Current-Driven Wire Planning for Electromigration Avoidance in Analog Circuits

Jens Lienig
Dresden University of Technology
Dresden, Germany
jens@ieee.org

Göran Jerke
Robert Bosch GmbH
Reutlingen, Germany
goeran.jerke@ieee.org

Abstract - Electromigration due to insufficient wire width can cause the premature failure of a circuit. The ongoing reduction of circuit feature sizes has aggravated the problem over the last couple of years, especially with analog circuits. It is therefore an important reliability issue to consider current densities already in the physical design stage. We present a new methodology capable of routing analog multi-terminal signal nets with current-dependent wire widths. It is based on current-driven wire planning which effectively determines all branch currents prior to detailed routing. We also discuss successful applications of our methodology in commercial analog circuit design.

I. Introduction

Layout for analog circuits has historically been a manual, time-consuming, trial-and-error task. A primary reason for the lack of automation is the vast amount of expert knowledge typically required to meet constraints such as electrical/thermal symmetry, current densities (including electromigration), voltage drops, temperature gradients, etc.

Unlike digital circuits, analog circuits must handle a multitude of different current levels, including extremely large currents in some applications (such as automotive circuits). Hence, the interconnect must be designed with the current that will be imposed on it in mind. Interconnect with an insufficient width (we assume the height to be a constant as given with many processes) may be subject to electromigration and eventually might cause the failure of the circuit at any time during its lifetime [1]-[3].

The ongoing reduction of circuit feature sizes has aggravated the problem over the last couple of years. For this reason it is becoming crucial to address the problems of current densities and electromigration during the routing of the interconnect.

A current-driven router must solve the problem of only partially known currents in net topologies during a sequential routing process. For example, the width of a net connecting two terminals can be easily derived from the terminals' currents. However, if a third terminal is subsequently connected with this net using a Steiner point, the current flow would change and might make the previous route obsolete (e.g., if the wire cannot be widened). Generally speaking, any new connection to a previously routed sub-net may alter the currents imposed on the sub-net's paths and hence alter the correctness of its topological layout.

Most approaches to automatic routing of power and ground nets address this problem with a separate post-processing step that includes layout modifications [4]-[11]. While this is feasible in power and ground routing due to its planar nature, limited number of nets and (still) unoccupied layers, current-driven routing of signal nets requires a different approach.

In this paper we present a new methodology for current-driven routing that has been successfully tested in industrial design flows. Our approach consists of current characterization, current-driven wire planning and a conventional detailed routing with variable wire widths. Wire planning determines the estimated routing path of a net by calculating a routing tree with minimized, current-dependent wire area. Since currents have already been taken into account during this planning phase, the detailed routing can then be considered point-to-point routing with known terminal currents at both end points, thus avoiding the above mentioned problems (such as post-route modifications of the layout).

The contributions of this paper are:

- a fast, yet sufficiently exact current characterization method based on reduced current vectors,
- a reliable wire width determination which includes a flexible handling of temperature variations,
- current-driven wire planning which minimizes the surface area of wires (rather than only the lengths) and effectively determines *all* branch currents *prior* to detailed routing, and
- a verification of our method on "real world" analog circuits.

This approach is to our knowledge the first commercially applicable wire planning algorithm focused on current-driven wire widths.

II. The Electromigration Problem

The copper or aluminum interconnects of a chip are polycrystalline, i.e., they are made up of grains of lattice. While conducting a current though this interconnect, the electrons will interact with the lattice imperfections. Specifically, atoms can be transported at the boundaries between the grains as a result of the "electron wind". In the direction of the electron flow, copper or aluminum atoms will be deposited over time (so-called "hillocks"), while in the opposite direction "voids" will grow between some grain boundaries (Fig. 1). While the hillocks might introduce

shorts to neighboring wires, voids will reduce the conductivity of the interconnect over time which eventually could stop the interconnect to conduct at all.

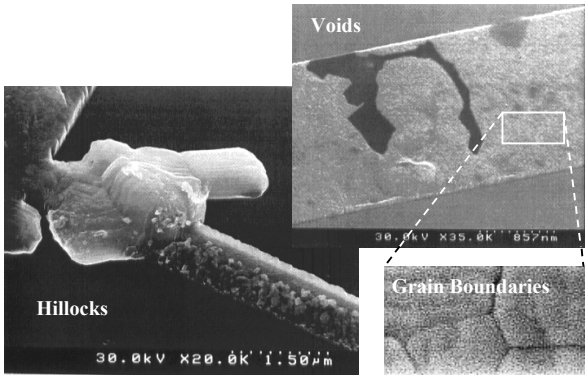

Fig. 1: Hillock and void formations in wires due to electromigration (Photo courtesy of G. H. Bernstein and R. Frankovic, University of Notre Dame)

The described mass transport in metals due to the electron wind is labeled with the term "electromigration". The failure of a single wire due to electromigration will often cause the entire chip to fail.

