VLSI Design, 2002 VOL. 15 (3), pp. 587–594

Taylor & Francis
Taylor & Francis Group

Simultaneous Buffer-sizing and Wire-sizing for Clock Trees Based on Lagrangian Relaxation

YU-MIN LEE[a], CHARLIE CHUNG-PING CHEN[a,*], YAO-WEN CHANG[b] and D.F. WONG[c]

[a]*Department of Electrical and Computer Engineering, University of Wisconsin, Madison, WI 53706, USA;* [b]*Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, ROC;* [c]*Department of Computer Sciences, University of Texas, Austin, TX 78712, USA*

(Received 15 March 2001; Revised 30 January 2002)

Delay, power, skew, area and sensitivity are the most important concerns in current clock-tree design. We present in this paper an algorithm for simultaneously optimizing the above objectives by sizing wires and buffers in clock trees. Our algorithm, based on Lagrangian relaxation method, can optimally minimize delay, power and area simultaneously with very low skew and sensitivity. With linear storage *overall* and linear runtime *per iteration*, our algorithm is extremely economical, fast and accurate; for example, our algorithm can solve a 6201-wire-segment clock-tree problem using about 1-minute runtime and 1.3-MB memory and still achieve pico-second precision on an IBM RS/6000 workstation.

Keywords: VLSI CAD; Interconnect optimization; Lagrangian relaxation; Buffer-sizing; Wire-sizing; Clock trees

INTRODUCTION

Delay, skew, power, area and skew sensitivity are the most important concerns in current clock-tree design. With the increasing complexity of synchronous ASICs, clock skew and clock-signal delay have become important factors in determining circuit performance [2,4,10,17]. Wire width process variations during fabrication can significantly impact the delay and skew; thus, it is important to consider the sensitivity of a design to inter-chip process variations [13]. As reported in Ref. [7], power dissipation of a clock tree play a key role in overall chip's power dissipation. Therefore, it is desirable to simultaneously consider delay, skew, power, area and sensitivity in clock-tree design.

Algorithms for routing-tree optimization have been proposed in much of the literature recently [3,4,5,12,13,15,17]. The works in Refs. [3,5,12,15] are designed for general routing tree, hence, they cannot handle clock tree issues such as skew and sensitivity. Although Refs. [4,13,14,17] consider sensitivity, skew and/or delay, most of these algorithms only size wires and do not minimize power and area. Moreover, existing algorithms suffer long runtime and large storage requirements. For example, Refs. [13,17] convert the skew minimization problem into the least-squares minimization problem. However, due to the storage and inversion of large gradient matrices, their respective runtime *per iteration* and storage requirements are about cubic and quadratic in the problem size.

We present in this paper an algorithm for simultaneously optimizing the above-mentioned objectives by sizing wires and buffers in clock trees. Our algorithm, based on the Lagrangian relaxation method, can simultaneously optimize delay, power and area with very low skew and sensitivity; it relaxes the constraints scaled with Lagrangian multipliers into its objective function and then iteratively solve the subproblems resulted from dynamically adjusting the Lagrangian multipliers. Our algorithm is extremely fast, economical and accurate; it requires only linear storage *overall* and linear runtime *per iteration* for adjusting wire and buffer sizes. For example, we can solve a 6201-wire-segment clock-tree problem in about 1-min runtime and 1.3-MB memory and still guarantee pico-second precision on an IBM RS/6000 workstation.

*Corresponding author.

ISSN 1065-514X print/ISSN 1563-5171 online © 2002 Taylor & Francis Ltd
DOI: 10.1080/1065514021000012200

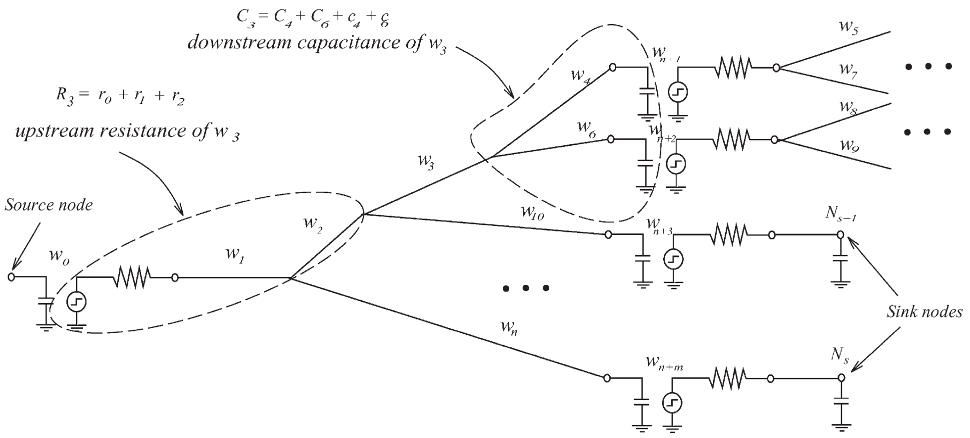

FIGURE 1 Upstream resistance and downstream capacitance.

PRELIMINARIES

We use the following notations in this paper.

