Simultaneous Driver and Wire Sizing for Performance and Power Optimization

Jason Cong, *Member, IEEE,* and Cheng-Kok Koh

***Abstract*— In this paper, we study the *simultaneous driver and wire sizing (SDWS) problem* under two objective functions: i) delay minimization only, or ii) combined delay and power dissipation minimization. We present general formulations of the SDWS problem under these two objectives based on the distributed Elmore delay model with consideration of both capacitive power dissipation and short-circuit power dissipation. We show several interesting properties of the optimal SDWS solutions under the two objectives, including an important result (Theorem 5) which reveals the relationship between driver sizing and optimal wire sizing. These results lead to polynomial time algorithms for computing the lower and upper bounds of optimal SDWS solutions under the two objectives, and efficient algorithms for computing optimal SDWS solutions under the two objectives. We have implemented these algorithms and compared them with existing design methods for driver sizing only or independent driver and wire sizing. Accurate SPICE simulation shows that our methods reduce the delay by up to 12%–49% and power dissipation by 26%–63% compared with existing design methods.**

I. Introduction

DELAY MINIMIZATION and power dissipation minimization are two important objectives in the design of the high-performance, portable, and wireless computing and communication systems. We believe that both device design (i.e., transistor/cell design) and interconnect design have to be considered and optimized simultaneously in order to achieve these two objectives. This new design methodology is especially important as the VLSI fabrication technology advances to submicron device dimension and gigahertz clock frequency. In this case, interconnect delay becomes an important factor in determining circuit speed, the *distributed* nature of the interconnect structure must be considered, and the capacitive power dissipation on charging/discharging interconnect structures takes a significant portion of overall system power. The objective of this paper is to study the simultaneous driver and wire sizing problem for both delay and power optimization.

In the past, two methods are commonly used to improve the performance of long interconnect lines. One method is driver sizing, which uses a large driver or a series of cascaded drivers of increasing sizes to drive long interconnect lines [2]. Another method is to break long interconnect lines into shorter segments by inserting repeaters. These repeaters can also be sized properly for further reduction in interconnect delay [2]. Both methods are effective for interconnect delay reduction but with substantial increase in power consumption.

Recent studies show that interconnect delay can also be reduced by interconnect topology optimization and wiresizing optimization. A number of interconnect topologies have been proposed for interconnect performance optimization, including bounded-radius bounded-cost trees [7], AHHK trees [1], maximum performance trees [6], A-trees [11], low-delay trees [3], [4], and IDW/CFD trees [16]. Wiresizing optimization was first proposed in [11], [12] to further minimize interconnect delay by optimally assigning different wire width to each wire segment in routing trees and substantial delay reduction was achieved for submicron IC design and MCM design. Follow-up work on wiresizing includes the use of convex programming technique [20] and interleave of wiresizing with routing tree construction [15]. Both interconnect topology optimization and wiresizing optimization are effective when resistance ratio, i.e., the driver resistance versus unit wire resistance, is small in the design [11].

Very recently, Cong, Koh, and Leung [8] explored the possibility of both driver sizing and wire sizing. A simple heuristic algorithm was used to size drivers according to a fixed constant ratio, and an *independent* wire sizing optimization is performed for each driver sizing solution. Although encouraging experimental results were reported, it is not difficult to see that this method often produces suboptimal solutions since diver sizing and wire sizing were carried out independently.

In this paper, we study the *simultaneous driver and wire sizing (SDWS) problem* under two objective functions: i) delay minimization only, or ii) combined delay and power dissipation minimization. We present general formulations of the SDWS problem under these two objectives based on the distributed Elmore delay model with consideration of both capacitive power dissipation and short-circuit power dissipation. We show several interesting properties of the optimal SDWS solutions under the two objectives, including an important result (Theorem 5) which reveals the relationship between driver sizing and optimal wire sizing. These results lead to polynomial time algorithms for computing the lower and upper bounds of the optimal SDWS solution under the two objectives, and efficient algorithms for computing optimal SDWS solutions under the two objectives. We have implemented these algorithms and compared them with existing design methods for driver sizing only or independent driver and wire sizing. Accurate SPICE simulation shows that our methods reduce the delay by up to 12%–49% and power dissipation

---

Manuscript received June 15, 1994; revised August 23, 1994. This work was supported by ARPA/CSTO under Contract J-FB1-93-112, the National Science Foundation Young Investigator Award MIP-9357582, and a Grant from Intel Corporation.

The authors are with the Department of Computer Science, University of California, Los Angeles, CA 90024 USA.

IEEE Log Number 9406370.

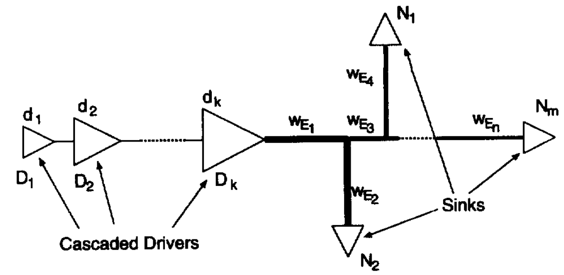


Fig. 1. A k-stage cascaded drivers driving an interconnect tree T with sinks $\{N_1, N_2, \ldots, N_m\}$. w_{E_i} denotes the width of the wire segment E_i, $i = 1 \cdots n$ and d_i denotes the size of driver D_i at ith-stage, $i = 1 \cdots k$.

by 26%–63% compared with existing design methods. To the best of our knowledge, this is the first work which presents in-depth study of the SDWS problem for both delay and power optimization.

The remainder of this paper is organized as follows. In Section II, we present the general formulation of the simultaneous driver sizing and wiresizing problems under the delay minimization objective and the combined delay and power minimization objective. In Section III, we present results on optimal SDWS solutions for delay minimization. In Section IV, we present results on optimal SDWS solutions under the combined delay and power minimization objective. Section V shows the experimental results obtained by our SDWS algorithms and the comparative study with other existing methods. Section VI concludes the paper with discussions of future work. An extended abstract of this paper was presented in ICCAD'94 [10].

