

Prime : A Timing-Driven Placement Tool using A Piecewise Linear Resistive Network Approach

Takeo Hamada, Chung-Kuan Cheng, and Paul M. Chau

University of California at San Diego, La Jolla, CA 92093-0407

Abstract

An approach toward path-oriented timing-driven placement is proposed. We first transform the placement with timing constraints to a Lagrange problem. A primal-dual approach is used to find the optimal relative module locations. In each primal dual iteration, the primal problem is solved by a piecewise linear resistive network method, while the dual process is used to update the Lagrange multiplier. The sparsity of the piecewise linear resistive network is exploited to obtain dramatic improvement on the efficiency of the calculation. Up to 22.0% of clock cycle reduction was observed for Primary2 test case.

1 Introduction

There have been extensive studies on timing-driven placement in recent years. The approaches toward the problem are often categorized into two groups; *net-based* and *path-based*. In a typical net-based approach, potential critical paths and acceptable delays at each cell are calculated, from which slack of a net on each path is derived. The slack of a net gives the upper and lower bounds on the net size, which serves as constraints during the subsequent placement phase. The basic idea of a net-based approach can be combined with ideas such as iterative improvement [13], constructive placement [18, 12], incremental timing analysis [19], giving rise to several variations.

Net weighting [5, 18] is a technique often used with this approach; it puts heavier weight on nets with smaller slack, or puts priority on critical nets, thus turning a constrained optimization problem into an unconstrained one. It has been reported that these techniques can achieve favorable results, however it should also be noted that the net weights tend to be heuristic, which makes it difficult to apply proper mathematical analysis to the problem.

In the path-based approach all or a subset of paths are taken into account in the formulation of the problem, often with a set of linear constraints. It is expected that the problem can be handled more mathematically in this approach, since the timing in VLSI is inherently path-oriented.

Jackson and Kuh [10] proposed an approach based on linear programming. Gao et al. [7] proposed it-

erative modification of net bounds to take advantage of the computational efficiency of the net-based approach. Srinivasan et al. [17] proposed an approach based on Lagrangian Relaxation. They observed that only a small subset of timing requirements are active as constraints at one time, thus the problem of a large number of paths can be effectively avoided. They represented timing requirements by a set of linear inequalities. When the corresponding constrained optimization problem is turned into a Lagrangian, these linear inequalities make Lagrangian non-differentiable. Sub-gradient method [2] was used to update Lagrange multipliers on the non-differentiable Lagrangian.

This paper presents an approach toward the path-oriented timing-driven placement. We adopt a nonlinear timing model to estimate the delay. For the problem of the nonlinear system, we propose a transformation to formulate the system with a nonlinear resistive network. An efficient piecewise linear resistive network approach is devised to solve the nonlinear problem. The approach of the resistive network analogy exploits the sparsity of the network and thus drastically reduces the complexity of the problem.

2 Timing Model

The timing of a chip, i.e. the clock cycle of a chip, is determined by a path with the worst delay time, which is also called a critical path. A path starts at a primary input or at an output of a latch, and terminates at a primary output or at an input of a latch. The delay of a path is obtained by aggregating cell delays on the path.

The delay between two successive logic stages is composed of three elements; (i) intrinsic delay due to switching a cell on/off. (ii) delays due to charging fanout and load capacitance up/down. (iii) delay due to distributed RC of interconnection.

Scaling rule [1] suggests that the wiring delay, which are delay(ii) and (iii), will be dominant for the circuits with larger chip size and smaller geometry, and especially delay (iii), which is delay due to distributed RC, can be quite significant for submicron circuit since it

30th ACM/IEEE Design Automation Conference®

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

©1993 ACM 0-89791-577-1/93/0006-0531 1.50

grows super-linearly with the scaling factor and the chip dimension.

Suppose v_i, v_j be logic cells of two successive stages, where v_i precedes v_j . We define cell-to-cell delay $d(v_i, v_j)$ as a time interval between arriving rise/fall edges at respective stages.

