

An Efficient Layout Style for Three-Metal CMOS Macro-Cells

F. Moraes

PUC-RS - Av. Ipiranga, 668

CEP 90619-900, Porto Alegre, Brazil

Phone: +55(0)513391511, FAX: +55(0)513391564

e-mail: moraes@andros.pucrs.br

R. Reis, F. Lima

UFRGS - CPGCC - Av. Bento Gonçalves 9500

CP 15064 - CEP 91501-970, Porto Alegre, Brazil

e-mail: reis@inf.ufrgs.br

Abstract

This paper presents a new layout style for macro-cells, using three-metal layers CMOS processes with stacked vias. The routing method between leaf cells reduces the track density up to 50% (compared to a double-layer router), allowing an important area reduction. The motivation to develop this work is the requirement in submicron technologies: smaller area, small delay and less power consumption. To attain these requirements, the main cost function in all algorithms (cell generation, placement, routing) is the parasitic capacitance reduction. Our contribution is the new layout style and the associated routing method.

Key-Words

Automatic Layout Synthesis, Layout Style, CAD, Routing.

1. INTRODUCTION

The standard-cell approach is currently considered by VLSI designers the best solution to synthesize large random logic blocks. Since the cells in a library are pre-characterized, it is easy for the designer to simulate large blocks without synthesizing the layout. This leads to accurate delay estimation of the final layout. Then, why using and developing automatic layout synthesis tools? The answer is performance and power reduction. The main advantage of automatic layout synthesis is the possibility to individually size each transistor in the circuit according to delay constraints. In this way, the designer chose transistor sizes according to the area-delay-power trade-off.

The basic leaf cell layout style was defined by Uehara (1981), which we call from now on *linear-matrix*. The linear-matrix style was the basis for many other styles in the 80's. In this style, the main cost function is the diffusion gap minimization, aiming at area and side-wall capacitance reduction. A cell with no diffusion gaps has minimal area and minimal parasitic capacitance values. Another example of a popular layout style in the 80's is the gate-matrix (Lopez, 1980). The gate-matrix style has a great number of diffusion gaps and consequently poor performance due to the associated parasitic capacitances. It is important to remark that the parasitic capacitance reduction was not so relevant in "micron" technologies, where the cell delay is normally much bigger than the associated routing delay. However, in submicron technologies, cell and routing delay are comparable, justifying new cell synthesis and routing methods. Also, in submicron technologies there are several routing layers and stacked vias, which increases the need to change the traditional concepts of layout style and routing.

THEDA system (Hwang, 1993) adopts a layout style with power/ground rails routed in the first metal layer between, instead of outside, the diffusion rows. The main advantages of this style are: (i) the possibility to create large transistors, with smaller impact in the silicon area (the transistor can be extended in the channel routing); (ii) the polysilicon length is reduced (as it has poor conductivity, it is important to avoid wires in this layer); (iii) the routing is simplified. At present, some technologies are using silicides over polysilicon (polycides) and contacts, increasing its conductivity, allowing to create local wires with polysilicon.

In Lin (1994 and 1996) is considered the transistor sizing problem, using the THEDA system (Hwang, 1993). This style was used due to the facility to create large transistors. Traditional layout styles require all cells in a row with the same height. This constraint lead to either wasted area in small cells or layout failure in large cells. Because the power/ground rails are interposed among transistors in the layout style of THEDA this constraint can be relaxed. This allows that transistors widths be arbitrarily increased as needed.

In Kim (1994) is presented a cell model, also with power/ground rails among transistors, with input/output pins between power/ground rails, in the middle of the cell. The first metal layer is used at cell level to create internal connections and metal2/metal3 for connection between cells. The routing is performed over transistors, resulting in a channelless approach.

In Fukui (1995) is presented a two-dimensional transistor arrangement, instead of a linear arrangement. In this style, the main cost function is the optimal transistor placement (horizontally and vertically), instead of the number of diffusion gaps. Cell area is comparable to hand crafted design, but the number of diffusion gaps is high. The authors take into account only silicon area, and there is no electrical simulation data in the paper. In Kim (1992) another layout style is presented, with transistor placement perpendicular to power rails. This style allows to individually size each transistor, keeping the layout area as small as possible. The handicap of it is the routing area: there are vertical wires to connect cell gates and horizontal wires to connect different cells, resulting in small transistor densities.