II. PROBLEM FORMULATION

A. Performance Optimization

Assume that we are given a routing tree T implementating a signal net which consists of a source N_+, and a set of m sinks $\{N_1, N_2, \ldots, N_m\}$. A node in T refers to the source, or a sink, or a Steiner node, and a segment connects two nodes in T. Assume that $\{E_1, E_2, \ldots, E_n\}$ is the set of segments forming the tree T, where n is the total number of segments in the tree. Each wire segment has a set of discrete choices of wire widths $\{W_1, W_2, \ldots, W_r\}$ $(W_1 < W_2 < \cdots < W_r)$. We use w_{E_i} to denote the width of the wire segment E_i, $i = 1 \cdots n$.

Furthermore, we assume that the signal net is driven by a chain of cascaded drivers of k stages at the source as shown in Fig. 1. We use $\mathcal{D} = \{d_1, d_2, \ldots, d_k\}$ to denote a driver sizing solution, where d_i denotes the size of driver D_i at i-th stage $(i = 1 \cdots k)$. We assume that driver D_1 is of minimum size, i.e., $d_1 = 1$ (after normalization). Given the above definitions, the problem of simultaneous driver and wire sizing (SDWS) for performance optimization can be defined as follows:

Formulation 1: Given a routing tree T, the SDWS problem for delay minimization (SDWS-D) is to determine the number of stages k, a driver sizing solution $\mathcal{D}$, and a wiresizing solution $\mathcal{W}$ on T, such that the performance measure t $(T, k, \mathcal{D}, \mathcal{W})$ is optimized.

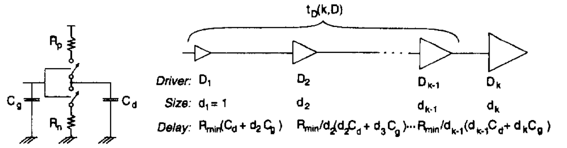


Fig. 2. (a) A switch-level RC model of a driver (b) Inter-stage delay of a k-stage cascaded drivers until the gate of the k-th driver.

If we fix the number of stages k, a restricted version of the SDWS-D problem called the k-SDWS-D problem can be defined as follows: Given a routing tree T and a chain of k drivers, the k-SDWS-D problem is to determine the optimal driver and wire sizing solution $\mathcal{D}$ and $\mathcal{W}$, such that the performance measure t $(T, k, \mathcal{D}, \mathcal{W})$ is optimized.

The performance measure t $(T, k, \mathcal{D}, \mathcal{W})$ approximates the delay of the signal net from the source to one or several critical sinks, and it can be decomposed as follows:

$$t(T, k, \mathcal{D}, \mathcal{W}) = t_D(k, \mathcal{D}) + t_I(T, k, \mathcal{D}, \mathcal{W}) \tag{1}$$

where the first term $t_D(k, \mathcal{D})$ measures the delay due to the drivers and the second term $t_I(T, k, \mathcal{D}, \mathcal{W})$ measures the interconnect delay. We estimate $t_D(k, \mathcal{D})$ based on a switch-level RC model of drivers. The interconnect delay $t_I(T, k, \mathcal{D}, \mathcal{W})$ is measured by the distributed Elmore delay model [14].

RC Delay Model for Drivers: We estimate the delay of a driver based on a switch-level RC model of driver. Fig. 2(a) shows a minimum-size inverter (driver) with a p-transistor resistance R_p and a n-transistor resistance R_n. We assume that $R_p = R_n = R_{\min}$. Let C_g denote the gate capacitance and C_d denote the capacitance due to the source and drain diffusion of the minimum-size driver.

Fig. 2(b) illustrates the delay due to a sequence of k cascaded drivers $\mathcal{D}$. Ignoring the interconnect resistance and capacitance between drivers (since cascaded drivers are placed closed to each other), the delay from driver D_i to D_{i+1} $(1 \leq i \leq k-1)$ is the product of the resistance of D_i and its capacitive load:

$$\frac{R_{\min}}{d_i} \cdot (C_d \cdot d_i + C_g \cdot d_{i+1}) = R_{\min} \cdot C_d + R_{\min} \cdot C_g \cdot \frac{d_{i+1}}{d_i} \tag{2}$$

The total delay up to gate of the last driver $t_D(k, \mathcal{D})$ (excluding the last driver) can be expressed as follows:

$$\begin{aligned} t_D(k, \mathcal{D}) &= (k-1) \cdot R_{\min} \cdot C_d + \sum_{i=1}^{k-1} \frac{R_{\min} \cdot C_g \cdot d_{i+1}}{d_i} \\ &= \mathcal{J}_1 + \mathcal{J}_2 \cdot \sum_{i=1}^{k-1} \frac{d_{i+1}}{d_i} \end{aligned} \tag{3}$$

where $\mathcal{J}_1 = (k-1) \cdot R_{\min} \cdot C_d$, and $\mathcal{J}_2 = R_{\min} \cdot C_g$. Notice that delay through the k-th driver will be counted as part of interconnect delay in Section II-A-3.

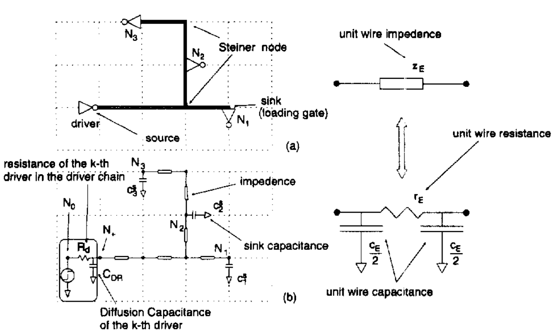


Fig. 3. A uniform grid structure for the distributed *RC* interconnect model. (a) The layout of an interconnect tree $T$with three sinks N_1, N_2 and N_3. (b) The corresponding distributed *RC* interconnect model of T. Each grid edge E in T connecting two adjacent nodes is modeled as a π-type *RC* circuit containing a resistor of r_E and two capacitors of $\frac{c_E}{2}$ each. Each sink has an extra loading capacitance.

Distributed Elmore Delay Model for Interconnect: We use the distributed Elmore delay model [14] for interconnect delay measure. The formulations used in this section are based on those in [12]. In order to model a routing tree as a distributed *RC* circuit accurately, a uniform grid structure is superimposed on the routing plane, and each wire segment in the routing plane is divided into a sequence of wires of unit length as shown in Fig. 3. In this case, T consists of a set of unit-length wire segments, each may have a different width[1].

