CELLERITY: A Fully Automatic Layout Synthesis System for Standard Cell Libraries

Mohan Guruswamy, Robert L. Maziasz, Daniel Dulitz, Srilata Raman, Venkat Chiluvuri, Andrea Fernandez, and Larry G. Jones

Unified Design System Laboratory
Motorola, Inc.
Austin, Texas

Abstract

This paper describes a fully automatic standard-cell layout synthesis system, CELLERITY. The system is flexible in supporting a wide variety of process technologies and a range of library template styles. The tool is fully automatic and provides several options to the user to customize the layout template. The tool considers performance and yield and generates dense, design-rule correct layouts. Experimental results indicate that the area of CELLERITY-generated standard cells is competitive with manually designed cells in a majority of circuits. In block-level tests of industrial circuits, standard-cell blocks generated using CELLERITY cells are about equal to the block area produced by using a manually-designed library. Recently, an embedded microcontroller in a state-of-the-art sub-micron process technology was fabricated using CELLERITY-generated standard cells.

1 INTRODUCTION

Standard-cell methodology is widely used in IC design. Automation of standard-cell mask layout generation significantly improves cycle time for creating new standard-cell libraries, provides a testbed for evaluating new process technologies by rapidly synthesizing blocks, and enables rapid retargeting of designs to new process technologies.

Standard cell layout synthesis presents new optimization problems such as transistor folding to meet a specified library height, optimal placement of input/output ports on a wiring grid, and satisfying cell boundary conditions for block-level design rule correctness.

Technology independent synthesis is vital to supporting rapidly advancing processes. The key components such as transistor placement, detailed routing, and layout compaction must be flexible to support a wide variety of processes and standard-cell template requirements. Deep sub-micron design requires additional functionality such as performance-driven transistor placement, antenna diode placement to protect transistor gates from charge accumulated during fabrication steps, area-efficient placement of substrate and well ties to prevent latch-up, performance-driven detailed routing, and layout compaction with preference to critical nets. The CELLERITY layout synthesis system supports all of the above requirements. This tool is currently being used to generate layouts for real designs and to steer development of new sub-micron process technologies; it also enables design rule changes to existing processes.

Section 2 presents a brief description of other layout synthesis tools and their limitations. Section 3 presents a detailed description of CELLERITY. Section 4 presents key results at the standard-cell level and at the block level on real industrial designs. The paper concludes with Section 5.

2 BACKGROUND

Many papers have been published in the area of layout synthesis in the style originally proposed by Uehara and van Cleemput [1], such as [2]-[12],[15],[16],[18], and [19]. (See [6] for overview.) Most of these have focused on specific, fundamental problems related to transistor placement, routing, and compaction, and have described layout synthesis systems developed mainly to demonstrate their specific innovations. Such systems have ignored many practical problems essential to fully automatic standard cell layout synthesis, such as transistor folding to meet cell height requirements and minimize width, well and substrate tie insertion to meet tie coverage requirements in an area efficient way, input/output port placement on a routing grid for compatibility with place and route tools, circuit performance issues, and more flexible cell architectures, such as transistor stacking.

Systems developed in industry have addressed some of these practical problems [2]-[5], [7]-[10], [15]. In some cases, these solutions are not fully automatic, are inadequate for standard cell synthesis, or are obsolete due to rapidly changing process and design technology. For instance, boundary ports are used in nearly all prior cell synthesis systems. However, to take advantage of processes with three or more metal layers, internal ports are required so that intercell routing can occur mainly over the cells [11]. Furthermore, although transistor folding methods have been reported [12][16], none guarantees to find a solution meeting a target height having minimum width after compaction, an essential feature for fully automatic synthesis of dense standard cells. Finally, no system has been reported in the literature that describes robust solutions to the practical problems essential to synthesizing high performance, high yield cells in a fully automatic manner with handcrafted density in current process technologies.