Routing with three metal layers has been used mainly within the over-the-cell (OTC) approach (Terai, 1994; Kim, 1996). The goal of the OTC is to reduce track density, employing one metal layer over the cells. In OTC routing, the cell layout area as well as the channel area between two cell rows are used as routing resources. Leaf cells must have "transparency" to one metal layer (for example, metal3), usually with power/ground rails into the middle of the cell in order to reduce obstacles. In Kim (1996) is showed that an OTC router achieves area savings from 21 to 50% in benchmark circuits.

Our **objective** is to present a new layout style with the following features: (i) parasitic capacitance reduction; (ii) small silicon area; (iii) routing with 3 metal layers; (iv) minimum use of polysilicon layer; (v) transistors with arbitrary widths.

The rest of this paper is organized as follows: section 2 analyzes the layout style at cell level, section 3 presents the main algorithms to synthesize macro-cells (partition and routing), section 4 discusses some preliminary results as well as our conclusions and directions for future work.

2. THE NEW LAYOUT STYLE

In our approach, at the cell level, transistors are still placed using the linear-matrix style, i.e., two horizontal diffusion lines parallel to power/ground rails. In this style transistors are connected, as much as possible, by abutment, aiming to reduce diffusion capacitances. Figure 1 shows a serial connection between two transistors and the associated parasitic capacitance (1.0 μm technology, capacitances for the diffusion layer: $C_{\text{area}} = 0.31 \text{ fF}/\mu\text{m}^2$ and $C_{\text{side-wall}} = 0.45 \text{ fF}/\mu\text{m}$).

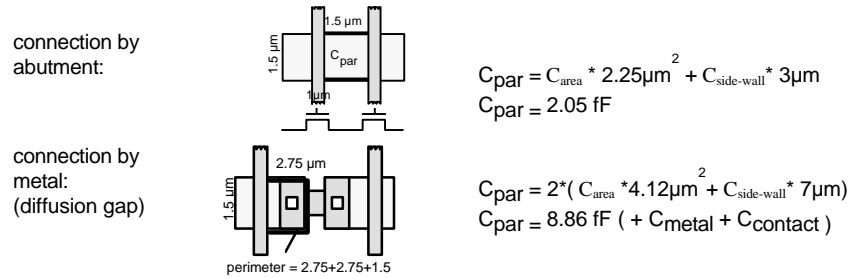


Figure 1 Parasitic capacitance of serial connected transistors

For this technology, the input capacitance of a minimum inverter is 4.31 fF, smaller than the capacitance generated by a diffusion gap. This example shows why transistors must be connected by abutment and how important it is to reduce parasitic capacitances in order to improve performance.

The power and ground rails are placed over the transistors, using the second metal layer (*metal2*). The connection between drains/sources to the power/ground rails is done with stacked vias (via1 over contact). The contacts to the substrate (body-ties) are placed in the routing region, above the PMOS diffusion row and below the NMOS diffusion row. Figure 2 illustrates the transistor placement, supply rails, contacts to the substrate and the polysilicon wires. This solution has the following advantages:

- The silicon area may be smaller, since now there is no exclusive area for supply rails (they are placed over the transistors).
- The length of the polysilicon wires is smaller, since the NMOS and PMOS transistors are placed close to each other. As the polysilicon has poor conductivity ($RC_{\text{poly}} \approx 5 \cdot RC_{\text{metal1}}$), this feature improves the electrical performance of the cells. In our layout style the polysilicon layer is only used to connect transistor gates, except for short segments to solve cyclic constraints in the routing region.
- The transistors can be freely sized.

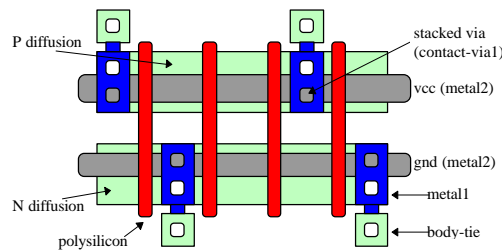


Figure 2 Transistors, body-ties, supply rails and polysilicon wires

The drains/sources connected to a cell output are routed with the first metal layer (*metal1*). A small vertical wire (*metal1*) connects the output nodes to a horizontal line (also in *metal1*) placed between N and P transistors. This topology is illustrated in figure 3.