Most chips must have a mean time to failure (MTTF) of at least 10 years. Failure due to electromigration of a single wire is usually expressed by Blacks equation [1]:

$$MTTF = \frac{A}{J^2} \cdot \exp\left(\frac{E_a}{k \cdot T}\right) \quad (1)$$

where A is a material constant, J is the current density, E_a is the activation energy, k is the Boltzmann constant, and T is the temperature.

As is obvious from Equation (1), the MTTF due to electromigration depends on two factors that can be influenced by the chip designer: temperature and current density. This article will focus on current density as the major parameter to address electromigration during physical design. Specifically, we will focus on DC currents where the metal is subject to an electron wind from a constant direction, such as in power supply wires and in analog signal wires. (AC currents, as common in most digital applications, have a "self-healing effect" due to the alternating current direction.)

The ongoing reduction of circuit feature sizes has aggravated the DC electromigration problem over the last couple of years. For this reason it is becoming crucial to consider current densities already during the physical design stage, i.e., during the routing of analog and power lines.

III. Previous Works

With the exception of the approaches in [12]-[14], current-driven routing has been applied so far only to layout generation of *power and ground nets* in digital circuits. In these cases, the generation of power supply interconnect is usually done prior to signal routing in order to achieve a planar (i.e., single layer) implementation.

The first automatic approaches to power and ground routing were presented in the 1980's [4]-[9], and usually involve three steps: interconnection topology construction, wire width determination, and layout generation. The interconnection topology is determined by using a standard wire width, and then based on that topology, branch currents are calculated. Afterwards, all wires are widened with respect to their current flow. This might result in DRC errors that must be resolved in a separate post-processing step which may require modification of the cell placement.

Recently, a floorplan-based planning methodology for power connections has been presented [10]. Here, a global power trunk and a block-level local power network are first generated from the floorplan, and then optimized regarding their widths. An alternative approach to optimizing power and ground networks is described in [11], where the authors present a fast linear programming method that optimizes the power and ground area subject to current density and IR-drops.

All these approaches to current-driven routing are limited to routing of power and ground nets and require some form of post-route modifications. These modifications are necessary because currents are only known after the entire topology has been laid out – a characteristic that prevents such methodologies from being applied to current-driven routing of signal nets.

The approaches of current-driven *signal* routing in [12]-[14] are focused on generating current-correct Steiner trees. The proposed current characterization method is very simplified (considering only the minimum and maximum currents at each pin), the Steiner tree generation is based on a connection graph which can only be applied to small problem sizes, and the solutions of [12][13] are restricted to planar layouts. These characteristics limit such approaches from being applied to "real world" commercial circuits.

IV. Design Flow

The design flow of our approach is illustrated in Fig. 2.

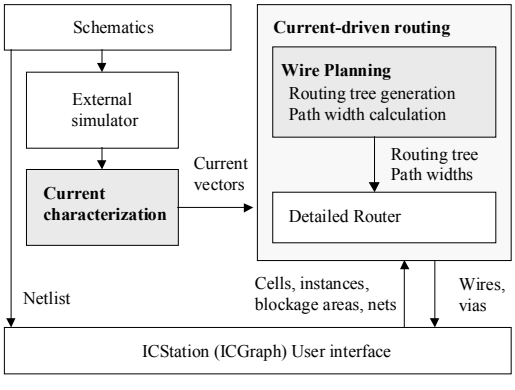

Fig. 2: Design Flow

During current characterization, current vectors attached to each terminal are obtained. These vectors are transferred

to the routing tools either as part of the schematic netlist or as an ASCII file.

Our new routing methodology has been integrated into the Mentor Graphic ICStation environment which reads the netlist from the schematic tool. After the initial placement of the cells is generated, the main layout components (cells, instances, blockage areas and nets) are forwarded to the routing tool.

During wire planning, a routing tree of the next net to be routed is generated that takes into account the current-correct widths of the connections. The objective of this routing tree is to minimize the wire area (rather than simply the length of the wires) and to calculate "intermediate" current vectors according to the topology of the routing tree. The resulting routing tree and the calculated path widths are transferred to a detailed router. Here a simple terminal-to-terminal path generation based on the sequence and width given by the routing tree is performed. Finally, the generated paths and vias are returned to the main layout tool (ICStation).

