A Current Driven Routing and Verification Methodology for Analog Applications*

Thorsten Adler†
Infineon Technologies AG
St.-Martin-Str. 76
D-81541 Munich, Germany
+49 89 234-28764
Thorsten.Adler@infineon.com

Hiltrud Brocke, Lars Hedrich, Erich Barke
Institute of Microelectronic Systems
University of Hanover
Appelstr. 4, D-30167 Hanover, Germany
+49 511 762-19691
{brocke|hedrich|barke}@ims.uni-hannover.de

ABSTRACT

We present a new methodology for current driven routing and layout verification for analog applications used to avoid defects due to electromigration.

The methodology presented uses a commercial simulator to calculate the current flow at all terminals of the analog circuit. Afterwards maximum currents per terminal are extracted and used as guidance for the Current Driven Router (CDR) which is capable of routing analog multiterminal signal nets with current driven wire widths.

The Current Density Simulator (CDS) is used to compute and verify current densities in layouts that were generated using a 'standard' routing methodology.

Keywords

Design methodology, electromigration, current density, routing, verification, multiterminal signal nets, Steiner tree

1. INTRODUCTION

Designing analog power ASICs in modern mixed signal BCD-processes (Bipolar, CMOS, DMOS) requires a large amount of expert knowledge in order to meet constraints like symmetry, voltage drops, current density, temperature gradients, piezo-electrical effects, electromigration, etc. Unlike in digital design it is not possible to treat all of these constraints automatically up to now.

One main difference with respect to signals in digital circuits is the presence of large currents. In order to avoid electromigration due to excessive current density one has to design wires according to the current imposed on them. Routing multiterminal nets with current driven wire widths is a problem which arises not only in power and ground routing of analog and digital circuits but also in routing of multiterminal signal nets in analog circuits.

The routing widths used to guide multiterminal net routing are normally determined by current properties specified by circuit designers or set to the minimum value which may lead to electromigration [5]. In order to avoid circuit malfunction due to electromigration the final layout is checked for excessive current densities which again may be inaccurate due to the manually specified current properties.

In this paper we present a current driven routing and verification methodology for analog applications. First, the currents at all terminals are calculated using a commercial simulator. Peak currents are extracted and inserted into the schematic. These currents guide the routing tool CDR in order to generate the design rule correct layout for all multiterminal signal nets. The remaining two-terminal signal nets are routed using an existing analog router or CDR.

The Current Density Simulator CDS is used to check the routing results for current density violations which has to be done for all layouts generated without using the proposed methodology.

In the next section we describe the proposed design flow. Section 3 illustrates the mechanism used to calculate all terminal currents of the analog circuit. In section 4 and 5 we present the Current Driven Router (CDR) [1] and the Current Density Simulator (CDS) [12], respectively. Section 6 illustrates some examples generated using the new design flow and finally we give some concluding remarks.

2. DESIGN FLOW

Figure 1 illustrates the proposed methodology currently being integrated into two commercial design flows. All data used within the design flow is stored in an object oriented database called MGEN [14] which provides easy access mechanisms for layout and netlist data.

An analog simulator is used to compute the currents at all terminals. Afterwards, MentorGraphics' SimPilot is used to extract the currents for all terminals. It postprocesses them and stores them in an ASCII file or within the schematic netlist from where they are read into MGEN.

The Current Driven Router CDR reads the netlist and the current properties from MGEN and performs the routes for all multi-terminal nets. Afterwards an existing analog router or CDR is used to route the remaining two-terminal nets.

* This work was supported by the German BMFT under contract No. 01 M 3034. The authors are responsible for the contents of this publication.

† This work was done while T. Adler was with IMS Hanover.