3 LAYOUT SYNTHESIS SYSTEM

3.1 Overview

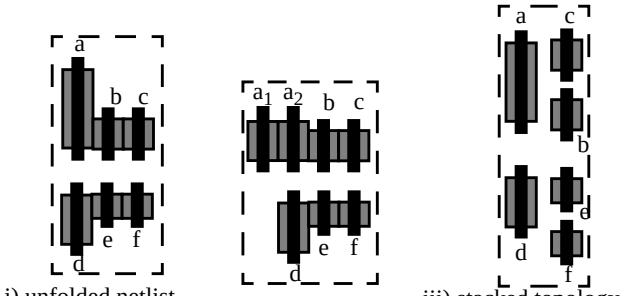
The inputs to the CELLERITY system are a process file which contains a description of the width/spacing rules for all mask layers, a netlist file containing a list of sized transistors and their interconnections, and a *template* file containing the control parameters for the layout topology (for example, cell height, *nwell* height, cell grid, supply rail layer, and supply rail location and size).

DAC 97, Anaheim, California

3.1.1 Design Flow

In the first step of layout synthesis, the input netlist is transformed into several physical netlists, each functionally similar but structurally distinct, by the *transistor folding* process (see Section 3.2.) *Transistor folding* is the process of converting one transistor into several smaller parallel transistors.

A *physical representation* of a netlist is defined as a combination of a specific folding (maximum P and N transistor size) and the cell topology which is given by the number of P transistor rows and N transistor rows. Figure 1 illustrates three instances of physical representations for the same netlist.



i) unfolded netlist representation ii) folded netlist iii) stacked topology (unfolded netlist)

Fig 1. Illustration of Physical Representation.

A typical standard cell with a high drive strength has several physical representations as a result of the folding process. Each physical representation is taken through the steps of component placement, routing, and compaction for cell topologies that could produce a cell at the required height and with the smallest width. If, at the end of the layout compaction, the selected physical representation does not meet the cell template requirements, the next physical implementation is selected for layout generation.

After selecting a specific physical implementation, the transistors are placed using simulated annealing with a cost metric that reduces routing within the cell and minimizes diffusion breaks. Transistor placement is followed by the port placement phase which identifies port locations that minimize the interconnect lengths of transistor connections. Antenna diodes are then placed at input ports as required by the design rules.

Once the transistors are placed, the locations of supply connections to source/drains of transistors are known. These locations, along with the channel column density information, are used to optimize substrate and well tie placement. The ties are placed on the supply rails and, if required, in the channel region (between P and N transistor regions) to meet the tie coverage requirements.

In the pre-routing step, adjacent source/drain connections are made with diffusion wiring and contacts are placed for source/drain terminals. Power supply routing is performed by placing horizontal supply rails and vertical taps to source/drain connections. Other special-case pre-routing may be performed for inverters and other high-drive-strength cells.

The multi-layer detailed area router, a derivative of Echelon [13], is then used to complete the routing of all remaining nets. This router effectively handles all the constraints imposed by the cell synthesis problem by using a non-uniform routing grid derived from the spacing rules of each routing layer.

The layout compaction step uses a one-dimensional constraint-graph-based compactor derived from SQUEEZE [14]. Supplemental constraints are added to satisfy template requirements. Automatic jog insertion is performed to minimize area and wire length, and external signal ports are placed on a routing grid.

The layout compaction step is followed by a post processing step that adds redundant ties and contacts to improve yield and enhance latch-up protection. The cell is then checked for tie coverage. If the layout satisfies tie rules, final post-processing steps are carried out to eliminate notch errors and add extra contacts to improve circuit performance. If the tie coverage is not sufficient, the tie placement is modified and the cell is routed and compacted again.

Each major component of the system has feedback loops to ensure success before proceeding to the next stage. For example, if the Echelon router detects an unroutable problem, the location of routing failure is used to modify the device placement to enable routing completion. The system is fully automated and *guarantees* a design-rule correct cell without any manual intervention. Feedback loops are used to ensure that the best possible cell is produced.

3.2 Transistor Folding

In standard-cell methodology, the height of the cell is critical. To obtain the specified cell height and reduce area, wide transistors are folded. Transistor folding is the process of splitting a transistor into multiple transistors of smaller widths connected in parallel.

Given a maximum size for PMOS and NMOS transistors and a cell netlist, our method synthesizes the cell using different folding combinations and terminates when it has found the narrowest possible cell that meets the target cell height. Where possible, our method splits nodes in the folded netlist into multiple equi-potential nodes in order to reduce the number of wires in the final cell layout.

Intelligent transistor folding is crucial to standard cell layout synthesis because it automatically generates area-optimized cells. In the absence of automated folding, a user must experiment with each cell layout in order to find the best foldings for PMOS and NMOS transistors, a very time-consuming exercise.