V. Current Characterization

A problem for a current-driven design methodology is the determination of realistic current values for each terminal. We utilize two approaches: One method uses a standard circuit simulator for simulation of the circuit netlist ignoring parasitic wiring resistances. The second approach uses current values manually attached to the terminals in the schematic netlist by the designer. Additionally, every terminal is labeled with its root mean square (RMS) current value (derived from *all* simulation values at this terminal).

	T_1	T_2	T_3 ... T_n
t_1	7	-2	6 ...
t_2	14	-4	3 ...
t_3	12	-2	-1 ...
:	:	:	:
t_{last}	2	1	-4 ...
I_{eq}	7.35	2.73	4.01 ...

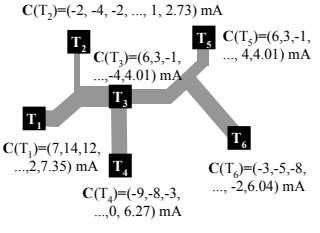

Fig. 3: A vector of current values $C(T_i)$ and their respective RMS current value I_{eq} are attached to each terminal T_i.

The unpredictable curve shape of arbitrary analog signals might result in a vast amount of simulation data. We introduce a simple method in order to achieve a noticeable reduction of vector elements. It is based on the observation that with increasing numbers of current vector elements their actual influence on the final RMS current value I_{eq} is decreasing. Hence, all "late" vector elements whose influence is limited such that the RMS current value stays within a given deviation +/- $\varepsilon/2$ can be ignored (Fig. 4).

The deviation ε should be smaller than (s–1), where s is the safety factor in Equations 2 and 3 (see Section VII). Usually ε can be set to $\varepsilon = (0.1 .. 0.2)$.

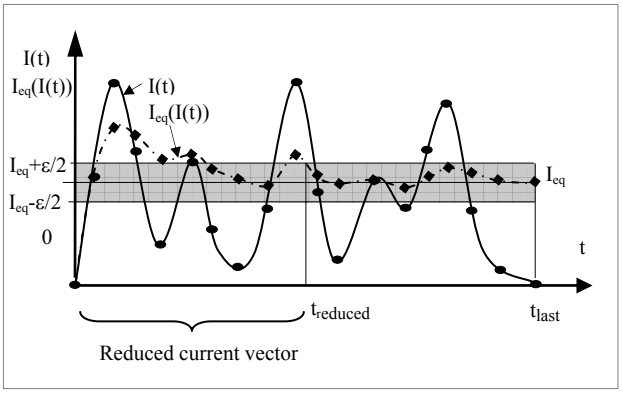

Fig. 4: A given terminal current $I(t)$ and its RMS current value $I_{eq}(I(t))$ are depicted for a time period ($t = 0 .. t_{last}$). The terminal current vector C(T) can be reduced to $I(t = 0 .. t_{reduced})$ if all corresponding $I_{eq}(t)$ "beyond" $t_{reduced}$ are within a specified interval $[I_{eq}-\varepsilon .. I_{eq}+\varepsilon]$.

The same time interval $t = t_1 .. t_{reduced}$ has to be considered for all terminals of a net in order to satisfy Kirchhoff's current law at any point of time within $t = t_1 .. t_{reduced}$. Hence, after calculating $t_{reduced\_1} .. t_{reduced\_n}$ for all terminals $T_1 .. T_n$, a common net-specific $t_{reduced}$ is determined by simply choosing the largest value among $t_{reduced\_1} .. t_{reduced\_n}$.

In the remainder of this paper, the term "current vector" denotes the reduced set of current values attached to each terminal.

VI. Propagation of Current Vectors

In order to generate wires which satisfy the current density restrictions for all possible current states (i.e., current vectors), the (reduced) current vector **C** must be propagated whenever (i) a new terminal is connected with a prior routed sub-net or (ii) a Steiner point is generated.

The involved current vectors (e.g., $\mathbf{C}(T_i)$, $\mathbf{C}(T_j)$, ..., $\mathbf{C}(T_k)$), are added up in order to generate a new current vector $\mathbf{C}(T_iT_j...T_k)$ at the respective terminal T_k or Steiner point (see Fig. 5). The RMS current value I_{eq} of the new terminal or Steiner point is calculated from $\mathbf{C}(T_iT_j...T_k)$.

