Parallel Algorithms for the Simulation of Lossy Transmission Lines

H. Holzheuer*, W. Rissiek**, O. Rethmeier**
* Universität–GH Paderborn/Cadlab, ** Siemens–Nixdorf Informationssysteme AG/Cadlab
Bahnhofstraße 32 – D-33094 Paderborn

Abstract

The simulation of lossy transmission lines in the time domain is a very time consuming task. It requires numerical convolutions and the solution of linear and nonlinear equation systems.

In this paper a model for lossy transmission lines and an algorithm for the simulation of these lines is presented. Based on this an approach for the parallel simulation of lossy transmission lines is developed. As this approach covers all time consuming tasks of the simulation a good speed up compared to a single processor solution is expected. Preliminary results are presented at the end of the paper.

1 Introduction

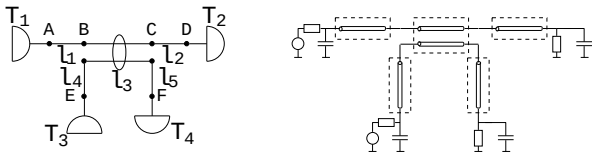
Due to the raising complexity and increasing clock rates of electronic designs the analysis of complex transmission line structures with respect to reflection and crosstalk effects is becoming increasingly important. This type of simulation requires a large computational effort especially if losses of transmission lines have to be considered.

The simulator *FREACS* provides an efficient analysis of reflection and crosstalk effects on lossless transmission lines considering the nonlinear characteristics of the terminations [3, 4]. An optimal adjusted step size for the evaluation of all transmission lines and termination networks is determined by a special simulation algorithm. Due to this specialisation the simulation time for complex examples can be reduced up to factor 10 compared to SPICE [8].

In this contribution new algorithms for the simulation of lossy transmission lines are described. As these algorithms require a lot of computation time a concept for the application of message based multiprocessor systems is discussed. The number of applied processors can be scaled according to the complexity of the transmission line structures that are analysed. In order to reduce the communication overhead the topology of the processor network matches the simulated transmission line structure.

2 Transmission Line Structures

Advanced tool environments [3] enable the electronic engineer to pre–analyse the layout data of a printed circuit board (PCB) with respect to reflection and crosstalk effects on transmission lines. The aim of this pre–analysis is a decomposition of the complex layout into small transmission line structures and a classification whether they are EMC–critical [7, 9] or not. Nevertheless, this pre–analysis of an average sized PCB results in a high number of critical transmission line structures and a simulation of all these structures needs an enormous amount of time. A typical example of a transmission line structure is presented in figure 1.



transmission line structure **equivalent circuit**

Figure 1: Transmission line example

For a simulation *FREACS* partitions such a transmission line structure into termination networks (terminations), transmission line sections (areas) and discontinuities. Therefore the drivers (T_1, T_3) and receivers (T_2, T_4) in the example are classified as terminations, and the four single transmission lines (l_1, l_2, l_4, l_5) as well as the coupled section of the two lines in parallel (l_3) are areas. Every time when two or more areas are connected together (B, C) we call it a discontinuity. In order to simulate such a structure Kirchhoff's current law is used to set up an equation system for every termination and every discontinuity. For the computation of the areas the '*model of delayed current components*' [4] is applied.

So two main parts have to be carried out for each timepoint to be calculated. First, the '*delayed current*

components' of both ends (ports) of each area have to be computed. The results are used to set up the equation systems for the terminations and discontinuities which are solved using a numerical integration method. Due to the delay behaviour of the transmission lines these terminations and discontinuities can be solved independently at each timepoint.

3 Fundamentals

Before describing the parallelisation a short introduction is given about the algorithms that are used for the simulation of a lossy transmission line structure. Here we focus on the '*model of delayed current components*', the discontinuities, and some basics of the simulation of terminations.

3.1 Delayed Current Components

As mentioned above, only the ports of an area are of interest for the simulation (see figure 2). Taking a look on such an area as a network component we have to consider its properties in form of equations like $\mathbf{i}_A = f(\mathbf{v}_A, \mathbf{v}_B)$ and $\mathbf{i}_B = f(\mathbf{v}_A, \mathbf{v}_B)$. Different approaches [4, 6] can be used to describe a lossy transmission line in such a way. We use the '*model of delayed current components*' [4] for this purpose. The model is based on the Quasi–TEM wave propagation and results from the telegraph equations describing coupled transmission lines.