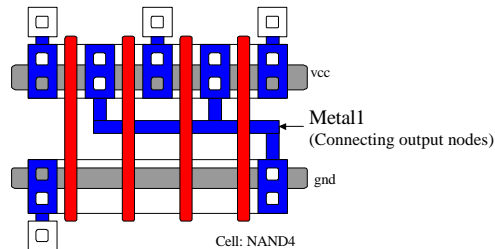


Figure 3 Output nodes

The number of contacts in each drain/source node is a function of the transistor width. In this way, more contacts will be placed in each drain/source, allowing the maximum current through the drain/source nodes to increase. To connect the drains/sources to the routing regions (upper and lower) we use a vertical stub in metal1, to connect the gates we use a vertical stub in polysilicon. In this way, we can see the leaf cells as boxes, with I/O pins at the extremities (figure 4).

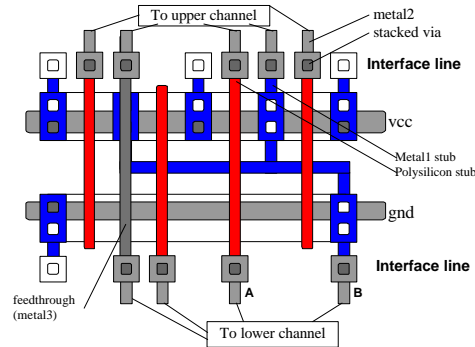


Figure 4 Stubs and "interface line"

This cell model is transparent to the third metal layer, which will be used vertically to make the feedthroughs. There are two types of feedthroughs:

- when there is a net which crosses a row and this net *is not* in this row, in this case it is used a vertical wire in metal3 over the row.
- when there is a net which crosses a row and the net *is* in this row, in this case it is used the I/Os pins of the cell which has the net (nets "A" and "B" in figure 4).

The line between transistors and the routing regions is called "interface line". In this line, stacked contacts will be placed, since non-adjacent layers will be connected (polysilicon to metal2, metal1 to metal3 and optionally polysilicon to metal3). In this line it will be placed contacts between nets of the routing regions with the I/O pins of the cells, feedthroughs and body-ties.

This model, with I/O pins in the extremities, can be inefficient for large transistors. The reason is the obstacle induced by the contact between the polysilicon wire and the routing region. A simple solution consists in placing a vertical wire, in metal1, over the polysilicon wire, with a contact in the middle of the cell (the horizontal routing of the output nodes must be now in metal2). In this way, the interface between gates and the channel routing will be done by a via, with no obstacles to transistor sizing.

This was not employed as we use uniformly sized transistors and buffer insertion (inverters with 2 or more parallel transistors) after nodes having the output capacitance that exceed the fanout limit (Turgis, 1995). This solution minimizes the parasitic capacitances, mainly the side-wall capacitance. Layouts with very large transistors tend to increase excessively the parasitic capacitances, increasing delay and power consumption.

To summarize, the layers are used in the following directions:

	<i>diffusion</i>	<i>polysilicon</i>	<i>metal1</i>	<i>metal2</i>	<i>metal3</i>
cell level	H	V	V/H	H	V
routing channel*	-	-	H	V	H

*H-horizontal, V-vertical, *preferential direction before optimization steps.*

3. OUTLINE OF THE MACRO-CELL SYNTHESIS TOOL

To generate a macro-cell, our system executes the following tasks: leaf cell extraction, cell generation (transistor pairing), partition and placement, global routing and detailed routing.

The leaf cell extraction isolates all basic cells of the input netlist (in Spice format, at the transistor or gate level). Basic cells are simple cells, represented by a dual graph, with n inputs and 1 (one) output. Our system deals with CMOS static gates and transmission gates. There are two advantages in using cell generation: (i) the first one is the possibility to individually size all transistors, according to the delay constraints, and (ii) to use complex gates (and-or-inverter or simply AOIs). The advantages in using AOI gates are: smaller area, delay and power. Comparing a standard-cell library with automatic cell generation (mapping with AOI gates, limited to 4 serial connected transistors), the average reduction in the transistor count is 35% (Reis, 1995).