For each edge E in the tree T, we use a π-type *RC* circuit to model E, where r_E and c_E are interconnect resistance and capacitance, respectively. Given an edge E, we use w_E to denote the width of the grid edge E. Assume that a unit-width unti-grid-length wire has wire resistance r_0, wire area capacitance c_0 and wire fringing capacitance c_1, then r_E and c_E for any grid edge E can be defined as follows:

$$\begin{aligned} r_E &= \frac{r_0}{w_E} \\ c_E &= c_0 \cdot w_E + c_1 \end{aligned} \tag{4}$$

Note that previous formulations in [11] and [12] did not consider fringing effect. Also, c_1 can be used to represent mutual capacitance between adjacent wires in the same layer or adjacent layers, when it is known.

We use c_u^s to denote the capacitance at sink u. To correctly model the driver resistance, we introduce an additional node N_0 and connect N_0 to N_+ via an additional segment with resistance $R_d = R_{\min}/d_k$ and capacitance $C_{\text{DR}} = C_d \cdot d_k$ (the resistance and capacitance of the last driver in the cascaded driver chain) in the later discussions.

Given a grid edge E, we use $\text{Des}(E)$ to denote the set of grid edges in the subtree rooted at E (excluding E), and $\text{Ans}(E)$ to denote the set of grid edges $\{E' \mid E \in \text{Des}(E')\}$ (again, excluding E). That is, $\text{Des}(E)$ is the set of "descendant" grid edges of E, and $\text{Ans}(E)$ is the set of "ancestor" grid edges of E. Also, we use $\text{sink}(E)$ to denote the set of sinks in the subtree rooted at E, and C_E to denote the *total* capacitance in the subtree rooted at E (including both the wire capacitances and the sink capacitances):

$$C_E = \sum_{u \in \text{sink}(E)} c_u^s + c_0 \cdot \sum_{E' \in \text{Des}(E)} w_{E'} + \sum_{E' \in \text{Des}(E)} c_1 \tag{5}$$

Furthermore, we use $\text{sink}(T)$ to denote the set of sinks in T and $P(u, v)$ to denote the unique path from u to v for any grid points u, v in the routing tree.

We use the Elmore delay model [14] as the objective function for delay optimization. Given a distributed *RC* circuit tree T, the signal delay at a particular node N_i, denoted as $t(N_i)$, is computed as follows:

$$t(N_i) = \sum_{E \in P(N_0, N_i)} r_E \cdot \left(\frac{c_E}{2} + C_E\right) \tag{6}$$

where the summation is taken over all the grid edges on the path from the driver N_0 to the node N_i.

Single Critical Sink Formulation: We first study the case where there is only one critical sink N_i in the net. Let R_d and C_{DR} be the resistance and diffusion capacitance of the last driver in the driver chain, respectively, i.e., $R_d = R_{\min}/d_k$, $C_{\text{DR}} = C_d \cdot d_k$. Also, we use $C_{\text{IL}}(T, \mathcal{W})$ to denote total capacitance of interconnect tree T (including the loading capacitance at the sinks). We define

$$\begin{aligned} C_{\text{IL}}(T, \mathcal{W}) &= \sum_{u \in \text{sink}(T)} c_u^s + c_0 \cdot \sum_{E \in T} w_E + \sum_{E \in T} c_1 \\ &= \sum_{u \in \text{sink}(T)} c_u^s + c_0 \cdot \sum_{E \in T} w_E + n \cdot c_1 \end{aligned} \tag{7}$$

which is the total capacitance that the driver chain will drive. According to (6) and following a similar derivation[2] in [11] and [12], the signal delay $t_i(T, R_d, \mathcal{W})$ at N_i under a given wiresizing solution $\mathcal{W}$ is:

$$\begin{aligned} & t_i(T, R_d, \mathcal{W}) \\ &= R_d \cdot (C_{\text{DR}} + C_{\text{IL}}(T, \mathcal{W})) \\ &\quad + \sum_{E \in P(N_+, N_i)} r_E \cdot \left(\frac{c_E}{2} + C_E\right) \\ &= R_d \cdot C_{\text{DR}} + R_d \cdot \sum_{u \in \text{sink}(T)} c_u^s \\ &\quad + R_d \cdot c_0 \cdot \sum_{E \in T} w_E + R_d \cdot n \cdot c_1 \\ &\quad + \sum_{E \in P(N_+, N_i)} \frac{r_0 \cdot c_0}{2} + \frac{r_0 \cdot c_1}{2} \cdot \sum_{E \in P(N_+, N_i)} \frac{1}{w_E} \\ &\quad + r_0 \cdot c_0 \cdot \sum_{E \in P(N_+, N_i)} \sum_{E' \in \text{Des}(E)} \frac{w_{E'}}{w_E} \\ &\quad + r_0 \cdot c_1 \cdot \sum_{E \in P(N_+, N_i)} \sum_{E' \in \text{Des}(E)} \frac{1}{w_E} \\ &\quad + r_0 \cdot \sum_{E \in P(N_+, N_i)} \sum_{v \in \text{sink}(E)} c_v^s \cdot \frac{1}{w_E} \end{aligned} \tag{8}$$

---

[1] In [11] and [12], a segment in T is defined to be an edge in the rectilinear Steiner tree. In that case, segments in T are of different lengths, and the wire width is uniform within each segment.

[2] In this paper, detailed derivation of (8), (12), (23), and (24) are omitted. The details can be found in the technical report [9].

