A Spacing Algorithm for Performance Enhancement and Cross-talk Reduction

Kamal Chaudhary Akira Onozawa* Ernest S. Kuh

Electronics Research Laboratory,
University of California, Berkeley.

*NTT LSI Laboratories,
Atsugi, Kanagawa, Japan.

Abstract

With the shrinking of feature size on silicon the coupled capacitance between adjacent wires is contributing a significant factor to the interconnect delay, which already dominates the circuit performance. In the near future coupled capacitance could contribute as much as 50-75% to the interconnect delay which has been largely ignored by performance oriented layout tools. Given a routed design, this work minimizes delay and the peak cross-talk by readjusting the space between interconnects. It can reduce the circuit delay significantly and in addition reduce the peak cross-talk problem. An efficient technique based on network simplex algorithm is used to solve the problem in the paradigm of compaction.

1 Introduction

As the performance requirement for the electronic system design is getting severer, LSI designers require advanced CAD tools to enhance the circuit performance at all the levels of the design. One of the important issues for the high performance LSI design is to reduce the interconnect delay in the circuits. Until recently the delay in the transistors had dominated the circuit performance, and the interconnect delay has not been so significant. However, as the device technology is improving the interconnect delays are increasingly dominating [1] the performance.

The interconnect delay is caused due to capacitance and resistance of wire segments. The capacitance for the wire can be classified into two groups: ground and coupled. It must be recognized that the coupled capacitance for a pair of unit length metal wires will be much larger than the ground capacitance in the future technology [1], as both the width of the wire and the space between wires is reduced.

A number of papers have been published for performance optimization during the physical design phase[2, 3]. Most of these papers propose placement and routing algorithms, essentially aiming to minimize the interconnection length on the critical path. The delay model for interconnection is assumed to be a functions of only the net length. This means that these algorithms only handle the ground capacitance directly, not the coupled capacitance which requires analysis of the neighboring wires. Before routing it is difficult to consider the effect of coupled capacitance accurately.

Our primary concern is to minimize interconnection delay for a routed layout of cell based circuit, using a compaction algorithm while considering both the coupled and ground capacitance for the wire segments on the critical path. Though a number of papers have been published for the compaction algorithms[4], they have largely ignored the performance issues so far. The issue has been indirectly and inadequately addressed by trying to minimize the total wire length [5].

Our secondary concern is the cross talk problem. The interconnect cross talk will be a critical problem as the coupled capacitance increases[1].

In the rest of the paper, we use the word *spacing* instead of *compaction* since we not only compact but expand the layout sometimes to have the optimal electrical performance.

1.1 Interconnect Capacitance Trend

In the single layer environment, the interconnection capacitance can be divided into three components: ground, fringing and coupled (wire-to-wire) capacitance. The fringing capacitance can be included into the ground capacitance. The ground capacitance per unit length of interconnection depends on the width of the wire w and the distance from the wire to the substrate s. The coupled capacitance, however, depends on metal's height h and the distance between two neighboring wires d. In case of multi layers, capacitance between two perpendicular running metal wires on different layers, can be considered as a ground capacitance, and that between two parallel metal wires as coupled capacitance.

The ground capacitance will decrease in the future with the reduction of width of the interconnection [1]. On the other hand, the coupled capacitance will increase as the space between the interconnections is being reduced and the height of the interconnections is kept constant to avoid increasing the interconnect resistance.

Figure 1, cited from [1], shows the trend curves of the interconnection capacitance. The main stream fabrication technology in the next decade will be $0.2\mu m$ CMOS technology. It is estimated from current status of CMOS that the coupled capacitance will become two or three times larger

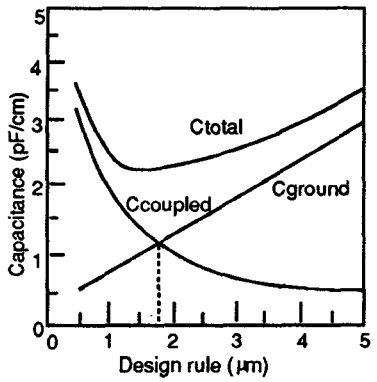

Figure 1: Interconnection capacitance trend

than the ground capacitance[6]. Clearly, the coupled capacitance will play an important role in determining the circuit delay in the future. Thus, it is inevitable to develop CAD tools for reduction of the coupled capacitance. The coupled capacitance besides increasing delay also creates the cross talk problem between interconnections.

