Incremental Capacitance Extraction and its Application to Iterative Timing-Driven Detailed Routing

Yanhong Yuan Prithviraj Banerjee

Center for Parallel and Distributed Computing
Department of Electrical and Computer Engineering
Northwestern University, Evanston, IL 60208
{yuanyh, banerjee}@ece.nwu.edu

ABSTRACT

In this paper, we consider delay optimization in multilayer detailed routing. Given a detailed routing by some detailed router, we iteratively improve the delays of critical nets or nets with timing violation in an aggressive way. A rip-up and reroute approach is employed to generate the alternate routes. The optimal route which satisfies the timing constraints is chosen. The net delays are calculated using the Elmore delay model. The coupling capacitances are computed using a 3D extractor by applying the process parameters. In such an approach, the interconnect delay can be most accurately modeled. However, this aggressive approach is likely to be computationally unaffordable, because the delay of each alternate route should be evaluated. To overcome this problem, the key is to efficiently handle the detailed 3-D extraction in each iteration step. In our approach, we represent the design changes in an incremental way. Then we compute the coupling capacitance using the incremental extraction algorithm which is specifically tuned to an iterative design environment. This makes the aggressive improvement approach computationally practical. Experimental results are presented to demonstrate the efficiency of the approach.

1. INTRODUCTION

As delay migrates to interconnect, the possibility that excess capacitance in routing violates a *timing* constraint or *cross-talk* constraint increases. Since accurate estimation of delay and cross-talk requires detailed understanding of wire adjacency information, much work has been done on the delay and cross-talk optimization in detailed routing[1, 2, 3].

While detailed interconnect information is available in the detailed routing stage, most existing approaches have considered the coupling capacitance in a simplified way and used simple delay or crosstalk models. For example, it was assumed that crosstalk only exists between wires in adjacent rows or columns, and crosstalk between two routes is proportional to the coupling length l between them[1, 2]. The linear delay model is assumed in [3]; that is, the propagation delay along a wire of length l is $O(l)$. A more accurate Elmore Delay model was used to compute the source-to-sink delays in [4], but the resistance and capacitance of a routing segment were still assumed to be proportional to the length of the routing segment.

However, in ultra-deep submicron(UDSM) designs, capacitances can no longer be extracted in a 1-D, 2-D or 2.5-D context. With dense multi-level metal exceeding 6 layers in advanced processes, these components must be accurately extracted in a 3D context in order to perform accurate verification[5, 6]. Moreover, due to the rising contribution of coupling capacitance to total load capacitance, the Miller effect can have a large impact on actual delay times on a chip. K-factors determined by switching conditions of wires should be taken into account to accurately model the delays[5, 6]. Switching correlation was considered in [7] and coupling capacitance was taken into account by using lookup table technique, but they addressed the performance issues in the pre-detailed routing stage rather than in the post-detailed routing stage.

In this paper, we consider the problem of delay optimization in multilayer detailed routing. Given a detailed routing by some detailed router, by applying the process parameters and calculating the *actual* net delays, we use an *iterative strategy* to reduce the delays of critical nets or nets with timing violation. The motivation is to utilize the detailed 3-D wire adjacency information given in the detailed routing to most accurately consider the actual net delays for timing optimization, which is essential in the UDSM technology. In contrast to previous approaches addressing performance issues in detailed routing, our approach has the following features.

(1) An aggressive, *iterative improvement strategy* is used to reduce the net delays and find the optimal routing for the nets. A set of alternate routes for the net under improvement

are produced using a rip-up and reroute approach. The route which results in the minimum delay for the net and has minimum impacts on neighbors is selected.

(2) We use the Elmore delay model to compute the *actual* net delays. Instead of simply considering the wire length as the delay constraints, the coupling capacitances are calculated using a *3-D field solution*, aiming to obtain the accurate delays for the nets. To account for the Miller effect between adjacent wires, we incorporate the K-factors into the Elmore delay model, so that we can consider both the best-case and worst-case net delays.

Because detailed coupling capacitances should be extracted each time the design is changed, even with the fastest 3-D extractors, such an aggressive improvement strategy is likely to be computationally unaffordable. The *incremental extraction algorithm* presented in [12] was shown to be an efficient method for accurate and fast computation of 3-D capacitances in an iterative design environment. Instead of starting over to compute the capacitance from the very beginning at each iterative step, the incremental algorithm can effectively re-utilize the computation results of a previous extraction step and rapidly re-compute the new parasitic parameters. The incremental algorithm is specifically tuned to be used in our iterative delay improvement approach. This makes the aggressive improvement strategy computationally practical. Its application is based on an incremental representation of the design changes.