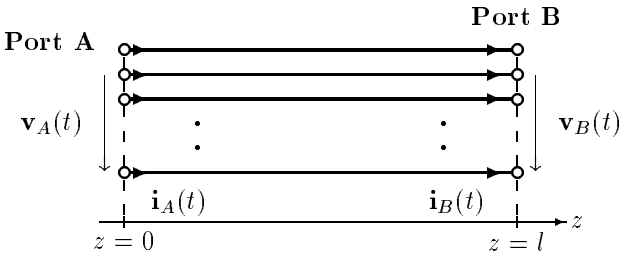


Figure 2: Area of n coupled transmission lines

For lossy transmission lines the calculation has to be done in the frequency domain. But as nonlinear terminations have to be considered during the simulation, the analysis itself cannot be performed in the frequency domain. By transforming the equations into the time domain, a multiplication in the frequency domain becomes a convolution in the time domain. Therefore, the simulation of lossy transmission lines leads to an enormous amount of computation time.

Since the resulting convolution integrals are solved numerically, a discretisation is used. To calculate the current $\mathbf{i}_A$ at the n^{th} timepoint we use equation (1) wherein T represents the overall simulation timestep (currents at port B are calculated respectively):

$$\mathbf{i}_{An} = \underbrace{\underbrace{T \cdot \mathbf{y}_{L0}}_{\mathbf{g}_A} \cdot \underbrace{\mathbf{v}_{An}}_{unknown}}_{\mathbf{i}_{0An}} + \mathbf{i}_{TAn} + T \cdot \sum_{m=1}^{k_{max}} \mathbf{y}_{Lm} \cdot \underbrace{\mathbf{v}_{An-m}}_{already\ known} \quad (1)$$

$$\mathbf{i}_{TAn} = T \cdot \sum_{m=k_{min}}^{k_{max}} \mathbf{g}_{Ltg_m} \cdot (2 \cdot \mathbf{i}_{Bn-m} - \mathbf{i}_{TBn-m}) \quad (2)$$

$\mathbf{y}_L(t)$ (see equation (1)) is the so called characteristic admittance matrix which describes the port behaviour of the area and $\mathbf{g}_{Ltg}(t)$ in equation (2) models the delay behaviour of the area. To solve equation (1) the delayed current component $\mathbf{i}_{TA}$ is required. The name results from the fact that its calculation refers only to currents from the opposite port being at least $k_{min} \cdot T$ ago.

As $\mathbf{v}_A$ is unknown for the calculation of the n^{th} timepoint, equation (1) is splitted into two parts. First the equation is solved only for $\mathbf{i}_{0A}$. Then this current and $\mathbf{g}_A$ are used to calculate the voltage $\mathbf{v}_A$ taking into account the neighboured discontinuities and terminations by Kirchhoff's current law. Finally the current $\mathbf{i}_A$ can be calculated.

3.2 Discontinuities

Concerning the calculation of voltages, discontinuities and terminations have to be distinguished.

The example presented in figure 3 shows how the transmission lines of different areas are connected to build a discontinuity. Every connection of lines is represented by one node (N_1, N_2, N_3). It is remarkable that the number of nodes describing a discontinuity can be higher than the maximum number of coupled transmission lines in the adjacent areas, but it will never exceed $\lfloor \frac{m}{2} \rfloor$ with m as number of transmission lines connected together in the discontinuity.

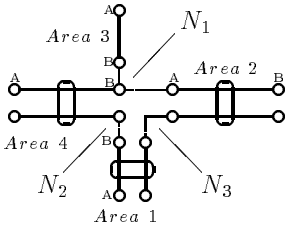


Figure 3: Discontinuity

To set up a linear equation system for the calculation of the required voltage vector $\mathbf{v}_D$, a simple equiva-

lent circuit can be given for every discontinuity (see figure 4). In this example the voltage vector $\mathbf{v}_D$ is of dimension $l = 2$ and contains the voltages at the nodes N_1 and N_2.