1.2 Overview

Given a routed layout, we consider a layout problem to optimize the circuit performance by changing only the spacing between the wires. As described in the previous sections increase in space between the wires will reduce the coupled capacitance thereby improving the performance. Increased spacing will increase the area and also the wirelength in the orthogonal direction. One can reduce the area penalty by exploiting the fact that only some portion of the layout is spatially critical, which determines the overall area. By distributing the space among wire segments carefully and considering the effect on vertical wire length increase one can improve the performance.

As the traditional compaction problem in general deals with the adjacency information of each interconnection segments, we think that the problem of coupled capacitance optimization can be best described in the paradigm of spacing, which is more general concept than compaction. The two-dimensional spacing is too time-consuming to take the timing issues into account, thus we restrict ourselves to the one-dimensional spacing.

Our first objective is to reduce the delay in the given routed circuit. In the proposed algorithm, PERFECT, a set of critical paths is extracted from the net list by performing a static timing analysis of the layout. The timing constraints on the critical paths are handled using Lagrangian relaxation technique. Here, the concept of the repulsive constraint proposed in [7] is used for the local increase of space. In addition to the coupled capacitance minimization, the ground capacitance is also reduced at the same time. The algorithm is based on the network simplex method, so the entire spacing process can be handled within a constraint graph.

Our secondary objective is to reduce the peak cross talk in the circuit. The cross talk is caused by the proximation of the long interconnects. We reduce the maximum peak cross talk in the chip by re-adjusting the space between the interconnects. Again, the repulsive constraints are used. We have established a table look-up method to estimate the cross talk rapidly in a routed circuit. We analyze the layout and select the nets whose cross talk exceeds the given threshold for applying repulsive constraints.

The rest of the paper is organized as follows. The timing model is described in Section 2, followed by the problem formulation in Section 3. Spacing algorithm based on network simplex algorithm is presented in Section 4. Section 5 discusses cross talk estimation and an algorithm for reduction. Section 6 presents experimental results followed by conclusions and future work in Section 7.

2 Timing Model

Delay in an integrated circuit may be viewed as consisting of two components: cell/module delay and net delay. Module delays are taken from the library and net delays are computed as described below. Timing analysis is computed by a block-oriented search [8].

Bakoglu in [1] has presented an interconnect delay model that is the basis for the model chosen in this work. Let R_g^{rise} and R_g^{fall} represent the resistance of the driving cell during a rising and falling output. The rising and falling waveform delays of the output net of G_g can be approximated by:

$$\begin{aligned} d_g^{rise} &= R_g^{rise}(C_{net} + C_{load}) + 0.5R_w(C_{net} + C_{load}) \\ d_g^{fall} &= R_g^{fall}(C_{net} + C_{load}) + 0.5R_w(C_{net} + C_{load}) \end{aligned} \quad (1)$$

C_{net} is the interconnect capacitance of the output net of G_g and C_{load} is the capacitance of the driven pins. R_w is the lumped interconnect resistance from the source to a sink. The factor of 0.5 which multiplies R_w is based on the analysis of [1] to model distributed RC delay.

Net capacitance C_{net} consists of two components: capacitance to ground C_{net}^{gnd} and coupled capacitance $C_{net}^{coupled}$. C_{net}^{gnd} is proportional to the routed net length. The $C_{net}^{coupled}$ however is a function of distance between the the adjacent wire segments which can be changed during compaction. The relation for coupled capacitance between two segments $s1, s2$ on layer $m1$ and $m2$ respectively is given by:

$$c_{s1s2}^{cou} = l_{s1s2} * K_{m1m2} * (1/d_{s1s2})^{1.34} [9] \quad (2)$$

where l_{s1s2} is the coupled length between the segments, d_{s1s2} is the distance between the segments and K_{m1m2} is the technology dependent constant. Using this eqn $C_{net}^{coupled}$ can be written as:

$$C_{net}^{coupled} = \sum_{s_i \in net} \sum_{s_j \notin net} c_{s_i s_j}^{cou} \quad (3)$$

3 Problem Formulation

In this section, we restrict ourselves to the vertical spacing. The horizontal spacing can be performed similarly.