The next step, cell generation, fixes the transistor order of all basic cells. The main cost functions are: minimization of diffusion gaps and minimization of intra-cell routing (cell height reduction). The algorithm tries to find the same Euler path in both graphs (N and P plan) of each cell. If there is at least one common path between N and P plan, the cell will be generated with no diffusion gaps. The cell area is proportional to the number of inputs.

Partition and placement are a single task. We use the quadrature algorithm (Fidduccia, 1982), with pin propagation (item 3.1). Global routing is executed after cell placement. The main function of the global routing is to determine where feedthroughs will be inserted. We can use vertical wires (metal3) over the cells or I/O pins of the cells, as explained in the previous section. The global routing generates a list of channels, which will be implemented according to the method described in section 3.2.

The result of these tasks is a symbolic description of the macro-cell. Except for the transistor sizing, this description is fabrication process independent, allowing the use of any CMOS three-metal layer with stacked contacts process. This symbolic description is translated into layout by a compactor tool, e.g, the ones included in the CADENCE or MENTOR frameworks.

3.1. Partition and Placement

A performance driven algorithm must consider the following cost functions:

- avoid long wires;
- avoid congestion areas into the channels, typically in the middle;
- reduce the distance between cells of the critical path(s).

We use the quadrature placement as basic algorithm. The quadrature placement (figure 5) alternatively divides the circuit into horizontal and vertical directions, minimizing the cutsize in each direction (Fidduccia, 1982). The circuit is initially partitioned in the vertical direction, using the min-cut algorithm, into two blocks with the same area. This first partition reduces the routing density in the middle of the channels, avoiding congestion in these areas. Next, each area is partitioned into two blocks with the area proportional to the number of rows. For example, for 7 rows there might be an area ratio 4:3, for 4 rows 2:2 and so on. The partition procedure stops when no more horizontal partitions are possible (area ratio 1:0). This process results in a set of quadrants, with few cells (typically 2-8) in each one. The cells in each quadrant can be placed using a simple algorithm. In our placement, cell order is obtained directly from the connectivity between cells.

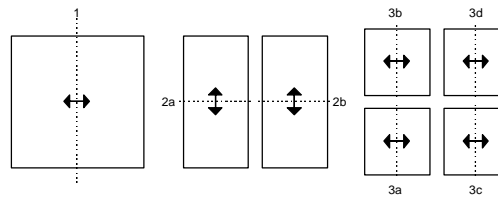


Figure 5 Quadrature placement

The main problem of the quadrature is the placement of cells with common nets in non neighbour quadrants. For example, in figure 5, consider two cells with a common net between quadrants 2a and 2b. After vertical partition, these cells might be placed in quadrants 3b-3d, 3b-3c, 3a-3d or 3a-3c. In order to reduce the wire length, it will be interesting to place these cells in quadrants 3b-3d or 3a-3c.

To improve the efficiency of the quadrature placement, we implemented a pin propagation method. Pin propagation tries to place cells with common signals in adjacent quadrants, reducing routing length. The main idea is the following: to make a partition within a quadrant, each quadrant already partitioned is also taken into account. For the horizontal partition, the quadrants are processed line by line, from left to right. For the vertical partition, the quadrants are processed column by column, from bottom to top.

This algorithm avoids long wires and congestion, distributing homogeneously the connections in both directions (vertical/horizontal), reducing the track density. This is a fast algorithm, it takes for example, 2.22 minutes for a circuit with 3156 gates in a Sun Sparc5.

The next function to include in this placement procedure is the information related to the critical paths. This information will guarantee that cells in the critical path will be placed closer.

3.2. Detailed Routing

The track density (lower bound) in each channel is defined by the clique of the horizontal graph. If the routing procedure uses only two layers to create wires, the track density will be equal to the clique, if there are no vertical cycles. If we have 3 layers, the channel can be folded in the middle, making it possible to reduce routing area up to 50%. As routing area is responsible for 60% of the circuit area (average value for random logic blocks), we can expect a reduction of 30% in the total circuit area.

The track density in our routing method is equal to the clique/2 (best case), superposing horizontally metal1 and metal3, using metal2 in the middle. Another advantage of our method is the great number of suppressed vias, typically 40-55% (see table 1), also reducing parasitic elements.