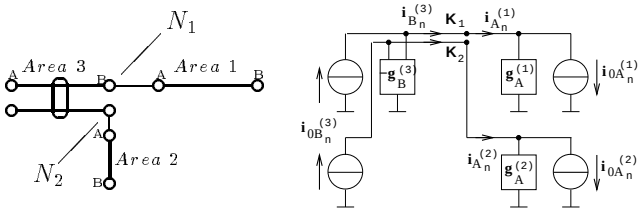


Figure 4: Equivalent circuit for a discontinuity

For each area a relation can be given containing the unknown voltage vector $\mathbf{v}_D$ by expanding equation (1) appropriate to the connection scheme of the discontinuity. An example for area 2 of figure 4 is presented in equation (3).

$$\underbrace{\begin{pmatrix} 0 \\ \mathbf{i}_{A_n}^{2} \end{pmatrix}}_{\mathbf{i}'^{\,2}_{A_n}} = \underbrace{\begin{pmatrix} 0 \\ \mathbf{i}_{0A_n}^{2} \end{pmatrix}}_{\mathbf{i}'^{\,2}_{0A_n}} + \underbrace{\begin{pmatrix} 0 & 0 \\ 0 & \mathbf{g}_A^2 \end{pmatrix}}_{\mathbf{g}'^{\,2}_{A}} \cdot \mathbf{v}_{Dn} \tag{3}$$

Corresponding to the 'Kirchhoff law' $\mathbf{i}_{B_{3n}}$ can be calculated by:

$$\mathbf{i}'^{\,3}_{Bn} = \mathbf{i}'^{\,1}_{An} + \mathbf{i}'^{\,2}_{An} \tag{4}$$

Considering equation (3), a linear equation system for the determination of voltages can be set up:

$$\underbrace{\mathbf{i}'^{3}_{0\mathrm{B}n} - \mathbf{i}'^{1}_{0\mathrm{A}n} - \mathbf{i}'^{2}_{0\mathrm{A}n}}_{\mathbf{i}_{0\mathrm{d}}} = \underbrace{\mathbf{g}'^{1}_{\mathrm{A}} + \mathbf{g}'^{2}_{\mathrm{A}} - \mathbf{g}'^{3}_{\mathrm{B}}}_{\mathbf{g}_D} \cdot \mathbf{v}_{Dn} \tag{5}$$

As the resistive circuits $\mathbf{g}_A$ and $\mathbf{g}_B$ of the areas are constant, the matrix $\mathbf{g}_D^{-1}$ that is required for the calculation of voltages can be set up initially.

3.3 Terminations

Termination circuits may contain linear or nonlinear macromodels for the description of the gate behaviour as well as simple discrete elements like resistors and capacitors. For the evaluation different methods like 'nodal analysis' or 'sparce tableau algorithm' [2] can be used. In the first version of the parallel simulator only the nodal analysis for linear termination is implemented. Later, nonlinear terminations will be integrated that are solved by sparce tableau and Newton–Raphson algorithm.

A linear termination can be described for timepoint n by:

$$\mathbf{i}_0 = \mathbf{Y}_{Lin} \cdot \mathbf{v}_{Tn} \tag{6}$$

In order to calculate the unknown voltage vector $\mathbf{v}_{Tn}$ of the termination the current vector $\mathbf{i}_0$ is set up using the port currents $\mathbf{i}_{0An}/\mathbf{i}_{0Bn}$ of equation (1) and source currents of the stimuli. $\mathbf{Y}_{Lin}$ consists of a constant part formed by the resistive circuits $\mathbf{g}_A$ and $\mathbf{g}_B$ (see figure 4) of the connected areas, and all constant elements in the termination, and a time dependent part for capacitances and inductances.

After the calculation the integration error is examined for capacitances and inductances. If the error is too large, the calculation for the last time interval is redone with a smaller timestep. The necessary port currents to calculate $\mathbf{i}_0$ are gained by the interpolation of given values.

4 The Parallelisation

Due to the high number of transmission line structures to be simulated and the enormous amount of simulation time for consideration of lossy lines an acceleration of this simulation is necessary. Some articles prefer a simplification of the basic fundamentals of lossy transmission lines, others try to improve the calculation by applying different mathematical techniques [6]. The usage of modern parallel computers allows both accurate line models and a reduction of computation time.

4.1 A Coarse–Grained Approach

In a first approach to parallelise the problem the partitioning of subnets into areas and terminations can be used as basis for parallel computation tasks.

```
Data Input;
Initial computation of g_Ltg and y_L;
DC analysis;
While (Actual timepoint < Simulation end)
  | For (all areas) Do
  |   | Compute the delayed current components;
  |   | Compute the current i_0(t);
  | End;
  | Send the current to all neighboured processors;
  | Receive required data;
  | For (all ports) Do
  |   | Compute the current and the voltage for all
  |   |  discontinuities and terminations;
  | End;
End;
```

Figure 5: Parallel simulation loop

The areas are distributed to different processors at the beginning of the simulation. For the evaluation at

time t the single processor elements carry out the necessary convolutions for all ports of the assigned areas. Then, an exchange of the data required for the calculation of voltage vectors $\mathbf{v}_D$ and $\mathbf{v}_T$ takes place. After this the simulation can continue with the analysis of the next timestep.