The rest of the paper is organized as follows: Section 2 presents the problem background. We describe the iterative improvement approach in Section 3. We also discuss the application of the incremental extraction in this section. Experimental results are presented in Section 4, and we conclude our paper with Section 5.

2. BACKGROUND

2.1. The Elmore Delay

RC trees, such as the one shown in Figure 1, have been widely used for modeling the gate and interconnect circuits. The Elmore Delay[9] is a convenient metric for RC trees. To apply the Elmore Delay for an RC tree $T = (V, E)$, each edge $e \in E$ is associated with a resistance and each node $v \in V$ is associated with a capacitance, denoted by c_v. Since there is only one unique edge between a node v and its parent node, we thus can denote such an edge as e_v. Let T_v denote the subtree rooted at node v. Let C_v be the total capacitance of subtree T_v, including c_v, and let r_v be the resistance of edge e_v. The Elmore Delay from a node u of T to a descendant node v of u, denoted by $d(u,v)$, is

$$d(u,v) = \sum_{u' \in path(u,v)} r_{u'} C_{u'} \qquad (1)$$

where $path(u,v)$ is the set of nodes along the unique path from u to v, excluding u.

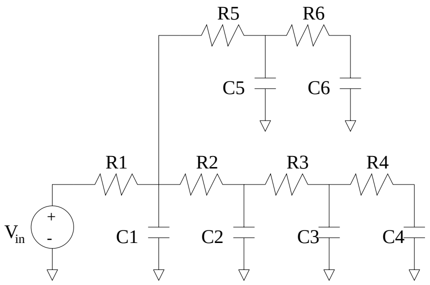


Figure 1. An RC tree.

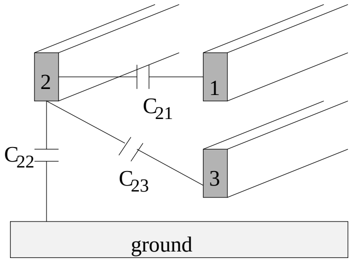


Figure 2. 3D coupling capacitance in multilayer metal

Due to the rising contribution of coupling capacitance to total capacitance, the Miller effect can have a large impact on actual delay times on a chip. Specifically, the actual delay associated with a net when switching from high-to-low depends on what is going on at its neighboring nets[5, 6].

For the example shown in Figure 2, if both neighboring nets 1 and 3 are stationary, then $C_2 = C_{22} + C_{21} + C_{23}$. However, if both are switching in the opposite direction to net 2, then $C_2 = C_{22} + 2C_{21} + 2C_{23}$. In general,

$$C_2 = C_{22} + K_{21}C_{21} + K_{23}C_{23} + \ldots + K_{2n}C_{2n} \qquad (2)$$

where the K-factors are determined by the switching conditions. Typically, $0 \leq K \leq 2$, but there are situations where $k > 2$.

Because in the RC tree, all the capacitances are grounded capacitances representing the parasitic interconnect effects, one approach to handling coupling effects is to ground all coupling capacitances after taking the K-factors into account.

2.2. Delay Minimization

The maximum tolerable delay of a net k is denoted by D^k_{max}. It is a given constant derived by the designer according to the electrical property of the circuit components. The actual delay of a net k is computed using the Elmore delay model and is denoted by D^k. We define the *delay slack* of net k as S^k, where $S^k = D^k_{max} - D^k$. If $S^k < 0$, net k is said to be a net violating the timing constraints. The minimum value of the *delay slacks* among all nets is called the *minimum delay slack*, denoted by S_{min}. In order to satisfy all the timing constraints, it is necessary that $S_{min} \geq 0$. Moreover, S_{min} measures the safety margin of the delays in the nets, and