When the effect of distributed RC is considered, Sakurai [15] derived a closed-form expression for inter-connection delay of a RC distributed line, and expressions given by Carter et al. [3, 9] fit this purpose. This model was also used by Prasitjutrakul and Kubitz [14] in their timing-driven global router. We simplify the model by decomposing the distance into x and y two portions and ignoring the interactive relation between x and y , i.e.

$$\begin{aligned} d(v_i, v_j) = & bd_i + \alpha \cdot (r_h c_h |x_i - x_j|^2 + r_v c_v |y_i - y_j|^2) \\ & + \beta \cdot C_j (r_h |x_i - x_j| + r_v |y_i - y_j|) \\ & + \beta \cdot R_i \sum_{i,j} (c_h |x_i - x_j| + c_v |y_i - y_j|) \\ & + \beta \cdot R_i C_L \end{aligned} \quad (1)$$

where constant bd_i is the intrinsic cell delay of cell v_i , C_j is the input load capacitance at cell v_j , R_i is the equivalent on-resistance of the output transistor of cell v_i , and C_L is the capacitive load of cell v_i . Constants c_h and c_v are unit length wire capacitance in horizontal and vertical layers, and constants r_h and r_v are unit length wire resistance in horizontal and vertical layers, respectively. When metal wiring is used both for horizontal and vertical layers, the resistance $r_h = r_v$ and the capacitance $c_h = c_v$ are typically $0.05 [\Omega/\mu m]$ and $2.0 \times 10^{-4} [pF/\mu m]$, respectively [21].

For the coefficients of α and β , the following values are commonly used [15, 3, 9, 14]: $\alpha = 1.02$, $\beta = 2.21$ for 90% threshold, $\alpha = 0.59$, $\beta = 1.21$ for 70% case, and $\alpha = 0.5$ and $\beta = 1.0$ for 62% case.

Points (x_i, y_i) , (x_j, y_j) represent positions of the terminal points. Although this timing model is non-differentiable at points $x_i = x_j$ and $y_i = y_j$ of term (3), it is convex and piecewise-differentiable. The second term of equation (1) corresponds to the distributed RC delay. When this term is dropped, the timing model can be written as a set of linear inequalities, and it reduces to the timing-requirements used by Jackson and Kuh [10] and others. The forth term represents horizontal and vertical wire length, respectively. Since exact wire length can not be known at the time of placement, they have to be approximated, either by half-perimeter wire length or single-trunk rectilinear Steiner tree (ST-RST), or some other approximation method.

3 Problem Formulation

We formulate the problem of path-oriented timing-driven placement based on the Lagrangian problem. Let us assume that all nets are two pin nets to simplify the description. In the implementation, we use

a weighted star to approximate the Steiner tree. Following the strategy of [10], we also decompose the two dimensional placement problem into two problems in x and y dimensions, respectively. Since the processes of x and y directions are symmetrical, our discussion concentrates on the process on the x direction in the sequel. Let c_{ij} be the connectivity between cell i and cell j . We want to minimize the total wire length while keeping the clock cycle no larger than τ , i.e.

$$\min z = \sum_{(i,j)} c_{ij} |x_i - x_j| \quad h_k(\mathbf{x}) \leq \tau, \quad k = 1 \dots \pi \quad (2)$$

where $\mathbf{x}$ is the vector of cell locations in x dimension, $h_k(\mathbf{x})$ is the delay equation of a path, and π is the number of paths in the circuit. Note that each of the objective function and path delay functions $h_k(\mathbf{x})$ is a convex function. The number π can be quite a large number, usually exponential to the number of cells in the circuit. We introduce slack variables s_k such that the inequalities are written as follows;

$$h_k(\mathbf{x}) + s_k = \tau. \quad (3)$$

3.1 The Lagrangian and Its Properties

Using Lagrangian multipliers, we turn (2) into an unconstrained optimization problem,