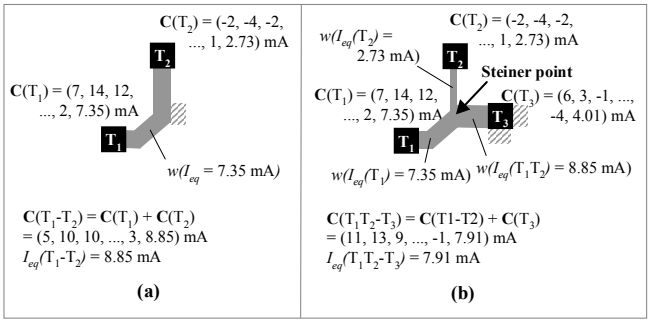

Fig. 5: The new current vector $C(T_1\text{-}T_2)$ in (a) is determined by simply adding up the terminal current vectors of $C(T_1)$ and $C(T_2)$. The new RMS current value $I_{eq}(T_1\text{-}T_2)$ is derived from $C(T_1\text{-}T_2)$. A similar strategy is applied when encountering Steiner points as seen in (b).

VII. Wire Width Determination

Based on the definition of current density ($dJ=dI/dA$), we determine the minimum wire width w_{min} using the maximum and the root mean square (RMS) current values as follows:

$$w_{min} = max \begin{cases} \dfrac{I_{eq} \cdot s}{d_{Layer} \cdot J_{max}(T_{ref}) \cdot f(T)}, & (2) \\ \dfrac{|I_{max}| \cdot s}{J_{peak\_Layer} \cdot d_{Layer}}, & (3) \\ w_{min\_process} & (4) \end{cases}$$

where

I_{eq} = the root mean square (RMS) current on this path,
s = safety factor ($s \approx 1.1 .. 1.2$),
d_{Layer} = thickness of routing layer,
$J_{max}(T_{ref})$ = maximum current density allowed by this manufacturing process for temperature T_{ref} ($J_{max}(150°C) \approx 1..2$ mA/µm^2),
$f(T)$ = temperature correction factor if maximum working temperature T is different than T_{ref} (see Equation (5)),
I_{max} = the maximum current on this path,
J_{peak_Layer} = layer dependent peak current density (process dependent),
$w_{min\_process}$ = minimum wire width determined by manufacturing process.

The safety factor s is used to account for (i) small deviations of d_{Layer} due to process variations, (ii) terminal currents not caught during simulation, and (iii) reduced accuracy of the current vector due to vector element reduction.

In order to allow for a flexible handling of temperature variations, we derived a temperature correction factor $f(T)$ from Blacks law [15]: If the maximum working temperature $T \neq T_{ref}$, $f(T)$ in Equation (2) is set as follows:

$$f(T) = exp\left(-\frac{Q}{n \cdot k \cdot T_{ref}} \left(1 - \frac{T_{ref}}{T} \right) \right) \quad (5)$$

where

Q = experimentally determined activation energy for electromigration failure mechanism ($Q \approx 0.5 .. 1.4$ eV),
n = 2 ("Black's law", [15]),
k = Boltzmann's constant (k = 1.38e-23 J/K),
T_{ref} = reference temperature (used for $J_{max}(T_{ref})$ in Equation (2), usually T_{ref} = 150°C),
T = maximum working temperature.

VIII. Wire Planning

A current-considering detailed router must solve the problem of altering current strengths in a prior routed sub-net whenever a new terminal is linked to it. In order to allow for a current calculation based on Kirchhoff's current laws even when routing only the first segments of a net, the sequence of *all* terminals to be connected must be known. Added connections which *directly* link a new terminal with a predefined target terminal will then have no influence on current strengths calculated in the prior routed sub-net.

Our wire planning step establishes a routing tree which defines this routing sequence for a subsequently applied detailed router. Another objective of wire planning is to minimizes the wire area of a net. Wire planning also provides a straightforward method of calculating the branch currents by applying Kirchhoff's current laws.

The *current connection area* CCA represents the "surface area" of a wire of a particular terminal-to-terminal connection:

$$CCA(T_n, T_m) = d_{n,m} \cdot w_{n,m} \quad (6)$$

where

$d_{n,m}$ = distance between terminals T_n and T_m,
$w_{n,m}$ = wire width of the connection $T_n - T_m$ (see Section VII).

The main objective of our wire planning algorithm is to minimize the overall CCA value for each net.