Let

$$\begin{aligned}
\mathcal{K}_0^i &= R_d \cdot C_{\mathrm{DR}} + R_d \cdot \sum_{u \in \mathrm{sink}(T)} c_u^s + R_d \cdot n \cdot c_1 \\
&\quad + \sum_{E \in P(N_+, N_i)} \frac{r_0 \cdot c_0}{2} \\
\mathcal{K}_1 &= R_d \cdot c_0 \\
\mathcal{K}_2 &= r_0 \cdot c_0 \\
\mathcal{K}_3 &= r_0 \cdot c_1 \\
\mathcal{K}_4 &= r_0 \\
\mathcal{K}_5 &= \frac{r_0 \cdot c_1}{2}
\end{aligned} \tag{9}$$

Note that given a fixed $R_d, \mathcal{K}_0^i$ is a constant with respect to the timing-critical sink N_i, and $\mathcal{K}_1$ to $\mathcal{K}_5$ are all constants depending only on the technology. Also, we define the functions $f_i(E, E'), g_i(E)$ and $h_i(E)$ as follows:

$$f_i(E, E') = \begin{cases} 1 & \text{if } E \in P(N_+, N_i) \text{ and } E' \in \mathrm{Des}(E) \\ 0 & \text{otherwise} \end{cases}$$

$$g_i(E) = \begin{cases} \sum_{v \in \mathrm{sink}(E)} c_v^s & \text{if } E \in P(N_+, N_i) \\ 0 & \text{otherwise} \end{cases}$$

$$h_i(E) = \begin{cases} 1 & \text{if } E \in P(N_+, N_i) \\ 0 & \text{otherwise} \end{cases}$$

Rearranging, we can rewrite (8) as follows:

$$\begin{aligned}
t_i(T, R_d, \mathcal{W}) = \mathcal{K}_0^i &+ \mathcal{K}_1 \cdot \sum_{E \in T} w_E \\
&+ \mathcal{K}_2 \cdot \sum_{E, E' \in T} f_i(E, E') \cdot \frac{w_{E'}}{w_E} \\
&+ \mathcal{K}_3 \cdot \sum_{E, E' \in T} f_i(E, E') \cdot \frac{1}{w_E} \\
&+ \mathcal{K}_4 \cdot \sum_{E} g_i(E) \cdot \frac{1}{w_E} \\
&+ \mathcal{K}_5 \cdot \sum_{E \in T} h_i(E) \cdot \frac{1}{w_E}
\end{aligned} \tag{10}$$

Multiple Critical Sinks Formulation: Let $\mathrm{sink}(T)$ denote the set of sinks in T. When there are several critical sinks of different priorities in the routing tree, the previous formulation can be generalized as follows:

$$t_i(T, R_d, \mathcal{W}) = \sum_{N_i \in \mathrm{sink}(T)} \lambda_i \cdot t_i(T, R_d, \mathcal{W}) \tag{11}$$

where λ_i is the weight of the delay penalty to sink N_i [3], [12]. The larger λ_i is, the more critical sink N_i is. We normalize λ_i's such that $\sum_{N_i \in \mathrm{sink}(T)} \lambda_i = 1$. We can rewrite (11) as follows:

$$\begin{aligned}
t(T, R_d, \mathcal{W}) = \mathcal{K}_0 &+ \mathcal{K}_1 \cdot \sum_{E \in T} w_E \\
&+ \mathcal{K}_2 \cdot \sum_{E, E' \in T} F(E, E') \cdot \frac{w_{E'}}{w_E} \\
&+ \mathcal{K}_3 \cdot \sum_{E, E' \in T} F(E, E') \cdot \frac{1}{w_E} \\
&+ \mathcal{K}_4 \cdot \sum_{E \in T} G(E) \cdot \frac{1}{w_E} \\
&+ \mathcal{K}_5 \cdot \sum_{E \in T} H(E) \cdot \frac{1}{w_E}
\end{aligned} \tag{12}$$

where

$$\begin{aligned}
\mathcal{K}_0 &= \sum_{N_i \in \mathrm{sink}(T)} \lambda_i \cdot \mathcal{K}_0^i \\
F(E, E') &= \sum_{N_i \in \mathrm{sink}(T)} \lambda_i \cdot f_i(E, E') \\
G(E) &= \sum_{N_i \in \mathrm{sink}(T)} \lambda_i \cdot g_i(E) \\
H(E) &= \sum_{N_i \in \mathrm{sink}(T)} \lambda_i \cdot h_i(E)
\end{aligned} \tag{13}$$

Using a similar argument as in [13], we have

$$\begin{array}{rll}
F(E_1, E_2) & \geq F(E_1, E_2') & \text{if } E_2 \in \mathrm{Des}(E_2') \\
F(E_1, E_2) & \geq F(E_1', E_2) & \text{if } E_1 \in \mathrm{Ans}(E_1') \\
G(E_1) & \geq G(E_1') & \text{if } E_1 \in \mathrm{Ans}(E_1') \\
H(E_1) & \geq H(E_1') & \text{if } E_1 \in \mathrm{Ans}(E_1') \\
H(E_1) & \geq F(E_1, E_2) & \text{if } E_2 \in \mathrm{Des}(E_1)
\end{array} \tag{14}$$

Performance Optimization Objective of the SDWS Problem: Let $t_I(T, k, \mathcal{D}, \mathcal{W}) = t(T, R_d, \mathcal{W})$. Hence, the performance measure on both driver and interconnect delay $t(T, k, \mathcal{D}, \mathcal{W})$ in (1) can be written as

$$\begin{aligned}
&t(T, k, \mathcal{D}, \mathcal{W}) \\
&\quad = t_D(k, \mathcal{D}) + t_I(T, k, \mathcal{D}, \mathcal{W}) \\
&\quad = \left\{ \mathcal{J}_1 + \mathcal{J}_2 \cdot \sum_{i=1}^{k-1} \frac{d_{i+1}}{d_i} \right\} + \left\{ \mathcal{K}_0 + \mathcal{K}_1 \cdot \sum_{E \in T} w_E \right. \\
&\qquad + \mathcal{K}_2 \cdot \sum_{E, E' \in T} F(E, E') \cdot \frac{w_{E'}}{w_E} + \mathcal{K}_3 \cdot \sum_{E, E' \in T} F(E, E') \\
&\qquad \left. \cdot \frac{1}{w_E} + \mathcal{K}_4 \cdot \sum_{E \in T} G(E) \cdot \frac{1}{w_E} + \mathcal{K}_5 \cdot \sum_{E \in T} H(E) \cdot \frac{1}{w_E} \right\}
\end{aligned} \tag{15}$$

where $\mathcal{K}_0, \mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3, \mathcal{K}_4$, and $\mathcal{K}_5$ are constants defined in (9) and (13).