- T: A clock tree with a driver w_0 at the root (source) and a set of s sinks $\{N_1, N_2, \ldots, N_s\}$.
- w_i: i-th wire segment or buffer. w_i is a wire segment when $1 \le i \le n$, or a buffer when $n+1 \le i \le n+m$ or $i = 0$.
- x_i, l_i: Size and length of w_i, respectively.
- λ: $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_s)$ is the Lagrange-multiplier vector.
- $\mathbf{x}$: $\mathbf{x} = (x_0, x_1, x_2, \ldots, x_{n+m})$ is a wire- and buffer-sizing solution.
- ρ_i: Resistance of wire per unit length at unit width, when $1 \le i \le n$; resistance of unit-size buffer, when $i = 0$ or $n+1 \le i \le n+m$.
- ϵ_i: Area capacitance of wire per unit square, when $1 \le i \le n$; capacitance of unit-size buffer, when $i = 0$ or $n+1 \le i \le n+m$.
- r_i: Resistance of w_i. $r_i \approx \rho_i l_i / x_i$, when $1 \le i \le n$; $r_i \approx \rho_i / x_i$, when $n+1 \le i \le n+m$ or $i = 0$.
- c_i: Capacitance of w_i. $c_i \approx \epsilon_i l_i x_i$, when $1 \le i \le n$; $c_i \approx \epsilon_i x_i$, when $n+1 \le i \le n+m$ or $i = 0$.
- U_i, L_i: Upper bound and lower bound of the size of w_i, respectively, i.e. $L_i \le x_i \le U_i$, $0 \le i \le n+m$.
- P_i: All wires and buffers on the path from the source to sink N_i (including N_i).
- T_i: All wires and buffers in the subtree of T rooted at w_i (excluding w_i).
- parent(w_i): Parent of w_i.
- Child(w_i): *Set* of w_i's children.
- Ans(w_i): All wires or buffers on the path from w_i to the nearest upstream buffer or the root (excluding w_i).
- Dec(w_i): All wires, buffers or sinks on the paths from w_i to the neighboring downstream buffers or sinks (excluding w_i).
- R_i: Upstream resistance of w_i; $R_i = \sum_{w_j \in \mathrm{Ans}(w_i)} r_j$.
- C_i: Downstream capacitance of w_i; $C_i = \sum_{w_j \in \mathrm{Child}(w_i)} (C_j + c_j) + \sum_{N_j \in \mathrm{Child}(w_i)} \tilde{c}_j$, where $\tilde{c}_j$ is the capacitance of sink N_j, $1 \le j \le s$.
- A: Area of a clock tree; $A = \sum_{i=1}^{n} x_i l_i + \sum_{i=n+1}^{n+m} x_i + x_0$.

See Fig.1 for an illustration of R_i and C_i.

We use a distributed resistance–capacitance (RC) segment to represent a branch of a clock tree (see Fig. 2(a)). The distributed RC segment can be modeled as an equivalent lumped π-circuit. The lumped resistance and capacitance of the π-model of an RC segment w_i are approximated by $\rho_i l_i / x_i$ and $\epsilon_i x_i l_i$, respectively. We use the switch-resistor model to compute buffer delays (see Fig. 2(b)) and apply the Elmore delay model [8] to approximate signal delays in a subtree. Given a distributed RC routing tree T, its signal delay at sink N_i is computed by

$$D_i = \sum_{w_j \in P_i,\, 1 \le j \le n} r_j \left(C_j + \frac{c_j}{2} \right) + \sum_{w_j \in P_i,\, n+1 \le j \le n+m} r_j C_j + r_0 C_0.$$

In practical CMOS applications, capacitive dissipation (due to charging and discharging of load capacitances)

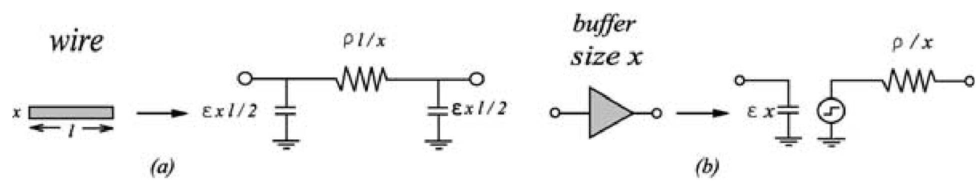

FIGURE 2 RC model for wire and buffer.

usually dominates the other types of power dissipation [5]. Hence, we consider only the capacitive dissipation in this paper. Given a clock tree, its power dissipation P can be approximated by $P \approx fC_{\text{tot}}V_{dd}^2$, where f is the clock frequency and C_{tot} is the total capacitance of the tree.

Clock skew is defined as the maximum difference in the delays from the clock source to clock sinks; that is, the skew of a clock tree, $S = \max_{i,j}|D_i - D_j|$. Given wire width w, the *skew sensitivity*, Δ, is defined as the maximum difference between skews under varying values of w due to process variations [4]. The goal of sensitivity minimization is to find an optimal w such that Δ is minimized.

This paper addresses the clock-tree wire- and buffer-sizing problem, targeting multiple objectives such as delay, skew, power, area and sensitivity. We give the formulation for the wire- and buffer-sizing problem as follows:

The *Clock-Tree Wire- and Buffer-Sizing Problem*

Given: A clock tree T with the source N_0 and sinks $\{N_1, N_2, \ldots, N_s\}$, wire segments $\{w_1, w_2, \ldots, w_n\}$, buffers $\{w_0, w_{n+1}, w_{n+2}, \ldots, w_{n+m}\}$, upper bounds $\{U_0, U_1, \ldots, U_{n+m}\}$, and lower bounds $\{L_0, L_1, \ldots, L_{n+m}\}$.

Objective: Find an $\mathbf{x}$ that minimizes $\max_{1\le i\le s} D_i$, S, P, A and/or Δ.