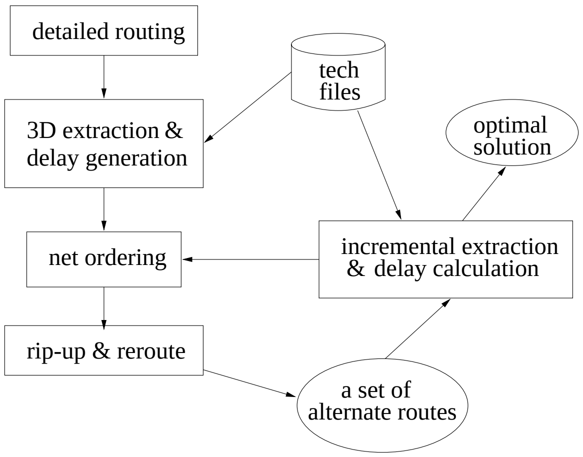

detailed routing
tech files
3D extraction & delay generation
optimal solution
net ordering
incremental extraction & delay calculation
rip-up & reroute
a set of alternate routes

Figure 3. The iterative improvement strategy.

a good routing algorithm should generate routing solutions with S_{min} as large as possible.

2.3. Capacitance Computation

We use the direct BEM method[11] to compute the capacitance. The following integral equation can be obtained using the BEM method:

$$c_s u_s + \int_\Gamma q^* u \delta\Gamma = \int_\Gamma u^* q \delta\Gamma, \qquad (3)$$

where u_s is the potential at a source point $S(x_s, y_s, z_s)$, and c_s is a constant depending on the geometry of the boundary Γ near the point S. In a 3-D problem, the fundamental solution u^* is $\frac{1}{4\pi r}$. r is the distance between the source point S and the field point (x, y, z).

Only the boundary Γ need to be discretized. By applying interpolation and numerical integration on the boundary elements, a linear equation system can be obtained as: $Ax = f$, where the column vector x has n unknown variables. Solving this linear system, we can *directly get the normal electric field q* on the Dirichlet boundary, and finally, we can calculate the capacitance(C) as: $C = \int_{\Gamma_u} \varepsilon q \delta\Gamma$, where ε is the permittivity.

3. THE ITERATIVE APPROACH

3.1. Problem Formulation

Given a multilayer detailed routing, our objective is to find a routing solution for the nets such that the timing constraints are satisfied, and their delays are reduced as much as possible. We define the problem as follows.

Problem 1. Timing-Driven Multilayer Detailed Routing *Given a multilayer detailed routing solution S, n nets and their timing constraints, find an optimal routing solution S' such that the delay slack $S^k \geq 0$, and S^k is maximized for all $1 \leq k \leq n$.*

To find the optimal solution for all nets simultaneously is hard. Therefore, we use an iterative improvement strategy which is shown in Figure 3. It is a sequential approach, that is, the optimization is done net by net. The approach consists of three steps. The first step orders the nets according to their *actual* delays. In the second step, the nets with timing violations are ripped-up and rerouted in the order decided in the first step. The third step evaluate the delays of the new routes and the optimal solution is selected. They are formulated as the following problems.

Problem 2. Net Ordering *Given the detailed routing topology and the process parameters, calculate the actual net delay slack S^k for all $1 \leq k \leq n$. Sort the delay slacks $S^1, S^2, \ldots, S^n$ in increasing order, i.e., Π:($S^{\pi_1}, S^{\pi_2}, \ldots, S^{\pi_n}$), such that $S^{\pi_i} \leq S^{\pi_j}$, if $i \leq j$.*

The delays are calculated using the Elmore delay model. The capacitances are computed using a 3-D field solution. K-factors are taken into account for worst case delay estimation. In this way, we are able to model the actual net delays most accurately.

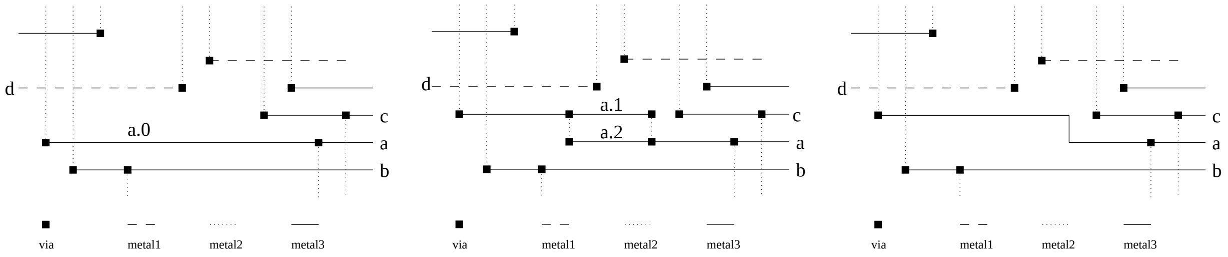
Nets with timing violations are ripped up and rerouted in the order determined in the net ordering step. The net with the largest delay violation is processed first. When a net is rerouted, besides the coupling capacitances changes on the current net, the coupling capacitances of its neighboring nets will also be affected. Therefore, we need to solve the following problem.