3.3 Device Placement

The system supports both single height (Figure 2) and double height standard cells (Figure 3). In the single height style, one or

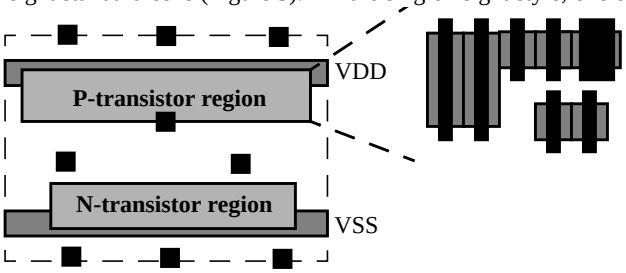


Fig 2. Single Height Standard-Cell.

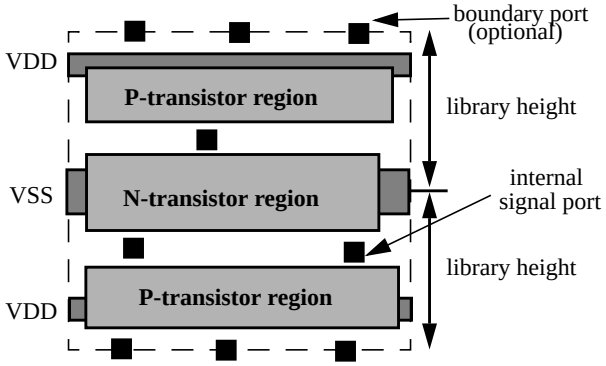


Fig 3. Double Height Standard-Cell.

two rows of transistors are allowed to be placed within each P and N transistor region, to make effective use of the standard-cell height. In the double-height style, shown in Figure 3, the N-transistor region in the middle of the cell can have up to four rows of NMOS transistors and the outer transistor regions (P-transistor regions) can have one or two rows of PMOS transistors. Alternately, in a double-height cell, the P-transistor region can be present in the middle of the cell with the N-transistor regions placed on the outside. Signal ports can be placed both internal to the cell and external, at the top and/or bottom of the cell boundary. The system also supports a standard-cell style which allows diffusion abutment between adjacent cells.

3.3.1 Transistors

CELLERITY utilizes a very general transistor placement algorithm that produces optimal or very nearly optimal results for all standard cells of practical size. It efficiently handles any netlist of arbitrary topology and transistors of any width and length.

The problem of finding a transistor placement for complementary circuits that simultaneously minimizes diffusion breaks (width) and channel density (height) is computationally intractable for deterministic algorithms [17], although such algorithms, e.g., TrailTrace [6], can find minimum width layouts with minimum height for circuits of limited size in a reasonable amount of time. Prior non-deterministic transistor placement methods, such as those using simulated annealing, have avoided using the best method of estimating cell height, namely channel density, because it has been thought to be too expensive to compute for each move. Instead, they have relied on other methods that are more quickly computed, such as total wire length [18] or total column density [9]. Thus, no prior method finds minimum or near-minumum diffusion breaks and channel density for circuits of arbitrary size.

CELLERITY uses a simulated annealing placement algorithm with a cost function that includes channel density. The algorithm finds transistor placements that have minimum or very nearly minimum cost based on all the metrics historically deemed to be the most effective: diffusion breaks, total wire length, channel density, as well as gate and source/drain alignments. Consequently, the algorithm routinely produces transistor placements that are equal to or better than hand-crafted placements for standard cells of practical size. Performance issues are considered by more heavily weighting time-critical nets.

In addition to the traditional two-row layout style (Figure 1(i)), CELLERITY can also generate transistor placements in a stacked layout style as introduced by TOPOLOGIZER [19] (Figure 1 (iii)). Such layouts can be very area efficient and are frequently generated by expert layout designers. CELLERITY's flexible layout style and resulting area efficiency make its placement approach superior to all previous transistor placement methods.

3.3.2 Signal Ports

After the placement of transistors, signal ports are placed. The goal of port placement is to minimize the additional cell area required for a port and its associated interconnect. Internal signal ports are placed in the routing channel between regions of P and N transistors. The algorithm prefers to place ports in locations with low column density and no other port in the same column. It also prefers to align ports with transistors having the same net as the port; furthermore, it tries to place the port within the overlapping span of the horizontal segments of the net on both sides of the routing channel. Ports can be vertically staggered to guarantee port access to the block-level router in both directions. Boundary ports are placed within their port net span where possible, as well as close to transistors along the boundary that connect to the port net.