Let $H = \{h_i\}$ be a set of the horizontal interconnection segments and let $h_t, h_b \in H$ be two virtual segments put at

the top and bottom of the layout respectively to bound the layout vertically. The maximum[minimum] x-coordinates of those segments are defined to be larger[smaller] than the x-coordinates of the other segments.

Let y_i be the y-coordinate of the segment h_i. The constraint induced by the process technology between two segments h_i and h_j is written as

$$y_i - y_j \geq R_{ji}, \tag{4}$$

where R_{ji} is the constraint value, usually the design rule value such as spacing or margin. Many types of constraints can be expressed by inequality (4). The upper-bound constraint is expressed with negative R_{ji}. The equality constraint $y_i - y_j = R_{ji}$ is replaced by two inequalities $y_i - y_j \geq R_{ji}$ and $y_j - y_i \geq -R_{ji}$. The set of ordered segment-pairs (h_j, h_i) with the constraints of inequality (4) is denoted by F.

In addition to the design rule constraints there are timing constraints on the critical paths. The timing constraints are too complex to be handled directly and we use the Lagrangian Relaxation technique[2] to them into the cost function.

With each critical path π we associate a Lagrangian multiplier λ_π. To begin with λ_π is set to be 0. We update λ_π according to the following equation:

$$\lambda_\pi = max(0, \lambda_\pi^{old} + t * (delay_\pi - cycle\ time)) \tag{5}$$

where $delay_\pi$ is the total delay on path π, cycle time is the specified cycle time and t is the damping factor, which is initially set to 1 and then reduced slowly. From the path Lagrangian multipliers we derive Lagrangian multipliers for each cell n_i as follows:

$$\lambda_{n_i} = \sum_{n_i \in \pi_j} \lambda_{\pi_j} \tag{6}$$

The spacing problem with the Lagrangian Relaxation can be defined as below:

Problem 3.1 *Vertical spacing problem*

Objective function

$$max\lambda\ min \sum_{\pi \in C} \lambda_\pi * \sum_{g \in \pi} (d_g^{intrinsic} + max(d_g^{rise}, d_g^{fall})) \tag{7}$$

Constraints

$$y_i - y_j \geq R_{ji}\ (h_j, h_i) \in F \tag{8}$$
$$y_t - y_b \leq specified\_height\_limit \tag{9}$$

Here C is a set of the critical paths and $d_g^{intrinsic}$, d_g^{rise} and d_g^{fall} are the intrinsic delay of a cell g, rise and fall delay of the net driven by g, respectively (See eq.(1)).

3.1 Linear Programming Formulation of the Problem

The non-linear objective function makes it difficult to develop an efficient algorithm to solve Problem 3.1 directly. Our basic idea is to solve it as an approximate linear programming problem for a fixed set of λ and then update λ based on the current solution. We transform the problem to one in which the linear term (ground capacitance delay) is the objective function and the non-linear term (coupled capacitance delay) is handled as a repulsive constraint[7]. The idea is to introduce a 'repulsive force' between two coupled critical segments. The repulsive constraint is defined based on the concept of the degree of proximity.

Let us define the degree of proximity $f(y)$ in our case as follows:

$$f(y) = \min_{(h_l, h_k) \in P} (y_k - y_l)/\alpha_{lk}, \tag{10}$$
$$\alpha_{lk} = \lambda_g \cdot D_2 \cdot l_{kl}/\delta_{lk}^{2.34}, \tag{11}$$

where δ_{lk} is the initial space between *critical segment pair* $(h_l, h_k) \in P$, D_2 is the technology dependent coupling capacitance constant, and α_{lk} is the sensitivity of the overall circuit delay to the δ_{lk}. The constraint $f(y) \rightarrow max$ is called repulsive constraint. Maximization of $f(y)$ ensures that the minimum weighted space between h_i and h_j expands as far as possible reducing the coupled capacitance.

