Robust Methods for EMC–Driven Routing

D. Theune, R. Thiele [1], W. John, T. Lengauer [2]

Abstract

Due to the application of fast device technologies and the increasing complexity of printed–circuit boards, electromagnetic phenomena, e.g., reflections and crosstalk, gain more and more importance and may even disturb the function of a circuit. In the future, it will be indispensable to consider phenomena of *electromagnetic compatibility* (EMC) already during layout synthesis. In this paper, robust methods are presented that make it possible for the first time to incorporate complex EMC–constraints and cost criteria into printed–circuit board routing. This includes both, concepts for the development and specification of EMC–design models and robust and efficient algorithms for EMC–driven routing which can handle these models.

1 Introduction

The technological progress in manufacturing integrated circuits, modules, and boards yields higher frequency rates and more complex systems. As a result, electromagnetic phenomena, e.g., reflections and crosstalk, increasingly reduce performance, cause logic errors or even destroy a circuit. John [13] presents a general approach to *electromagnetic compatibility* (EMC) on printed–circuit boards (PCB). Bakoglu [1] focuses on the technology and design of high–speed VLSI circuits. So far, during layout synthesis EMC–aspects are largely neglected. The layout

[1] Current address: *German National Computer Science Research Center (GMD–SCAI), Schloß Birlinghoven, D 53754 Sankt Augustin, Germany*

[2] Current address: *German National Computer Science Research Center (GMD–SCAI), Schloß Birlinghoven, D 53754 Sankt Augustin, Germany* and *Department of Computer Science, University of Bonn, Römerstraße 164, D 53117 Bonn, Germany*

phase is succeeded by an EMC–verification phase, making use of an *EMC–analysis workbench* [15], for example. If the layout phase itself does not incorporate EMC–phenomena, a number of time–consuming redesigns has to be made, in general, to obtain a failsafe layout.

To render the design process as time- and cost–efficient as possible, EMC–constraints and cost criteria have to be integrated directly into layout synthesis. The approaches for EMC–constrained routing for ICs [5, 8, 28, 29] and PCBs [21, 38] presented so far make use of *maze–running* and *line–search* methods [23]. The effects of EMC–phenomena are mapped onto path–cost functions. Choudhury and Sangiovanni–Vincentelli [4], and Wawryn [38] incorporate parasitic capacitance and inductance couplings into the cost functions. The disadvantages of these approaches result from their sequential nature (routing net by net) and their large memory and run–time requirements for *area routing* and for computing the cost functions. As a consequence, they can be applied only to small circuits. Thus, the methods of [8, 28, 29] are used for the design of small analog integrated circuits such as operational amplifiers. Kiefl [21] extends line–search algorithms in order to incorporate constraints for maximal net lengths and maximal lengths of parallel wiring. Choudhury and Sangiovanni–Vincentelli [3], Gyurcsik and Jeen [11] extend channel routing algorithms to deal with parasitic couplings on integrated circuits. Cong et al. [5] develop an efficient algorithm for constructing approximate *Steiner trees* with bounded longest path length for timing–driven routing. Mitra et al. [30] made a promising approach for routing mixed-signal ASICS, dealing with similar constraints to those relevant in our context. They use simulated annealing for global and detailed routing, which is practicable only for small and midrange circuits. All approaches presented so far (mainly [21]) consider only partial aspects of the intricate EMC–problem.

Typical printed–circuit boards consist of more than one thousand nets, several thousand pins and up to one thousand devices (ICs, resistors, capacitors). For typical boards the routing

area is quite large (e.g., for a 34 $cm \times 24$ cm board with 0.2 mm wire width and minimum wire distance the routing grid has 510 000 points). Routing on PCBs differs from IC routing in several respects.

- Devices (ICs) are mounted on the top and on the bottom of the board. Therefore, they are no obstacles for routing, and the routing area cannot be naturally subdivided into *channels*.
- The number of routing layers is much higher on PCBs than on ICs (8, 12, 16 or even more).
- In general, layer changes (*vias*) need more routing space than wires. Typical values are 0.6 mm via diameter and 0.2 mm wire width.
- Vias are realized by drilling holes. Therefore, vias cover routing space on all layers, or at least on all layers above or below the layers to be connected.

In this paper we present new methods for the development of EMC–design models and their incorporation into algorithms for PCB–routing. Our main contributions consist of:

- Carrying the two–phase approach to routing, in which global routing is followed by detailed routing over to PCBs. This approach has been very successful on chips.
- Providing a hierachical method for global routing which is both robust and efficient (section 5). A prime advantage of this framework is that nets can be routed simultaneously and thus their interdependence can be taken into account directly.
- Providing an extremely general path–search framework for detailed routing, which can model a wide variety of EMC–cost functions (section 6).

Sections 2 and 3 introduce EMC–design models and rules which consist of design constraints and cost criteria for the purpose of avoiding or limiting noise effects resulting from EMC–phenomena. Section 4 summarizes the requirements for the routing algorithm to be developed. Section 7 reports on experimental results obtaimed by the HERO system, which implements the method presented in this paper [17, 36]. Our methods make it possible for the first time to incorporate more complex EMC–constraints and cost criteria into routing even of large–scaled printed–circuit boards.

2 EMC–design models

Considering EMC–phenomena in algorithms for placement and routing requires precise EMC–design models. Such models include constraints as well as cost criteria to avoid or limit noise effects already during the layout phase of a printed–circuit board.

The effects of EMC–phenomena on printed–circuit boards depend on a large number of parameters. The dependencies between the various factors of influence are quite complex and unstructured. Therefore, it is not possible to describe all EMC–phenomena in exact models. In addition, during layout design only simplified models can be handled efficiently. Therefore, the models can only cover partial aspects and approximate descriptions. In [4, 38] EMC–cost functions (e.g., coupling capacities) are derived directly from the electrical behaviour of the circuit by extensive and time consuming circuit analysis. However, it is our experience that the resulting routings are still not failsafe, in general. Therefore, a generated layout always has to be submitted to a subsequent EMC–verification phase.

In order to render the routing process as efficient as possible we map the complex EMC–design rules onto simple geometric constraints and cost criteria. This contrasts with the approaches of [4, 38]. The EMC–design rules are the result of extensive simulations and analytic calcula-

tions [13, 16, 17] which, however, need be done only once for a given technology and not again and again for each circuit during routing. Therefore, we can use accurate models of the electrical behaviour of basic circuit elements, such as input gates, lines, and output gates.

Kiefl [21] presents a similar approach, which considers geometric constraints for the transmission–line length and the coupling length between transmission lines.

For the purpose of layout design the parameters that are significant for EMC–phenomena on PCBs can be divided into variable and fixed parameters. Signal–specific parameters (e.g., rise and fall time, DC and AC noise margins) and PCB specific parameters (e.g., characteristic phase velocity and impedances of the layer structure) are fixed during layout synthesis, for example. In contrast, the geometric parameters like transmission–line length, width and distance between different transmission lines are variable during layout synthesis.

We use a rule–driven approach in order to specify EMC–design models by EMC–design rules containing restrictions and evaluations for the variable geometric parameters in dependence on the fixed parameters. This approach offers the following advantages:

- Rule–driven design models offer a robust and flexible framework for specifying todays and future EMC–requirements. By adding further rules or modifying existing rules, models can be restricted to the essential dependencies and can easily be adapted to the requirements of the actual application and continuous technological developments.

- Rule–driven design models help to structure EMC–phenomena. For instance, different EMC–phenomena like reflection and crosstalk are separated into different rules. An important advantage of this separation is that different EMC–phenomena can be considered in different (namely, their relevant) design phases.

A problem of all rule–driven systems is that different rules may contain contradictory or only

partly realizable requirements. Therefore, a weighting or priority scheme has to be imposed on the rule set.

3 EMC–design rules

In this section we derive our EMC–model in the following sequence of steps:

1. Structuring of occurring EMC–phenomena
2. Identifying relevant variable and fixed parameters for each EMC–effect
3. Evaluating parameter–combinations for each EMC–effect
4. Deriving rules from the evaluation
5. Ranking the rules
6. Validating the model

These steps are implemented by analytic calculations, simulations and measurements. In the following we exemplify the steps 2 to 4 with two design rules pertaining to two typical noise effects on PCBs (reflections and crosstalk).

3.1 Reflections

Signal reflections are caused by transmission–line inhomogeneities. On printed–circuit boards such inhomogeneities are discontinuities like bends, layer changes (vias) and junctions as well as mismatches of the characteristic transmission–line impedance and the input and output impedances of the connected devices. For current device and connection technologies on PCBs, however, reflections are mainly caused by the mismatch of the characteristic transmission–line impedance and the impedances of the input and output digital gates.