Our routing algorithm is divided into four steps:

- initial double-layer routing;
- metal2-to-metal1 and metal2-to-metal3 track transformation (vertical filter);
- metal1-to-metal2 and metal3-to-metal2 track transformation (horizontal filter);
- cycles solution.

The first step uses a conventional double-layer greedy router (Rivest, 1982). When the double-layer router is finished, the odd tracks are superposed to the even tracks. Figure 6a shows the double-layer solution (in fact, there are 3 layers, two horizontal and one vertical), and figure 6b the layer superposition. A 3D illustration of the channel is shown in figure 6c. The channel can be seen as 2 channels, an upper channel (metal3) and a lower channel (metal1), with a connecting layer between them (metal2).

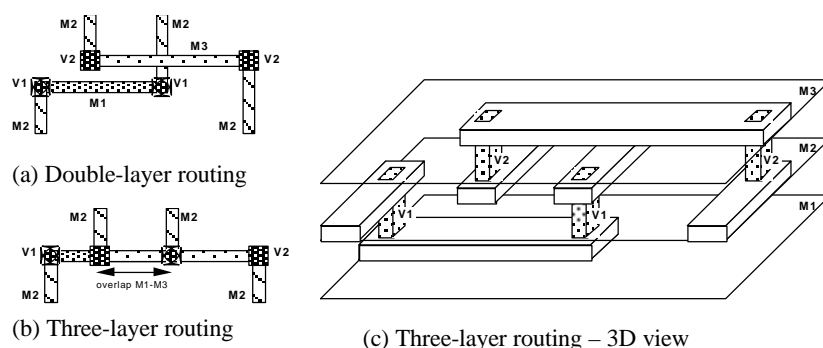


Figure 6 Routing approach

The second step transforms metal2 vertical wires into metal1 or metal3 vertical wires, suppressing unnecessary vias. The average suppression rate is 35 to 40%, in each channel. The third step transforms metal1/metal3 horizontal wires into metal2 horizontal wires. The average suppression rate in this step is 5%. In the channel routing there are no stacked vias. All stacked vias are in the "interface line" or in the power/ground connections. The "interface line" is responsible for making the connections between cells and channels.

The fourth step solves via cycles. They arrive when there are adjacent contacts in the same column, resulting in a via2 superposed to a via1, and consequently a short-circuit. In the greedy router, there is a procedure to avoid cycles, since the track density is not increased. In this way, cycles can happen because the procedure to avoid them in the router may fail. These cycles are solved in the following manner:

- usually, at least one via can be suppressed (vertical filter);
- if the cycle remains, we use an "underground" solution, changing the vertical metal2 wire, connected to the horizontal metal1 wire, to polysilicon (a fourth layer). In this way, we will have a contact under a via2, without electrical connection. Experiments under benchmark circuits show that this case occurs occasionally, usually up to 4 times in each channel. This vertical polysilicon is a short wire, with no impact in the parasitic capacitance of the cell.

Table 1 illustrates in the third column the number of vias when using a greedy router with no optimizations (filters), in the fourth column the number of vias after vertical and horizontal optimizations, in the fifth column the via reduction rate and finally the total number of "underground" solutions in these circuits. The number of vias is in the channel routing, without the vias of the "interface line".

Table 1 Number of vias after horizontal/vertical filters (C^* are ISCAS benchmarks)

<i>Circuit</i>	<i>Transistor Number</i>	<i>Vias# Greedy</i>	<i>Vias# Optimized</i>	<i>Reduction %</i>	<i>"underground" solutions</i>
adder 2 bits (AOIs)	28	31	8	0.74	0
adder 2 bits (gates)	40	34	0	1.00	0
c432	150	122	18	0.85	0
alu 4 bits (AOIs)	260	320	138	0.57	5
alu 4 bits (gates)	424	451	224	0.50	1
ripple carry adder 12 bits	448	624	233	0.63	3
carry lookahead adder 12 bits	528	725	270	0.63	3
booth multiplier 4 bits	824	831	286	0.66	5
c880	1750	2173	1086	0.50	30
c1355	2244	2710	1269	0.53	28
c1908	3156	4126	2124	0.49	53

This routing procedure can be also used in cell-based approaches, aiming area and delay reduction. Compared to a double-layer router, this method guarantees a **half number** of tracks. As polysilicon conductivity can be increased using silicides, it will be interesting to study the resulting number of tracks for a 4 layers routing.