For a fixed set of λ the spacing problem 3.1 can be approximately transformed in the following parametric linear programming problem with parameter z.

Problem 3.2 *LP Formulation of Problem 3.1*

Objective function

$$d_G = \sum (\lambda_g \cdot n_{ij} \cdot D_1 \cdot (y_i - y_j)). \rightarrow min. \tag{12}$$

Constraints

$$y_i - y_j \geq R_{ji}\ ((h_j, h_i) \in F), \tag{13}$$
$$y_b - y_t \geq -specified\_height\_limit, \tag{14}$$
$$y_k - y_l \geq \alpha_{lk} \cdot z\ ((h_l, h_k) \in P), \tag{15}$$

Parameter

$$z(\rightarrow max). \tag{16}$$

where, D_1 is the technology dependent ground capacitance constant and d_G represents the delay due to the ground capacitance.

We could solve Problem 3.2 by a conventional method based on the simplex algorithm[10]. However, since the pivoting operation in a large sparse matrix is inefficient, we use the algorithm in [7] based on the concept of *network simplex method.*

4 Spacing Algorithm

4.1 Constraint Graph

The first step of the spacing algorithm is creation of the constraint graph. Let us denote the constraint graph by

$G(V, E)$, where V is the set of vertices and E is the set of edges. A vertex v_i corresponds to a horizontal segment h_i in the layout and there is an directed edge (v_j, v_i) if the segment h_i is above segment h_j and is visible from it.

Let us denote the set of the segments pairs (h_j, h_i) having the term $(y_i - y_j)$ in the objective function(12) by Q. We refer a critical path on the constraint graph G as a *spatial critical path* to differentiate it from the critical path in the timing domain.

We assign two weights $w_1(e)$, $w_2(e)$ and cost $c(e)$ for each edge $e \in E$. The weight $w_1(e)$ is set to R_{ji} for $e \in F$, 0 for the others. and the weight $w_2(e)$ is set to α_{ji} for $e \in P$, 0 for the others. The total weight for an edge e is expressed by $w_1(e) + w_2(e) \cdot z$. The edge weight increases as z is increased by the algorithm. The cost $c(e)$ is set to $\lambda_g \cdot n_{ij} \cdot D1$ for $e \in Q$, 0 for the others.

4.2 Graph Based Algorithm to Resolve Repulsive Constraints

In [7], a network simplex algorithm which solves a class of problems like Problem 3.2 was proposed and the algorithm is adapted to our case. Here we directly present the algorithm, for details see[7].

Algorithm 4.1 *(An Algorithm for Solving Problem3.2[7])*

1. *Let $z = 0$. Construct a tree T of G corresponding to a minimum wire length layout (initial fesaible solution) by [5]. Let d_G be the value of the objective function and $C_1(e_i, T)$ $[C_2(e_i, T)]$, and $D(f, T)$ be the sum of edge weights $w_1(e)[w_2(e)]$ and edge costs in the fundamental circuit and cutset w.r.t. a cotree edge e_i and a tree edge f respectively.*

2. *Let $I = \{e | C_2(e, T) > 0, e \in E - T\}$.*

3. *Find δz and an edge $r \in I$ which satisfy*

$$\delta z = \min_{e \in I}(-C_1(e, T)/C_2(e, T)) = -C_1(r, T)/C_2(r, T)). \quad (17)$$

4. *Let*

$$w_1(e) \leftarrow w_1(e) + \delta z \cdot w_2(e) \ (for\ all\ e \in E) \quad (18)$$

$$z \leftarrow z + \delta z, \quad (19)$$

$$d_G \leftarrow d_G + \delta z \cdot \sum_{e \in Q} c(e) \cdot C_2(e, T). \quad (20)$$