An example of *Clock-Tree Wire- and Buffer-Sizing Problem*

Figure 3 illustrates an example of clock trees with source N_0. There are three sinks (N_1, N_2 and N_3), five wires (w_1, w_2, w_3, w_4 and w_5), and two buffers (w_0, w_6) in this clock tree. The goal is to find a set of wire and buffer sizes to minimize $\max_{1\le i\le s} D_i$, S, P, A and/or Δ.

DELAY/POWER/AREA MINIMIZATION

We formulate the wire- and buffer-sizing problem for simultaneous delay, power and area minimization as follows:

$$\begin{aligned} \mathscr{M}:\ \text{Minimize}\quad & \alpha D_{\max} + \beta P + \gamma A \\ \text{Subject to}\quad & D_i(\mathbf{x}) \le D_{\max}, \quad 1 \le i \le s, \\ & L_i \le x_i \le U_i, \quad 0 \le i \le n+m, \\ & D_{\max} > 0, \end{aligned}$$

where α, β and γ are the given constants. Note that $D_{\max}$ is a variable we introduced to minimize maximum delay. As shown above, there are two sets of inequalities. The first set of s inequalities is used to ensure that every sink satisfies its delay constraint. The second set of inequalities is used to ensure that the size of every wire segment and buffer satisfies its size constraints.

By dividing both sides of the delay, lower bound, and upper bound constraints by $D_{\max}$, x_i and U_i, respectively, we can rewrite these constraints as $(D_i(\mathbf{x})/D_{\max}) \le 1$, $(L_i/x_i) \le 1$ and $(x_i/U_i) \le 1$. Hence, $\mathscr{M}$ becomes a geometric programming problem which can be reduced to a convex programming problem by an exponential transformation [6]. However, since general geometric programming solvers usually involve gradient matrices inversions, their storage and runtime requirements are at least quadratic and cubic in the problem size, respectively. Therefore, it is desirable to develop an efficient algorithm for solving this problem.

Our approach for solving $\mathscr{M}$ is based on Lagrangian relaxation [1,9]. We relax the delay constraints into

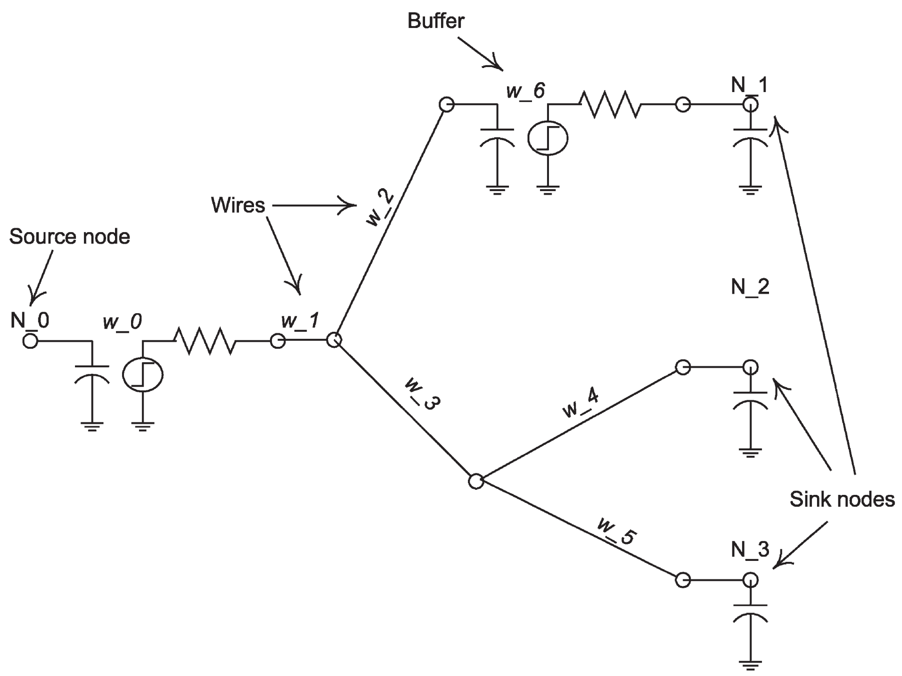

FIGURE 3 An example of *Clock-Tree Wire- and Buffer-Sizing Problem.*

the objective function by introducing Lagrange multipliers λ_i's, $1 \le i \le s$, one for each delay constraint $D_i(\mathbf{x}) \le D_{\max}$. We have the Lagrangian-relaxation subproblem for $\mathscr{M}$ as follows:

$$\begin{aligned} \mathscr{M}':\ \text{Minimize} \quad & \alpha D_{\max} + \beta P + \gamma A + \textstyle\sum_{i=1}^{s} \lambda_i (D_i(\mathbf{x}) \\ & - D_{\max}) \\ \text{Subject to} \quad & L_i \le x_i \le U_i, \quad 0 \le i \le n+m, \\ & D_{\max} > 0. \end{aligned}$$

For each λ, let $\mathscr{L}(\lambda)$ be the optimal objective function value of $\mathscr{M}'$. It is well known that $\mathscr{L}(\lambda)$ is a lower bound of the optimal objective value of $\mathscr{M}$ [1,9]. On the other hand, any feasible solution of $\mathscr{M}$ is an upper bound of the optimal objective value. Hence, we can use these two bounds to evaluate the quality of a current solution and to determine the termination criteria. By the Kuhn–Tucker theory [11] and the fact that $\mathscr{M}$ is equivalent to a convex programming problem, we have the following theorem.