Input: Terminal topology of a net and attached terminal current vectors $\mathbf{C}(T)$

Output: Routing tree $\boldsymbol{R}$ that contains the sequence of terminals to be connected and the current strengths of the respective routing paths

1. **Create a mesh graph $\boldsymbol{G}$** of terminal mid points by triangulation:
   - 1.1 Store mid points of all net terminals in a terminal mid point list $\boldsymbol{L}$;
   - 1.2 Choose an arbitrary start edge $\boldsymbol{E}$ from convex hull of all mid points in $\boldsymbol{L}$ and put $\boldsymbol{E}$ in a frontier edge list $\boldsymbol{F}$;
   - 1.3 *While* number of edges in frontier edge list $\boldsymbol{F} > 0$:
     - (a) *From* $\boldsymbol{F}$: get the frontier edge $\boldsymbol{E}$ with nodes $\boldsymbol{N_L}$ and $\boldsymbol{N_R}$;
     - (b) *For edge* $\boldsymbol{E}$: determine the nearest mid point $\boldsymbol{P}$ ($\boldsymbol{P} \neq \boldsymbol{N_L}$, $\boldsymbol{P} \neq \boldsymbol{N_R}$) in $\boldsymbol{L}$ so that (distance ($\boldsymbol{N_L}$, $\boldsymbol{P}$) + distance ($\boldsymbol{N_R}$, $\boldsymbol{P}$)) → min
     - (c) Create two new frontier edges ($\boldsymbol{N_L}$, $\boldsymbol{P}$) and ($\boldsymbol{N_R}$, $\boldsymbol{P}$) and put them in $\boldsymbol{F}$;
     - (d) Include frontier edge $\boldsymbol{E}$ in the mesh graph $\boldsymbol{G}$ and remove it from frontier edge list $\boldsymbol{F}$;
     - (e) Remove nodes $\boldsymbol{N_L}$ and/or $\boldsymbol{N_R}$ from $\boldsymbol{L}$ if they are not used by other frontier edges in $\boldsymbol{F}$;
2. **Store all net terminals T in a terminal list $\boldsymbol{S}$**;
3. **In terminal list $\boldsymbol{S}$: Choose a start terminal $\mathbf{T_1}$** with mid point located at the boundary of mesh graph G. Find its nearest neighbor $\mathbf{T_{1,next}}$ in $\boldsymbol{G}$ and assign ($\mathbf{T_1}$, $\mathbf{T_{1,next}}$) to route tree $\boldsymbol{R}$;
4. **Perform generation of routing tree $\boldsymbol{R}$** by taking three terminals into account at each iteration: the current terminal $\mathbf{T_i}$, its preceding terminal $\mathbf{T_{i-1}}$ and the terminal to be considered next $\mathbf{T_{next}}$:

   While number of terminals in terminal list $\boldsymbol{S} > 0$:
   - (a) *For each terminal* $\mathbf{T_i}$ *and* $\mathbf{T_{i-1}}$: get nearest terminals $\mathbf{T_{i,next}}$ and $\mathbf{T_{i-1,next}}$ in $\boldsymbol{G}$ (respectively $\boldsymbol{S}$) which are not yet assigned to $\boldsymbol{R}$;
   - (b) *For connections* $\mathbf{T_{i-1}}$-$\mathbf{T_i}$-$\mathbf{T_{i,next}}$ *and* $\mathbf{T_i}$-$\mathbf{T_{i-1}}$-$\mathbf{T_{i-1,next}}$: calculate the current connection area CCA, e.g., for connection $\mathbf{T_{i-1}}$-$\mathbf{T_i}$-$\mathbf{T_{i,next}}$:
     $CCA(\mathbf{T_{i-1}},\mathbf{T_i},\mathbf{T_{i,next}}) = distance(\mathbf{T_{i-1}}\text{-}\mathbf{T_i}) * w(\boldsymbol{I_{eq}}(\mathbf{T_{i-1}}\text{-}\mathbf{T_i})) + distance(\mathbf{T_i}\text{-}\mathbf{T_{i,next}}) * w(\boldsymbol{I_{eq}}(\mathbf{T_i}\text{-}\mathbf{T_{i,next}}))$;
   - (c) *If* $CCA(\mathbf{T_i},\mathbf{T_{i-1}},\mathbf{T_{i-1,next}}) \leq CCA(\mathbf{T_{i-1}},\mathbf{T_i},\mathbf{T_{i,next}})$:
     - assign terminal $\mathbf{T_{i-1,next}}$ and edge ($\mathbf{T_{i-1}}$,$\mathbf{T_{i-1,next}}$) with current strength $\boldsymbol{I_{eq}}$ to $\boldsymbol{R}$;
     - remove current terminal $\mathbf{T_i}$ from terminal list $\boldsymbol{S}$;
     - set next current terminal : $\mathbf{T_i} = \mathbf{T_{i-1}}$;