3.3.3 Ties and Antenna Diodes

Placement of well and substrate ties is a difficult problem for layout synthesis tools that aspire to be truly technology independent. To achieve density similar to creative manual cell designers, our system uses a two step tie placement process.

The first step occurs immediately after transistor placement but before routing. Here critical cell dimensions are estimated and, based on these estimates, a bare minimum number of substrate and well ties are placed using a one-dimensional algorithm that operates independently on each row of transistors. The tie placement algorithm prefers to place ties at supply taps, where the tie can be merged with active nitride. If no supply tap is available near the required location, the algorithm will try to place the tie at other preferred locations based on column density, diffusion breaks, and existing diffusion contacts; if no preferred location is found, a tie is simply placed wherever needed.

After routing and compaction, ties are filled as necessary into available space. After tie filling, an internal subsystem automatically checks the tie coverage of the cell; if coverage is insufficient, synthesis is repeated with more ties added in the first tie placement step. In technologies with less demanding tie placement rules, no ties need be placed in the first tie placement step at all, as there is often enough space available after compaction to insert ties and achieve design rule correctness.

CELLERITY supports synthesis of antenna diodes to protect transistor gates against charge accumulation during the fabrication process. Antenna diodes are added only if required by the process technology and user of the library.

The placement of antenna diodes affects the area of the circuit layout; therefore, it is important to place the diodes for each transistor gate at a location where it has the least impact on routing and circuit area. After the transistors and signal ports have been placed, a greedy algorithm is adopted to place the antenna diodes. Typically, a diode can be placed in any location that is immediately adjacent to the input port. A cost is associated with each possible location for the diode and the algorithm generates a solution that minimizes the total cost of placing all the diodes. The cost function captures impact on cell routing, accessibility of the signal port from the corresponding gate in the presence of the diode at that location, and the effect on the compaction step of layout synthesis.

3.4 Routing

Once the transistors and other structures have been placed in the symbolic layout, the next step is to connect the layout using wires in available layers according to the netlist. Routing has a profound impact on the quality of final compacted cell layout. Poor routing of nets includes unnecessary crossover of wires, circuitous routes, and redundant vias and contacts, all of which impair electrical performance and adversely affect yield and area.

Typically, for a standard cell layout, two layers are available for routing: *polysilicon* and *metal*. Owing to the high resistivity of diffusion, it is primarily used to interconnect adjacent transistors that share common signal nets. Similarly, in processes with high polysilicon resistivity, polysilicon wires are limited to nets that connect transistor gates. Cell synthesis systems reported in the literature typically use channel routing algorithms [4][15]. These restrict the cell architecture because they can route only a single row of P and N transistors and cannot effectively use the space over the devices. Moreover, a two-layer channel router typically relies too heavily on poly wiring and poly contacts, resulting in poor performance and reliability.

CELLERITY uses an area routing scheme which does not have fixed routing directions for layers and is both performance and area

driven unlike prior art [8] [16]. The CELLERITY router allows user-specified layers for interconnection, and it can route in more than the two conventional layers, polysilicon and metal. For example, if the technology permits a second layer of metal or local interconnect to be used within the standard cell, the router will use it.

Detailed routing of nets is handled by a multi-layer area router [13] which is designed to handle prerouted wires and rectilinear obstacles. The area router minimizes a cost function that takes into account the length of the wiring in various layers, the number of bends in the routes, and the number of vias. Electrical performance is addressed by according preferential routing to critical nets (either identified by the user or by the tool). The router attempts to route these critical nets in shorter wire spans and in a layer of least resistivity. The router also aims at reducing polysilicon jumpers (which hurt performance) in output nets. In addition, special nets can be routed *a priori* in specified layers. While the router can route in any direction in a layer, a preferred direction (horizontal or vertical) can be set for each layer. The router generates wiring that honors these preferred direction requirements to the greatest extent possible while attempting to optimize the area.