THEOREM 1 $(\mathbf{x}^*, D^*_{\max})$ is an optimal solution if and only if there exists a vector $\lambda^* = (\lambda_1{}^*, \lambda_2{}^*, \ldots, \lambda_s^*)$ such that

(1) $\sum_{i=1}^{s} \lambda_i^* = \alpha$;
(2) $\lambda_i^*(D_i(\mathbf{x}^*) - D_{\max}{}^*) = 0, 1 \le i \le s$;
(3) $D_i(\mathbf{x}^*) - D_{\max}{}^* \le 0, 1 \le i \le s$;
(4) $\lambda_i^* \ge 0, 1 \le i \le s$;
(5) $x_i^* = \min(U_i, \max(L_i, \Phi_i))$, where

$$\Phi_i = \sqrt{(\rho_i \mu_i C_i)/(\beta p_i + \gamma + \epsilon_i \textstyle\sum_{w_j \in \mathrm{Ans}(w_i)} r_j \mu_j)},$$

$$p_i = f \epsilon_i V_{dd}^2,$$

and

$$\mu_i = \textstyle\sum_{N_j \in T_i} \lambda_j, \quad 0 \le i \le n+m.$$

Proof Since the objective function is a posynominal and the delay constraints are also posynominals after dividing both the sides with $D_{\max}$. $\mathscr{M}$ is a geometric programming problem which is equivalent to a convex programming problem under the following transformation $x_i = e^{y_i}$. Hence, a local minimum of $\mathscr{M}$ is a global minimum of $\mathscr{M}$.

We write down Kuhn–Tucker conditions [11] for $\mathscr{M}'$ as follows:

$$\frac{\partial \mathscr{L}}{\partial D_{\max}{}^*} = 0, \tag{1}$$

$$\frac{\partial \mathscr{L}}{\partial x_i} = 0, \quad 0 \le i \le n+m, \tag{2}$$

$$\lambda_i(D_i(\mathbf{x}^*) - D_{\max}{}^*) = 0, \quad 1 \le i \le s, \tag{3}$$

$$D_i(\mathbf{x}^*) - D_{\max}{}^* \le 0, \quad 1 \le i \le s, \tag{4}$$

$$D_{\max}{}^* > 0, \tag{5}$$

$$\lambda_i \ge 0, \quad 1 \le i \le s. \tag{6}$$

By Eq. (1), we get

$$\sum_{i=1}^{s} \lambda_i = \alpha \tag{7}$$

We can also rewrite Eq. (2) as follows:

$$\begin{aligned} \frac{\partial \mathscr{L}}{\partial x_i{}^*} &= \frac{\partial \beta P + \gamma A + \sum_{j=1}^{s} \lambda_j D_j + (\alpha - \sum_{j=1}^{s} \lambda_j) D_{\max}{}^*}{\partial x_i} \\ &= \frac{\partial \beta P + \gamma A + \sum_{j=1}^{s} \lambda_j \left[\sum_{w_k \in P_j, 1 \le k \le n} r_k \left(C_k + \frac{c_k}{2}\right) + \sum_{w_k \in P_j, n+1 \le k \le n+m} r_k C_k + r_0 C_0 \right]}{\partial x_i} \\ &= \frac{\partial \beta P + \gamma A + \left[\sum_{w_k \in T, 1 \le k \le n} r_k \left(C_k + \frac{c_k}{2}\right) \sum_{N_j \in \mathrm{dec}(w_k)} \lambda_j + \sum_{w_k \in T, n+1 \le k \le n+m} r_k C_k \sum N_j \in \mathrm{dec}(w_k) \lambda_j + r_0 C_0 \right]}{\partial x_i} \\ &= \frac{\partial \beta P + \gamma A + \left[\sum_{w_k \in T, 1 \le k \le n} r_k \mu_k \left(C_k + \frac{c_k}{2}\right) + \sum_{w_k \in T, n+1 \le k \le n+m} r_k \mu_k C_k + r_0 C_0 \right]}{\partial x_i}. \end{aligned}$$

Note that the terms that involve x_i come from $\sum_{w_k \in \mathrm{ans}(w_i)} r_k \mu_k C_k$. In fact, only the term $\epsilon_i l_i x_i$ (the wire capacitance of w_i) and $\epsilon_i x_i$ (the buffer capacitance of w_i) in C_k contribute to the terms with x_i, hence

$$A_i(\mathbf{x}^*) = \begin{cases} l_1 \left(\beta p_i + \gamma + \epsilon_i \sum_{w_j \in \mathrm{ans}(w_i)} r_j \mu_j \right) & 1 \le i \le n, \\ \beta p_i + \gamma + \epsilon_i \sum_{w_j \in \mathrm{ans}(w_i)} r_j \mu_j & i = 0 \text{ or } n+1 \le i \le n+m. \end{cases}$$

Since the terms that involve $(1/x_i)$ only coming from $r_i\mu_i C_i$, we have

$$B_i(\mathbf{x}^*) = \begin{cases} l_i\rho_i\mu_i C_i & 1 \le i \le n, \\ \rho_i\mu_i C_i & i = 0 \text{ or } n+1 \le i \le n+m. \end{cases}$$