Problem 3. Rip-up and Reroute for Global Delay Minimization *Given a detailed routing solution S for nets 1, 2, ..., m-1, with $S^k \geq 0$ for all $1 \leq k \leq m-1$, find a solution for nets 1, 2,..., m, such that routing topologies of nets 1, 2, ..., m-1 are maintained, $S^k \geq 0$, and S^k is maximized for all $1 \leq k \leq m$.*

In order to take into account the neighboring nets, we keep track of the neighboring nets of the current net under delay improvement, and compute their delays as well. The solution which satisfies the timing constraints of the neighboring nets is selected. In fact, in a 3-D field solution, all coupling capacitances are computed in the context of its neighborhood, therefore all the coupling capacitances among them can be computed in the same pass of the field solution.

Remark 1. Due to the shielding effect, it is only necessary to consider the neighbors of a net within certain distance, depending on the surrounding net density.

Generally, in the multilayer routing, there may be a *set* of alternate routes for the ripped-up net. We prefer to consider all these alternate routes in a global way. By having a set of possible routes for the net in mind, we can determine which route has lower delay globally during the optimization. In order to select the route which results in the minimum net



(a) Initial routing (b) Two alternate routes for net *a* (c) Another route for net *a*

Figure 4. An example of the iterative delay improvement.

delay for the net, detailed coupling capacitances should be extracted and the delay calculated for each alternate route, within the context of its corresponding neighborhood. We need to efficiently solve the following problem.

Problem 4. Delay Evaluation of Alternate Routes *Given a set of p alternate routes for net k, denoted by routes k.1, k.2, ..., k.p, rapidly compute the actual delays of these alternate routes within its corresponding 3-D context.*

With the existance of many alternate routes, the delay computation may be very timing consuming. The key in our approach is to consider the alternate routes in an *incremental* way, and rapidly compute the coupling capacitances and delays in an incremental manner. We illustrate the incremental approach using Figure 4.

Figure 4 shows a 3 metal layer routing example. An initial routing is given in (a), and net *a* is under delay improvement consideration. We denote the initial route of net *a* as *a.0*. Two alternate routes for net *a*, routes *a.1* and *a.2* are shown in (b) where some segment of route *a.0* is moved away from the route of net *b* for potential coupling capacitance decrease between the parallel segments of nets *a* and *b*.

We need to calculate the delay for each of the alternate routes of the net *a*. Instead of considering them independently, we consider them in an incremental way. Route *a.1* can be considered as an incremental design resulting from changing some portion of route *a.0*. The coupling capacitances of the new route *a.1* can be rapidly computed based on the previous extraction results of route *a.0*. This is very efficient compared to computing the new values from the very beginning. Similarly, route *a.2* can be considered as an incremental design of route *a.1*, and its coupling capacitances can be incrementally computed from the results of route *a.1*. The incremental capacitance extraction is discussed in the next section.

Remark 2. Some heuristics can be employed to speed up the optimization. For example, by considering the relative position of the net to its neighboring nets, we can remove some large-delay redundant routes early before the detailed extraction. In Figure 4(b), route *a.2* can be removed easily because it has a longer segment in parallel to the route of net *b*, which suggests a larger delay than that of route *a.1*.

Remark 3. In the reserved layer routing model, vias may be introduced in the rerouting. As it is known, vias will introduce additional capacitance and resistance. While, in the multilayer design, the delay decrease by removing large parallel coupling capacitances can more than offset the delay increase due to the introducing of vias. If unreserved model is used, no additional vias need to be introduced. Figure 4(c) shows an alternate route of net *a* under unreserved layer model.

3.2. Incremental Representation

As discussed above, the rerouting of a net can be considered as an incremental design of the previous route. In our approach, the incremental design change is represented by the following basic operations:

MOVE: Move a conductor;
ADD: Insert a conductor;
DELETE: Delete a conductor;
STRETCH: Resize a conductor;

A design change can be an arbitrary combination of these basic operations. By representing the design changes in an incremental sequence, we are able to utilize the incremental algorithm for capacitance extraction.