5. *Let $J = \{e \in T | e$ is a tree edge which is included in the fundamental circuit defined by co-tree edge r and has opposite direction in this circuit$\}$. If $J = \phi$ then stop.*

6. *Find an edge $s \in J$ satisfying $\min_{e \in J} D(e, T) = D(s, T)$.*

7. *Let $T \leftarrow (T - \{s\}) \cup \{r\}$. Go to step2.*

Starting with a minimum wire length solution at $z = 0$, Algorithm4.1 increases z by an amount such that a new spatial path becomes critical. Then, the weight $w_1(e)$ is increased by $\alpha_{ji} \cdot z$ to expand the space between the critical horizontal segments. This process is iterated until no further increase of z is possible.

Each step in Algorithm4.1 except step1 takes at most $O(|E|)$, and it can be shown that we can initialize $C_1(e, T)$, $C_2(e, T)$ and $D(e, T)$ for all e with a T in $O(|E|)$. Note that it is only necessary to re-calculate the fundamental circuit value for the co-tree edges in the fundamental cutset defined by s in each iteration, and the recalculation can be done in constant time for each edges in the cutset. Thus, the complexity for Algorithm4.1 is at most $O(|E| \cdot N_I)$, where N_I is the number of iterations. $N_I < |V|$ in all of our experimental circuits.

4.3 A Performance Driven Spacing Algorithm

The overview of the algorithm is as follows.

Algorithm 4.2

```
construct_vertical_constraint_graph;
do {
    timing_analysis ;
    update_lagrangian_multipliers;
    set_repulsive_constraints ;
   while(there exist repulsive constraints){
        vertical_spacing ;
     delete_critical_repulsive_constraints ;
    }
} while (improvements > e);
```

In `set_repulsive_constraints`, we set the repulsive constraints to the constraint graph G, using the result of `timing_analysis`. In the next step, we solve the Problem3.2 using the Algorithm 4.1. We set all α_{ji} on the spatial critical paths to 0 in `delete_critical_repulsive_constraints` and repeat Algorithm4.1. (See Figure 2) These procedures are iterated until there is no more repulsive constraints with positive repulsive coefficients. The same process is repeated in the horizontal direction.

Note that each step in Algorithm 4.2 except `vertical_spacing` takes $O(|E|)$. The overall complexity depends on the number of repulsive constraints and the value of `e`, given by designers.

5 Cross Talk Reduction by Spacing

In this section, we address the problem of reducing peak cross talk in the chip by using the same spacing algorithm.

5.1 Cross Talk Estimation

We need to estimate/compute the peak cross talk voltage for the given layout in order to reduce it. Current transmission line simulators for ex. spice etc. are too slow to simulate thousands of nets in the circuit, thus we need some fast way of computing the cross-talk. Number of papers have been published for the cross-talk computation[11, 12],

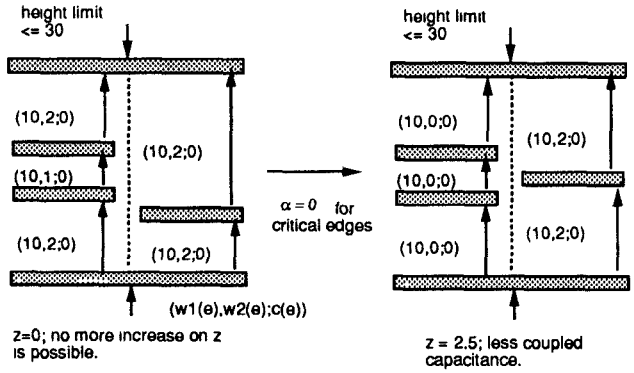

Figure 2: Deleting repulsive constraints for further optimization

however, they make simplifying assumptions about either losslessness of the interconnect and/or deal with the lines terminated by characteristic impedance. But the real interconnects are in general lossy, and terminated by various impedance.