$$\begin{aligned} \max_{\lambda \geq 0} \min_{\mathbf{x}} L(\mathbf{x}, \lambda) \\ L(\mathbf{x}, \lambda) = \sum_{(i,j)} c_{ij} |x_i - x_j| + \sum_{k=1}^{\pi} \lambda_k (h_k(\mathbf{x}) - \tau) \end{aligned} \quad (4)$$

where $L(\mathbf{x}, \lambda)$ is the Lagrangian. From Kuhn-Tucker theory [8] we obtain the following complementary slackness theorem. Let λ_k^* and s_k^* be their respective values at global optimum.

Theorem 3.1 *At global optimum, $\lambda_k^* s_k^* = 0$.*

The theorem implies that there are active constraints ($\lambda_k > 0$) and inactive constraints ($\lambda_k = 0$), and we can effectively ignore those inactive constraints in the neighborhood of the global optimum. As observed by Srinivasan et al. [17], the number of active constraints are kept small throughout timing-driven placement.

We can look at the Lagrangian as a function of two sets of variables, primal variables $\mathbf{x}$ and dual variables λ . We can use $L(\mathbf{x})$ or $L(\lambda)$ to denote the Lagrangian when the other variable set is fixed. The following two optimization problems can be formulated using the above definitions: the *primal* problem

$$\min_{\mathbf{x}} L(\mathbf{x}) \quad \text{s.t.} \quad h_k(\mathbf{x}) \leq \tau, \quad k = 1, 2, \dots, \pi$$

and the *dual* problem

$$\max_{\lambda} L(\lambda) \quad \text{s.t.} \quad \lambda \geq 0.$$

A *saddlepoint* of the Lagrangian is defined as such a pair $(\mathbf{x}^*, \lambda^*)$ that satisfies

$$L(\mathbf{x}^*, \lambda) \leq L(\mathbf{x}^*, \lambda^*) \leq L(\mathbf{x}, \lambda^*)$$

for all $\lambda \geq 0$ and all $\mathbf{x}$. As customary, we assume constraint qualification, differentiability, and convexity of $h_k(\mathbf{x})$. The following *duality* theorem is due to Wolfe [22].

Theorem 3.2 *If $\hat{\mathbf{x}}$ is the solution of the convex primal problem, then dual variables $\hat{\lambda}$ exist, such that $(\hat{\mathbf{x}}, \hat{\lambda})$ solves the dual problem and the extremes of two problems are equal, that is*

$$L(\hat{\mathbf{x}}, \hat{\lambda}) = \min_{\mathbf{x}} \max_{\lambda \geq 0} L(\mathbf{x}, \lambda) = \max_{\lambda \geq 0} \min_{\mathbf{x}} L(\mathbf{x}, \lambda).$$

The main idea of the primal-dual method [2] is to find a saddlepoint of the Lagrangian by solving primal problem and dual problem alternatively.

4 Primal Solution Using A Piecewise Linear Resistive Network Approach

We propose a piecewise linear resistive network approach to solve the primal problem. In the primal problem, the current λ and the set of critical paths are given. We search for the cell positions to minimize the Lagrangian. Let K_{ij} denote the set of the critical path passing net (i, j) . The primal problem can be rewritten as

$$L(\mathbf{x}) = \sum_{(i,j)} \left[c_{ij} |x_i - x_j| + \sum_{k \in K_{ij}} \lambda_k d(x_i, x_j) \right] + A$$

where A represents the constant contributed by τ , the cell delay bd_i , the wire segment in the y direction, etc. The contribution of a net (i, j) to $L(\mathbf{x})$ can be written as a function of the coordinates of its two adjacent cells i , and j , i.e.