Briefly, there are five main steps involved in the detailed routing stage. First, nets are ordered based on their criticality; the area router processes nets sequentially based on this ordering. Second, the router maps the symbolic layout problem onto a non-uniform two-dimensional virtual routing grid whose objective is to maximize the number of routing tracks that can be used in all the routing layers. Third, a maze router determines a coarse route for each net. Fourth, based on the coarse routes, each group of wires is assigned to a layer with the objective of minimizing a cost function and ensuring that wires belonging to different nets do not overlap in the same layer. The algorithm used in this step is based on simulated annealing. The fifth and final step is that of rip-up and reroute in which the objective is to improve routing by rerouting nets using shorter wires, fewer vias and bends.

In cases where all the nets cannot be completely routed in the given area, a feedback loop determines where and how to expand the routing space. This step provides information about the location at which a routing failure has occurred. Then, a heuristic algorithm determines the additional space that must be added in the vicinity of the routing failure. Using the augmented routing space, the router attempts to route the circuit again so that all the nets are successfully routed.

3.5 Compaction

3.5.1 Template Constraints

Standard cell libraries are often optimized to achieve best results with a particular combination of place and route tool, process technology, and transistor sizing. This optimization occurs over many template parameters, including cell height and well height. Some of these key parameters have not been discussed in the literature. Since layout-synthesized cells may be used to augment an existing hand-designed library, it is important for a layout synthesis system to support a wide range of template parameters.

Calculating the cell boundary is a straightforward matter of spacing most elements by half their spacing distance to the bounding box so that abutting cells will not violate design spacing rules. Supply rails and diffusion-sharing structures are not spaced from the bounding box. Structures that simulate abutting cells may be added, depending on the technology, so that abutting cells will not violate design minimum width and notch rules.

Nwell and *pwell* spacing rules are often much larger than spacing rules on other layers, so drawing the boundary outside of the well region is very inefficient (Figure 4 i). Nwell and pwell regions near the left and right cell boundaries should therefore be confined to user-specified intervals in order to avoid spacing violations between the well of one cell and the well of an abutting cell (Figure 4). The CELLERITY compactor satisfies both cell height and well height of each cell.

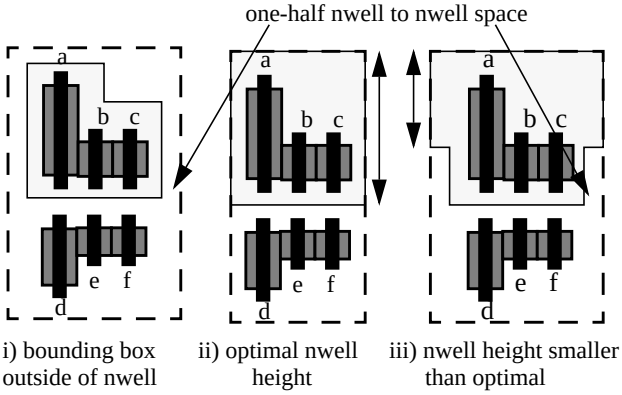


Fig 4. Advantage of Correct Well Height.

Most standard-cell place-and-route tools use gridded routers, so maximum block-level place-and-route efficiency is achieved when the input and output ports of each cell are placed on the routing grid and the cell boundary is placed on the placement grid. The compactor implements fully independent grids [21] for each port, as well as a grid for the bounding box, and the grids can be independently offset from the bounding box origin.

3.5.2 Compaction Directions

In a standard cell environment, achieving the library height is of the utmost importance, followed closely by minimizing the cell width. To achieve these goals with a one-dimensional compactor [14][22], several sequences of compaction directions are used. A Y-first (YXY) compaction is first performed in order to minimize the cell height. If the library height is met, an X-first (XYX) compaction is performed on the original uncompacted layout. If the X-first compaction also meets the standard cell height, it will usually result in a narrower cell than the Y-first compaction and, if so, it is selected in preference to the Y-first result.

Since even small size reductions which are a fraction of a grid may achieve the library height or reduce the cell width by an entire grid, it is desirable to spend much optimizing effort in the compaction phase. Typically, three compaction passes (XYX, YXY) are used. When the user requests the best cell possible, regardless of runtime, up to five compaction passes (XYXYX, YXYXY) are used, achieving a width reduction in a small but significant percentage of cells. Conversely, when speed is of primary importance, only two compaction passes (XY, YX) are used.