It is clear that $A_i(\mathbf{x}^*)$, and $B_i(\mathbf{x}^*)$ are independent of x_i. Hence, we can rewrite $(\partial\mathscr{L}/\partial x_i)$ as follows:

$$\frac{\partial\mathscr{L}}{\partial x_i} = \frac{\partial A_i(\mathbf{x}^*)x_i + \frac{B_i(\mathbf{x}^*)}{x_i} + E_i(\mathbf{x}^*)}{\partial x_i} = A_i(\mathbf{x}^*) - \frac{B_i(\mathbf{x}^*)}{x_i^2},$$

where $E_i(\mathbf{x}^*)$ is independent of x_i, since while fixing other variables, $(\partial\mathscr{L}/\partial x_i)$ is a convex function respect to a single variable x_i. We know that the optimal x_i^* satisfies following equation:

$$x_i^* = \min\left(U_i,\ \max\left(L_i, \sqrt{\frac{B_i(\mathbf{x}^*)}{A_i(\mathbf{x}^*)}}\right)\right),$$
$$0 \le i \le n+m. \tag{8}$$

Theorem 1 thus follows. □

Based on the above analysis, we need to find $\mathbf{x}^*$ and λ^* to solve Problem $\mathscr{M}$. Once λ_i's are assigned, we can compute $\mathbf{x}^*$ based on Theorem 1(5). Hence, we can adopt a two-level approach to solve this problem: in the outer loop, we dynamically adjust sink weights λ_i's; weight associated with each sink is proportional to the signal delay of the sink. In the inner loop, we find an optimal wire- and buffer-sizing solution for the given λ_i's. With this in mind, we present the Lagrangian-relaxation-based algorithm shown in Fig. 4; the algorithm iteratively adjusts the multipliers based on the delay information associated with sinks and solves the corresponding Lagrangian relaxation subproblems. Our algorithm runs in $O(pqn)$ time using $O(n)$ storage, where p is the number of iterations (**A3**–**A6**) in OWBA and q is the number of iterations (**S2**–**S3**) in LRS. Empirically, the overall runtime approaches linear. We have the following theorem.

Algorithm: OWBA (Optimal Wire- and Buffer-sizing Algorithm)
A1 Let $k = 0$, $x_i = L_i$, $0 \le i \le n+m$.
A2 $\lambda_i = 1/s$, $1 \le i \le s$.
A3 Call Subroutine LRS.
A4 Recursively compute all sink delays D_i's; let $D_{max} = \max_i(D_i(\mathbf{x}))$.
A5 Adjust sink weights λ_i's according to the formula $\lambda_i = \lambda_i + \theta_k(D_i(\mathbf{x}) - D_{max})$, $1 \le i \le s$, where step size θ_k satisfies $\lim_{k\to\infty}\theta_k = 0$ and $\sum_{j=1}^{k}\theta_j \to \infty$.
A6 $k = k+1$.
A7 Repeat **A3**–**A6** until $D_{max} - \mathcal{L}(\lambda) \le$ error bound.

Subroutine: LRS (Lagrangian-Relaxation Subroutine)

S1. Compute all the wire-segment weights in a bottom-up manner using the formula: $\mu_i = \sum_{w_j\in Child(w_i)}\lambda_j$.
S2. Compute the downstream capacitance in a bottom-up manner using the formula $C_i = \sum_{w_j\in Child(w_i)}(C_j + c_j)$.
S3. Traverse the clock tree in the dept-first-search order; During visiting w_i, keeping other wire and buffer sizes fixed, compute $R_i' = R_{parent_i} + \mu_{parent_i} r_{parent_i}$, $C_i' = \mu_i C_i$, and $x_i = min\left(U_i, max\left(L_i, \sqrt{\frac{P_i C_i'}{\beta P_i + \gamma + \varepsilon_i R_i'}}\right)\right)$
S4. Repeat **S2**–**S3** until no improvement.

FIGURE 4 The optimal wire- and buffer-sizing algorithm.

THEOREM 2 Algorithm OWBA converges to a global optimal solution.

SKEW AND SENSITIVITY MINIMIZATION

By definition, clock skew $S = \max_{i,j}|D_i - D_j|$. To reduce clock skew, we need not only to reduce signal delays but also to balance delays. We have the following formulation to minimize clock skew:

$$\begin{aligned} \mathscr{M}1: \text{ Minimize } \quad & \alpha D_{\max} + \beta P + \gamma A + \delta(D_{\max} - D_{\min}) \\ \text{Subject to } \quad & D_i(\mathbf{x}) \le D_{\max}, \quad 1 \le i \le s, \\ & D_i(\mathbf{x}) \ge D_{\min}, \quad 1 \le i \le s, \\ & L_i \le x_i \le U_i, \quad 0 \le i \le n+m, \\ & D_{\max} > 0, \quad D_{\min} > 0. \end{aligned}$$

Since $\mathscr{M}1$ introduces negative coefficients, it is no longer a geometric programming problem and hence there is no guarantee of convexity. For a non-convex problem, global optimal solution may not be found easily. We resort to the following heuristic approach. Following the Lagrangian relaxation procedure, we relax the delay constraints by bringing them into the objective function with associated Lagrange multipliers λ_i's and σ_i's, $1 \le i \le s$, where λ_i and σ_i are the Lagrange multipliers associated with the delay constraint $D_i(\mathbf{x}) \le D_{\max}$ and $D_i(\mathbf{x}) \ge D_{\min}$, respectively. We have the Lagrangian relaxation subproblem for $\mathscr{M}1$ as follows:

$$\begin{aligned} \mathscr{M}1': \text{ Minimize } \quad & \alpha D_{\max} + \beta P + \gamma A + \delta(D_{\max} - D_{\min}) \\ & + \textstyle\sum_{i=1}^{s} \lambda_i(D_i(\mathbf{x}) - D_{\max}) \\ & + \textstyle\sum_{i=1}^{s} \sigma_i(D_{\min} - D_i(\mathbf{x})) \\ \text{Subject to } \quad & L_i \le x_i \le U_i, \quad 0 \le i \le n+m, \\ & D_{\max} > 0, \quad D_{\min} > 0. \end{aligned}$$

Hence, by repeatedly solving the Lagrangian relaxation subproblems, we can minimize clock skew.