DAC 2000, Los Angeles, California

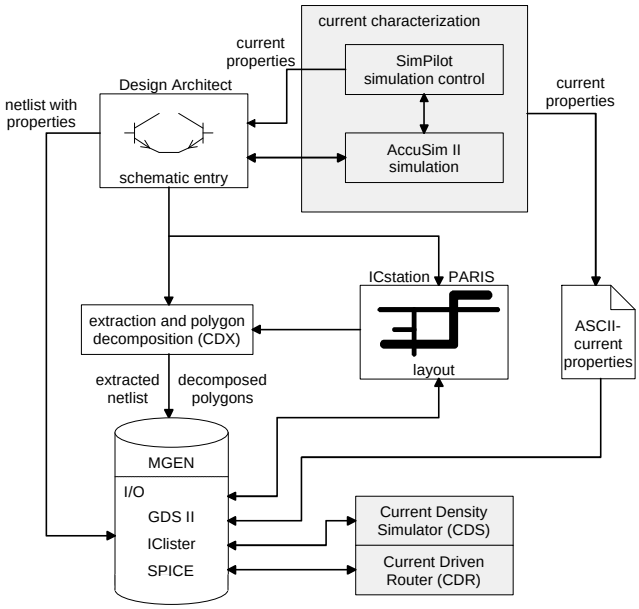

Figure 1: The design flow

In order to check existing layouts for current density violations an 'extraction' step is performed (CDX) which decomposes all wires into simple rectangles and irregularities. The resistance of the rectangles is calculated and then used to build an 'extracted' netlist which incorporates references to the corresponding geometric dimensions. This enhanced netlist is stored into MGEN and used by the Current Density Simulator CDS. CDS reads this modified netlist, calculates the current densities on all wires and writes these results into MGEN where they are used for visualization purposes.

3. CURRENT CHARACTERIZATION

A problem for the current driven design methodology is the determination of realistic current values for each terminal. Our approach uses a standard circuit simulator for simulation of the circuit netlist while parasitic wiring resistances are neglected. The resulting current values for all terminals are stored into MGEN and used by the Current Driven Router and the Current Density Simulator, respectively. The concept of this approach is shown in Figure 2:

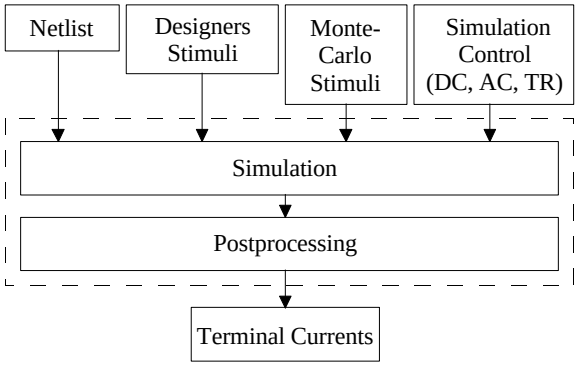

Figure 2: Concept of terminal current characterization

Two different types of input stimuli for the circuit simulation can be considered. First, the realistic but possibly error prone designer stimuli in combination with the necessary simulations are used. The designer delivers some manually generated stimuli, driving the circuit in well known operating regions which have already been used for circuit verification. The second method uses automatically generated input stimuli. Considering some information about the inputs like input voltage range the input sources generating analog Monte-Carlo-Stimuli are setup and simulated with the netlist in an appropriate transient simulation. This method tends to deliver more pessimistic results, i.e. larger terminal currents.

In both cases the results from one or more simulations are postprocessed by calculating a set of current vectors satisfying Kirchhoff's current laws. They represent a snapshot of the circuits operation at a particular point of time. For each terminal the vector with the terminals positive and negative maximum value is stored in the set of current vectors which reduces the simulation results to a set of 'worst case' currents vectors. For a net with *n* terminals that may lead to up to *2n* current vectors.

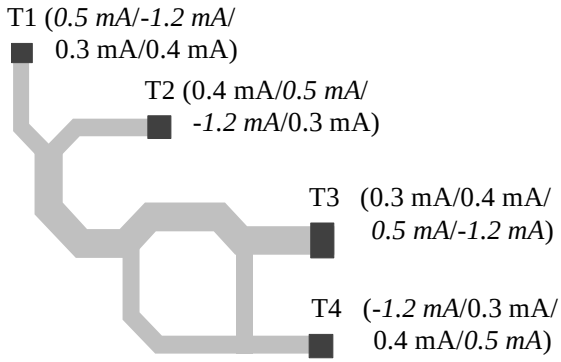

Figure 3: Current properties

In the example shown above each terminal has four current properties with two of them (printed *italic*) representing the terminals maximum positive and negative current value. Due to correlation only four different time points from circuit simulation are needed to represent the nets eight different maximum current values (two time points per terminal).

4. CURRENT DRIVEN ROUTER (CDR)

CDR [1] is a routing algorithm capable of routing multiterminal signal nets with current driven wire widths which is a problem similar to the problem of routing power and ground nets ([2], [10], [11], [13]).