3.5.3 Jog Insertion

Jog insertion, the process of inserting bends in straight wires during compaction, is needed in order to find (noncyclic) solutions to complex template constraints. Jog insertion is also critical to achieving hand-packed density even on the smallest cells, since ties, diodes, and internal ports cause serious congestion in the channel area between transistor rows. Stacked transistor placements can rarely be implemented within the standard cell height without effective jog insertion.

The CELLERITY compactor iteratively inserts jogs in wires on the critical path; the jog locations are chosen using a variant of the shadow-based approach [14][20]. Unique algorithms are used to support arbitrary gridded objects within the graph-based jog insertion approach. Both area and runtime are reduced as compared to

prior implementations which inserted jogs before each compaction pass. The use of jog insertion is strictly controlled during the first direction of compaction to avoid interlock problems in the second direction. Critical path jog insertion algorithms can fail to minimize the length of some wires. Our compactor overcomes this problem by constraining critical wires to minimum length and relaxing the wire length constraints if the graph has no solution even after jog insertion.

3.5.4 Wire Minimization

Wire minimization is performed after the cell is compacted in each direction. Minimizing wire length improves circuit performance and reduces cell area after a subsequent compaction. A unique algorithm has been implemented for minimizing wires that are connected to layout objects, such as I/O ports, that must be placed on a grid. Overall cell area and performance characteristics are improved as compared to previous wire minimization algorithms based on [23] that could not move gridded objects while satisfying the grid constraint. The gridded wire minimization algorithm has reduced the area of some standard cells by 15%.

In order to overcome the limitations of one-dimensional constraint-graph based compaction, we implemented several heuristics to achieve better performance characteristics. For example, diffusion wire length between series transistors must be minimized for improved circuit performance. Since transistors are oriented vertically during the placement stage, diffusion wire minimization typically takes place during X compaction. However, if the compaction sequence is started with the Y direction, some layout elements may be moved between transistor gates preventing diffusion wire minimization during subsequent X compaction. This movement is inhibited in our compactor on a heuristic basis. During wire minimization, the weights of wires on critical nets are adjusted to preferentially minimize the length of those wires without greatly affecting the length of other wires.

4 RESULTS

Figure 5 illustrates sample cell layouts with stacked and unstacked transistors generated by CELLERITY using the MOSIS 2.0µ SCMOS technology. For reasons of proprietary information, this technology was used for illustration and does not reflect the submicron process capability of the tool.

4.1 Cell Level Results: Comparisons with Manual Library

A sub-micron standard cell library widely used within Motorola, consisting of 201 high-performance cells, was generated by CELLERITY. The total area of the automatically synthesized library was within 5% of the area of the manually designed library. Estimated cell performance based on diffusion and polysilicon area was identical to manually designed cells. A standard template was used to generate the synthesized cells, and no effort was made to individually enhance any synthesized cell.

To further illustrate the density of cells generated by CELLERITY, Table 1 shows a comparison of individual cell area between the manual layout and the automated layout for a representative sample of cells from the library. Each cell layout consists of one P/N-transistor region pair. The cell height for the CELLERITY cells and the manual cells are the same. The column *% Smaller* indicates how much smaller the automatically generated layout is compared to the manual layout. (A negative amount indicates the manual layout is smaller than the CELLERITY layout.)

Table 1: Individual Cell Comparisons

Cell Type	CELLERITY Area	Manual Area	% Smaller	Total # of Transistors
inv_1	24206	26754	10	2
inv_2	28028	33124	15	2
buf_1	10192	10192	0	4
buf_2	12740	14014	9	4
ind	25480	26754	5	8
inr_1	11466	11466	0	6
inr_2	11466	12740	10	6
exnor	36946	43316	15	12
exor	20302	31850	36	12
mux_1	17836	20384	12	12
mux_2	54782	61152	10	12
nand3_1	15288	29302	48	6
and3_1	17836	19110	7	8
and2_1	12740	11466	-11	6
nor2	21658	24206	11	4
or2_1	12740	11466	-11	6
or2_2	12740	14014	9	6
or3_1	15288	14014	-9	8
oa_1	17836	19110	6	8
oa_2	17836	20384	12	8
ao_1	17836	17836	0	8
ao_2	20384	22932	11	10
oai_1	66248	90454	27	8
aoi_1	87360	103194	15	8
clk_1	85358	98098	13	2
dffscn_1	67522	64974	-4	36
dffscn_2	71344	64974	-10	36