     Else :
     - assign terminal $\mathbf{T_{i,next}}$ and edge ($\mathbf{T_i}$,$\mathbf{T_{i,next}}$) with current strength $\boldsymbol{I_{eq}}$ to $\boldsymbol{R}$;
     - remove current terminal $\mathbf{T_{i-1}}$ from terminal list $\boldsymbol{S}$;
     - set next current terminal : $\mathbf{T_i} = \mathbf{T_i}$;

Fig. 6: Outline of the wire planning algorithm

As described in Fig. 6 and illustrated in Fig. 7, the wire planning approach is based on a mesh graph which connects each terminal with its nearest terminal not connected yet (step 1 in Fig. 6). The first two nodes of the routing tree are an arbitrarily selected terminal on the "outer boundary" of

the mesh graph and its nearest neighbor terminal (step 3 in Fig. 6). Adjacent terminals to both nodes are weighted according to the resulting CCA value. The terminal which results in a minimal CCA, i.e., which leads to the smallest *overall* wire area increase, is chosen as terminal to be linked next to the routing tree, and so on (step 4 in Fig. 6). During each tree extension, the current vector of the new connection, including the RMS current value, are updated as described in Section VI.

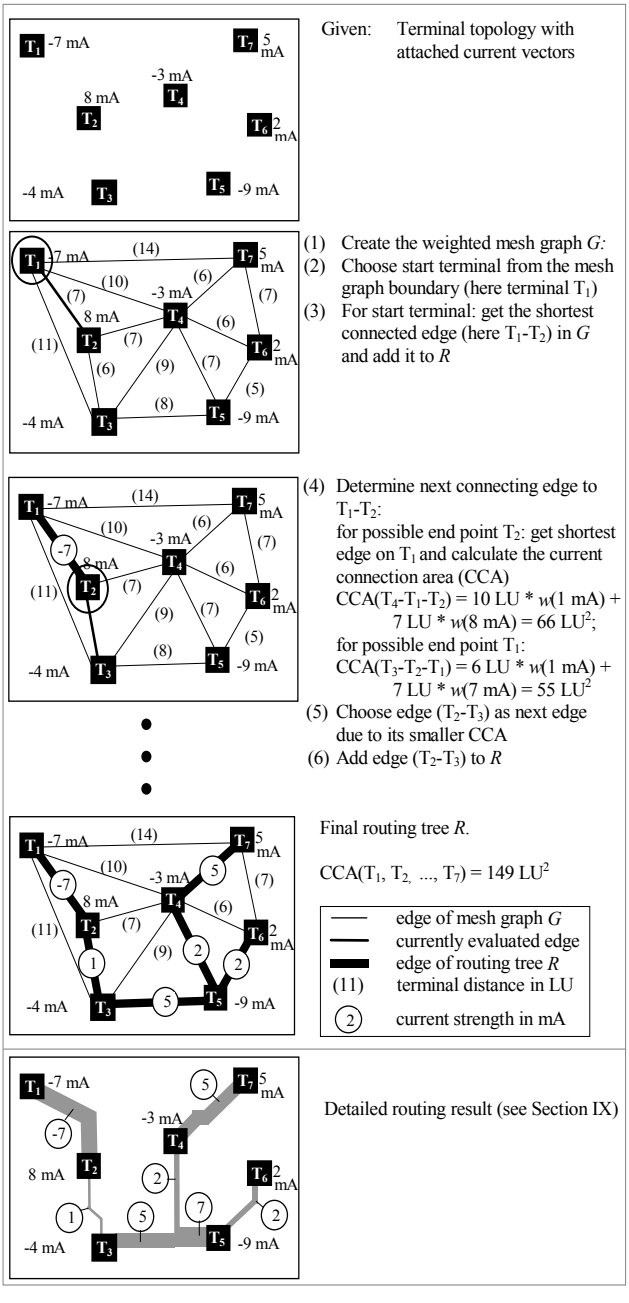

Fig. 7: Graphical illustration of the main steps of our wire planning algorithm. Depicted current values are only examples, in reality the entire current vector must be considered. In this example we assume that an RMS current I_{eq} of 1 mA requires a wire width of 1 length unit (LU).