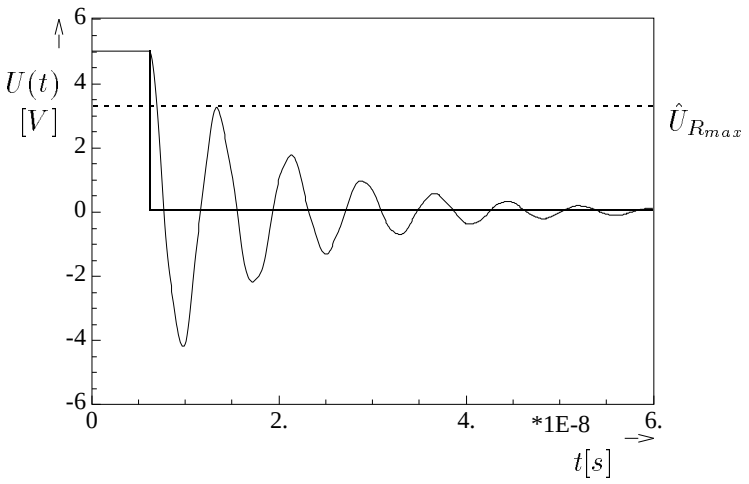

Figure 1: Typical voltage curve at the input digital gate (B) of a transmission line

(transmission–line length $l = 30cm$, ACMOS–technology)

For this purpose, simulators such as FREACS [16] are available. Figure 1 shows the voltage curve at the input digital gate of a single transmission line determined by simulation. Figure 2 shows the used simulation model and basic transmission–line structure. The simulation result indicates the over– and undershoots caused by reflections which may cause a fault reaction of the digital input gate.

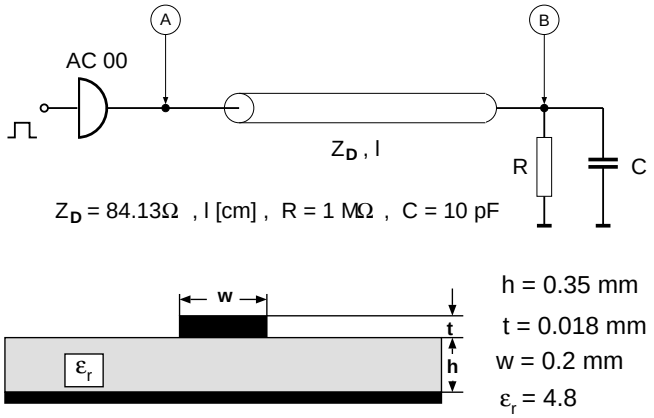

Figure 2: Simulation model and transmission–line structure

By simulation of well chosen transmission–line structures with different parameter combinations it is possible to identify the parameters relevant for reflections and to evaluate the relevant parameter combinations with respect to their noise impact. In this paper the development of EMC–design models and rules is restricted to the main parameters. Among the variable parameters the transmission–line length is considered. This parameter determines the noise voltage caused by signal reflections. The main fixed parameters are the device technology dependent parameters (e.g., the rise and fall time, the input and output impedance and the noise margin) and the PCB specific parameters [15] (e.g., layer structure, transmission–line width, transmission–line thickness, dielectric constant etc.).

Figure 3 shows the maximal amplitude $\hat{U}_{R_{max}}$ of the over– and undershoot values as a function of the transmission–line length for a fixed device technology (ACMOS) and fixed PCB–specific parameters (given in figure 2). This curve is a rsult of simulations.

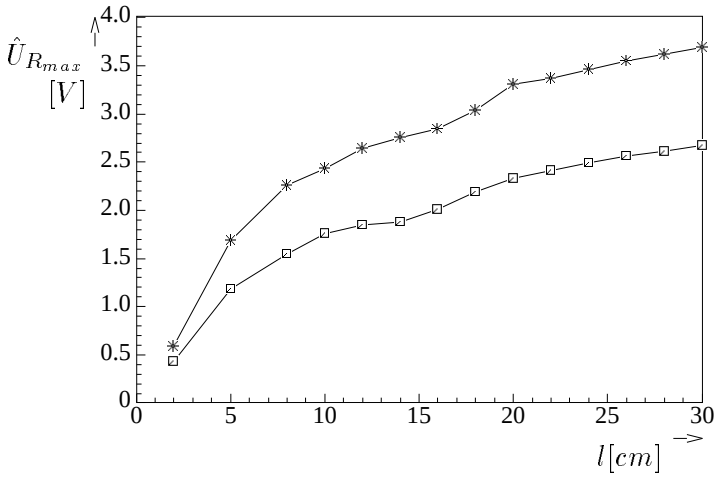

Figure 3: Max. ringing amplitude $\hat{U}_{R_{max}}$ as a function of the transmission–line length
($*$: *high/low*–transition, □: *low/high*–transition, ACMOS–technology)

By evaluating such $\hat{U}_{R_{max}}/l$ diagrams, a critical transmission–line length l_k can be deduced for each possible combination of fixed parameters.

EMC-Design Rule 1

In order to avoid faulty functions of digital gates due to reflections, no transmission line between output and input gate should exceed the critical line length l_k.

For the development of critical line lengths, extensive simulation results have to be processed. We also use simple approximations based on theoretical calculations [14] in order to determine and validate the critical line lengths. In table 1, the critical transmission–line lengths l_k are depicted for various device technologies and the PCB specific parameters shown in figure 2. The values of this table are based on the assumption that only noise due to reflection has to be considered. In practice, the noise effects caused by the other EMC–phenomena (e.g. crosstalk, simultaneous switching, current spike, radiation) have to be accounted for, too. Therefore, the values from the table have to be decreased accordingly. This fact will be discussed in more detail in subsection 3.3.

Technology	$l_k[cm]$	Technology	$l_k[cm]$
AC	17.0	S	14.0
ACT	12.7	LS	41.0
HC	33.0	ALS	22.7
HCT	26.4	AS	14.0
BCT	14.0	FAST	19.5

Table 1: Critical transmission–line length for different device technologies

Up to now, we have neglected several relevant dependencies. For example, parameters like rise and fall time and noise margins are device dependent. Currently, we assume that these parameters are identical for all devices of the same device technology. Another problem not mentioned yet is the development of critical transmission–line lengths for multiterminal nets. In this case the transmission–line topology has to be considered, as well. Therefore, EMC–design

rule 1 is only a first step to consider noise effects caused by reflections.

3.2 Crosstalk

Noise voltages on a transmission line can also be caused by parallel transmission lines. Figure 4 shows the voltage curve at the input (A) and output digital gate (B) of a *passive* transmission

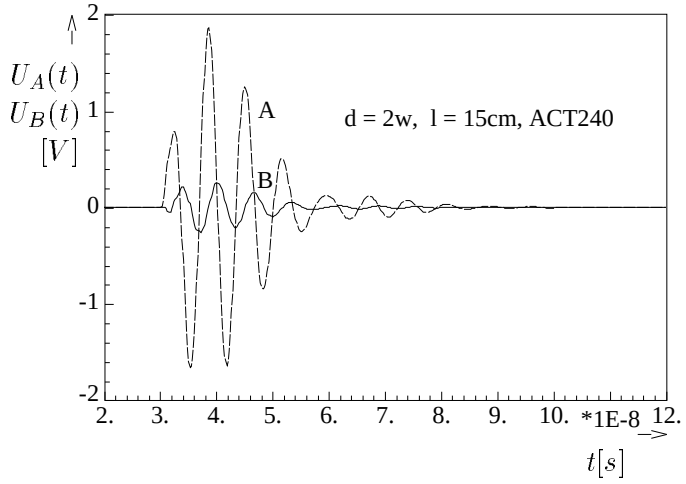

Figure 4: Typical voltage curves $U_A(t)$, $U_B(t)$ at the input (A) and output digital gate (B) of a passive transmission line (ACMOS–technology)

line in a coupled transmission–line structure determined by simulation. Figure 5 shows the underlying simulation model and the transmission–line structure used as input for the simulation.

The parameters relevant for the crosstalk problem can be determined and possible parameter combinations can be evaluated similar to the method used for reflections. Again, we have limited our consideration to the main parameters, so far. The most important variable parameters for crosstalk are the distance d and the coupling length l of the coupled transmission lines. For the fixed parameters the device technology dependent parameters and PCB specific data (see figure 5) are taken into account. In the following, only effects from near–end crosstalk are considered.