3.3. Incremental Extraction

The 3-D extractors compute the capacitance based on a given 3-D structure[10]. First, a discretization of the problem is computed. Then, a linear system of equations are generated. Finally, the linear system is solved for the capacitance and resistance. Usually, an iterative solution method is employed, such as GMRES. When the structure is changed, traditional extractors will have to treat the changed structure as a totally new one and start every thing from beginning.

The incremental extraction recomputes the capacitances in a different way[12]. We describe two main steps of the algorithm: *incremental system re-configuration* and *incremental re-solution*.

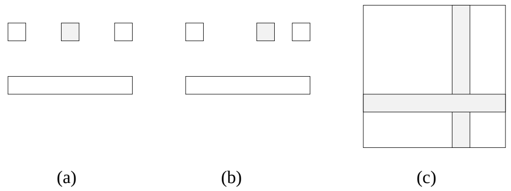


Figure 5. An incremental MOVE operation: (a) The original structure(cross section); (b) The changed structure after moving the shaded conductor(cross section); (c) The part of the linear system to be recalculated(shaded area).

Incremental Re-configuration When the structure is changed, the linear system will also change. One key idea in the incremental algorithm is to *locally* modify only part of the linear system according to the design change. When a small change is made to the structure, the new system reflecting the new design can be re-configured rapidly by performing some local re-calculation of the system. Figure 5 illustrates a re-configuration when a conductor is moved. To reduce the size of the problem and improve the accuracy, an *adaptive mesh* is employed with finer discretization at or near edges, contacts and corners.

Incremental Solution To avoid the $O(n^3)$ cost of solving the linear system with Gaussian elimination, a conjugate residual like algorithm GMRES[13] is commonly used. An initial guess x_0 to the linear system is required. An all-zero vector is usually used in conventional linear system solvers. In an incremental approach, because only a small portion of the problem is changed, it is reasonable to assume that the new solution will not change significantly compared to the previous solution. Consequently, in the incremental algorithm, instead of using an all-zero vector as x_0 each time, the solution of the previous extraction is used as x_0 for the modified linear system. This approach results in significantly fast convergence to the final solution in much less number of iteration steps.

Let A_0 be the initial linear system, and A be the new system after the design change. It is unnecessary to explicitly store both A_0 and A. Instead, A is stored in an incremental way, that is, $A = A_0 + \Delta A$, where ΔA represents the modified part. Therefore, the incremental algorithm is memory efficient.

4. EXPERIMENTAL RESULTS

The iterative improvement approach was implemented in C. Experiments were conducted on an HP C-110 workstation with a 120MHz PA-7200 CPU and 128MB memory and using the HP-UX 10.2 operating system.

A set of nets from the triple-layer OTC routing of Deutsch's difficult problem in [14] were randomly selected for delay improvement. In order to compute the actual net delays, we employ the process parameters for interconnects predicted in [6]. Some of the parameters used in our experiments are shown in Table 1 and Table 2.

Table 1 lists the dimensions for the lower 2 levels of metal for each process. Copper is used for all sheet resistance calculations, with a resistance of $2.2\mu\Omega - cm$. It was assumed that the aspect ratio (height/width) $AR = 2$ for all the processes. Table 2 shows projections for global interconnect throughout scaling. For intermediate metal layers not included in the tables, it was assumed that the minimum pitches and thickness are between those presented in the tables. For instance, metals 3 and 4 have a minimum pitch that is approximately double that of metals 1 and 2.

Table 1. Interconnect Characteristics for metals 1 and 2.

Process(μm)	0.18	0.1	0.05
Height(μm)	0.46	0.26	0.14
Width/Space(μm)	0.23	0.13	0.07
Sheet Resistance($\Omega/\Box$)	0.048	0.085	0.16
T_{ins}(μm)	0.5	0.32	0.21
Dielectric Constant	2.7	2.0	1.5

Table 2. Top metal layer parameters for all generations.

Height (μm)	Width/Space (μm)	Sheet Resistance ($\Omega/\Box$)	T_{ins} (μm)
2.5	2.0	0.009	1.4

Table 3 summarizes the results of the delay improvement using the aggressive iterative improvement approach. The resistances are computed using the sheet resistance given in Table 1. We assume the wire thickness, width and spacing for metal 3 are double that of metals 1 and 2. K-factors are set to 2 for all the coupling capacitances so we are considering the worst-case delay of the nets. Because no delay constraints are given in [14], the maximum tolerable delays D^k_{max} are assigned randomly, and we are more interested in the improvement of delays.