By this the complete simulation process is divided into toggling blocks of 'calculation' and 'communication' which is sometimes called 'lock step synchronisation' in literature. This way of parallelisation is very efficient if a good load balancing can be found and the communication overhead is small.

4.2 Redefinement of the Approach

A first examination shows that a lot of transmission line structures on PCB's often consist only of a few lines. Furtheron, the size of coupled areas varies so that a good load balancing is not easy to be found. Therefore a refinement of the parallelisation described above is necessary. For this, the time consuming calculation of the convolutions for the determination of delayed current components $\mathbf{i}_{TA}(t)$ and the currents $\mathbf{i}_A(t)$ is examined first.

4.2.1 Parallelisation of the Convolution

A load analysis of a sequential simulator prototype shows, that most of the time is spent calculating the convolutions (see sum in equation (1) and (2)). A closer view on these discrete convolutions shows two main characteristics which are important for the parallelisation.

- both convolutions result in a vector with the dimension of the respective area,
- both convolutions are calculated by summing up a matrix vector product.

In order to parallelise this problem a general approach presented in the following can be used. It is $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{n \times n}$ and $\mathbf{x} = (x_j) \in \mathbb{R}^n$. The calculation of the matrix vector product $\mathbf{y} = (y_j) \in \mathbb{R}^n$ is elementarily defined by:

$$y_i = \sum_{j=1}^{n} a_{ij} \cdot x_j \tag{7}$$

A matrix vector product is set up in parallel, when the vector $\mathbf{x}$ is pushed from the left and the rows of matrix $\mathbf{A}$ are pushed from above through a processor array so that each processor j can process its element y_i after $2n-1$ steps by summing up the product of the elements arriving after each step (see figure 6, [5]).

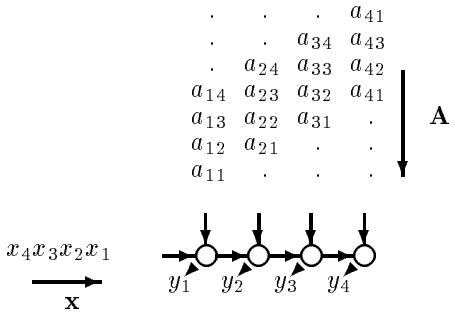


Figure 6: Parallel matrix vector product

A significant speed up of the calculation is obtained if each processor j has direct access to the vector element x_j and the rows (a_j) of the matrix $\mathbf{A}$. The resulting parallel algorithm uses a ring of n processors where the elements of $\mathbf{x}$ are pushed n times to the right through the ring to calculate $\mathbf{y}$. This algorithm shown in figure 7 needs n multiplication/addition steps.

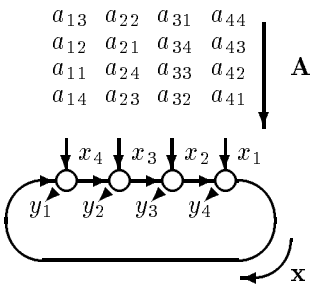


Figure 7: Acceleration of the matrix vector product

Besides this calculation a summation of vector elements y_α is necessary for all discrete points of the convolutions defined in the equations (1) and (2). This can be achieved if all these matrix vector element operations are performed sequentially on the same processor element.

If l ($l = k_{max} - k_{min} + 1$ see equation (2)) is the number of the required matrix vector multiplications of dimension n, the calculation time for the convolution is reduced from $O(n^2 \cdot l)$ in the sequential case to $O(n \cdot l)$ with n processor elements in parallel.

In order to reduce the communication overhead, the order of 'discrete convolution' and 'matrix vector multiplication' is changed.

$$z_i = \sum_{m=k_{min}}^{k_{max}} \sum_{j=1}^{n} a_{ij}^m \cdot x_j^m = \sum_{j=1}^{n} \underbrace{\sum_{m=k_{min}}^{k_{max}} a_{ij}^m \cdot x_j^m}_{(\star)} \tag{8}$$

Let's assume that all columns i of a_{ij} are located at processor i of the ring. In this case $k_{max} - k_{min} + 1$ multiplications can be performed to compute term

($\star$) of equation (8) before all vector elements x_j^m have to be transmitted to the next processor to compute for the next j. After n communication/calculation periods the solution of the convolution is obtained.