Figure 6 shows the voltage curves $U_A(t)$ on a passive transmission line for various coupling

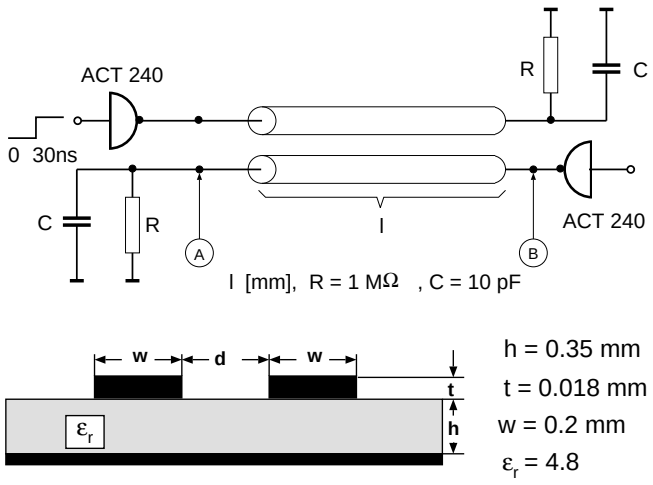

Figure 5: Crosstalk simulation model and transmission–line structure

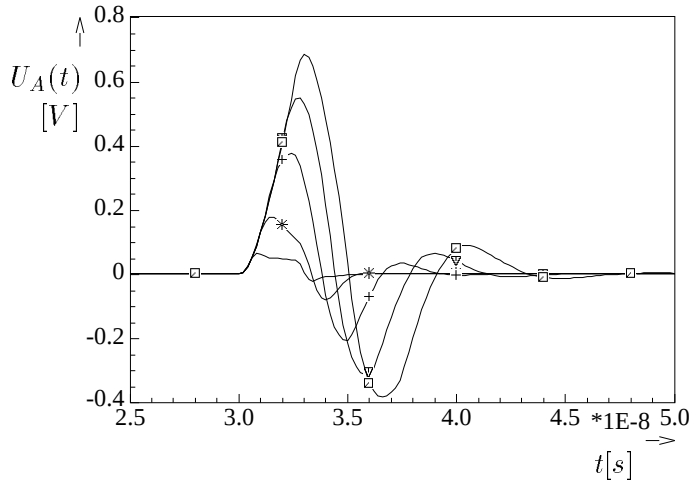

Figure 6: Noise voltage $U_A(t)$ as a function of the coupling length l
$\square\hat{=}20cm$, $\triangle\hat{=}15cm$, $+\hat{=}10cm$, $*\hat{=}5cm$, $-\hat{=}2cm$, $d = w$, HCMOS–technology

lengths l. The gates of the active and passive transmission line are in HCMOS–technology. The underlying transmission–line structure and the geometrical dimensions are shown in figure 5. Figure 6 illustrates a dependency between the coupling length and the noise voltage amplitude. Figures 7 and 8 show the maximal noise voltage amplitude $\hat{U}_{C_{max}}$ for two device technologies (ACMOS, HCMOS) and for several coupling distances d as a function of the coupling length l.

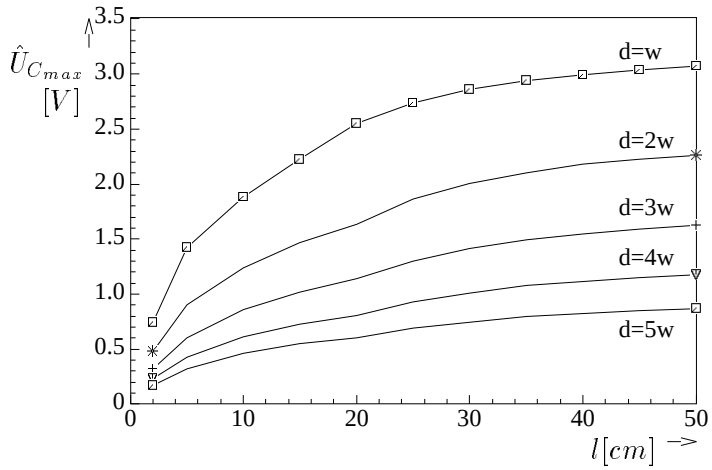

Figure 7: Max. crosstalk amplitude $\hat{U}_{C_{max}}$ as a function of the coupling length l and the coupling distance d (near–end, node A, ACMOS–technology)

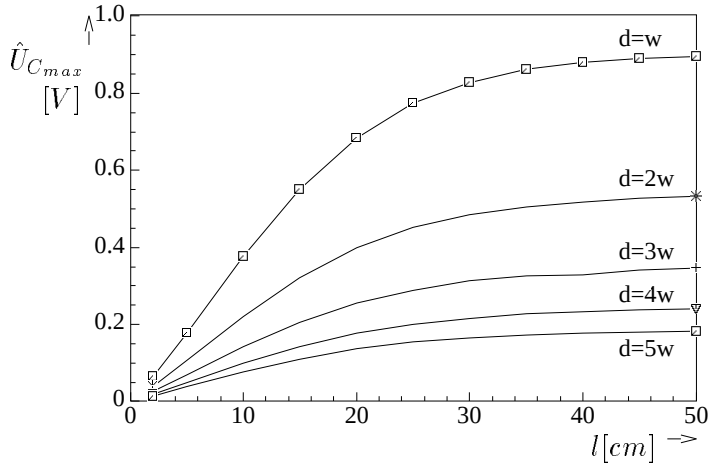

Figure 8: Max. crosstalk amplitude $\hat{U}_{C_{max}}$ as a function of the coupling length l and the coupling distance d (near–end, node A, HCMOS–technology)

In order to provide a model for the crosstalk noise for a transmission line N, the transmission–line structure is subdivided into coupling regions R_i (see figure 9).

Only those line segments S_i directly adjacent to N form a coupling region with N. These may also be segments on different layers. Our simulations showed that nonadjacent line segments which are separated from line N by another line can be neglected. The parameter l_i describes

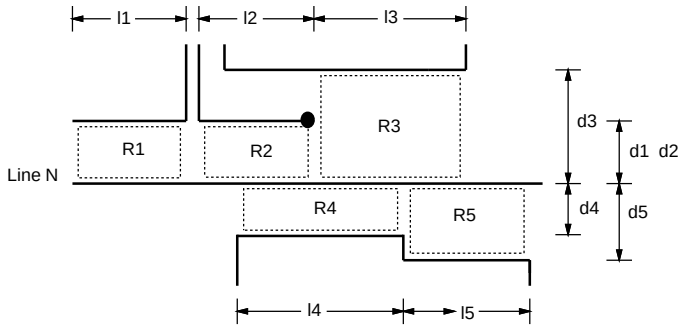

Figure 9: Subdivision of a transmission line into coupling regions

the length of each coupling region R_i and d_i describes the distance between N and the interfering line segment S_i. For each coupling region R_i the coupling level $C_{R_i}(N)$ is a measure for the noise voltage for line N due to the coupling.

$$C_{R_i}(N) = \frac{l_i}{l_c(T, T_i) \cdot f(d_i, T_i)} \tag{1}$$

The term $l_c(T, T_i)$ represents the maximal coupling length for the minimum coupling distance $d_{\min}$ (in the context of this paper $d_{\min}$ has been chosen as $d_{\min} = 0.2mm$). The length l_c depends on the device technologies T of line N and T_i of the interfering line segment S_i. Currently for the determination of l_c only the coupling between two transmission lines is considered. The function $f(d_i, T_i)$ describes the increase of the admissible coupling length with increasing coupling distance. The term $l_c(T, T_i) \cdot f(d_i, T_i)$ represents the admissible coupling length for the coupling distance d_i.

The overall coupling level $C(N)$ of line N is obtained by summing over all $C_{R_i}(N)$. This level $C(N)$ can be used for evaluating the crosstalk noise effects of transmission lines on printed–circuit boards. Failures due to crosstalk can be avoided, if the maximum coupling level $C_{\max}(N) = 1$ is not exceeded. Thus the following EMC–design rule to avoid crosstalk effects is derived:

EMC-Design Rule 2

The overall coupling level C of a transmission line should not exceed the maximum coupling level $C_{max}(N) = 1$.

The determination of the maximum coupling length l_c and the function f requires extensive data based on a large number of simulations.

In table 2, the maximum coupling lengths $l_c(T, T_i)$ for several technology combinations are given. Again, the values of this table are based on the assumption that only noise caused by crosstalk has to be considered. These values have to be decreased accordingly, if other EMC–phenomena have to be considered, too. This fact is discussed in more detail in subsection 3.3.

$l_c(T,T_i)$		T_i						
$[cm]$		AC	LS	ALS	S	FAST	HC	BCT
T	AC	15.5	50.0	50.0	50.0	50.0	50.0	50.0
	LS	2.5	20.5	9.5	20.5	20.5	33.5	9.5
	ALS	4.5	50.0	19.2	50.0	19.2	50.0	19.2
	S	3.4	49.0	13.7	49.0	13.7	50.0	13.7
	FAST	4.5	50.0	19.2	50.0	19.2	50.0	19.2
	HC	15.5	50.0	50.0	50.0	50.0	50.0	50.0
	BCT	3.0	29.0	12.0	29.0	12.0	50.0	12.0

Table 2: Maximum coupling lengths $l_c(T, T_i)$ for various device technology combinations ($d_i = d_{\min} = 0.2mm$)

In table 3, the values of the distance function $f(d_i, T_i)$ are given for different device technologies.