$$\begin{aligned} f(x_i, x_j) &= c_{ij} |x_i - x_j| + \sum_{k \in K_{ij}} \lambda_k d(x_i, x_j) = \\ &= c_{ij} |x_i - x_j| + \left(\sum_{k \in K_{ij}} \lambda_k \right) [\alpha \cdot r_h c_h |x_i - x_j|^2 \\ &\quad + \beta \cdot C_j r_h |x_i - x_j| + \beta \cdot R_i c_h |x_i - x_j|]. \end{aligned} \quad (5)$$

We rewrite $f(x_i, x_j)$ in a simpler form:

$$f(x_i, x_j) = a_{ij} (x_i - x_j)^2 + b_{ij} |x_i - x_j| \quad (6)$$

where positive constants a_{ij} , b_{ij} are given by

$$\begin{aligned} a_{ij} &= \left(\sum_{k \in K_{ij}} \lambda_k \right) \alpha r_h c_h \\ b_{ij} &= c_{ij} + \left(\sum_{k \in K_{ij}} \lambda_k \right) \beta (C_j r_h + R_i c_h) \end{aligned}$$

Let E_i be the set of cells adjacent to cell i . The necessary condition [8] for the optimal solution requires that the gradient of $L(\mathbf{x})$ be equal to zero, i.e.

$$\partial L(\mathbf{x}) / \partial x_i = \sum_{j \in E_i} \partial f(x_i, x_j) / \partial x_i = 0. \quad (7)$$

Suppose that $x_i \neq x_j$. Let $\delta_{ij} = 1$, if $x_i > x_j$; $\delta_{ij} = -1$, otherwise. The partial derivative of $f(x_i, x_j)$ with respect to x_i can be expressed as

$$\partial f(x_i, x_j) / \partial x_i = 2a_{ij}(x_i - x_j) + \delta_{ij} b_{ij}. \quad (8)$$

4.1 The Analogy of A Nonlinear Resistive Network

We transform the primal problem to a nonlinear resistive network. The network has each node i correspond to each cell i , and the voltage of node i correspond to the location x_i of cell i . For each net (i, j) , we construct a nonlinear resistor connecting nodes i and j with a conductance, ϵ , σ_{ij} . Given a constant ϵ , σ_{ij} is defined as follows:

$$\sigma_{ij} = \begin{cases} 2a_{ij} + b_{ij}/|x_i - x_j| & \text{if } |x_i - x_j| > \epsilon \\ 2a_{ij} + b_{ij}/\epsilon & \text{if } |x_i - x_j| \leq \epsilon \end{cases} \quad (9)$$

The current flowing from node i to node j is equal to the product of the voltage difference $x_i - x_j$ and the conductance σ_{ij} , i.e.

$$\begin{aligned} (x_i - x_j) \cdot \sigma_{ij} &= \\ &= \begin{cases} 2a_{ij}(x_i - x_j) + \delta_{ij} b_{ij} & \text{if } |x_i - x_j| > \epsilon \\ 2a_{ij}(x_i - x_j) + b_{ij}(x_i - x_j)/\epsilon & \text{if } |x_i - x_j| \leq \epsilon \end{cases} \end{aligned} \quad (10)$$

which is an approximation of the partial derivative of $f(x_i, x_j)$ with respect to x_i , equation (8). Consequently, the Kirchhoff's current law [4] in the nonlinear resistive network corresponds to the necessary condition equation (7). Therefore we can claim the analogy between the primal problem and the resistive network.

Theorem 4.1 *Given a constant Υ , there exists an ϵ for conductance equation (9) such that the voltage solution of the transformed nonlinear resistive network is an approximate solution of the primal problem with an error bound Υ on the value of $L(\mathbf{x})$*

Note that this analogy is an extension of the approach proposed by Sigl et al. [16]. In [16], a quadratic equation with nonlinear coefficients is derived to match the value of the objective function. Our approach derives the equivalence of the resistive network solution by matching the gradient constraint (7) of the original problem to the current law of the new problem, rather than by matching only on the value of the objective function.