Sensitivity is used to measure the influence of production variations. It can be measured by the first derivative of the signal delay with respect to wire (buffer) size which can be shown to be $|\varepsilon_i R_i - (\rho_i l_i C_i/x_i^2)|$ $(|\epsilon_i R_i - (\rho_i C_i/x_i^2)|)$. Restricting the sensitivity of every wire (buffer) to be smaller than $\Delta_{\max}$, we get $|\epsilon_i R_i - (\rho_i l_i C_i/x_i^2)| \le \Delta_{\max}$ $(|\epsilon_i R_i - (\rho_i C_i/x_i^2)| \le \Delta_{\max})$. In our algorithm, we dynamically add the following constraints into Step **S3** of LRS during execution:

TABLE I Experimental results in delay, skew and sensitivity

Ckt	# Nodes	Delay (ns) Initial	Final	Red%	Skew (ps) Initial	Final	Red%	Δ_{max} (10^{-15} sec/μmm) Initial	Final	Red%	Runtime (sec)	Stor (kb)	Err (ps)
r1	533	0.775	0.161	481	64	16	400	7.96	0.53	1501	3.50	148	0.2
r2	1195	2.108	0.379	556	221	12	1842	15.86	0.65	2436	13.38	280	0.4
r3	1723	3.376	0.572	590	154	36	427	20.58	0.68	3039	17.25	388	0.6
r4	3805	9.087	1.376	660	716	92	778	42.13	1.48	2850	54.87	812	1.4
r5	6201	15.864	2.312	686	974	102	955	63.51	2.06	3085	67.04	1300	2.3
Avg	–	–	–	595	–	–	691	–	–	2582	–	–	–

- For $1 \leq i \leq n$

$$x_i \leq \min\left(U_i,\ \max\left(L_i, \sqrt{\frac{\rho_i l_i C_i}{\epsilon_i R_i - \Delta_{max}}}\right)\right),$$

$$\text{if}\ \ \epsilon_i R_i - \frac{\rho_i l_i C_i}{x_i^2} \geq 0,$$

$$x_i \geq \min\left(U_i,\ \max\left(L_i, \sqrt{\frac{\rho_i l_i C_i}{\epsilon_i R_i + \Delta_{max}}}\right)\right),$$

$$\text{if}\ \ \epsilon_i R_i - \frac{\rho_i l_i C_i}{x_i^2} < 0.$$

- For $i = 0$ or $n + 1 \leq i \leq n + m$

$$x_i \leq \min\left(U_i,\ \max\left(L_i, \sqrt{\frac{\rho_i C_i}{\epsilon_i R_i - \Delta_{max}}}\right)\right),$$

$$\text{if}\ \ \epsilon_i R_i - \frac{\rho_i C_i}{x_i^2} \geq 0,$$

$$x_i \geq \min\left(U_i,\ \max\left(L_i, \sqrt{\frac{\rho_i C_i}{\epsilon_i R_i + \Delta_{max}}}\right)\right),$$

$$\text{if}\ \ \epsilon_i R_i - \frac{\rho_i C_i}{x_i^2} < 0.$$

While the above approaches reduce skew and sensitivity, they also tend to increase delay, power, area and runtime at the same time. In fact, we observe that Algorithm OWBA already significantly reduces skew and sensitivity while optimizing delay, power and/or area. Since Algorithm OWBA tends to allocate higher weights to sinks with longer delay and smaller weights to the ones with shorter delay. Consequently, the longer paths get more resources than the shorter ones. This effect directly balances the delays between different sinks and hence reduces clock skew. We observed that OWBA is a good heuristic for sensitivity minimization as well. To see this, let us consider delay minimization (i.e. $\alpha = 1$, $\beta = \gamma = 0$). Our algorithm essentially iteratively sizes all buffers and wire segments, one at a time (in Step S3 of LRS) while keeping the sizes of all other buffers/wire segments fixed. It can be proved that S3 not only optimally size a buffer/wire segment, it also simultaneously minimizes the sensitivity with respect to average delay.

EXPERIMENTAL RESULTS

We implemented our algorithm and tested on the five circuits $r1-r5$ used in Ref. [16] on an IBM RS/6000 workstation. The per micron resistance and capacitance used are 3 mΩ and 0.02 fF, respectively. The lower and upper bounds for wire widths are 1 and 10 μm, respectively. Table I lists the names of the circuits, numbers of wire segments in the circuits, delays, skews, sensitivity, runtimes and storage requirements. It shows that our algorithm, on the average, reduced