From tables 2 and 3 and equation (1) it can be deduced that two transmission lines with gates of ACMOS–technology for $d_i = 0.2\,mm$ should not exceed the coupling length of $15.5\,cm$.

3.3 EMC–Balance

In subsections 3.1 and 3.2, design rules for avoiding faulty functions due to reflections and crosstalk are developed. In these subsections, reflection and crosstalk effects are analyzed separately.

$f(d_i, T_i)$	d_i $[mm]$				
Technology T_i	0.2	0.4	0.6	0.8	≥ 1
AC	1	2.38	5.56	10	∞
LS	1	1.49	2.50	6.67	∞
ALS	1	2.23	4.35	∞	∞
S	1	1.49	2.50	6.67	∞
BCT	1	2.23	4.35	∞	∞
FAST	1	2.23	4.35	∞	∞
HC	1	1.67	2.86	10	∞

Table 3: Distance evaluation function $f(d_i, T_i)$ for different device technologies

In practice, however, for each net the sum of the noise voltages caused by all occurring EMC–effects should not exceed a given maximum value, the *maximum overall noise level*. This value depends on the noise margin of the digital gates of the analyzed net, for instance. Therefore, it is necessary to account for the noise impact resulting from all occurring noise effects together. We do so by specifying an EMC–balance for each net. The EMC–balance determines for each net, what portion of the overall noise is allowed for reflections, crosstalk, and so on. The assignment of portions may vary during the synthesis process, as more information on the layout becomes available. This flexibility is important, because it is difficult to determine a meaningful EMC–balance for each net, offhand. As an example, it is very hard to predict, whether crosstalk or reflection problems are to be expected and thus what portion of the overall EMC–balance has to be assigned to each effect. By splitting the layout design into several phases and by treating EMC–effects in different phases the EMC–balance can be changed dynamically. For instance, after global routing improved estimates of the expected transmission–line length and thus also of the noise effects caused by reflections are available. Therefore, the EMC–balance can be adjusted appropriately.

In [17] we describe the design of EMC–design rules thoroughly. Further design rules (e.g., net topology rules, thermic rules and electromagnetical radiation rules) are points for future work.

4 Robust and efficient methods for EMC–driven routing

The incorporation of EMC–design models into the layout design process results in very complex optimization problems. Our way to solve these optimization problems efficiently is to decompose them into small subproblems. For this reason, the EMC–router HERO [36] partitions the routing phase into *global* and *detailed routing*, as common in chip layout. During global routing, the rough route of each net is determined. This is done hierarchically, i.e., the routes of the nets are refined in a sequence of steps. During detailed routing the exact routes are determined inside each block resulting from the hierarchical partitioning of the board. The detailed routing phase works sequentially, i.e., net by net.

The hierarchical decomposition of the board during global routing induces a decomposition of the routing problem into small subproblems. In each subproblem of the global routing phase, it is possible to consider all nets simultaneously. Because of the large number of wiring layers, a tentative layer assignment is determined already during global routing. In this aspect, our system contrasts with the approaches common in chip layout and the approaches presented by Kessenich and Jackoway [20], and Kiefl [21].

With the use of hierarchical routing methods in combination with layer assignment, global dependencies between nets can be taken into account during global routing. In contrast, existing printed–circuit board routers work in a purely sequential manner. As a result, the routes of nets being routed late underlie strong restrictions imposed by the numerous routes that have been placed earlier. In many cases, there is no route for a late net, or only a route with poor electrical behavior (for example a route exceeding the critical line length, which will cause reflection

problems) is available.

In order to be able to incorporate EMC–phenomena in detailed routing, *maze–running* methods have been appropriately extended [25, 26]. This extension offers a greater flexibility with respect to the cost functions that can be considered than classical maze–running techniques or line–search algorithms do, which are applied in [21]. For example, a realistic model of crosstalk problems by the cost functions is possible with this extension.

In order to avoid noise effects right from the start, we incorporate EMC–criteria already in the global routing phase. In this point of view, our approach contrasts with all existing approaches for EMC–driven layout.

5 Hierarchical global routing

In this section, hierarchical routing methods developed for chip layout are extended to EMC–driven layout of printed–circuit boards. Up to now, only Kiefl [21] applied hierarchical methods—namely the method presented by Burstein and Pelavin [2]—to this problem. In the approach presented by Kessenich and Jackoway [20], global routing is done nonhierarchically and sequentially.

Our approach for global routing uses linear programming techniques. The first hierarchical routing method based on linear programming was developed by Burstein and Pelavin [2] in the context of gate–array layout. Their integer linear programming approach for 2×2 grids selects an admissible wiring alternative for all nets of a subproblem simultaneously, by computing the number of nets which can use a certain wiring alternative with respect to the given capacity restrictions. Luk et al. [27] extend this approach to general–cell layout and to *slicing floorplans*, i.e. floorplans that can be generated with mincut bipartitioning. Karp et al. [18] set up the global routing problem for gate arrays as a single system of linear equations. In [19] they present a

worst-case analysis for the channel width needed for routing and a new rounding algorithm for obtaining integral approximations for the solutions of the system of linear equations. Hu and Shing [12] propose a bottom–up method for hierarchical routing on the basis of binary cut trees. They set up the global routing problem as a hierarchical system of integer programming problems, suggesting that these should be solved using standard techniques of 0–1 programming. Their bottom–up approach is unsuitable for EMC–driven routing, because the essential global view of the problem is not considered until critical decisions have already been made. Thaik et al. [34] use integer programming techniques for global routing in chip layout. In contrast to us, they reduce the problem complexity by solving the problem in three phases and not by a hierarchical decomposition into subproblems.

The solution of the subproblems by linear programming techniques offers a robust framework which permits the incorporation of various EMC–design rules. This is done by the restriction to *admissible* wiring alternatives and by the appropriate choice of the cost functions and constraints in the optimization problems. Of course, it is not possible (and not meaningful) to consider all EMC–constraints during global routing. For example, a precise model of crosstalk effects can only be applied during detailed routing (see section 6.3). During global routing we only use some simple heuristics to largely avoid crosstalk (see sections 5.3 and 5.4). Since, for each net, the net length is essentially determined during global routing, the critical length has to be considered mainly in this routing phase. Therefore, whether a transmission line exceeds the critical line length, can be discovered already during global routing. For this reason, protocols of rule violations are generated, which can be used as a feedback to the placement phase. When the global routing phase is finished, the wiring lengths and the resulting noise voltages caused by reflections can be estimated appropriately. Therefore, it is possible to consider reflections and crosstalk separately and to adjust the EMC–balance appropriately after global routing (see

subsection 3.3).

5.1 Outline of Global Routing

Figure 10 depicts an example of the stepwise refinement of the route for a net during global routing. A procedure similar to the method proposed by Luk et al. [27] is used. In addition, for each net, a tentative layer assignment is determined already during global routing.

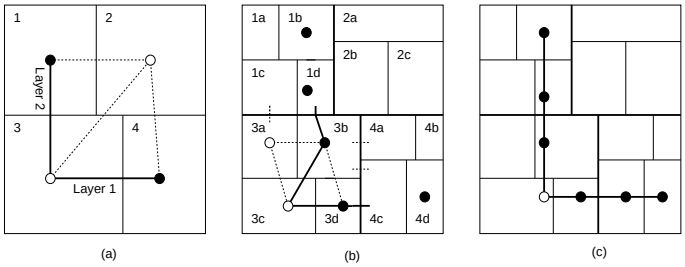

Figure 10: Global hierarchical routing

On the first level of the hierarchy, the printed–circuit board is partitioned into at most four subblocks (see figure 10(a)). For each net, the route between these blocks is determined. The computation of global wirings between at most four blocks is called a *primitive wiring problem*. After solving the primitive wiring problem, each wiring segment of a net is assigned to a wiring layer. In figure 10(a), the global wiring of a net with pins inside block 1 and block 4 is shown. Each wiring segment is labeled with a layer.

During the refinement, each block is partitioned into at most four subblocks (see figure 10(b)). In the first step of the refinement, all wiring segments are refined to the next level of the hierarchy. The wiring segment between blocks 1 and 3 may be refined to a wiring segment between blocks 1c and 3a or 1d and 3b, for example. The local nets inside each block are refined by the solution of four independent primitive wiring problems (see figure 10(b)). In figure 10(c), the global wiring of a net for the second level of the hierarchy is depicted.

The refinement process is continued through several levels. Each refinement step consists of three steps: the solution of the primitive wiring problems, the layer assignment and the refinement of the wiring segments to the next level of the hierarchy (if present).

5.2 Hierarchical partitioning

Before global routing, the printed–circuit board is hierarchically partitioned into rectangular blocks. Each block is partitioned into at most four subblocks. Therefore, for each net there are maximally eight wiring alternatives, and the primitive wiring problems can be solved efficiently.