Sakurai, et.al[12] discuss a model for the cross talk in RC interconnect with lossy interconnect and capacitive coupling. This model can be used within the CMOS chip. However the formula presented in there for cross talk computation gives an unusually big cross talk compared to our simulation results by SPICE and SWEC[13]. To have a realistic and accurate cross talk estimation, we construct an empirical model based on the simulation results using industrial parameters. We have done the simulation with the transmission line simulator SWEC[13] and derived a number of tables for peak cross talk voltage as a function of various circuit parameters. The device and transmission line parameters employed in the simulation are taken from the industrial process[6]. The cross talk estimation is done using the table look-up and interpolation.

The parameters used for the simulation are total quiet line length L_t, coupled length L_c, coupled locations x_l, x_r, line space s, and rise/fall time of drivers T_r, T_f.

During the spacing we compute the cross talk voltage for the rising signal according to the following equation.

$$V_c = F_r(T_r, L_t, s)L_c/L_t \qquad (21)$$

Here, the function F_r is computed by the interpolation of the 3-dimensional table obtained from SWEC simulation.

In general the segments composing a net forms a tree in the layout and the maximum possible cross talk for each fanout pins in this tree is required. We simply take a summation of the cross talk on each path from the output pin to each input pin.

5.2 Spacing Algorithm for Reduction of Maximum Cross talk

After routing we can adjust the spacing to reduce the cross-talk. Increased spacing can increase not only the area, but coupled lengths in the orthogonal direction as well. We have to carefully distribute the increased spacing among the segments so as to minimize the area increase by exploiting the non-critical spatial areas while not increasing the coupled lengths too much. This can be easily done by the approach described in previous sections. By setting repulsive constraints between the wire segments we can increase the spacing and by including the coupled length in the objective function we can reduce the coupled length.

The overview of the algorithm is as follows.

Algorithm 5.1 *Repeat the following procedure until no nets are selected at step 2.*

1. *Compute cross talk for each fanout pin on each net.*
2. *Select nets whose cross talk exceeds the threshold.*
3. *Set repulsive coefficient proportional to the cross talk excess.*
4. *Spacing with the repulsive constraints[7].*

6 Experimental Results

We have implemented PERFECT in C language on a DEC-Station 5000, and performed experiments on standard cell layouts. These examples are taken from the MCNC logic synthesis layout benchmarks. They were mapped using Berkeley tools MisII and placed and routed using OCT-TOOLS. In these experiments, we performed only vertical spacing.

Here, unit length capacitance, etc. for $0.2\mu m$CMOS were estimated[6] from scaling of $0.5\mu m$CMOS. The cell delays are also adjusted such that sum of the cell delay on a critical path is about 50% of its total delay, which is considered a realistic assumption. Table 1 shows the results comparing ordinary min area based spacing[14] with PERFECT. CPU times are in sec. Note that the tabulated circuit delay in ns is the worst path delay for the circuit including both cell and interconnect delay. The average improvement of the circuit delay for 0.5μ technology is about 5% and for 0.2μ technology is about 10%. The maximum allowable area was set to 15% in these experiments. The average area increase is 9%, but it can be reduced at the cost of a slightly smaller delay improvement. Figure 3 shows the area-delay trade off for Example C5315.

Table 3 shows result for cross-talk reduction. On average our algorithm reduces peak cross-talk by 15-20%. This reduction is achieved at about 8% increase in the area.

7 Conclusions and Future Work

Coupled capacitance is going to be an important limiting factor in achieving high performance circuits in near future. We have presented an effective algorithm to minimize the coupled capacitance to enhance the circuit performance. It can reduce the net delay by as much as 20%. In addition, the algorithm also reduces the cross-talk in the circuit. The cross talk problem is an inevitable problem in MCM design. Applying our algorithm to a routed MCM circuit is the topic of our future work.

Acknowledgements

This research was sponsored by the State of California MICRO Program, under Grant Nos. 93-085 and 93-086.