An important feature of this current-driven approach is the objective of minimizing the current-dependent surface area of the wire rather than simply the wire length. In the example in Fig. 7, the wire area of the final routing tree is 149 LU^2 (LU = length units). Applying a minimal spanning tree algorithm to the same terminal topology would lead to a current-correct wire area of 225 LU^2.

IX. Detailed Routing

The exact path and layer allocations of the wire segments of the routing tree are determined during detailed routing. We use a simple, commercially available point-to-point router which can consider variable wire widths. Its only extension is a variable Steiner point handling: If a Steiner point insertion requires wire widening of a previously routed net segment, then the Steiner point is only created if the wire widening can be accommodated (as in Fig. 7, connection $T_3 - T_5$).

X. Implementation and Results

Our approach has been integrated into Mentor Graphics' ICStation environment using C/C++.

Several commercial analog circuits (ranging from 100 to 500 devices per cell) were routed using our methodology (Table 1 lists some of the examples). The routing results have been verified with an in-house Current Density Simulator [16]. The layouts achieved were current-density-correct in all cases.

TABLE I
Characteristics of some analog cells routed with our algorithm

CELLS	DEVICES	TERMINAL-TO-TERMINAL CONNECTIONS ("FLYLINES")	NETS
supply	101	118	64
wala	134	194	57
imux	135	224	86
receiver	284	389	241
dcdriver	498	476	281

The presented approach can be used in either semi-automatic mode (with specific nets selected for routing) or full-automatic mode (where all nets within a window are routed). Analog circuit designers usually prefer the first method, hence, run times vary widely. Using an UltraSPARC 10 workstation, routing of specific nets is performed within seconds; full-automatic routing of the entire layout requires a run-time of minutes (up to one hour).

Unfortunately, there are no benchmarks available to evaluate current-driven analog routing. We compared our routing results with layouts that were semi-automatically routed by experienced designers (using Mentor Graphic's IRoute tool) and afterwards manually adjusted for current-density. We also compared our results to two approaches presented in [14]. In order to allow a fair comparison, the same detailed router was used in all cases (except for the manually generated designs).

As can be seen in Table 2, the wire area and routing area consumption of our approach was less as compared to both

the manually adjusted layouts and the strategies presented in [14]. (The approaches in [14] minimize only the routing length and not the overall wire area.) The larger routing area of the manual approach is mainly due to the additional area a layout designer will reserve when routing in order to perform wire widening *after* the topology of the entire net has been laid out (i.e., after all branch currents are known). It is important to note that the run time of our approach is only a fraction of the time needed for an experienced designer to route and manually adjust the interconnect for current density.

TABLE 2
Comparison of results between our approach, the conventional method of semi-automatically routing with manual wire width adjustment, and two strategies "Steiner tree" and "Terminal tree" presented in [14].

CELLS	METHOD	WIRE AREA (μM^2)*	VIAS	ROUTING AREA (%)**	ROUTING TIME (MIN)
supply	**Our approach**	**50,220**	**145**	**100**	**3**
	Manually	51,440	138	102.6	≈ 125
	Steiner tree	n/a	149	102.0	22
	Terminal tree	n/a	148	102.2	4
wala	**Our approach**	**76,337**	**136**	**100**	**3.5**
	Manually	76,340	130	100.3	≈ 150
	Steiner tree	n/a	142	103.0	8
	Terminal tree	n/a	139	104.0	5
imux	**Our approach**	**78,880**	**178**	**100**	**4.5**
	Manually	80,320	178	103.4	≈ 180
	Steiner tree	n/a	198	103.8	9
	Terminal tree	n/a	193	103.9	5
receiver	**Our approach**	**54,604**	**180**	**100**	**6**
	Manually	58,380	178	102.5	≈ 185
	Steiner tree	n/a	199	102.0	13
	Terminal tree	n/a	197	102.8	7
dcdriver	**Our approach**	**102,275**	**455**	**100**	**14**
	Manually	108,880	460	104.1	≈ 240
	Steiner tree	n/a***	n/a***	n/a***	n/a***
	Terminal tree	n/a	483	106.0	20

* *Wire Area = Current connection area (CCA) of all wires according to Equation (6)*
** *Routing Area = Die area used for routing*
*** *Not applicable due to memory and run time limitations*