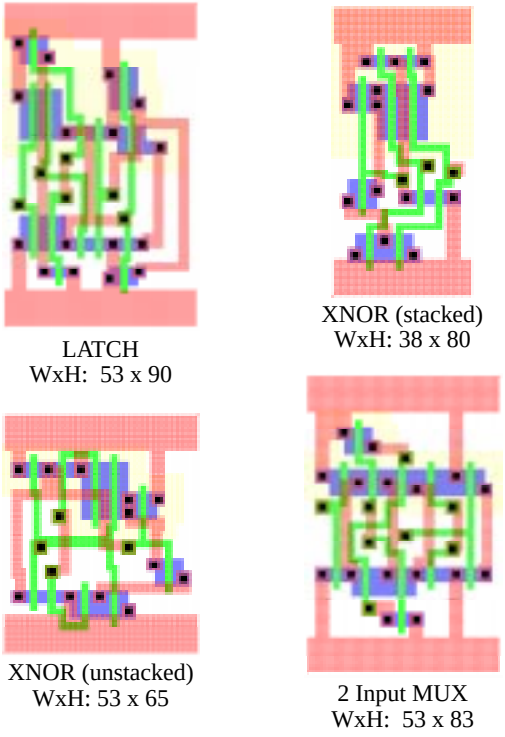


Fig 5. Example Layouts in MOSIS 2.0µ Technology.

4.2 Block Level Results

To compare the quality of CELLERITY-generated libraries with manually designed libraries, commercial place-and-route tools were used to generate standard-cell blocks. We have used a fully-synthesizable industrial microcontroller block to compare the block-level area results. This block has 2223 nets, 58 unique standard cells and 2153 instances of these cells. This block was synthesized using both a manually-designed library and an automatically synthesized library. The CELLERITY-generated block area was the same as the block area using the manually-designed library.

4.3 Run Time

The results reported in this section are for a Sun SPARC 20/61 workstation. CELLERITY run time is dependent on the number of transistors in the netlist, the size of transistors in the netlist, and the level of optimization chosen by the user. On average, the run time varies from 1-15 minutes for cells with 2-40 transistors. The library mentioned in Section 4.1 was generated in 2 days. The block-level experiments, which included generation of standard cells for the block, cell-level netlist and design rule verification, floorplanning, and place and route of the block using commercial place and route software, took less than 1 day. The block was functional, design rule correct, and the same as the manual library area.

4.4 New Process Technologies

CELLERITY has been a key enabling technology in reducing the time to market for new processes. As an example, a brand new standard-cell library for a new sub-micron process technology was generated in less than a week and a test chip (shown in Figure 6) was sent for fabrication within a few weeks. This test chip comprised 22,268 standard-cell instances and 23,940 nets with 93 unique cell types. Because of the quick turn-around time, it was possible to experiment with different heights for the library and select the optimal one for the chip. To the best of our knowledge, this block/chip

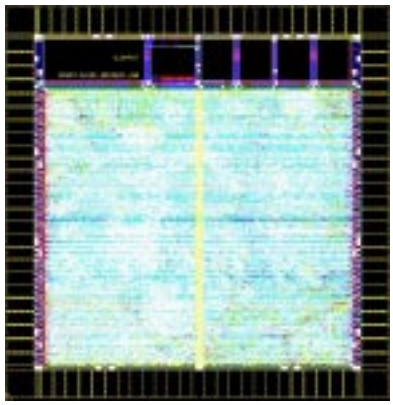


Fig 6. A Design Targeted for a new Process using CELLERITY.

specific library optimization methodology has never before been used on any real industrial circuit.

5 CONCLUSIONS

In this paper, we have described a fully-automatic standard cell layout synthesis system. The tool is flexible enough to handle many process technologies and a wide variety of layout styles. In addition to area, both performance and yield are considered when generating the layout. Experimental results demonstrate the effectiveness of the tool in generating standard cell libraries which are as dense as manually designed libraries, with very quick turn-around time and no manual intervention. This technology has also proved to be invaluable in optimizing the cell library template for improved chip density. CELLERITY is currently in use for augmenting existing standard cell libraries as well for generating new libraries for new processes.