This partitioning of the board can be derived from a partitioning resulting from an EMC–driven placement procedure, which is not available yet, or it can be computed from scratch by two methods. In the first method, blocks are partitioned such that the sum of the weights of nets running between subblocks is minimized, in a mincut–like process. The objective of this method is to obtain a large number of local wirings. The second method partitions the placed circuit according to the pattern of a uniform grid.

5.3 Solution of the primitive wiring problems

As mentioned above, for each net, there are at most eight wiring alternatives within a primitive wiring problem. Therefore, all nets can be wired simultaneously. The solution of a primitive wiring problem is divided into several phases. At first, for each net, the *admissible* (i.e., EMC–noncritical) wiring alternatives are selected from the set of possible alternatives. For example, to take into account the critical line length (see section 3), only those wiring alternatives are chosen as admissible, which do not exceed the critical length l_k. If no wiring alternative is admissible (for example, if all alternatives exceed the critical length because it already is exceed by the distance of the pins), the best wiring alternative (with minimum rule violations) is chosen. This is to ensure the completeness of wiring.

In the next step, the primitive wiring problem is formulated as an integer linear program—similar to the approach mentioned by Luk et al. [27]. Since, in general, nets differ in their electrical behavior, it is not possible to group geometrically equivalent nets. This is in contrast to chip layout [27, 32]. Hence, for each admissible wiring alternative j of every net N an explicit variable $x_{(N,j)}$ is needed.

The uniqueness of the route for each net N is guaranteed by constraints $\sum_{j=1}^{8} x_{(N,j)} \leq 1$. Capacity constraints ensure that the wiring capacities between and inside the blocks are not violated. In addition, these constraints reserve more wiring space for critical nets. The idea is to prevent crosstalk effects by enabling the detailed routing phase to wire these nets with a wider coupling distance to other nets.

The cost of each wiring alternative reflects its quality $\gamma(N, j)$ w.r.t. EMC–phenomena and its length $\lambda(N, j)$. For the EMC–quality $\gamma(N, j)$ the following criteria are presently taken into account:

- In order to reduce signal reflections $\gamma(N, j)$ mainly reflects the relation between the length of the wiring alternative and the critical length l_k, appropriately weighted.

- Critical blocks (with high levels of radiation, for instance) should not be passed by sensitive nets. Therefore, the wiring alternatives passing these blocks are penalized by adding a term to $\gamma(N, j)$.

- Each net may be restricted to be of a certain *wiring type*, such as *daisy chaining*, i.e., serial arrangement of pins. For this reason, alternatives realizing the required wiring type are made more attractive by decreasing $\gamma(N, j)$.

These cost criteria may also be considered for the selection of admissible wiring alternatives. This has to be done, for example, if a wiring alternative has to be strictly avoided. The objective

function of the integer program aims at a complete global wiring of all nets with minimum total cost. Because of the large number of variables we do not solve the integer program but its linear relaxation—in contrast to [27, 34]. The integer constraints are relaxed to $x_{(N,j)} \geq 0$. To relax capacity constraints violations of them are penalized in the objective function. Capacity violations of blocks b and block boundaries e are designated as $V_{c(b)}$ and $V_{c(e)}$, respectively. The penalty factor p is chosen very large, so violations of capacity constraints are very rare. Therefore, the objective function is the following:

$$min \left(\sum_{(N,j)} x_{(N,j)} \cdot (\gamma(N,j) + \lambda(N,j)) + \sum_{e} p \cdot V_{c(e)} + \sum_{b} p \cdot V_{c(b)} \right) \tag{2}$$

Using the *randomized rounding*–techniques proposed by Raghavan et al. [31, 33] we obtain an integer solution from the linear relaxation.

5.4 Layer assignment

In order to take into account global dependencies between nets, a layer assignment is determined already during global routing. The layer assignment aims at distributing the nets to the layers in such a way, that wirings of nets on the same layer do not interfere with each other electrically. For this purpose, several EMC–constraints have to be considered during layer assignment.

In order to avoid crossing conflicts already at high levels of abstraction, the layer assignment is done hierarchically during each refinement step. For each refinement step, the layer assignment of the previous level of the hierarchy is retained. (For instance, in figure 10, the wiring segment between blocks 1d and 3b will be assigned to the same layer as the wiring segment between blocks 1 and 3.) As a result, the layer assignment is subdivided into a series of independent subproblems. There is one subproblem corresponding to each primitive wiring problem. This decomposition makes an efficient heuristic solution of the NP–complete layer assignment problem

possible.

Let k be the number of available layers. The layer assignment problem is formulated as a graph k–coloring problem. The *conflict graph* to be colored consists of one vertex for each wiring segment of a net. There is a weighted edge between two vertices, if the corresponding wiring segments should not be assigned to the same layer, because they are in conflict. The criticality of conflicts is modeled heuristically via appropriately chosen edge weights.

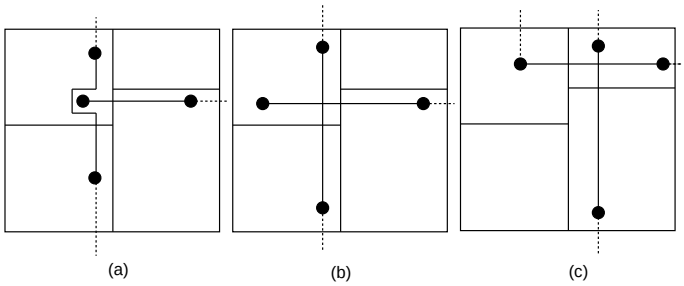

Figure 11: Evaluation of crossing conflicts

An edge between two wiring segments, that can be wired on one layer by using a short detour (see figure 11(a)), receives a low edge weight. In contrast, two wiring segments that intersect in the way shown in figure 11(b) should not be wired on the same layer. For this reason, there is an edge between these wiring segments with a large edge weight. The largest edge weight is chosen, if — as in figure 11(c) — additional vias would become necessary. Note that, in most cases, we can make only heuristic assumptions about conflicts, because we do not know the exact net courses during global routing.

The vertices of the conflict graph have to be colored with k colors. We use a heuristic similar that proposed by Turner [37]. First a k–clique is determined and colored with k colors. Then the remaining vertices are colored in a sequential manner. In each step the vertex with the smallest number of colors not used for adjacent vertices is chosen. If several vertices meet this criterion, the vertex with the highest adjacent edge weight is chosen. The selection of the color

for this vertex takes a series of cost criteria into account:

1. The layer assignment should avoid conflicts, that is, the sum of all edge costs between vertices of the same color should be minimal.

2. For each layer, a preferred wiring direction (horizontal or vertical) has to be obeyed. By alternating wiring directions crosstalk effects between adjacent layers can be largely avoided. In this design step crosstalk effects can only be handled by this simple heuristic.

3. For each net, the number of vias should be as small as possible. Therefore, colors already used for the corresponding net are preferred. The objective is to reduce production time and cost, and also to reduce reflections caused by layer changes.

4. If a layer change cannot be avoided, its cost depends on the difference between the characteristic impedances of the two connected layers.

5. Loads on different layers should not differ too much from each other. This is realized by balancing the distribution of the colors.

6. Each net may have preferred layers, for example critical nets should be routed on inner layers. This is considered during the selection of the color.

Obviously, some of these criteria contradict each other, so all these criteria enter the cost function appropriately weighted. The determination of appropriate weights is complex and requires a lot of further experience. For the examples in section 7, these criteria are used in the order given above to determine the color for a selected vertex. The critical transmission–line length l_k is not considered explicitly by the layer assignment.

In order to improve the coloring, an iterative process is executed after computing the initial k–coloring. This improvement ameliorates the disadvantages of the sequential coloring heuristic.

Nevertheless, the layer assignment during global routing cannot resolve all crossing conflicts. Unresolved conflicts are passed on to the detailed routing phase.

5.5 Refinement of wiring segments to the next level of hierarchy

Each wiring segment running between two blocks b_1 and b_2 has to be refined to two adjacent subblocks of b_1 and b_2, respectively. In general, there are several (at most seven) alternatives for every net. Similar to Lauther [22], and Lengauer and Müller [24], a cost is computed for each alternative. The cost depends on the pin positions of the net. Alternatives which result in short and straight routes receive small costs.