4.2 A Piecewise Linear Resistive Network Algorithm

Figure 1 illustrates the concept of the piecewise linear resistive network, where m is the number of cells, and $n - m$ is the number of I/O pads. The node voltages of m free terminals correspond to the cell positions whereas the $n - m$ independent voltage sources correspond to the positions of the I/O pads.

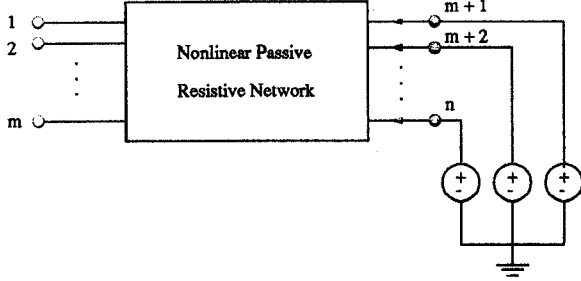


Figure 1: An n -terminal Passive Resistive Network

Let $\mathbf{x}_1$ denotes cell positions, and $\mathbf{x}_2$ denotes fixed I/O pad positions. Let G_{11} and G_{12} denote the submatrices of the admittance matrix corresponding to vectors $\mathbf{x}_1$ and $\mathbf{x}_2$. The primal problem reduces to solving the following equation

$$G_{11}\mathbf{x}_1 = -G_{12}\mathbf{x}_2$$

Note that the admittance matrix contains a nonzero entry (i, j) only if there is a net (i, j) in the placement problem. Utilizing the sparsity of the network, the complexity of the problem is drastically reduced. Since all the conductances are positive, the admittance matrix is symmetric and positive definite. Furthermore, after the approximation of the partial derivative of $f(\mathbf{x}_i, \mathbf{x}_j)$ by equation (9), the resistive network satisfies the Lipschitz condition [11].

We can approximate the nonlinear resistance by a piecewise linear resistive network. The piecewise linear resistive network approach proposed by Katzenelson, Fujisawa and Kuh ($K-F-K$) [11, 6] can be exploited to obtain the solution. The Lipschitz condition and the positive definite condition of the admittance matrix constitute a sufficient condition for the convergence of the algorithm [6]. We state this fact as the following theorem.

Theorem 4.2 *The $K-F-K$ piecewise linear resistive network approach proposed by [6] converges to a global minimum solution on the resistive network.*

It is however costly to apply $K-F-K$ algorithm exactly to the placement problem, since the solution process must stop at the boundary of the regions, and the Hessian of the Lagrangian must be recalculated at

the point to search for a new direction of the solution curve.

We improve the efficiency of the algorithm by skipping the discontinuous points and jump to the local optimal solution. Our piecewise linear resistive network method starts from an initial node voltage vector $\bar{\mathbf{x}}$. Given a current voltage vector $\bar{\mathbf{x}}$, we define the conductance of the resistors according to (9). Next, we treat the circuit as a linear resistive network, and find the voltage solution with the Successive Over Relaxation (SOR) method [20]. We then update $\bar{\mathbf{x}}$ to the derived voltage vector and repeat the process of the linear resistive network construction and the voltage solution derivation. The iteration continues until the voltage vector $\bar{\mathbf{x}}$ converges or the number of iterations is beyond a limit. In our implementation, we set this limit as a constant factor of the region size.

Algorithm 4.1

Piecewise Linear Resistive Network Approach {
 1. Set initial point $\mathbf{x}$ vector.
 2. Calculate σ_{ij} according to the given $\mathbf{x}$ vector.
 3. Solve $G_{11}\mathbf{x}_1 = -G_{12}\mathbf{x}_2$ as a linear resistive network.
 4. Repeat 2.-3., until convergence or number of iterations exceeds the limit.
 } /* end of algorithm. */

We devise a potential function $\Psi(\mathbf{x})$ and show the convergence of the algorithm.