6 REFERENCES

[1] T. Uehara and W. M. vanCleemput, "Optimal layout of CMOS functional arrays," *IEEE Transactions on CAD*, vol. 30, No. 5, May 1981, pp. 305-312.

[2] Dwight D. Hill, "Sc2: A Hybrid automatic layout system," *Proceedings of ICCAD-85*, pp. 172-174.

[3] A. Domic, S. Levitin, N. Phillips, C. Thai, T. Shiple, D. Bhavsar, and C. Bissel, "CLEO: a CMOS layout generator," *Proceedings of ICCAD-89*, pp. 340-343.

[4] K. Tani, K. Izumi, M. Kashimura, T. Matsuda and T. Fujii, "Two-dimensional layout synthesis for large-scale CMOS circuits," *Proceedings of ICCAD-91*, pp. 490-493.

[5] C. C. Chen and S.-L. Chow, "The layout synthesizer: An automatic netlist-to-layout system," *26th ACM/IEEE Design Automation Conf.*, 1989, pp. 232-238.

[6] R. L. Maziasz and J. P. Hayes, *Layout Minimization of CMOS Cells*, Boston, Kluwer Academic Publishers, 1992.

[7] S. Wimer, R. Pinter, and J. Feldman, "Optimal chaining of CMOS transistors in a functional cell," *IEEE Transactions on CAD*, vol. 6, No. 5, September 1987, pp. 795-801.

[8] C. J. Poirier, "Excellerator: custom CMOS leaf cell layout generator," *IEEE Transactions on CAD*, vol. 8, No. 7, July 1989, pp. 744-755.

[9] T. Sadakane, H. Nakao, and M. Terai, "A new hierarchical algorithm for transistor placement in CMOS macro cell design," *Proceedings of CICC-95*, pp. 461-464.

[10] A. Gupta, S. The and J. P. Hayes, "XPRESS: A cell layout generator with integrated transistor folding," *Proc. of European Design and Test Conference*, March 1996, pp.393-400

[11] B. Guan and C. Sechen, "An area minimizing layout generator for random logic blocks," *Proc. of CICC-95*, pp.457-460.

[12] T. Her and D. Wong, "Cell area minimization by transistor folding," *Proceedings of Euro-DAC*, 1993, pp.172-177.

[13] M. Guruswamy and D. F. Wong, "Echelon: A multilayer detailed area router," *IEEE Transactions on CAD*, vol. 15, No. 9, September 1996, pp. 1126-1136.

[14] S. Shah, "SQUEEZE: A graph based one-dimensional compactor," Master's Thesis, Dept. of Electrical and Computer Engineering, Univ. of Texas at Austin, 1991.

[15] C.-L. Ong, J.-T. Li. and C.-Y. Lo, "GENAC: An automatic cell synthesis tool," *26th ACM/IEEE DAC*, 1989, pp.239-244.

[16] Y.-C. Hsieh, C.-Y. Hwang, Y.-L. Lin, and Y.-C. Hsu, "Lib: A CMOS cell compiler," *IEEE Transactions on CAD*, vol. 10, No. 8, August 1991, pp. 994-1005.

[17] S. Chakravarty, X. He, and S. Ravi, "Minimum area layout of series-parallel transistor networks is NP-Hard," *IEEE Transactions on CAD*, vol. 10, July 1991, pp. 943-949.

[18] A. Stauffer and R. Nair, "Optimal CMOS cell transistor placement: a relaxation approach," *Proceedings of ICCAD-88*, pp. 364-367.

[19] P. Kollaritsch and Neil H. E. Weste, "TOPOLOGIZER," *IEEE Journal of Solid-State Circuits*, vol. sc-20. No.3, June 1985. pp.799-803.

[20] W.H. Crocker, Ravi Varadarajan, and Chi-Yuan Lo, "MACS: A module assembly and compaction system," *Proceedings of ICCD-1987*, pp. 205-208.

[21] J.-F. Lee and D. T. Tang, "VLSI layout compaction with grid and mixed constraints," *IEEE Transactions on CAD*, vol. 6, No. 5, September 1987, pp. 903-910.

[22] David G. Boyer, "Symbolic layout compaction review," *25th ACM/IEEE Design Automation Conf.*, 1988, pp. 383-389.

[23] W. L. Schiele, "Improved compaction by minimized length of wires," *20th DAC*, 1983, pp. 121-127.