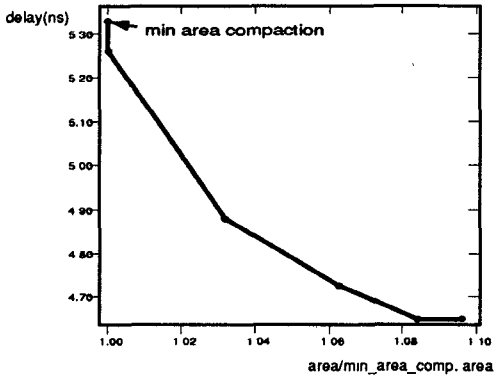

Figure 3: Area-Delay tradeoff for C5315

circuit	cells	segs	Min Area		PERFECT	
			ctalk	area	ctalk	area
0.5μ CMOS Technology						
C1355	241	1920	0.12	1.00	0.10	1.07
C2670	428	3988	0.31	1.00	0.24	1.10
C5315	1001	8808	0.35	1.00	0.28	1.08
C7552	1400	14670	0.50	1.00	0.42	1.09
0.2μ CMOS Technology						
C1355	241	1920	0.17	1.00	0.15	1.07
C2670	428	3988	0.59	1.00	0.47	1.09
C5315	1001	8808	0.68	1.00	0.54	1.06
C7552	1400	14670	0.85	1.00	0.79	1.10

Table 2: Cross Talk Reduction

circuit	cells	Min Area			PERFECT		
		delay	area	CPU	delay	area	CPU
0.5μ CMOS Technology							
C1355	241	6.53	1.00	2	6.40	1.08	18
C2670	428	9.41	1.00	6	8.90	1.09	107
C5315	1001	12.90	1.00	18	12.06	1.08	240
C7552	1400	33.12	1.00	36	31.40	1.08	822
0.2μ CMOS Technology							
C1355	241	2.55	1.00	2	2.45	1.08	18
C2670	428	3.85	1.00	6	3.48	1.09	109
C5315	1001	5.33	1.00	18	4.64	1.08	257
C7552	1400	13.84	1.00	36	12.30	1.10	787

Table 1: Experimental results

References

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

[2] A. Srinivasan, K. Chaudhary, and E.S. Kuh. RITUAL : A performance driven placement algorithm. *IEEE Trans. Circuits and Systems*, pages 825–839, November 1992.

[3] S. Prasitjutrakul and W. J. Kubitz. A performance-driven global router for custom VLSI chip design. *IEEE Trans. Computer-Aided Design*, 11(8):1044–1051, August 1992.

[4] D.G. Boyer. Symbolic compaction review. In *Proc. 25th Design Automation Conference*, pages 383–389, 1988.

[5] T. Yoshimura. A graph theoretical compaction algorithm. In *Proc. ISCAS 85*, pages 1455–1458, June 1985.

[6] K. Nishimura and Y. Ohtomo. Private communication. NTT LSI Laboratories, 1992.

[7] A. Onozawa. Layout compaction with attractive and repulsive constraints. In *Proc. 27th Design Automation Conference*, pages 369–374, 1990.

[8] R. B. Hitchcock, G. L. Smith, and D. D. Cheng. Timing analysis of computer hardware. *IBM Journal of Research and Development*, 26(1):100–105, 1983.

[9] T. Sakurai and K. Tamaru. Simple formulas for two- and three- dimensional capacitances. *IEEE Trans. on Electron Devices*, ED-30:183–185, Feb. 1983.

[10] K. Murty. *Linear Programming*. John Wiley and Sons, 1983.

[11] A. Feller, H.R. Kaupp, and J.J. Digiacomo. Crosstalk and reflections in high-speed digital systems. In *AFIPS proceedings*, volume 27, pages 511–525, 1965.

[12] T.Sakurai, S.Kobayashi, and M.Noda. Simple expressions for interconnection delay, coupling and crosstalk in VLSI's. In *Proc. ISCAS*, pages 2375–2378, 1991.

[13] S. Lin, M. Marek-Sadowska, and E. S. Kuh. Swec: A Step Wise Equivalent Conductance timing simulator for CMOS VLSI circuit. In *Proc. EDAC*, pages 142–148, 1991.

[14] Y.-z. Liao and C. K. Wong. An algorithm to compact a VLSI symbolic layout with mixed constraints. *IEEE Trans. Computer-Aided Design*, CAD-2(2):62–69, April 1983.