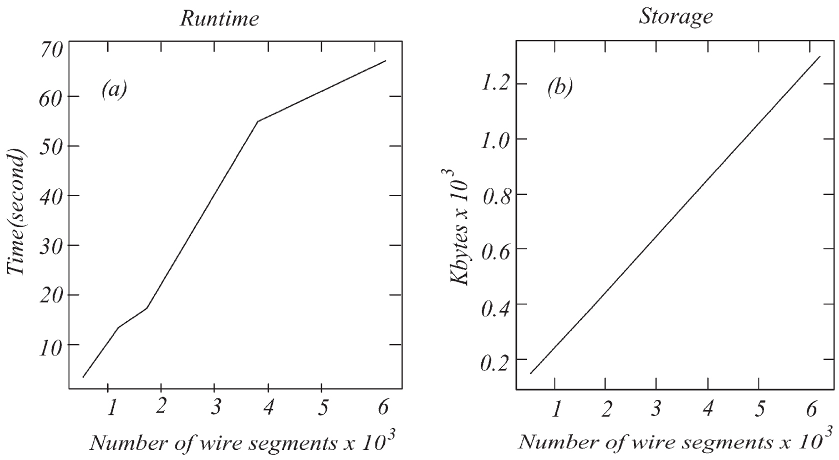

FIGURE 5 Runtime and storage requirements.

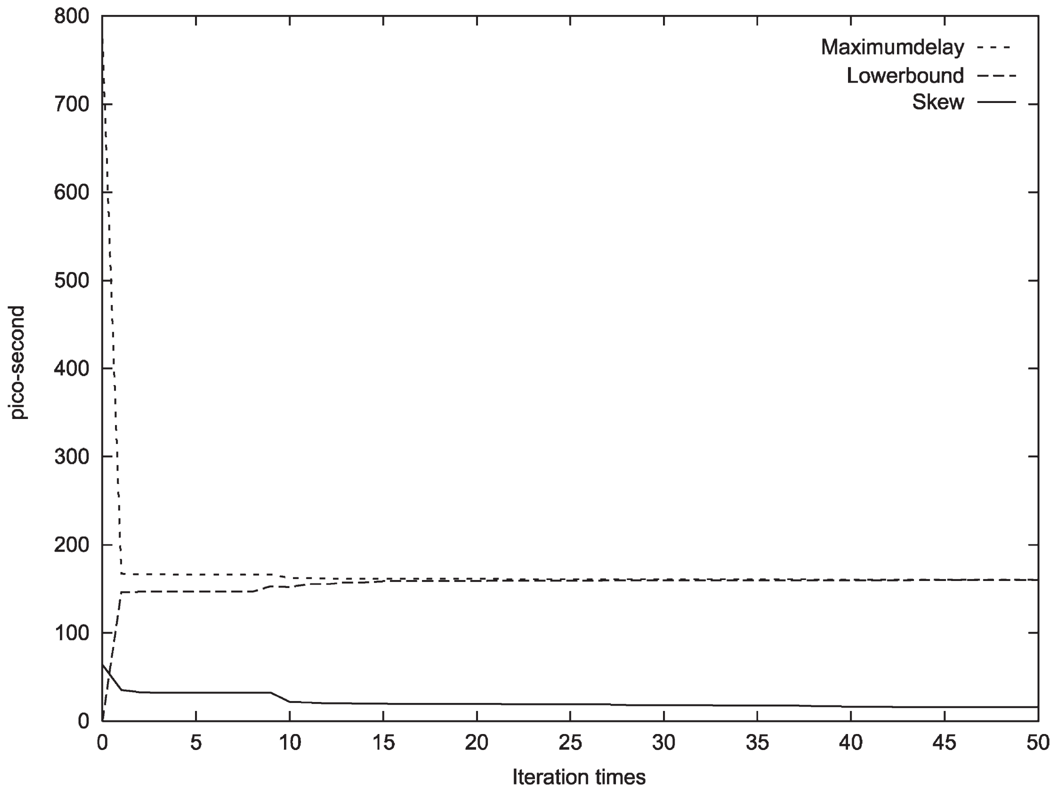

FIGURE 6 The values of D_{max} (upper bound), $\mathscr{L}(\Lambda)$ (lower bound) and clock skew during execution on r1.

the respective delay, skew and sensitivity by 595, 691 and 2582% after wire-sizing. Further, our algorithm is extremely fast and economical. For example, for the circuit r5 with 6201 wire segments, our algorithm needed only 67-second runtime and 1.3-MB storage to achieve 2.3-ps precision. In Fig. 5(a),(b), the runtime and storage requirements, respectively (represented by the vertical axis), are plotted as a function of the number of wire segments in a circuit (denoted by the horizontal axis). It shows that the runtime and storage requirements of our algorithm approach linear in the number of wire segments. Figure 6 shows the relationship among the maximum delays (D_{max}), the value of the $\mathscr{L}(\lambda)$ and clock skew at each iteration. The horizontal axis and the vertical axis represent the number of iterations and D_{max}, $L(\lambda)$, and skew (in pico second), respectively. The gap between D_{max} and $\mathscr{L}(\lambda)$ is the error bounds of our algorithm.

Acknowledgements

The authors thank Prof. Leon S. Lasdon, Prof. Patrick Jaillet, Prof. Ross Baldick, Prof. Jonathan Bard and Prof. Jayant Rajgopal for their invaluable help and comments.

References

[1] Lasdon, Leon S. (1970) Optimization Theory for Large Systems (Macmillan Publishing Co., Inc., New York).

[2] Bakoglu, H. (1990) Circuits, Interconnections, and Packaging for VLSI (Addison-Wesley Publishing Company Inc., Reading, MA).