The resulting macro-cell has no transparency because all layers are being used while routing. So, at the circuit level it will not be possible to implement the routing over macro-cells. One solution is to estimate the number of horizontal and vertical feedthroughs in each macro-cell, at the floorplane step of the chip.

4. RESULTS

It will be used the layout synthesis tool TROPIC (Moraes, 1994) for area and delay comparisons. This tool uses the linear-matrix layout style, without channel routing between rows. Horizontal routing is implemented in *metal1*, between transistors, and the vertical routing in *metal2*. TROPIC was compared to LAS (Cadence, 1991), an industrial layout generator, obtaining equivalent values for area and delay. Then, TROPIC will be the reference for two metal layer layout generation.

Table 2 presents values for area and transistor density for TROPIC and our new layout style (*3 metal*). As expected, silicon area was reduced by 20 to 30%. The average transistor density is 4600 tr/mm² (w=10μm) and 5500 tr/mm² for minimal sized transistors. The exception was the ripple carry circuit, only 10%. To reduce this difference, it is necessary to insert jogs in the compaction step.

Table 2 Area Comparison

<i>Circuit</i>	<i>Transistor</i>	<i>Area (mm²)</i>		<i>Density (Tr/mm²)</i>		<i>Difference</i>
	<i>Number</i>	<i>TROPIC</i>	<i>3metal</i>	<i>TROPIC</i>	<i>3metal</i>	<i>3metal/TROPIC</i>
adder 2 bits (AOIs)	28	0.0080	0.0057	3500	4912	0.71
adder 2 bits (gates)	40	0.0110	0.0069	3636	5800	0.62
c432	150	0.0446	0.0282	3363	5319	0.63
alu 4 bits (AOIs)	260	0.0764	0.0633	3403	4106	0.82
alu 4 bits (gates)	424	0.1480	0.1158	2864	3662	0.78
ripple carry adder 12 bits	448	0.1179	0.1068	3800	4195	0.90
carry lookahead 12 bits	528	0.1943	0.1315	2717	4012	0.67

1.0 μm technology - l=1μm, w=10μm - Compaction without jog insertion - C_{load} = 100 fF

The macro-cell height is drastically reduced because the track number is reduced to half and the power/ground rails are implemented over transistors. However, the macro-cell length tends to increase, due to the great number of contacts in the “interface line”. To increase the transistor density, it is important to develop a specific layout compactor, adapted to the proposed layout style. The existing compactors are generic tools, highly time and memory consuming.

Table 3 presents values for parasitic capacitances (routing and diffusion), delay of the critical path and average power consumption. As expected, the sum of parasitic capacitances was reduced (28% for alu AOI and 30% for carry lookahead). Consequently, delay and power are also reduced. The small difference in delay and power (5 to 13%) is due to the transistor sizing.

Table 3 Delay Comparison (*1.0 μm technology - $l=1\mu\text{m}$, $w=10\mu\text{m}$ - $C_{load} = 100\text{ fF}$)*

<i>Circuit</i>	<i>Transistor Number</i>	<i>Total Cpar (pF)</i>		<i>Delay (ns)</i>		<i>Power (mw)</i>	
		<i>TROPIC</i>	<i>3metal</i>	<i>TROPIC</i>	<i>3metal</i>	<i>TROPIC</i>	<i>3metal</i>
adder 2 bits (AOIs)	28	0.356	0.310	1.85	1.62	0.36	0.34
adder 2 bits (gates)	40	0.708	0.654	1.61	1.51	0.33	0.27
alu 4 bits (AOIs)	260	4.985	3.541	5.18	4.96	2.54	2.35
alu 4 bits (gates)	424	9.800	7.645	7.99	6.42	3.59	3.10
ripple carry adder 12 bits	448	6.115	5.604	16.24	15.51	8.48	8.18
carry lookahead 12 bits	528	9.139	6.360	13.43	12.70	8.53	8.02

In our examples all transistors are sized to $10\mu\text{m}$, minimizing the influence of parasitic capacitances. To observe the real impact of parasitic capacitances in the layout style, it is recommended to use minimal sized transistors.