Theorem 4.3 *For each iteration of steps 2 & 3 of the algorithm, the potential function $\Psi(\mathbf{x})$ strictly decreases.*

5 Dual Solution

To solve the dual problem, we apply the Newton method to update dual variables λ . Suppose the dual function $L(\lambda)$ is twice differentiable regarding λ . Let λ^k be the k -th iterative solution of λ by the primal-dual loop, and consider a second-order Taylor series of $L(\lambda)$ about λ^k ,

$$L(\lambda) \approx L(\lambda^k) + (\lambda - \lambda^k) \nabla L(\lambda^k) + \frac{1}{2} (\lambda - \lambda^k)^T \text{Hess}(\lambda^k) (\lambda - \lambda^k) \quad (11)$$

where $\text{Hess}(\lambda^k)$ represents the Hessian of $L(\lambda)$ at λ^k . From Theorems 3.1 and 3.2 $\text{Hess}(\lambda^k)$ is negative semi-definite in the neighborhood of a saddle-point. We want to achieve the maximum of $L(\lambda)$ such that $\nabla L(\lambda) = 0$. This leads us to choose the next dual variables λ^{k+1} by

$$\lambda^{k+1} - \lambda^k = -[\text{Hess}(\lambda^k)]^{-1} \nabla L(\lambda^k). \quad (12)$$

Note that $\lambda \geq 0$ must be observed, which requires that any of the variables of λ^{k+1} that turn negative

during the update are set to 0, and the corresponding path turns inactive.

Deriving from equations (3) and (4), we have

$$\nabla L(\lambda^k) = -s^k$$

where s^k is a vector of slack variables at the k -th iteration given by (3). We further assume that $Hess(\lambda^k)$ is diagonally dominant, which implies that the paths corresponding to active constraints are relatively independent of each other. This assumption can be true only when the number of active paths is relatively small compared to the dimension of $Hess(\lambda^k)$, which equals to the number of movable cells in the circuit. In fact, the number of active paths are usually kept small, only a fraction of the number of movable cells, throughout the primal-dual iterations. We also assume that $Hess(\lambda^k)$ is negative definite, such that a unique maximizer of (11) is found by (12). When $Hess(\lambda^k)$ is negative definite, so is its inverse $[Hess(\lambda^k)]^{-1}$. The inverse $[Hess(\lambda^k)]^{-1}$ is then approximated by

$$[Hess(\lambda^k)]^{-1} \approx D(\lambda^k) + \alpha I$$

where $D(\lambda^k)$ is a diagonal matrix, whereas α is a small negative constant to keep $[Hess(\lambda)]^{-1}$ negative definite. The diagonal matrix $D(\lambda^k)$ is calculated numerically as follows:

$$D_{ii}(\lambda^k) = -\frac{\lambda_i^k - \lambda_i^{k-1}}{s_i^k - s_i^{k-1}}$$

where s_i^k, s_i^{k+1} and $\lambda_i^k, \lambda_i^{k+1}$ are slack variables and dual variables of path p_i , corresponding to the k -th and $k+1$ -th iterations, respectively.

6 Prime : A Path-Oriented Timing-Driven Placement Tool

Algorithm 6.1 illustrates the proposed path-oriented timing-driven placement algorithm via pseudo-code.

This algorithm is composed of three parts; timing-driven global placement, initial detail placement by linear placement, and iterative refinement by pairwise swapping. The heart of timing-driven global placement is *primal-dual* iterations. The placement is performed within a region. Initially, there is only one region which covers the whole chip. In each iteration of the loop, a region with the largest size is taken.

First, global placement algorithm without timing constraints is applied to obtain an initial primal solution, from which the target clock cycle is calculated as 98% of the current critical timing. $L(x, \lambda)$ represents the Lagrangian, whose saddle point gives a solution to the path-oriented timing-driven placement.