For the selected nets, we are able to improve the delays by 25% on average. It is interesting that the process parameters do not affect the delay improvement too much. The main reason for this is that, the wire thickness, width and space are scaled in approximately the same factor as the sheet resistance, as it can be seen from the interconnect parameter tables. We also find that the wire densities have large impacts on the performance of the iterative improvement approach. The delay of net 6 is improved by a large margin. This is because the wire density around that net is quite low, and there is much freedom for improvement. For a two-layer routing, the resulting layout is usually quite

Table 3. Delay improvement from iterative improvement strategy.

process	0.18μm			0.1μm			0.05μm		
	initial	improved	% timing	initial	improved	%timing	initial	improved	% timing
net	delay(ps)	delay(ps)	improved	delay(ps)	delay(ps)	improved	delay(ps)	delay(ps)	improved
net 1	1.12	0.91	18	1.02	0.83	18	1.04	0.84	19
net 2	3.50	2.65	24	3.42	2.56	25	3.38	2.50	26
net 3	1.46	1.15	22	1.32	1.03	22	1.27	0.97	23
net 4	2.57	2.06	20	2.33	1.86	20	2.26	1.78	21
net 5	0.69	0.56	18	0.63	0.51	19	0.60	0.48	19
net 6	0.88	0.53	40	0.80	0.47	41	0.76	0.44	41

compact, leaving little space for iterative improvement. For a multilayer routing, in general, there is much freedom for rerouting a net for performance improvement.

Table 4. Speedup from incremental computation.

net	w/o incr	w/ incr	speedup
net 1	190	16	12
net 2	268	17	16
net 3	145	12	12
net 4	212	19	11
net 5	55	7	8
net 6	69	3	22

The incremental extraction has made the iterative improvement approach computationally practical. Table 4 presents the running times (seconds) with full extraction versus incremental extraction. Compared to the computation without incremental features, an order of magnitude speed up has been archived by using the incremental approach.

5. CONCLUSION

In this paper, we presented an iterative improvement approach for delay optimization in multilayer detailed routing. The coupling capacitances of rerouted nets are computed using an incremental extraction algorithm. The actual delays of candidate routes are calculated using the *Elmore delay model*. The approach can also be applied to the iterative improvement of the crosstalks in the UDSM designs.

REFERENCES

[1] T.Gao and C.L.Liu, "Minimum Crosstalk Channel Routing," *ICCAD*, 1993.

[2] H.Zhou and D.F.Wong, "An Optimal Algorithm for River Routing with Crosstalk Constraints," *ICCAD*, 1996.

[3] S.Natarajan, N.Sherwani, N.Holmes, and M.Sarrafzadeh, "Over-the-cell Routing for High Performance Circuits," *DAC*, 1996.

[4] K.Zhu and D.F.Wong, "Switch Bound Allocation for Maximizing Routability in Timing Driven Routing of FPGA's," *IEEE Trans. CAD*, 1998, pp316-323.

[5] R.Saleh, D.Overhauser and S.Taylor, "Full-Chip Verification of UDSM Designs," *ICCAD*, 1998, pp453-460.

[6] D. Sylvester and K.Keutzer, "Getting to the Bottom of Deep Submicron," *ICCAD*, pp.203-211, 1998.

[7] H.P.Tseng, L.Scheffer and C.Sechen, "Timing and Crosstalk Driven Area Routing," *DAC*, 1998.

[8] P.Penfield and J.Rubinstein, "Signal Delay in RC Tree Network," *DAC*, 1981, pp.613-617.

[9] W.C.Elmore, "The Transient Response of Damped Linear Networks with Particular Regard to Wide Band Amplifiers," *J. Appl. Phys,.*, vol.19, pp.55-63, 1948.

[10] K. Nabors and J. White, "FASTCAP: A Multipole-Accelerated 3-D Capacitance Extraction Program," *IEEE Trans. CAD*, pp. 1447-1459, Nov. 1991.

[11] C. A. Brebbia, *The Boundary Element Method for Engineers*, London : Pentech Press, 1978.

[12] Y.Yuan and P.Banerjee, " ICE: Incremental 3-Dimensional Capacitance and Resistance Extraction for an Iterative Design Environment," *GLSV*, 1999.

[13] Y.Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, 1996.

[14] J.Kim and S.M.Kang, "A New Triple-Layer OTC Channel Router," *IEEE Trans. CAD*, vol.15, no.9, pp.1059-1070, Sept. 1996.