[3] Chung-Ping, Chen and Wong, D.F. (1996) "A fast algorithm for optimal wire-sizing under Elmore delay model", *Proc IEEE ISCAS*.

[4] Chung, J. and Cheng, C.-K. (1994) "Skew sensitivity minimization of buffered clock tree", *Proc. ICCAD*, 280–283.

[5] Cong, J. and Leung, K.-S. (1995) "Optimal wiresizing under Elmore delay model", *IEEE TCAD* **14**(3), 321–336.

[6] Duffin, R.J., Peterson, E.L. and Zener, C. (1967) Geometric Programming—Theory and Application (John Wiley and Sons, Inc., New York).

[7] Dobberpuhl, D. and Witek, R. (1992) "A 200 MHz 64B dual-issue CMOS microprocessor", *Proc. IEEE ISSCC*, 106–107.

[8] Elmore, W.C. (1948) "The transient response of damped linear networks with particular regard to wide band amplifiers", *J. Appl. Phys.* **19**(1).

[9] Fisher, M.L. (1985) "An applications oriented guide to Lagrangian relaxation", *Interfaces* **15**(2), 10–21.

[10] Jackson, M.A.B., et al. (1990) "Clock routing for high performance ICs", *Proc. DAC*, 573–579.

[11] Luenberger, D.G. (1984) Linear and Nonlinear Programming (Addison-Wesley Pub. Company Inc., Reading, MA).

[12] Menezes, N., Baldick, R. and Pillage, L.T. (1995) "A sequential quadratic programming approach to concurrent gate and wire-sizing", *Proc. ICCAD*.

[13] Menezes, N., Pullela, S., Dartu, F. and Pillage, L.T. (1994) "RC interconnect syntheses—a moment fitting approach", *Proc. ICCAD*.

[14] Pullela, S., Menezes, N. and Pillage, L.T. (1993) "Reliable non-zero skew clock trees using wire width optimization", *Proc. 30th ACM/IEEE Design Automation Conf.*, 165–170.

[15] Sapatnekar, S.S. (1994) "RC interconnect optimization under the Elmore delay model", *Proc. ICCAD*, 387–391.

[16] Tasy, R.-S. (1993) "Exact zero skew", *IEEE TCAD*.

[17] Zhu, Q., Dai, W.W.-M. and Xi, J.G. (1993) "Optimal sizing of high-speed clock networks based on distributed RC and lossy transmission line models", *Proc. ICCAD*, 628–633.

Yu-Min Lee was born in Taiwan in 1969. He received the B.S. and M.S. degrees in communication engineering from the National Chiao-Tung University, Taiwan, in 1991

and 1993, and the M.S. degree in electrical and computer engineering from the University of Wisconsin at Madison in 2000. He is currently working towards the Ph.D. degree at the University of Wisconsin at Madison. His research interests include electrical design automation, low-power and high-performance circuit design and signal integrity analysis and optimization.

Charlie Chung-Ping Chen received his B.S. degree in computer science and information engineering from the National Chiao-Tung University, Hsinchu, Taiwan, in 1990. He received his M.S. and Ph.D. degrees in computer science from the University of Texas at Austin in 1996 and 1998, respectively. Between 1997–1999 he was with the Intel Corporation as a senior CAD engineer with Strategic CAD Labs. He actively participated in several high-speed interconnect optimization and circuit synthesis projects. He received the D2000 Award from Intel Corp. in 1999. Currently, he is an assistant professor in the Electrical and Computer Engineer Department at the University of Wisconsin, Madison. His research interests are in the areas of computer-aided design and microprocessor circuit design with an emphasis on interconnect and circuit optimization as well as signal integrity analysis and optimization. Dr Chen received Faculty Early Career Development Award (CAREER) from National Science Foundation in 2001 for his work on VLSI interconnect modeling, simulation and optimization.

Yao-Wen Chang received his B.S. degree in computer science and information engineering from the National Taiwan University, Taipei, Taiwan, in 1988, and the M.S. and the Ph.D. degrees in the computer science from the University of Texas at Austin in 1993 and 1996, respectively. He was with IBM T.J. Watson Research Center, Yorktown Heights, NY, in the VLSI group, during the summer of 1994. Currently, he is an Associate Professor in the Department of Electrical Engineering at the National Taiwan University, Taipei, Taiwan. His research interests lie in design automation, architecture and systems for VLSI and combinatorial optimization. Dr Chang received the Best Paper Award of the CAD tract at the 1995 IEEE International Conference on Computer Design (ICCD-95) for his work on FPGA routing. He is a member of IEEE Circuits and System Society, ACM, and ACM/SIGDA.

D.F. Wong received the B.Sc. degree in mathematics from the University of Toronto, Canada, and the M.S. degree in mathematics and the Ph.D. degree in computer science from the University of Illinois at Urbana-Champaign. He is currently a Professor of the computer science department at the University of Texas at Austin. His main research interest is computer-aided-design (CAD) of very-large-scale integration (VLSI). He has published more than 140 technical papers in this area. He is a coauthor of Simulated Annealing for VLSI Design (Norwell, MA: Kluwer, 1988). He is the Technical Program Chair of the 1998 ACM International Symposium on Physical Design (ISPD-98). He has also served on the Technical Program Committees of a number of other VLSI CAD conferences (e.g. ICCAD, ED&TC, ISCAS, FPGA). Dr. Wong received the Best Paper Awards at DAC-86 and ICCD-95 for his work on floorplan design and field programmable gate array routing, respectively. He is an Editor of IEEE Transactions on Computers.