B. Performance and Power Optimization

Driver and wire sizing are effective approaches to reduce interconnect delay. However, larger driver size and additional routing capacitance also increase the power dissipation. In practice, high-speed circuit design requires a careful trade-off between performance and power. We define the SDWS problem for both delay and power optimization (SDWS-DP) as follows:

Formulation 2: Given a routing tree T, the SDWS problem for both delay and power optimization (SDWS-DP) is to determine the number of stages k, a driver sizing solution $\mathcal{D}$, and a wiresizing solution $\mathcal{W}$ on T, such that the objective function $\mathrm{obj}(T, k, \mathcal{D}, \mathcal{W})$ defined below is minimized:

$$\begin{aligned}
\mathrm{obj}(T, k, \mathcal{D}, \mathcal{W}) &= \alpha \cdot \mathrm{Power}(T, k, \mathcal{D}, \mathcal{W}) \\
&\quad + \gamma \cdot t(T, k, \mathcal{D}, \mathcal{W})
\end{aligned} \tag{16}$$

where $t(T, k, \mathcal{D}, \mathcal{W})$ is the performance measure, Power $(T, k, \mathcal{D}, \mathcal{W})$ is the power dissipation, and α and γ are adjustable non-negative parameters controlling the trade-off between performance and power dissipation.

Parameters α and γ can either be given by designer or higher level synthesis program. Similarly, if we fix the number of stages k, we can define the *k-SDWS-DP* problem as follows: Given a routing tree T and a chain of k drivers, determine the optimal driver and wire sizing solution $\mathcal{D}$ and $\mathcal{W}$, such that the combined objective function obj$(T, k, \mathcal{D}, \mathcal{W})$ is minimized.

Power Dissipation Formulation: There are two components that determine the amount of power dissipated in a CMOS circuit, namely *static* dissipation due to leakage current, and *dynamic* dissipation due to switching transient current (short-circuit dissipation) and charging and discharging of load capacitances (capacitive dissipation) [21]. We consider only dynamic dissipation in our formulation since static dissipation is usually 2 to 3 order of magnitude smaller. Given a loading capacitance C_L, we can write the capacitive and short-circuit dissipation of a single driver as follows [21]:

$$\text{Power}_{\text{cap}} = f \cdot C_L \cdot V_{dd}^2 \tag{17}$$

$$\text{Power}_{\text{sc}} = f \cdot \frac{\beta}{12} \cdot (V_{dd} - 2V_t)^3 \cdot t_{\text{rf}} \tag{18}$$

where f is the switching frequency of the input signal, β is the MOS transistor gain factor, V_t is the threshold voltage, and t_{rf} is the rise and fall time of the input signal which are assumed to be equal.

Consider a chain of k drivers. For driver $D_i (i = 1 \cdots k-1)$, the capacitive load of the driver is assumed to be the sum of its diffusion capacitance $C_d \cdot d_i$ and the gate capacitance $C_g \cdot d_{i+1}$ of the next driver D_{i+1}. The last driver D_k has a capacitive load of $C_d \cdot d_k + C_{\text{IL}}(T, \mathcal{W})$. Hence the capacitive power of the cascaded drivers is

$$\begin{aligned}
&\text{Power}_{\text{cap}}(k, \mathcal{D}, C_{\text{IL}}(T, \mathcal{W})) \\
&\quad = f \cdot V_{dd}^2 \cdot \left(C_d + \sum_{i=2}^{k-1} (d_i \cdot C_d + d_{i+1} \cdot C_g) \right. \\
&\qquad \left. + \; d_k \cdot C_d + C_{\text{IL}}(T, \mathcal{W}) \right) \\
&\quad = \mathcal{L}_0 + \mathcal{L}_1 \cdot \left(\sum_{i=2}^{k} d_i + \frac{C_{\text{IL}}(T, \mathcal{W})}{C_g + C_d} \right)
\end{aligned} \tag{19}$$

where $\mathcal{L}_0 = f \cdot V_{\text{dd}}^2 \cdot C_d$ and $\mathcal{L}_1 = f \cdot V_{\text{dd}}^2 \cdot (C_g + C_d)$.

Let $\beta_{\min}$ be the gain factor of a minimum-size driver. The gain factor of a driver of size d can be defined as $d \cdot \beta_{\min}$. Hence, we can write the short-circuit power of the cascaded drivers as follows [21]:

$$\begin{aligned}
\text{Power}_{\text{sc}}(k, \mathcal{D}, C_{\text{IL}}(T, \mathcal{W})) &= f \cdot \sum_{i=1}^{k} \frac{\beta_{\min} \cdot d_i}{12} \cdot (V_{dd} - 2V_t)^3 \cdot t_{\text{rf}} \\
&= \mathcal{L}_2 + \mathcal{L}_2 \cdot \sum_{i=1}^{k} d_i
\end{aligned} \tag{20}$$

where $\mathcal{L}_2 = f \cdot \frac{\beta_{\min}}{12} \cdot (V_{dd} - V_t)^3 \cdot t_{\text{rf}}$. Note that one limitation of this simplified formulation is that it assumes t_{rf} to be constant for all drivers along the chain. In reality, the rise and fall times of the transition at the input to the gate depend on the driving capability of the previous stage driver, the loading capacitance of the gate concerned and the interconnect between the drivers.

Summing up the capacitive and short-circuit power, the dynamic power dissipation of the circuit can be written as:

$$\begin{aligned}
\text{Power}(T, k, \mathcal{D}, \mathcal{W}) &= \text{Power}_{\text{cap}}(k, \mathcal{D}, C_{\text{IL}}(T, \mathcal{W})) \\
&\quad + \text{Power}_{\text{sc}}(k, \mathcal{D}, C_{\text{IL}}(T, \mathcal{W})) \\
&= \left\{ \mathcal{L}_0 + \mathcal{L}_1 \cdot \left(\sum_{i=2}^{k} d_i + \frac{C_{\text{IL}}(T, \mathcal{W})}{C_g + C_d} \right) \right\} \\
&\quad + \left\{ \mathcal{L}_2 + \mathcal{L}_2 \cdot \sum_{i=2}^{k} d_i \right\} \\
&= \mathcal{L}_0 + \mathcal{L}_1 \cdot \frac{C_{\text{IL}}(T, \mathcal{W})}{C_g + C_d} + \mathcal{L}_2 \\
&\quad + (\mathcal{L}_1 + \mathcal{L}_2) \cdot \sum_{i=2}^{k} d_i
\end{aligned} \tag{21}$$