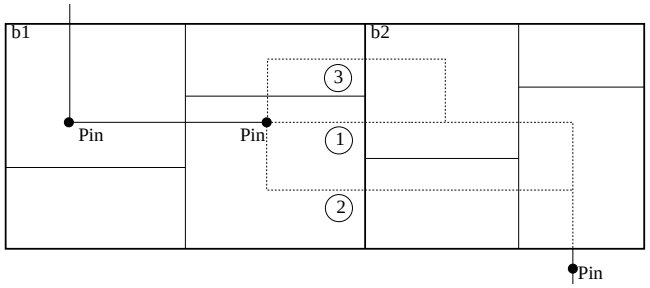

Figure 12: Different refinement alternatives for a net

For example, in figure 12, the net between the two blocks can be refined to the alternatives 1, 2, or 3. The best alternative is number 1, because it is the shortest and straightest one. Alternative 2 induces the same wiring length, but its realization in the detailed routing phase requires one bend more than alternative 1. The additional bend is caused by the positions of the connected pins and by the upper boundary of the lower left subblock of $b2$. The worst alternative is number 3, because this alternative corresponds to the one with the largest detour and with the largest number of bends.

The wiring segments are distributed among the subblocks such that capacities on each layer between blocks are not violated and such that the total cost of all alternatives is minimum

(*linear assignment problem*). As mentioned in section 5.3, for critical nets, more wiring space than usual is reserved. This problem is formulated as a *mincost–flow problem* (see [22]) and is solved by classical techniques.

6 Sequential detailed routing

During detailed routing, the exact route is determined for each net. The detailed routing phase is done block by block—as in [20] and in chip layout. Each net is split into *subnets*. There is one subnet for each block through which the net passes. Routing a block amounts to a *switchbox problem* [23]. There is a multitude of heuristic algorithms for this problem, which achieve good results in chip layout. However, these methods cannot be applied here, because they use detailed routing models (*Manhattan routing, knock–knee routing*) not capable of handling complex EMC cost criteria. For this reason, HERO applies an extension of sequential *maze–running* methods to general cost criteria.

Before a block is routed, the entry positions (*terminals*) of wires are still moveable along those block sides for which the adjacent block has not been routed before (*variable* block sides). Therefore, in [20] the exact entry positions are determined in a first step. However, it is hard to predict the best position for each wire. Therefore, we do not assign exact coordinates to the wires but compute a *wire ordering* for each variable block side. We use the *topological routability* as the primary criterion for ordering. Our objective is to obtain an ordering such that, using the given layer assignment, all blocks can be routed with few crossings. Up to now, only simple heuristics for solving this task are used. Before a block is routed, a wire ordering for variable block sides is determined such that the block is topological routable. As a second criterion we use heuristics to avoid conflicts for the blocks which are subsequently routed. We take notice of the pin positions and the global routing in the blocks not routed yet. For example, nets which

turn left in the upper neighboring block should be arranged left of those nets which run straight through this block, and nets running straight through this block should be arranged left of nets turning right.

We are working on improved heuristics for wire ordering based on [9, 10]. This approach takes a more global view on the problem by computing wire orderings for all blocks and block sides before any block is routed. We aim at combining wire ordering with layer assignment. Details of this approach will be described in a later publication.

6.1 Ordering the blocks

We use a two–step criterion for ordering the blocks. As the main criterion, HERO routes the blocks in an order, such that for each block at least one of the two block sides opposite to each other (top/bottom and left/right) is variable. In our experience such an ordering maximizes the chance that blocks can be routed to completion. As a secondary criterion, *difficult* blocks are routed with preference. As a result difficult blocks have variable entry positions on as many sides as possible. The difficulty of a block is measured by computing a figure of *congestion* which takes into account the block size, an estimation of the total internal wiring length, an estimation of the number of vias and the number of pins. The figure 13(a) exemplifies the block order. Blocks are labeled with their congestion (in percent) and the resulting routing order. Every variable block side is highlighted with a dotted line.

6.2 Ordering the nets

Each block is routed sequentially, i.e., net by net. This is done in three passes. During the first pass, each net is restricted to those layers proposed by the layer assignment, and the wire orderings have to be strictly observed. Here, only a subset of topologically routable nets is considered, which is output of the wire ordering phase. First experiences with our improved

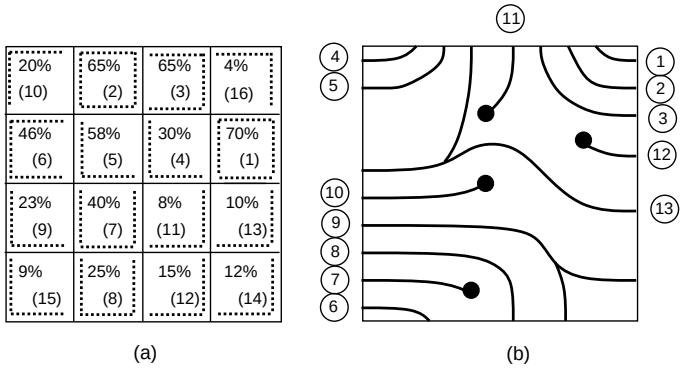

Figure 13: Block order and net order

heuristic for wire ordering indicate that most nets are in this subset. During the second pass each net not routed yet may use all layers. The third pass permits to change the layer assignment and wire ordering of variable block sides.

The sequential approach requires the determination of an appropriate net ordering. Taking into account the exact coordinates of pins, of terminals, and the wire ordering net ordering can be largely avoid nets blocking each other. During the first pass, the nets are ordered *from the corners inward*. This is possible because a subset of topologically routable nets is considered. As a result, nets can be routed along the contours of previously routed nets. Figure 13(b) exemplifies this heuristic for one layer. An extension of this heuristic can be applied, if not all terminals of a net are assigned to the same layer. During the second and third pass we use a random net ordering. This is because, at this time, the block is already highly congested—about 85% of the nets are routed to completion during the first pass—and the dependence between net ordering and completion rate is very hard to predict.

The question arises whether the net order should favor critical nets—similar to [29]. This strategy may result in an increase of blockings. In the future, *rip-up and reroute* techniques [6] will be applied, if a net cannot be routed. After global layer assignment and wire ordering, this

probably will be necessary for only few nets.

6.3 Detailed routing of a net

For detailed routing of a single net, HERO applies an extension of maze–running techniques to complex cost criteria. The block to be routed is represented by a 3–dimensional grid graph. Obstacles result from previously routed wires, vias, pins and other blocked areas. Routing of a net is performed by determination of an *optimal path* between its pins and terminals. EMC–constraints and cost criteria are mapped onto path costs, that is, a path with optimal cost corresponds to a route that is EMC–noncritical.

In [8, 28, 29, 38], classical shortest–path methods (with path costs over the real numbers) are used. Each edge cost depends on the noise that is caused, if the route of the net makes use of this edge. The cost of a path, defined by the sum of its edge costs, is a measure of the noise level of the corresponding route. Mapping realistic EMC–design rules onto path costs, however, leads to more complex path cost functions. We use the methods presented in [25, 26] which efficiently solve path problems for complex path cost functions not necessarily having the algebraic properties of the real numbers with addition. Using this extension of classical maze–running techniques, complex EMC–design rules can be mapped onto path costs. Different rules can be considered in concert by suitable multicriteria path costs. We apply methods of *goal–directed search* [23, 25] to execute the path search more efficiently.

The global EMC-design rules have to be mapped to rules that apply locally to each block. In order to make sure that, on the one hand, path costs reflect the EMC–cost criteria adequately and, on the other hand, an efficient determination of *optimal* paths is possible, the mapping has to be chosen carefully. We will exemplify this process with the crosstalk example (see subsection 3.2). Since the coupling level is additive, local rules can be set up for every net inside

the block, mostly depending on the portion of the coupling level not consumed in previously routed blocks. Let N be the net to be routed. In order to consider crosstalk phenomena, the cost of a path p has to reflect the overall coupling level $C(N)$ of net N and the overall coupling levels $C(N_i)$ of the nets N_i $(1 \leq i \leq k)$ routed before, which may have coupling regions with p. As a result, path costs are vectors with one component for the coupling level of net N, caused by the nets N_i, and one component for the coupling level of each net N_i, caused by N. Let e be an edge of the routing graph. If a coupling region of a line segment $S_{j(i)}$ of net N_i and edge e exists, let $R_i^{(e)}$ be this coupling region, for $1 \leq i \leq k$. Let T and T_i be the device technologies of net N and the nets N_i, respectively. Furthermore, let l_e be the length of e and d_i be the distance between e and $S_{j(i)}$, if e and N_i form a coupling region, and infinity otherwise. The component

$$c_0^{(e)} := \sum_{i=1}^{k} C_{R_i^{(e)}}(N) = \sum_{i=1}^{k} \frac{l_e}{l_c(T, T_i) \cdot f(d_i, T_i)} \tag{3}$$

of the edge cost of e gives the coupling level for net N on edge e, caused by the nets N_i. For $1 \leq i \leq k$ the component

$$c_i^{(e)} := C_{R_i^{(e)}}(N_i) = \frac{l_e}{l_c(T_i, T) \cdot f(d_i, T)} \tag{4}$$

gives the coupling level of net N_i, caused on segment $S_{j(i)}$ by net N, if routed on edge e. Here, T and T_i are the device technologies of net N and the nets N_i, respectively. The parameter l_e is the length of e, d_i is the distance between e and N_i, if e and N_i form a coupling region, and infinity else. Each edge can have coupling regions with at most four nets (up to two adjacent nets on the same layer and up to two on the layers above and below). Therefore, we have to store only a four–dimensional cost vector for each edge instead of a $k+1$–dimensional one. Path costs are defined by componentwise addition of edge costs. We are looking for a path such that the maximum of all cost components is minimal.