Routing of power and ground nets consists of three tasks: Construction of interconnection topology, wire width determination and layout generation. [10] computes the power and ground net topology using a combination of Hightower's line-search algorithm [4] and Lee's maze-routing algorithm [6] using a standard wire width. Based on that topology all unknown currents, i.e. the currents of wires connecting Steiner points, are calculated. Afterwards, all wires are widened with respect to their calculated current flow. This may lead to DRC errors that have to be resolved in a separate post-processing step which modifies device placement.

Unlike in power and ground routing CDR calculates the unknown wire widths 'on the fly' during Steiner tree based layout con-

struction. Therefore, no post-processing steps are needed to generate design rule correct layout.

In order to achieve good routing results even in congestioned layout regions CDR uses more advanced detailed routing algorithms than those used by power and ground routers normally working on an empty routing area.

4.1 CDR's Database and Basic Algorithms

CDR's database and path searching algorithms are based on those presented in [16]. A connection graph G_C is used for layout representation. It can be obtained by extending the horizontal and vertical edges of each obstacle until another obstacle or the boundary is reached, in addition to generating a horizontal and vertical line through all terminals (see Figure 4). For a more formal definition of G_C refer to [16] or [8].

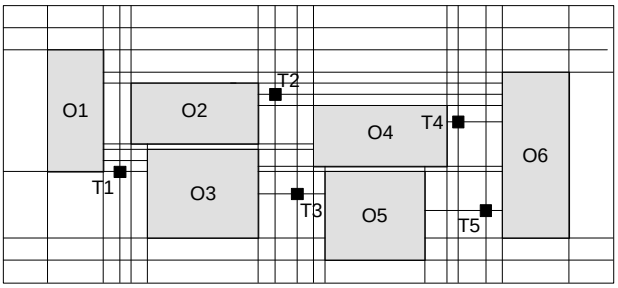

Figure 4: An example connection graph G_C

It is known that the shortest path between source and target is a path in the connection graph [8], [16]. Therefore, a shortest path algorithm such as Dijkstra's algorithm can be employed to find a path of minimum length between source and target.

The algorithm used to accomplish this is similar to the one presented in [16]. It is basically a line-search version of the Minimum Detour (MD) [3] algorithm with a generalized detour number concept.

4.2 CDR's Steiner Tree Construction

Due to the more generalized problem of Steiner tree based layout construction with non-uniform wire widths CDR has to cope with additional difficulties with respect to basic Steiner tree algorithms.

The current flow on wires connecting two Steiner points is not known as the topology of the net is unknown prior to layout construction. Figure 5 illustrates an example net with five terminals using only one current value per terminal for simplicity:

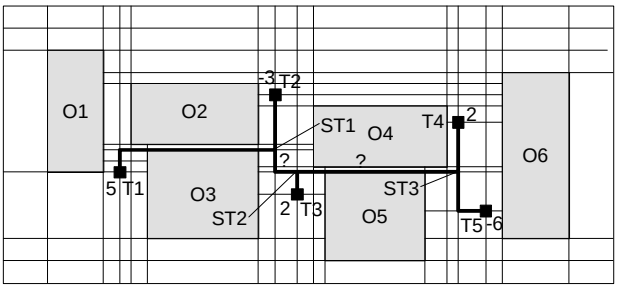

Figure 5: An example net topology

The net shown above has two current sources (T2 and T5) and three current sinks (T1, T3, T4) which are indicated by the positive current value shown. A 'standard' Steiner tree algorithm using a uniform (i.e. minimum) wire width would lead to the net topology shown in Figure 5 (Steiner points ST1, ST2, ST3).

The current flow on wires connecting two Steiner points (e.g. ST1 and ST2) is unknown prior to topology construction and has to be computed afterwards in order to widen all wires according to the currents imposed on them. However, this may lead to improper layouts due to design rule violations. In the example shown in Figure 5 obstacles O2/O3 and O4/O5 would have to be moved in order to generate the final layout.