Trade-Off Between Performance and Power: We can now write the trade-off between performance and power in (16) as follows:

$$\begin{aligned}
&\text{obj}(T, k, \mathcal{D}, \mathcal{W}) \\
&\quad = \alpha \cdot \text{Power}(T, k, \mathcal{D}, \mathcal{W})) + \gamma \cdot t(T, k, \mathcal{D}, \mathcal{W}) \\
&\quad = \left\{ \alpha \cdot \left(\mathcal{L}_0 + \mathcal{L}_1 \cdot \frac{C_{\text{IL}}(T, \mathcal{W})}{C_g + C_d} \right. \right. \\
&\qquad \left. \left. + \mathcal{L}_2 + (\mathcal{L}_1 + \mathcal{L}_2) \cdot \sum_{i=2}^{k} d_i \right) \right\} \\
&\quad + \left\{ \gamma \cdot \left(\mathcal{J}_1 + (\mathcal{J}_2 \cdot \sum_{i=1}^{k-1} \frac{d_{i+1}}{d_i} + \mathcal{K}_0 + \mathcal{K}_1 \cdot \sum_{E \in T} w_E \right. \right. \\
&\quad + \mathcal{K}_2 \cdot \sum_{E, E' \in T} F(E, E') \cdot \frac{w_{E'}}{w_E} \\
&\quad + \mathcal{K}_3 \cdot \sum_{E, E' \in T} F(E, E') \cdot \frac{1}{w_E} \\
&\quad \left. \left. + \mathcal{K}_4 \cdot \sum_{E \in T} G(E) \cdot \frac{1}{w_E} + \mathcal{K}_5 \cdot \sum_{E \in T} H(E) \cdot \frac{1}{w_E} \right) \right\}
\end{aligned} \tag{22}$$

There are two terms that links the driver chain and the interconnect in both the power formula and delay formula, namely the last driver d_k and the capacitive load $C_{\text{IL}}(T, \mathcal{W})$. Suppose a wire-width assignment $\mathcal{W}$ is given for a routing tree T, the capacitive load $C_{\text{IL}}(T, \mathcal{W})$ can be computed. Eliminating the constant terms, the drivers are sized to minimize the following:

$$\begin{aligned}
\text{obj}_{T,k,\mathcal{W}}(\mathcal{D}) &= \alpha \cdot (\mathcal{L}_1 + \mathcal{L}_2) \cdot \sum_{i=2}^{k} d_i \\
&\quad + \gamma \cdot \mathcal{J}_2 \cdot \left(\frac{C_{\text{IL}}(T, \mathcal{W})}{d_k \cdot C_g} + \sum_{i=1}^{k-1} \frac{d_{i+1}}{d_i} \right)
\end{aligned} \tag{23}$$

Given a driver size assignment, the wires are sized to minimize the following tradeoff between routing area and interconnect delay after eliminating the constant terms:

$$\begin{aligned}\mathrm{obj}_{T,k,\mathcal{D}}(\mathcal{W}) = {} & \left(\alpha \cdot \mathcal{L}_1 \cdot \frac{1}{C_g + C_d} \cdot c_0 + \gamma \cdot \mathcal{K}_1\right) \cdot \sum_{E \in T} w_E \\ & + \gamma \cdot \mathcal{K}_2 \cdot \sum_{E,E' \in T} F(E,E') \cdot \frac{w_{E'}}{w_E} \\ & + \gamma \cdot \mathcal{K}_3 \cdot \sum_{E,E' \in T} F(E,E') \cdot \frac{1}{w_E} \\ & + \gamma \cdot \mathcal{K}_4 \cdot \sum_{E \in T} G(E) \cdot \frac{1}{w_E} \\ & + \gamma \cdot \mathcal{K}_5 \cdot \sum_{E \in T} H(E) \cdot \frac{1}{w_E} \qquad (24)\end{aligned}$$

The formulation is similar to (12) except with difference in constant coefficients, which implies that the wiresizing results for delay minimization can also be applied to simultaneous power and performance optimization.

Since Power$(T, k, \mathcal{D}, \mathcal{W})$ and $t(T, k, \mathcal{D}, \mathcal{W})$ are usually of different orders of magnitude. The choice of α and γ in (22) might be difficult. We fine in our study that it is convenient to optimize the following objective:

$$\begin{aligned}\mathrm{obj}_\alpha(T, k, \mathcal{D}\mathcal{W}) = {} & \alpha \cdot \frac{\mathrm{Power}(T, k, \mathcal{D}, \mathcal{W})}{\mathrm{Power}_{\min}(T)} \\ & + (1 - \alpha) \cdot \frac{t(T, k, \mathcal{D}, \mathcal{W})}{t_{\min}(T, \mathcal{D}, \mathcal{W})} \qquad (25)\end{aligned}$$

where Power$_{\min}(T)$ is the minimum power required for driving an interconnect tree T and $t_{\min}(T, \mathcal{D}, \mathcal{W})$ is the minimum delay achievable by a chain of drivers driving an interconnect tree T.

Note that given a SDWS-DP problem, Power$_{\min}(T)$ can be computed easily by assuming a single driver driving an interconnect tree where all interconnect grid edges are assigned with minimum wire width. On the other hand, $t_{\min}(T, \mathcal{D}, \mathcal{W})$ can be computed using the algorithms to be presented in Section III. Therefore, they can be treated as constants.

Let k_α^*, $\mathcal{D}_\alpha^*$ and $\mathcal{W}_\alpha^*$ denote the optimal stage number and the optimal driver and sizing solution, respectively for a given α. Note that as α increases from 0 to 1, Power$(T, k_\alpha^*, \mathcal{D}_\alpha^*, \mathcal{W}_\alpha^*)$ will decrease monotonically and $t(T, k_\alpha^*, \mathcal{D}_\alpha^*, \mathcal{W}_\alpha^*)$ will increase monotonically[3]. Therefore, we can easily adjust parameter α between 0 and 1 to achieve smooth trade-off between performance and power.

III. SIMULTANEOUS DRIVER AND WIRE SIZING FOR PERFORMANCE OPTIMIZATION (SDWS-D PROBLEM)

A. *Properties of Optimal SDWS-D Solutions*

We consider the simultaneous driver and wire sizing problem for performance optimization which minimizes the performance measure $t(T, k, \mathcal{D}, \mathcal{W})$ specified by (15). The lemmas and theorems in this subsection are used to prove the correctness of the bound computation algorithm and optimality of the optimal algorithm to the SDWS-P problem. If the reader is only interested to learn the algorithms, one may proceed directly to Section III-B.