5. CONCLUSION AND FUTURE WORK

In this paper a new layout style was presented. It minimizes the diffusion capacitances and the polysilicon length by using three-metal layers for routing. As shown, the sum of parasitic capacitances was reduced, allowing to reduce area, delay and power. Future work includes:

- Develop a specific layout compactor for the presented layout style.
- Study the track reduction when using 4 (or n) metal layers for routing.
- Improve the transmission gate topology in order to avoid waste area.
- Improve the precision of parasitic capacitance estimation.
- Take into account the critical path(s) in the placement algorithm.
- In order to route over the macro-cells, insert feedthroughs to allow crossing wires.

REFERENCES

- CADENCE (1991) Virtuoso Layout Synthesizer - LAS - User Guide. *CADENCE Version 4.2*, October 1991.
- Fiduccia, A.E. and Matheyses, R.M.. (1982) A linear time heuristic for improving network partitions. *19th Design Automation Conference*, pp. 175-181.
- Fukui, M.; Shinomuya, N. and Akini, T. (1995) A New Layout Synthesis for Leaf Cell Design. *Asia South Pacific DAC*, pp. 259-264.
- Hwang, C.; Hsieh, Y.; Lin, Y. and Hsu, Y. (1993) An efficient layout style for two-metal CMOS leaf cells and its automatic synthesis. *IEEE Transactions on CAD*, Vol. **12**, No 3, March 93, pp. 410-423.
- Kim, J. and Kang, S.M. (1996) A New Triple-Layer OTC Channel Router. *IEEE Transactions on CAD*, Vol. **15**, no. 9, September 1996, pp. 1059-1070.
- Kim, J.; Kang, S.M. and Sapatnekar, S. (1994) High Performance CMOS Macromodule Layout Synthesis. *ISCAS'94*, pp. 179-182.
- Kim, S.; Owens, R.M. and Irwin, M.J. (1992) Experiments with a performance driven module generator. *29th Design Automation Conference*, pp. 687-690.
- Lin, H.; Chou, C.; Hsu, Y. and Hwang, T. (1994) Cell Height Driven Transistor Sizing in a Cell Based Module Design. *EDAC'94*, pp. 425-429.
- Lin, H.; Hsu, Y. and Hwang, T. (1996) Cell Height Driven Transistor Sizing in a Cell Based Static CMOS Module Design. *IEEE Journal of Solid-State Circuits*, Vol. **31**, no.5, May 1996, pp. 668-676.
- Lopez, A.D. and Law, H.S. (1980) A Dense Gate-Matrix Layout for MOS VLSI. *IEEE Transactions on Electron Device*, Vol. **ED-27**, No. 8, August 1980, pp. 1671-1675.
- Moraes, F.G.; Robert, R.; Auvergne, D. and Reis, R. (1994) An Efficient Layout Synthesis Approach for CMOS Random Logic Circuits. *IX SBMICRO - Congresso da Sociedade Brasileira de Microeletrônica*, pp. 48-57.
- Reis, A.; Robert, M.; Auvergne, D. and Reis, R. (1995) Associating CMOS Transistors with BDD Arcs for Technology Mapping. *Electronic Letters*, Vol. **31**, No 14, July 1995.
- Rivest, R.L. and Fiduccia, C.M. (1982) A Greedy channel router. *19th Design Automation Conference*, pp. 208-219.
- Terai, M.; Nakajima, K.; Takahashi, K. and Sato, K. (1994) "A New Approach to Over-the-Cell Channel Routing with Three Metal Layers". *IEEE Transactions on CAD*, Vol. 13, No 2, February 94, pp. 187-200.
- Turgis, S.; Azemard, N. and Auvergne, D. (1995) Design and Sizing of Tapered Buffers for Minimum Power-Delay Product. *PATMOS'95*, pp. 74-90.
- Uehara, T. and Cleemput, W. (1981) Optimal Layout of CMOS Functional Arrays. *IEEE Transactions on Computers*, Vol. **C-30**, No. 5, May 1981, pp. 305-312.

More information: <http://www.inf.pucrs.br/~moraes/tropic.html>