Therefore, CDR's Steiner tree algorithm has to build the Steiner tree in a greedy, sophisticated fashion to compute the unknown wire widths 'on the fly' during Steiner tree construction. CDR's Steiner tree construction is based on a modification of the P3S algorithm [7]. The P3S algorithm sequentially adds the nearest terminal to the already routed subtree. That terminal is determined using simple Manhattan distances. Due to the presence of obstacles CDR's algorithm has to use a smarter method to add a terminal to the partially routed subtree:

```
CDR()
{
  sort all terminals in increasing x-order
  i:=0; source:=terminals[i]; source_width:=width[i]
  for (;i<num_of_terminals-2;i++)
   {
     steiner_point:=calc_steiner(source, terminals[i+1],
        terminals[i+2], source_width,width[i+1],
        width[i+2])
     detour(source,steiner_point,source_width)
     detour(terminals[i+1], steiner_point, width[i+1])
     source:=steiner_point
     source_width:=source_width+width[i+1]
   }
  detour(source,terminals[i],width[i])
  detour(source,terminals[i+1],width[i+1])
}
```

The algorithm calculates the Steiner tree layout by repeatedly computing an optimum Steiner point for three terminals at a time. At first *calc_steiner()* computes the optimum Steiner point for the first three terminals. After that Steiner point has been found, *detour()* is used to connect the first and the second terminal to the calculated Steiner point. Afterwards, *calc_steiner()* is called repeatedly to connect the last found Steiner point to the next two unconnected terminals, etc. The remaining two terminals are then connected to the last Steiner point calculated using *detour()*.

Using this greedy Steiner tree construction CDR is able to compute the unknown current flow on connections between two Steiner points by simply adding the current flows of the two wires connection to the Steiner point:

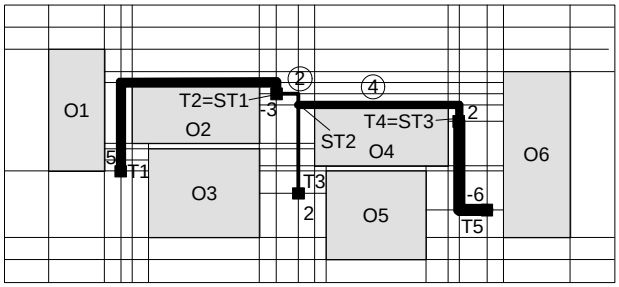

Figure 6: CDR's solution

At Steiner point ST1 CDR has to add the current flows of terminal T1 and terminal T2 to compute the unknown current on the wire which leaves Steiner point ST1. At Steiner point ST2 the current flow of terminal T3 is added to this sum and finally the current flow of terminal T4 is added to that value at Steiner point ST3. In order to generate wires which satisfy the current density restrictions for all possible circuit states (i.e. current vectors) CDR always has to choose that terminal current value from the computed current vectors that leads to the worst (i.e. largest) current sum at each Steiner point.

Calc_steiner() is guided by a cost function which basically computes the resulting routing area for each candidate Steiner point. The area formula consists of four terms. The first one represents the routing area for the route from the previous Steiner point to the current candidate Steiner point; the second and the third ones give the routing area for a route from the current candidate Steiner point to the first and second target terminal. The last term is an estimation for the routing area needed to connect the remaining terminals to the current candidate Steiner point. To improve the routing result some heuristics are used to choose the optimal Steiner point.

The resulting net layout is stored within MGEN and then used to update the layout editor.

5. CURRENT DENSITY SIMULATOR (CDS)

The developed validation tool Current Density Simulator (CDS) [12] is able to verify multiterminal nets as well as two-terminal nets with respect to correct current densities.

After current characterization the currents at all terminals are known (see Section 3). The unknown currents on wires connecting two Steiner points (see Figure 5) are calculated using an LU decomposition [9] in order to solve non-tree-shaped problems as shown in Figure 3.

In order to compute current densities CDS has to know the width of each wire and associate it with the appropriate current value. Therefore, a customized algorithm (CDX) is used to do a simple extraction of the interconnection network. CDX decomposes the wires into simple rectangles and irregularities:

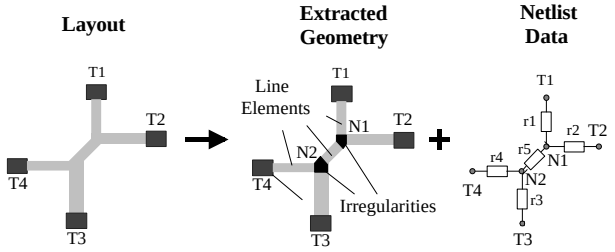

Figure 7: Geometry data of a net

The rectangular wire elements have a homogeneous current distribution. Their resistance value is calculated using the technology dependent sheet resistance. Irregularities such as wire bends, terminals as well as vias and contacts have an inhomogeneous current distribution [15] and can not be validated with the current version of CDS.

The resistors and their corresponding geometry are added to the original circuit netlist. This extended netlist is stored in MGEN and used by CDS to calculate the current densities for all rectangular wire segments.