Properties of Optimal Driver Sizing Solutions for Performance Optimization:

Theorem 1: [17], [21] Given the loading capacitance C_{IL}, a chain of k drivers and the minimum gate capacitance C_g, the optimal stage ratio is $s = \left(\frac{C_{\mathrm{IL}}}{C_g}\right)^{1/k}$. □

Note that Theorem 1 assumes that the interconnect capacitances between drivers are negligible. We also assume that the loading capacitance C_{IL} is larger than the minimum gate capacitance C_g. Otherwise, a single minimum size driver is enough to drive the load.

Properties of Optimal Wire Sizing Solutions for Performance Optimization: The results in [12] showed several interesting properties of optimal wiresizing solutions for performance optimization, including the *separability, the monotone property*, and the *dominance property*. But they did not model the driver capacitance and interconnect fringing capacitance. Our study shows that these results still hold after we model the driver capacitance and fringing capacitance in the interconnect delay formulation specified by (12). Proofs of these theorems can be found in [9]. They are similar to the proofs of these properties in [13], except that our proofs can handle the driver capacitance and fringing capacitance of the interconnects.

Theorem 2 (separability)*:* If the wire width assignment of a path P originated from the source is given, the optimal wire width assignment for each subtree branching off from P can be carried out independently. □

Definition 1: Given a routing tree T, a wiresizing solution $\mathcal{W}$ on T is a monotone assignment if $w_E \geq w_{E'}$ for any pair of segments E and E' such that $E \in \mathrm{Ans}(E')$.

Theorem 3 (Monotone Property)*:* For any given tree T, there exists a monotone optimal wire width assignment $\mathcal{W}^*$. □

Definition 2: Given two wire width assignments $\mathcal{W}$ and $\mathcal{W}', \mathcal{W}$ dominates $\mathcal{W}'$ if for any segment E, the width assignment of E in $\mathcal{W}$ is greater than or equal to that of E in $\mathcal{W}'$.

Definition 3: Given a routing tree T, a wire width assignment $\mathcal{W}$ of T, and any particular segment $E \in T$, a local refinement on E is the operation to optimize the width of E based on the objective function in (12), without changing the widths of other segments specified by $\mathcal{W}$.

Theorem 4 (Dominance Property)*:* Given a routing tree T, let $\mathcal{W}^*$ be an optimal wire width assignment. If a wire width assignment $\mathcal{W}$ dominates $\mathcal{W}^*$, then a local refinement of $\mathcal{W}$ on any edge E is T still dominates $\mathcal{W}^*$. Similarly, if a wire width assignment $\mathcal{W}$ is dominated by $\mathcal{W}^*$, then a local refinement of $\mathcal{W}$ on any edge E in T is dominated by $\mathcal{W}^*$. □

Relation Between Driver Sizing and Optimal Wire Sizing for Performance Optimization: Given a routing tree T with one or more critical sinks. Let R_d be the resistance of the last driver driving the routing tree and $\mathcal{W}^*$ be the corresponding optimal wire width assignment. Consider another chain of cascaded drivers such that R'_d is the resistance of the last driver and $\mathcal{W}'^*$ is the optimal wire width assignment. We shall show the

---

[3] Otherwise, we can swap the optional SDWS-DP solutions for α and $\alpha'(\neq \alpha)$ and show that either $\mathrm{obj}_\alpha(T, k_{\alpha'}^*, \mathcal{D}_{\alpha'}^*, \mathcal{W}_{\alpha'}^*) < \mathrm{obj}_\alpha(T, k_\alpha^*, \mathcal{D}_\alpha^*, \mathcal{W}_\alpha^*)$ or $\mathrm{obj}_{\alpha'}(T, k_\alpha^*, \mathcal{D}_\alpha^*, \mathcal{W}_\alpha^*) < \mathrm{obj}_{\alpha'}(T, k_{\alpha'}^*, \mathcal{D}_{\alpha'}^*, \mathcal{W}_{\alpha'}^*)$ to obtain a contradiction.

following result that $R_d < R'_d$ implies that $\mathcal{W}^*$ dominates $\mathcal{W}'^*$ (Theorem 5):

Lemma 1: If $R_d < R'_d$, then area $(\mathcal{W}^*) \geq \text{area}(\mathcal{W}'^*)$, where area $(\mathcal{W})$ denotes the total wire width used in wire width assignment $\mathcal{W}$.

Proof: Let $\Delta t(T, R_d, \mathcal{W} \rightarrow \mathcal{W}')$ denote the difference $t(T, R_d, \mathcal{W}') - t(T, R_d, \mathcal{W}')$. Clearly, $\Delta t(T, R_d, \mathcal{W}^* \rightarrow \mathcal{W}'^*) \leq 0$ and $\Delta t(T, R'_d, \mathcal{W}'^* \rightarrow \mathcal{W}^*) \leq 0$ since $\mathcal{W}^*$ and $\mathcal{W}'^*$ are the optimal wiresizing solutions for R_d and R'_d, respectively. We can write $\Delta t(T, R_d, \mathcal{W}^* \rightarrow \mathcal{W}'^*)$ and $\Delta t(T, R'_d, \mathcal{W}'^* \rightarrow \mathcal{W}^*)$ as follows according to (12):