Unfortunately, the weakly NP–complete *shortest–weight–constrained–path problem* [7] forms a

special case of this path cost function. Therefore, it is not possible to determine optimal paths for this path cost in acceptable time. There are two possibilities to circumvent this problem:

1. Approximation. We also developed fast algorithms to determine good approximate solutions for general path problems [35]. Note that, for the crosstalk example, we need not compute a path with minimal coupling levels, but only a path that does not exceed the maximal coupling level $C_{\max}(N) = 1$, for all involved nets.

2. Simplifying the path cost function, such that an efficient determination of optimal paths is possible. Note, that this need not result in a more inaccurate estimation of EMC–effects, if it is done carefully.

For the crosstalk example, we make use of the second option. For this purpose, every net is classified as *critical* or *uncritical*. Critical nets are those nets, for which the overall coupling level, caused by coupling regions in previously routed blocks is high (e.g., greater than $C_{\text{krit}}(N) = 0.8$). In order to obtain this classification, the exact coupling levels are determined, using the path cost function mentioned above. A large cost is assigned to every edge having a coupling region with some net N_i, if either net N_i or N is critical. As a consequence, critical nets will be routed with a minimum number of coupling regions. The results presented in section 7 illustrate that, using this heuristic, the maximum coupling level is not exceeded in most cases.

Up to now, beside couplings, only critical transmission–line lengths (see subsection 3.1) are considered during detailed routing. This is done by adding a component for path length to the path cost vectors. In the future we will integrate more EMC–design rules into our robust routing framework.

7 Experimental results

In this section, we present experimental results of our router HERO. HERO was programmed in C and runs on UNIX workstations. In table 4, the characteristics of the two test boards are depicted. The more complex board (`board 2`) has also been designed under industrial requirements by layout engineers with many years of experience. Placement was made manually. Both boards consist of six layers plus additional V_{gg}– and V_{cc}–layers.

Board	Nets	Pins	Devices	Grid points	Layers
1	587	2683	229	575×400	6
2	1370	5874	997	850×600	6

Table 4: Technical data of the test boards

Figure 14 depicts a cut out of the final routing of one layer of `board 1`. Vias and pins are represented by dots. The underlying partitioning is a uniform grid. From 128 blocks of `board 1` the figure depicts only 16 blocks and some parts of the neighboring blocks. Wires ending at a via are continued on other layers. Wires ending at block borders could not be completed by the router, yet.

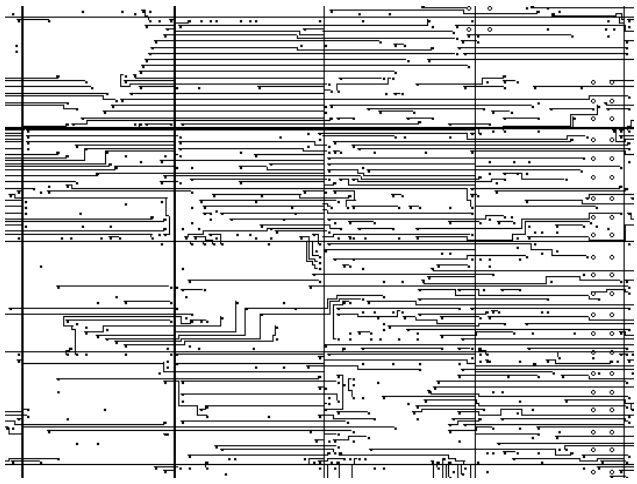

Figure 14: Cut out of the routing of `board 1`

In figure 15 the detailed routing for one layer of one of the 512 blocks of `board 2` is depicted. Its size is 2.08 $cm \times 1.48$ cm (51×36 grid points). Vias and pins are represented by filled circles. The nets depicted grey have been classified as crosstalk-critical, because their coupling levels, caused by couplings in previously routed blocks, are high. Clearly, it can be seen that these nets were routed with a safe distance (1 mm, three grid points) to other nets. Therefore, the nets routed after the critical nets—these are the nets above the net—have to make a small detour.

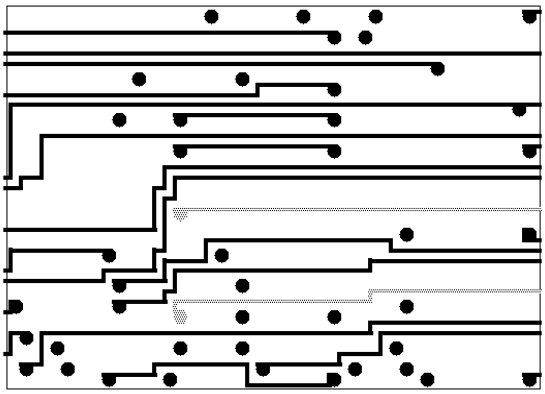

Figure 15: Detailed routing example

In table 5, the resulting net lengths (a) after global routing, (b) after detailed routing, and (c) the coupling levels are depicted. All values are normalized to the maximal admissible value γ, resulting from EMC–design rule 1 and 2, respectively. As a result of placement, the distance between the pins of many nets is larger than the respective critical transmission–line length. For this reason, we chose the maximum of the critical length and 1.2 times the maximum pin distance as the maximum admissible line length for every net. The results in table 5 indicate that the net lengths are often smaller after than before detailed routing. This is no contradiction, because after global routing we have only estimations of the net lengths. 90.8% of all nets from `board 1` and 95.4% of all nets from `board 2` do not violate the maximum admissible value for the net length. 98.5% and 99.7% of the nets, respectively, stay within 1.3 times the maximum

value. Using an improved placement procedure, even better results can be expected. 92.8% (95.8%) of all nets do not violate the maximum coupling level, 99.0% (99.2%) do not exceed 1.3 times the maximum coupling level. Since HERO did not route the boards to completion, for a complete routing the coupling levels will be slightly increased. For nets not completely routed the lengths of the missing parts were estimated in the Manhattan metric. The industrial PCB layout experts, with whom we cooperate, classify our results as being of very high quality. In addition, detailed reports of rule violations provide an effective input to the verification phases.

	$< 1.0\gamma$	$< 1.3\gamma$	$< 1.5\gamma$	$< 2\gamma$	$> 2\gamma$
`board 1`					
(a)	527	40	15	5	0
(b)	533	45	7	2	0
(c)	545	36	2	4	0
`board 2`					
(a)	1308	58	1	3	0
(b)	1298	65	5	2	0
(c)	1312	47	7	4	0
`board 2` (EMC-design rules turned off)					
(b)	1299	66	3	2	0
(c)	1186	41	18	37	88
`board 2` (manual)					
(b)	1097	210	46	16	1
(c)	802	92	53	105	318

Table 5: Classification of nets regarding to critical length and coupling level:
(a) net lengths after global routing and (b) after detailed routing, (c) coupling levels

A comparison of our results with other routers is not possible, because of the lack of benchmarks, because of missing standards in EMC–modeling and, none the least, because very few layout systems take EMC–phenomena into account at all.

For `board 2`, we compared the results to the layout made by our router with EMC–design rules 1 and 2 turned off and to the layout made manually by layout engineers. They did not consider EMC–design rules explicitly here, but brought in their practical experience. Hand layout is still the state–of–the–art in this segment of PCB design.

Table 5 indicates that our results—with respect to the coupling level (c)—exceed those of automatic layout with EMC–design rules turned off and of hand–made layout dramatically in quality. The comparison shows that HERO is able to deal with restrictive EMC–constraints. Some rule violations of the manual layout are caused by address and data buses that were realized as parallel wires. This is not critical for the current technologies. Buses are not considered by our router, yet. This is a point of future work. Note, that rule violations do not always lead to non–functional PCBs. This fact indicates, that our design rules have to be refined and to be weakened in some cases. For example, the critical coupling length for two nets depends on more net–specific data and not only the technologies of the nets.

In table 6, the run times for global and detailed routing are depicted. In comparison with other routers, the run time for detailed routing seems to be exceedingly high. Note, however that, up to now, commercial printed–circuit board routers gain only poor acceptance. This is, because they do not provide any EMC–criteria and produce a large number of vias. Up to now we have paid attention to run time aspects only marginally. For this reason we expect to improve our run times significantly in the future.