So the rearrangement in equation (8) leads to a reduction to $O(n-1)$ communications (n number of coupled transmission lines) of vectors containing $k_{max} - k_{min} + 1$ elements.

4.2.2 Discontinuities

To calculate the voltage vector $\mathbf{v}_D$ of a discontinuity using equation (5) a simple matrix vector operation is required. Therefore, some processors are collected to one 'supervised ring'. For this ring we can use some of the processors that just calculated $\mathbf{i}_{TA}(t)$ and $\mathbf{i}_A(t)$. Each of the processors contain one row of $\mathbf{g}_D^{-1}$ and the current vector element $\mathbf{i}_{0d}$. So the voltages can be calculated with regard to the already known parallel matrix vector algorithm. After 'sending back' the single voltages the currents of the ports at the neighboured areas can be calculated.

4.2.3 Terminations

To calculate the voltage vector $\mathbf{v}_T$ the linear equation system (6) has to be solved. Considering the small dimension of the appropriate terminations this can be done with the help of the '*Gaussian Elimination*' (LU–decomposition). As the admittance matrix $\mathbf{Y}_{Lin}$ is mainly diagonal dominant for linear circuits, a pivot strategy is not necessary. With $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{n \times n}$ and $\mathbf{x} \in \mathbb{R}^n$ the LU–decomposition is defined as follow:

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{L} \cdot \mathbf{U} \cdot \mathbf{x} = \mathbf{b} \;\rightarrow\; \mathbf{L} \cdot \mathbf{y} = \mathbf{b},\; \mathbf{U} \cdot \mathbf{x} = \mathbf{y} \qquad (9)$$

$$l_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}},\; a_{ij}^{(k+1)} = \begin{cases} a_{ij}^{(k)} - l_{ij} \cdot a_{kj}^{(k)} & i \geq k, j \geq k \\ a_{ij}^{(k)} & else \end{cases} \qquad (10)$$

For a parallelisation of this LU–decomposition a linear processor array with p processors is optimal [1]. Therefor the matrix $\mathbf{A}$ is stored cyclic after rows on p processors (see definition).

Def.: Matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is called cyclic stored after rows (columns) in a multiprocessor system with the processors $P_1 ... P_p$, if P_i contains exactly the rows (columns) j of $\mathbf{A}$, with $j \equiv i \; mod \; p$.

This mapping is also called 'interleaved storage', 'torus assignment', or 'wrapped mapping'.

For this storage technique it can be shown that the number of rows assigned to the processors differs uppermost at 1 (proof see [1]). This means a good load balancing for the processor elements. 12 possible forms of the algorithms are emerging by the variation of the indices i, j and k in the loops of the algorithm and the cyclic storing [1]. In the following we consider only the asynchron kij–form promising the highest efficiency for our application.

For $(k = 1$ to $n-1)$
| **If** $(k \in myrows)$
| | broadcast$(a_{kk}^{(k)} ... a_{kn}^{(k)})$;
| **Else**
| | receive$(a_{kk}^{(k)} ... a_{kn}^{(k)})$ from P_k;
| **End;**
| **For** $(i \geq k+1$ and $i \in myrows)$
| | $l_{ik} = a_{ik}^{(k)} / a_{kk}^{(k)}$;
| | **For** $(j = k+1$ to $n)$
| | | $a_{ij}^{(k+1)} = a_{ij}^{(k)} - l_{ik} \cdot a_{jk}^{(k)}$;
| | **End;**
| **End;**
End;

Figure 8: Parallel LU–decomposition

The above mentioned algorithm is directly applied to the linear equation system (6). After $\mathbf{y}$ (see equation (9)) is computed $\mathbf{v}_T$ can be determined by a simple backward substitution.

Because of small terminations only those processors are used in the LU–decomposition which are incident to the connected transmission lines. As the number of processors being involved in the algorithm is small, we can collect them to a '*clique*' to speed up the necessary broadcast operation.

