori, "A compaction method for full chip VLSI layouts", Proc. 30th DAC, pp. 407-412, 1993.

7. T. G. Matheson, C. Christensen, and M. R. Buric, "A software environment for building core-microcomputer compilers", Proc. International Conference on Computer Design: VLSI in Computers (ICCD '85), pp. 221-4, 1985.

8. J. C. Herbert, "An integrated design and characterization environment for the development of a standard cell library", Proc. IEEE 1991 CICC., pp. 25.6.1-25.6.5, 1991.

9. J. Duh, T. G. Matheson, and Ed Hepler, "Efficiently Embedding Expertise in High-Density Process-Portable Standard Cell Generators", Proc. IEEE 1995 Custom Integrated Circuits Conference, pp. 497-500, 1995.

10. R. A. Eesley and M. A. Tarsi, "ACAP - A system for the interactive graphical capture of generators", Proc. IEEE 1991 CICC., pp 22.1.1-22.1.4, 1991.