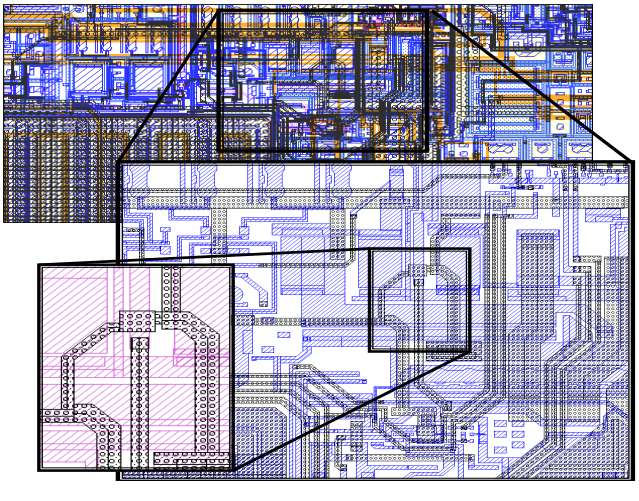

Fig. 8: Excerpt of the routed analog circuit "dcdriver"

XI. Conclusion

We have proposed a wire-planning-based routing methodology capable of routing analog multi-terminal signal nets with current-driven wire widths. This approach is to our knowledge the first *commercially applicable* wire planning algorithm focused on current-driven wire widths in order to prevent electromigration processes. Using our wire planning algorithm, a current-correct detailed routing can be performed without the need for a separate layout post-processing step.

Our algorithm has been successfully used to generate current correct designs of "real world" circuits. This has been achieved using much shorter design times and less routing area compared to conventional flows that include manual adjustment of wire widths. Currently, our approach is being integrated into commercial design flows of analog circuits for automotive applications. We believe that the proposed methodology can be used to address the problem of electromigration in other applications as well.

References

[1] J. R. Black, "Electromigration – a brief survey and some recent results", *Proc. IEEE Int. Reliability Physics Symposium*, Dec. 1968, pp. 338-347.

[2] F. M. D'Heurle, "Electromigration and failure in electronics: an introduction", *Proc. of the IEEE*, Vol. 59, no. 10, 1971, pp. 1409-1417.

[3] D. Young, A. Christou, "Failure mechanism models for electromigration", *IEEE Trans. on Reliability*, Vol. 43, no. 2, 1994, pp. 186-192.

[4] Z. A. Syed, A. Gamal, "Single layer routing of power and ground networks in integrated circuits", *Journal of Digital Systems*, vol. VI, no. 1, 1982, pp. 53-63.

[5] H.-J. Rothermel, D. A. Mlynski, "Automatic variable-width routing for VLSI", *IEEE TCAD*, vol. CAD-2, no. 4, Oct. 1983, pp. 271-284.

[6] A. S. Moulton, "Laying the power and ground wires on a VLSI chip", *Proc. Design Automation Conf.,* 1983, pp. 754-755.

[7] S. Haruyama, D. Fussel, "A new area-efficient power routing algorithm for VLSI layout", *Proc. ICCAD*, 1987, pp. 38-41.

[8] S. Chowdhury, "An automated design of minimum-area IC power/ground nets", *Proc. Design Automation Conf.*, 1987, pp. 223-229.

[9] T. Mitsuhashi, E. S. Kuh, "Power and ground network topology optimization", *Proc. Design Automation Conf.,* 1992, pp. 524-529.

[10] J.-S. Yim, S.-O. Bae, Ch.-M. Kyung, "A floorplan-based planning methodology for power and clock distribution in ASICs", *Proc. Design Automation Conf.*, 1999, pp. 766-771.

[11] X.-D. Tan, et.al., "Reliability-constrained area optimization of VLSI power/ground networks via sequence of linear programmings", *Proc. Design Automation Conf.*, 1999, pp. 78-83.

[12] T. Adler, E. Barke, "Single step current driven routing of multiterminal signal nets for analog applications", *Proc. Design, Automation and Test in Europe (DATE)*, 2000, pp. 446-450.

[13] T. Adler, et. al. "A current driven routing and verification methodology for analog applications", *Proc. Design Automation Conf.*, 2000, pp. 385-389.

[14] J. Lienig, G. Jerke, T. Adler, "Electromigration avoidance in analog circuits: two methodologies for current-driven routing ", *Proc. ASP-DAC/VLSI Design 2002*, pp. 372-378.

[15] J. R. Black, "Physics of electromigration", *Proc. IEEE Int. Reliability Physics Symposium, 1983*, pp. 142-149.

[16] G. Jerke, J. Lienig, "Hierarchical current density verification for electromigration analysis in arbitrarily shaped metallization patterns of analog circuits", *Proc. . Design, Automation and Test in Europe (DATE)*, 2002, pp. 464-469.