5 Implementation and Preliminary Results

The implementation of the introduced simulator has to cope the variety of size and complexity of real transmission line structures and an arbitrary number of processor elements. This was obtained by realising the above mentioned algorithms as independent threads (lightweight processes). These threads can be placed on any computer topology so that a good load balancing can be found. The interconnection structure of the transmission lines is initially mapped in a virtual topology to create all communication channels of the simulator. The whole process is working in a lock step manner of calculation and communication and is synchronised by sending and receiving the required data, which is often *called data* or *event driven*.

As the implementation is not yet finished only preliminary results achieved in a test environment can be

presented. This test environment provides the parallel simulation kernel with test data that emulates a real simulation. The values for $\mathbf{g}_{Ltg}$ and $\mathbf{y}_L$ as well as the DC–values for $\mathbf{i}$, $\mathbf{i}_T$ and $\mathbf{v}$ are initialised by predefined reals as the calculation of these variables has not been implemented up to now. Then the parallel transient simulation process is activated which includes the evaluation of the numerical convolution (1, 2) and the data exchange between all affected processors for all timesteps to be simulated.

The simulation of the complex example in figure 9 reflects all assumptions about the coarse– and fine–grained approach. Due to the time consuming evaluation of the area l_3 with five coupled lines a maximum speedup of 2.2 can be achieved if only the coarse–grained approach is used. The application of the fine–grained approach where every transmission line is assigned to one processor leads to an improvement of the speedup to 10 (53% efficency) using 19 processors. At last the different load for the calculation of the convolutions ($O(n \cdot l)$ see chapter 4.2.1) is taken into account to find a good load balancing. Here 11 processors attain a speedup of 10.1 (92% efficency).

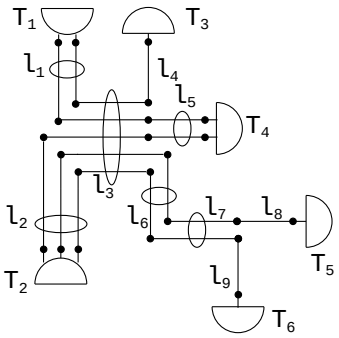


Figure 9: Complex example

Summary and Outlook

The consideration of losses in the simulation of transmission lines in the time domain requires an enormous amount of time and computational power. This results mainly from numerical convolution integrals. The parallelisation approach presented in this paper covers all necessary tasks of the simulation, so that a good speedup compared to a sequential solution is attained.

Actually we are working on the completion of the DC-analysis and the treatment of terminations so that realistical simulations are possible. Additionally, further work focusses on the improvement of good load balancing strategies, as this seems to be the major point to get a fast parallel simulator.

Acknowledgements

This work is part of the OPAL project (*On line placement on printed circuit boards*) which is supported by the German Government, Department of Economics under grant 9146 B. The responsibility for this publication is held by the authors only.

References

[1] A. Frommer. *Lösungen linearer Gleichungssysteme auf Parallelrechnern.* Vieweg Verlag, 1990.

[2] E.-H. Horneber. *Simulation elektrischer Schaltungen auf dem Rechner.* Springer Verlag, 1985.

[3] W. John. Support of Printed Circuit Board Design by an EMC–Workbench. *EMC–Zurich, 10th International Zurich Symposium on Electromagnetic Compatibility*, pages 185–194, 1993. Zurich, Switzerland.

[4] W. John and J. Hoener. Transient Analysis of Lossy Transmission Lines Systems with Respect to Reflection and Crosstalk Effects. *EMC–Zurich*, 1989. Zurich, Switzerland.

[5] F. Th. Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays–Trees–Hypercubes.* Morgan Kaufmann Publishers, San Mateo, California, 1992.

[6] S. Lin and E.S. Kuh. Transient Simulation of Lossy Interconnects. *DAC'92: Design Automation Conference*, pages 81–86, 1992.

[7] J. Müller and E. Griese. An Approach for a fast Preanalysis of Reflection Effects on Printed Circuit Boards. *EURO Electromagnetics : International Symposium on Electromagnetics*, 1994. Book of abstracts, Bordeaux 30.5. – 4.6.

[8] O. Rethmeier. Leistungsvergleich FREACS – Spice. *Internal Cadlab Report*, 1994.

[9] D. Wagenblaßt, J. Müller, W. John, and E. Griese. An Approach of Rule Develompent for Reflection and Crosstalk Effects on Printed Circuit Boards. *Proceedings of 12th International Wroclar Symposium on Electromagnetic Compatibility*, 1994. Wroclar, Poland, 28.6.–1.7., to appear.