Second, timing requirements are added to the Lagrangian. Third, the primal-dual iteration is applied to the Lagrangian. After each iteration, convergence

Algorithm 6.1

```

Prime {
  /* timing-driven global placement. */
  B: constant region size.
  while (region size > B) {
    take the largest region from the list;
    find initial primal solution;
     $\tau$ : 98% of the current critical timing.
    add timing requirements to
      Lagrangian  $L(x, \lambda)$ ;
    /* primal-dual iteration. */
    L : loop limit;
     $\epsilon$  : convergence error;
    for (loop = 0; loop < L; loop++) {
      /* dual problem. */
      maximize  $L(x, \lambda)$  w.r.t.  $\lambda$ ;
      /* primal problem. */
      minimize  $L(x, \lambda)$  w.r.t.  $x$ ;
      if ( $\|x^{new} - x^{old}\| < \epsilon$ ) {
        /*
          solution converged.
          end primal-dual loop.
        */
        break;
      }
      if (target cycle time is achieved) {
        set new target cycle  $\tau$ ;
        update Lagrangian;
      }
    }
    partition the current region;
  }
  /* initial detail placement. */
  linear placement of each region;
  /* iterative refinement. */
  pairwise swapping;
  row length equalization;
  orientation flipping;
} /* end of algorithm. */

```

of the solution is examined. The convergence error ϵ is set to a small value such that the solution is accurate enough to represent cell positions. If the target cycle time is achieved before its full convergence, new target cycle time is set, and the Lagrangian is updated, continuing the primal-dual iteration. Note that the target clock cycle originally set may be overconstrained, and the primal-dual iteration may not converge. We set loop limit L , beyond which the iteration is terminated. The constant L is set to a factor of the region size such that the iteration converges in normal cases. Lastly, the region is partitioned according to the positions of cells in the current solution, to satisfy slot constraints, forming a balanced slicing tree.

7 Experimental Results

Prime is implemented in C language. All the following experiments were performed on Sun 4/IX with

Table 1: Statistics on Test Cases

circuit	cell	pad	net	pin	latch
(I) Primary1	752	81	1266	5888	269
(II) Primary2	2907	107	3817	19195	603

Table 2: Experimental Results on Primary1 and Primary2

Algo.	H-P [mm]	ST-RST [mm]	cycle [ns]	delay [ns]	CPU [s]
(I) CV	1055.7	1348.2	45.43	24.58	144.1
(I) TD	1111.7	1379.4	38.32	17.46	234.5
(II) CV	4439.1	7063.7	92.63	52.95	797.2
(II) TD	4512.9	6830.0	72.25	31.55	2049.2

48M bytes of main memory. We applied *Prime* to Primary1 and Primary2 test cases from MCNC. Both test cases are sequential circuit. Table 1 shows statistics on the two test cases.

Table 2 summarizes experimental results on Primary1 and Primary2 test cases. We used *Prime* without timing constraints to obtain results by conventional placement (CV), and by timing-driven placement (TD) for respective test cases, Primary1 (I) and Primary2 (II). The table shows half-perimeter wire length (H-P), single-trunk rectilinear Steiner tree (ST-RST) wire length, clock cycle time, wire delay, which is the contribution from wire delay to the clock cycle time, and CPU time. We assumed metal wiring for both vertical and horizontal layers. We used the timing model of equation (1) to calculate cell-to-cell delay, and we used ST-RST to estimate load capacitance of wires. For Primary1 test case, 17.5% reduction in clock cycle time was obtained at the cost of 2.3% increase of ST-RST wire length and 1.63 times increase of CPU time. The percentage of the reduction in terms of the wire delay is 29.0%. For Primary2 test case, 22.0% reduction in clock cycle time was obtained at the cost of 2.57 times increase of CPU time. The percentage of the reduction in terms of the wire delay only is 40.0%.