The current density S is calculated using $S=|I/(w^{\prime}t)|$ with the current I, the line width w and the line thickness t. The layer dependent technology boundary $S^{*}_{max}=|I_{max}/w|$ is given as current density times the line width as the layer thickness is constant for the technology used. Normally ,this technology boundary includes some safety margin in order to overcome current density problems in 45 and 90 degree corners.

The LU decomposition is carried out for each net and each current vector stored in MGEN. All calculated current densities S^{*} are stored within MGEN and backannoted into the layout editor for visualization purposes.

6. RESULTS

Due to the ongoing integration into the commercial design flows CDR is currently not yet able to read industrial designs. The example shown in Figure 8 was routed without any necessary layout modification due to CDR's enhanced Steiner tree generation:

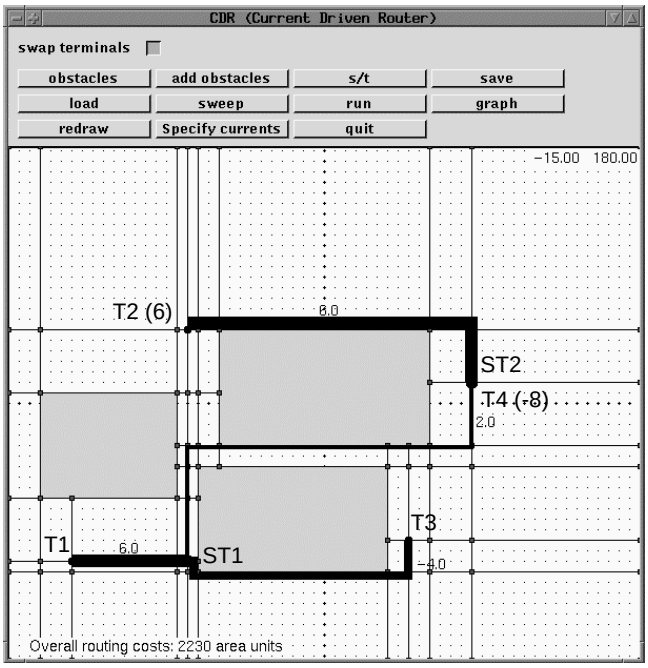

Figure 8: An example net routed by CDR

A standard power and ground routing algorithm would connect terminal T1 and terminal T2 at first which would lead to an resulting routing width of 12 units. In order to create a design rule correct layout one obstacle would have had to be moved after topology determination which is not applicable for analog designs.

Figure 9 shows the result of a circuit validation for an industrial analog design which was not routed using the proposed methodology. The two metal layers are drawn in light blue and pink. CDS was used to check single specific nets and showed that some wires were not properly dimensioned with respect to the current flow imposed on them. Those errors are colored yellow. Wire bends are drawn in light blue or pink as they are not yet checked with the current version of CDS.

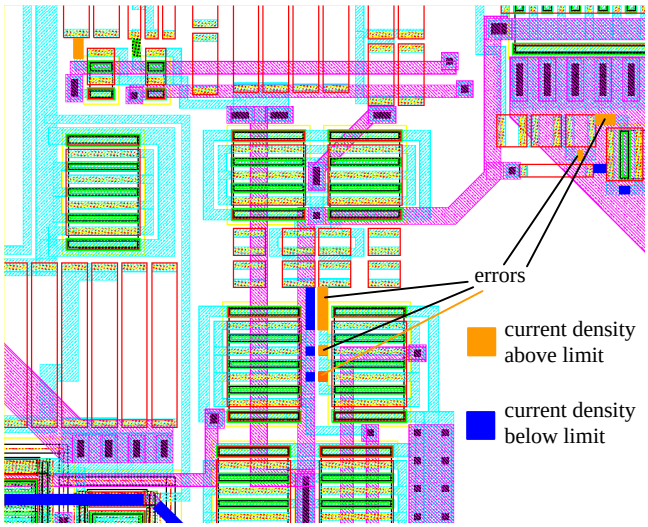

Figure 9: An industrial example checked with CDS

7. CONCLUSIONS

In our paper we presented a new routing and verification methodology for analog applications used to avoid circuit malfunction due to electromigration.

A commercial simulator environment is used to calculate the current flow at all terminals and extracts a set of 'worst case' current vectors representing the circuit behavior at some particular point of times.