	`board 1`	`board 2`
Global routing	17min	50min
Detailed routing	3h	12h
Completion rate	92.0%	92.6%

Table 6: Run time (real time, Sun4 Sparc SLC) and completion rate for `board 1` and `board 2`

The *completion rate* is defined as the ratio of the number of routed subnets through the number of all subnets. Although HERO does not use *rip–up and reroute*–techniques, the completion rate is quite good. This is a consequence of the global routing phase including layer assignment. The completion rate can certainly be increased by using improved heuristics during layer assignment and wire ordering as mentioned at the beginning of section 6. In the future, rip–up and reroute techniques will be applied, if a net cannot be routed.

In contrast to most PCB routers that work strictly sequentially, the total number of vias generated by HERO is not substantially greater than with manual layout. For `board 1` 1456 vias are required compared to 1409 vias generated manually. Note, however, that the board is not completely routed. In order to decrease the number of vias, future work will have to improve our heuristics for layer assignment and detailed routing.

8 Summary

In order to carry out the design of printed–circuit boards as time– and cost–efficient as possible, EMC–phenomena have to be taken into account during layout synthesis. EMC–design models consisting of geometric constraints and cost criteria are used to avoid noise effects resulting from EMC–phenomena. HERO offers a robust framework for incorporating complex EMC–design models into routing. In general, it will not be possible to obtain a completely failsafe layout. However, experimental results for typical boards prove that a great number of EMC–problems can be avoided during layout synthesis and that the effects of EMC–phenomena can be reduced substantially. Detailed reports of EMC–design rule violations provide effective input to the succeeding EMC–verification phase. The remaining violations of EMC–design rules are mainly caused by an inappropriate placement. We are also working on methods for EMC–driven placement. These methods will be presented in a later publication. It seems to be of

great promise, to combine hierarchical placement methods with our approach for hierarchical routing—similar to [24].

9 Acknowledgements

Anja Feldmann contributed substantially to early stages of this research. Furthermore, we thank Henrik Schmidt, Nina Graf and Udo Jacob for supporting our work. We are grateful to Siemens–Nixdorf Informationssysteme AG for providing test data.

References

[1] H.B. Bakoglu. *Circuits, Interconnections and Packaging for VLSI.* Addison Wesley, 1990.

[2] M. Burstein and R. Pelavin. Hierarchical wire routing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, CAD-2(4):223–234, 1983.

[3] U. Choudhury and A. Sangiovanni-Vincentelli. Constraint based channel routing for analog and mixed analog/digital circuits. *ICCAD*, pages 198–201, 1990.

[4] U. Choudhury and A. Sangiovanni-Vincentelli. Constraint generation for routing analog circuits. *Proceedings of the 27th ACM/IEEE DAC*, pages 561–566, 1990.

[5] J. Cong, A.B. Kahng, G. Robins, M. Sarrafzadeh, and C. K. Wong. Provably good performance-driven global routing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, CAD-11(6):739–752, 1992.

[6] W.A. Dees and P.G. Karger. Automated ripup and reroute techniques. *Proceedings of the 19th ACM/IEEE DAC*, pages 432–439, 1982.

[7] M.R. Garey and D.S. Johnson. *Computers and intractibility.* W.H. Freeman, 1979.

[8] D.J. Garrod, R.A. Rutenbar, and L.R. Carley. Automatic layout of custom analog cells in ANAGRAM. *ICCAD*, pages 544–547, 1988.

[9] P. Groeneveld. On global wire ordering for detailed routing. In *Proceedings of the 26th Design Automation Conference*, pages 155–161. ACM/IEEE, 1989.

[10] P. Groeneveld. Wire ordering for detailed routing. *IEEE Design & Test of Computers*, pages 6–17, December 1989.

[11] R.S. Gyurcsik and J.-C. Jeen. A generalized approach to routing mixed analog and digital signal nets in a channel. *IEEE Journal of Solid State Circuits*, pages 436–442, 1989.

[12] T.C. Hu and M. T. Shing. A decomposition algorithm for circuit routing. *Mathematical Programming Study*, 24:87–103, 1985.

[13] W. John. Remarks to the solution of EMC-problems on printed circuit boards. *7th International Conference on EMC, Univ. of York, UK*, 1990.

[14] W. John. Application of microelectronic devices on printed circuit boards under EMC-aspects (in German). *GME-Fachbericht 9, EMV Störfestigkeit leittechnischer Einrichtungen, GME-Fachtagung, Bad Nauheim*, pages 207–232, 1991.

[15] W. John. Support of printed circuit board design by an EMC-workbench. *10th International Zurich Symposium & Technical Exhibition on Electromagnetic Compatibility, Zurich*, 1993.

[16] W. John, J. Hoener, and O. Rethmeier. FREACS - A fast reflection and crosstalk simulator. *CompEuro 91*, pages 526 – 530, 1991.

[17] W. John, D. Theune, and R. Thiele. Layout design for printed-circuit boards under EMC-constraints. *Electromagnetic Compatibility 1993, 10th International Zurich Symposium & Technical Exhibition on Electromagnetic Compatibility, Zürich*, pages 225–232, 1993.

[18] R.M. Karp, F.T. Leighton, R.L. Rivest, C.D. Thompson, U.V. Vazirani, and V.V. Vazirani. Global wire routing in two-dimensional arrays. *Proceedings of the 24th Annual Symposium on Foundations of Computer Science*, pages 453–459, 1983.

[19] R.M. Karp, F.T. Leighton, R.L. Rivest, C.D. Thompson, U.V. Vazirani, and V.V. Vazirani. Global wire routing in two-dimensional arrays. *Algorithmica*, 2:113–129, 1987.

[20] J. Kessenich and G. Jackoway. Global Forced Hierarchical Router. *Proceedings of the 23rd ACM/IEEE DAC*, pages 798–802, 1986.

[21] S. Kiefl. A new hierarchical concept for the physical design of PCBs. *Int. Workshop on Placement and Routing, Research Triangle Park, North Carolina*, 1988.

[22] U. Lauther. Top down hierarchical global routing for channelless gate arrays based on linear assignment. In Carlo H. Sequin, editor, *Proceedings of VLSI'87*, pages 141–151. Elsevier Science Publ. B.V., Amsterdam, 1987.

[23] T. Lengauer. *Combinatorial algorithms for integrated circuit layout*. Teubner Wiley Series in Computer Science. John Wiley and Sons, New York, 1990.

[24] T. Lengauer and R. Müller. A robust framework for hierarchical floorplanning with integrated global wiring. *ICCAD*, pages 148–151, 1990.

[25] T. Lengauer and D. Theune. Efficient algorithms for path problems with general cost criteria. *18th Int. Symposium on Automata, Languages and Programming, Springer LNCS No. 510*, pages 314–326, 1991.

[26] T. Lengauer and D. Theune. Unstructured path problems and the making of semirings. *2nd Workshop on Algorithms and Data Structures, Ottawa, Canada, Springer LNCS No. 519*, pages 189–200, 1991.

[27] W. K. Luk, P. Sipila, M. Tamminen, D. Tang, L. S. Woo, and C. K. Wong. A hierarchical global wiring algorithm for custom chip design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, CAD-6(4):518–533, 1987.

[28] E. Malavasi, M. Chilanti, and R. Guerrieri. A general router for analog layout. *COMPEURO '89, Hamburg*, pages 549–551, 1989.

[29] E. Malavasi, U. Choudhury, and A. Sangiovanni-Vincentelli. A routing method for analog integrated circuits. *ICCAD*, pages 202–205, 1990.

[30] S. Mitra, S.K. Nag, R.A. Rutenbar, and L.R. Carley. System-level routing of mixed-signal ASICS in WREN. *ICCAD*, pages 394–399, 1992.

[31] A. P.-C. Ng, P. Raghavan, and C. D. Thompson. Experimental results for a linear program global router. *Computers and Artificial Intelligence*, 6(3):229–242, 1987.

[32] O.A. Olukotun and T.N. Mudge. Hierarchical gate–array routing on a hypercube multiprocessor. *Journal of Parallel and Distributed Computing*, 8:313–324, 1990.

[33] P. Raghavan and C. D. Thompson. Randomized rounding: A technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–374, 1987.

[34] R.W. Thaik, N. Lek, and S.-M. Kang. A new global router using zero-one integer linear programming techniques for sea-of-gates and custom logic arrays. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, CAD-11(12):1479–1494, 1992.

[35] D. Theune. *Robust and efficient methods for solving path problems (in German).* PhD thesis, University of Paderborn, 1994. (in preparation).

[36] D. Theune, R. Thiele, T. Lengauer, and A. Feldmann. HERO: Hierarchical EMC-constrained routing. *ICCAD*, pages 468–472, 1992.

[37] J.S. Turner. Allmost all k–colorable graphs are easy to color. *Journal of Algorithms*, 9:63–82, 1988.

[38] K. Wawryn. Layout including parasitics for printed circuit boards. *Int. Journal Circuit Theory Appl. (UK)*, 6(2):107–128, April 1988.