References

- [1] H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*, Reading, MA: Addison-Wesley, 1990.
- [2] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, San Diego, CA: Academic Press, 1982.
- [3] D. L. Carter and D. F. Guise, "Analysis of Signal Propagation Delays and Chip Level Performance Due to On-Chip Interconnections," *Proc. of Int. Conf. on Computer Design*, pp. 218-221, Nov. 1983.
- [4] C. A. Desoer and E. S. Kuh, *Basic Circuit Theory*, New York, NY: McGraw-Hill, 1969.
- [5] A. E. Dunlop, V. D. Agrawal et al., "Chip Layout Optimization Using Critical Path Weighting," *Proc. of 21st Design Automation Conf.*, pp. 133-136, 1984.
- [6] T. Fujisawa and E. S. Kuh, "Piecewise-linear Theory of Nonlinear Networks," *SIAM J. Appl. Math.*, vol. 22, no. 2, pp.307-328, Mar. 1972.
- [7] T. Gao, P. M. Vaidya, and C. L. Liu, "A Performance Driven Macro-Cell Placement Algorithm," *Proc. of 29th Design Automation Conf.*, pp.147-152, June 1992.
- [8] G. Hadley, *Nonlinear and Dynamic Programming*, Reading, MA: Addison-Wesley, 1964.
- [9] E.-H. Horneber and W. Mathis, "A Closed-Form Expression for Signal Delay in CMOS-Driven Branched Transmission Lines" in *VLSI'87*, ed. by C. H. Sequin, pp. 353-362.
- [10] M. A. B. Jackson and E. S. Kuh, "Performance-Driven Placement of Cell Based IC's," *Proc. of 26th Design Automation Conf.*, pp.370-375, June 1989.
- [11] J. Katzenelson, "An algorithm for solving nonlinear resistive networks," *Bell System Tech. J.*, vol. 44, pp.1605-1620, 1965.
- [12] I. Lin, D. H. C. Du, "Performance-Driven Constructive Placement," *Proc. of 27th Design Automation Conf.*, pp.103-106, June 1990.
- [13] Y. Ogawa, et al., "Efficient Placement Algorithms Optimizing Delay for High-Speed ECL Masterslice LSI's," *Proc. of 23rd Design Automation Conf.*, pp. 404-410, June 1986.
- [14] S. Prasitjutrakul and W. J. Kubitz, "A Timing-Driven Global Router for Custom Chip Design," *Proc. of Int. Conf. on Computer-Aided Design*, pp.48-51, Nov. 1990.
- [15] T. Sakurai, "Approximation of Wiring Delay in MOSFET LSI," *IEEE Journal of Solid-State Circuits*, vol. SC-18, no. 4, pp. 418-426, 1983.
- [16] G. Sigl, K. Doll, F. M. Johannes, "Analytical Placement: A Linear or a Quadratic Objective Function?," *Proc. of 28th Design Automation Conf.*, pp. 427-432, June 1991.
- [17] A. Srinivasan, K. Chaudhary, E. S. Kuh, "RITUAL: A Performance Driven Placement Algorithm for Small Cell IC's," *Proc. of Int. Conf. on Computer-Aided Design*, pp. 48-51, Nov. 1991.
- [18] S. Sutanthavibul and E. Shragowitz, "An Adaptive Timing-Driven Layout for High Speed VLSI," *Proc. of 27th Design Automation Conf.*, pp. 90-95, June. 1990.
- [19] S. Teig, R. L. Smith and J. Seaton, "Timing-Driven Layout of Cell-Based ICs," *VLSI Systems Design*, pp. 63-73, May 1986.
- [20] R. S. Tsay and E. S. Kuh, "Module Placement for Large Chips Based on Sparse Linear Equations," *Int. Journal of Circuit Theory and Applications*, vol. 16, pp. 411-423, 1988.
- [21] N. Weste, K. Eshraghian, *Principles of CMOS VLSI Design: A Systems Perspective*, Reading MA: Addison-Wesley, 1985.
- [22] P. Wolfe, "A Duality Theorem for Nonlinear Programming," *Quarterly of Applied Math.*, vol. 19, 239, 1961.