These values guide the current driven router CDR which is used to route multiterminal signal nets with current driven wire widths. CDR generates Steiner tree based layout in a greedy fashion in order to compute the unknown wire widths between two Steiner points 'on the fly'. The remaining two-terminal nets are routed using an existing analog router or CDR.

In order to check existing designs that were generated without using the proposed methodology a special algorithm (CDX) is used to 'extract' the wiring resistances. It associates them with the corresponding layout information and updates the original circuit netlist.

The Current Density Simulator CDS reads this modified netlist and verifies the current densities on all straight wire segments. The results are backannotated into the layout editor window for error visualization purposes.

8. FUTURE WORK

The methodology and tools presented are currently being integrated into two commercial design flows and will be used to enhance automatic layout generation for analog integrated circuits.

CDR will be used as a global router to compute optimum Steiner points while existing inhouse routers will be used to connect them.

Future versions of CDS will use parameterized models representing 'irregularities' in order to verify current densities on wire bends, junctions, terminals, contacts and vias.

ACKNOWLEDGEMENTS

The development of the methodology and tools presented was supported by German BMFT project SSE Parasitics in cooperation with aiss GmbH, Robert Bosch GmbH and TEMIC semiconductors. The authors would like to thank M. Gerbershagen from TEMIC Semiconductors for supporting our work with the MGEN environment and some industrial examples and M. Decker from aiss GmbH for providing us with CDX.

REFERENCES

[1] T. Adler, E. Barke, *Single Step Current Driven Routing of Multiterminal Signal Nets for Analog Applications*, Proc. Design, Automation and Test in Europe, 2000

[2] S. Chowdhury, *An Automated Design of Minimum-Area IC Power/Ground Nets*, Proc. Design Automation Conference, pp. 223-229, 1987

[3] F.O. Hadlock, *The shortest: Path algorithm for grid graphs*, Networks, vol. 7, pp. 323-34, 1977

[4] D.W. Hightower, *A Solution to line routing problems on the continuous plane*, Proc. Design Automation Workshop, pp. 1-24, 1969

[5] W. Hunter, *Self-Consistent Solutions for Allowed Interconnect Current Density – Part II: Applications to Design Guidelines*, IEEE Transactions on Electron Devices, Vol. 44, No. 2, 1997

[6] C.Y. Lee, *An Algorithm for Patch Connections and Its Applications*, IRE Trans. Electron. Comput., vol. EC-10, pp. 364-65, 1961

[7] J.H. Lee, N.K. Bose, and F.K. Hwang, *Use of Steiner's problem in suboptimal routing in rectilinear metric*, IEEE TCAS, vol. CAS-23, pp. 470-476, 1976

[8] T. Lengauer, *Combinatorial Algorithms for Integrated Circuit Layout*, John Wiley & Sons, pp. 405-406, 1990

[9] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical recipes in C*, Cambridge University Press, 1992

[10] H.-J. Rothermel, and D.A. Mlynski, *Computation of Power Supply Nets in VLSI Layout*, Proc. Design Automation Conference, pp. 37-47, 1981

[11] H.-J. Rothermel, and D.A. Mlynski, *Automatic Variable-Width Routing for VLSI*, IEEE TCAD, vol. CAD-2, no. 4, pp. 271-284, Oct. 1983

[12] F. Shaikh-Brocke, L. Hedrich, T. Adler, M. Laage, A. Stürmer, C. Roedel, *Berechnung der Stromdichten des Leitbahnsystems integrierter Schaltungen*, Analog '99, 5. ITG/GMM-Diskussionssitzung, Entwicklung von Analogschaltungen mit CAE-Methoden, Febr. 1999, Munich

[13] Z.A. Syed, and A. Gamal, *Single Layer Routing of Power and Ground Networks in Integrated Circuits*, Journal of Digital Systems, vol. VI, no. 1, pp. 53-63, 1982

[14] TEMIC, *The MGEN Reference Manual*, TEMIC Telefunken microelectronic GmbH, Revision 2.1, 1997

[15] K. Weide, *Untersuchungen von Stromdichte-, Temperatur- und Massenflussverteilung in Viastrukturen integrierter Schaltungen*, Ph.D. Thesis, VDI Verlag, Duesseldorf, Reihe 9: Elektronik, No. 184, 1994

[16] S.Q. Zheng, J.S. Lim and S.S. Iyengar, *Finding Obstacle-Avoiding Shortest Paths Using Implicit Connection Graphs*, IEEE TCAD, vol. CAD-15, no. 1, pp. 103-110, Jan. 1996