$$\begin{aligned}
&\Delta t(T, R_d, \mathcal{W}^* \rightarrow \mathcal{W}'^*) \\
&\quad = R_d \cdot c_0 \cdot \sum_{E \in T} (w^*_E - w'^*_E) \\
&\qquad + \mathcal{K}_2 \cdot \sum_{E, E' \in T} F(E, E') \cdot \left(\frac{w^*_{E'}}{w^*_E} - \frac{w'^*_{E'}}{w'^*_E} \right) \\
&\qquad + \mathcal{K}_3 \cdot \sum_{E, E' \in T} F(E, E') \cdot \left(\frac{1}{w^*_E} - \frac{1}{w'^*_E} \right) \\
&\qquad + \mathcal{K}_4 \cdot \sum_{E \in T} G(E) \cdot \left(\frac{1}{w^*_E} - \frac{1}{w'^*_E} \right) \\
&\qquad + \mathcal{K}_5 \cdot \sum_{E \in T} H(E) \cdot \left(\frac{1}{w^*_E} - \frac{1}{w'^*_E} \right) \leq 0 \\
&\Delta t(T, R'_d, \mathcal{W}'^* \rightarrow \mathcal{W}^*) \\
&\quad = R'_d \cdot c_0 \cdot \sum_{E \in T} (w'^*_E - w^*_E) \\
&\qquad + \mathcal{K}_2 \cdot \sum_{E, E' \in T} F(E, E') \cdot \left(\frac{w'^*_{E'}}{w'^*_E} - \frac{w^*_{E'}}{w^*_E} \right) \\
&\qquad + \mathcal{K}_3 \cdot \sum_{E, E' \in T} F(E, E') \cdot \left(\frac{1}{w'^*_E} - \frac{1}{w^*_E} \right) \\
&\qquad + \mathcal{K}_4 \cdot \sum_{E \in T} G(E) \cdot \left(\frac{1}{w'^*_E} - \frac{1}{w^*_E} \right) \\
&\qquad + \mathcal{K}_5 \cdot \sum_{E \in T} H(E) \cdot \left(\frac{1}{w'^*_E} - \frac{1}{w^*_E} \right) \leq 0
\end{aligned}$$

Summing up the above two inequalities, we obtain

$$\begin{aligned}
&\Delta t(T, R_d, \mathcal{W}^* \rightarrow \mathcal{W}'^*) + \Delta t(T, R'_d, \mathcal{W}'^* \rightarrow \mathcal{W}^*) \\
&\quad = c_0 \cdot (R_d - R'_d) \cdot \sum_{E \in T} (w^*_E - w'_E) \leq 0
\end{aligned}$$

Since $R_d < R'_d$, we have $\sum_{E \in T} (w^*_E - w'^*_E) \geq 0$, i.e. $\text{area}(\mathcal{W}^*) \geq \text{area}(\mathcal{W}'^*)$. □

Lemma 2: Consider an edge E in the routing tree. Let w^*_E and w'^*_E be the width of the edge in assignments $\mathcal{W}^*$ and $\mathcal{W}'^*$, respectively. If $w^*_E < w'^*_E$, then

$$\begin{aligned}
&\sum_{E' \in \text{Ans}(E)} H(E') \cdot \left(\frac{1}{w^*_{E'}} - \frac{1}{w'^*_{E'}} \right) \\
&\quad + \frac{H(E)}{w^*_E \cdot w'^*_E} \sum_{E' \in \text{Des}(E)} (w'^*_{E'} - w^*_{E'}) \\
&\quad > 0 \qquad (26)
\end{aligned}$$

Proof: Rewrite $t(T, R_d, \mathcal{W})$ in (12) with respect to E as follows:

$$\begin{aligned}
&t(T, R_d, \mathcal{W}) \\
&\quad = \mathcal{K}_0 + \left\{ R_d \cdot c_0 \cdot \sum_{E' \in T - \{E\}} w_{E'} + R_d \cdot c_0 \cdot w_E \right\} \\
&\qquad + \left\{ \mathcal{K}_2 \cdot \sum_{E', E'' \in T - \{E\}} F(E', E'') \cdot \frac{w_{E''}}{w_{E'}} \right. \\
&\qquad\quad + \mathcal{K}_2 \cdot \sum_{E' \in \text{Ans}(E)} F(E', E) \cdot \frac{w_E}{w_{E'}} \\
&\qquad\quad \left. + \mathcal{K}_2 \cdot \sum_{E' \in \text{Des}(E)} F(E, E') \cdot \frac{w_{E'}}{w_E} \right\} \\
&\qquad + \left\{ \mathcal{K}_3 \cdot \sum_{E' \in T - \{E\}, E'' \in T} F(E', E'') \cdot \frac{1}{w_{E'}} \right. \\
&\qquad\quad \left. + \mathcal{K}_3 \cdot \sum_{E' \in \text{Des}(E)} F(E, E') \cdot \frac{1}{w_E} \right\} \\
&\qquad + \left\{ \mathcal{K}_4 \cdot \sum_{E' \in T - \{E\}} G(E') \cdot \frac{1}{w_{E'}} + \mathcal{K}_4 \cdot G(E) \cdot \frac{1}{w_E} \right\} \\
&\qquad + \left\{ \mathcal{K}_5 \cdot \sum_{E' \in T - \{E\}} H(E') \cdot \frac{1}{w_{E'}} + \mathcal{K}_5 \cdot H(E) \cdot \frac{1}{w_E} \right\}
\end{aligned}$$

Let $\mathcal{W}/E : w_E \rightarrow w'_E$ denote the wire .sizing solution obtained by replacing w_E with w'_E in $\mathcal{W}$. The difference $\Delta t(T, R_d, \mathcal{W}^*/E : w^*_E \rightarrow w'^*_E)$, i.e., $t(T, R_d, \mathcal{W}^*) - t(T, R_d, \mathcal{W}^*/E\text{: } w^*_E \rightarrow w'^*_E)$ is given by:

$$\begin{aligned}
&\Delta t(T, R_d, \mathcal{W}^*/E : w^*_E \rightarrow w'^*_E) \\
&\quad = R_d \cdot c_0 \cdot (w^*_E - w'^*_E) \\
&\qquad + \mathcal{K}_2 \cdot \sum_{E' \in \text{Ans}(E)} F(E', E) \cdot \left(\frac{w^*_E - w'^*_E}{w^*_{E'}} \right) \\
&\qquad + \mathcal{K}_2 \cdot \sum_{E' \in \text{Des}(E)} F(E, E') \cdot w^*_{E'} \cdot \left(\frac{1}{w^*_E} - \frac{1}{w'^*_E} \right) \\
&\qquad + \mathcal{K}_3 \cdot \sum_{E' \in \text{Des}(E)} F(E, E') \cdot \left(\frac{1}{w^*_E} - \frac{1}{w'^*_E} \right) \\
&\qquad + \mathcal{K}_4 \cdot G(E) \cdot \left(\frac{1}{w^*_E} - \frac{1}{w'^*_E} \right) \\
&\qquad + \mathcal{K}_5 \cdot H(E) \cdot \left(\frac{1}{w^*_E} - \frac{1}{w'^*_E} \right) \\
&\quad \leq 0
\end{aligned}$$

Similarly, we can obtain $\Delta t(T, R'_d, \mathcal{W}'^*/E : w'^*_E \rightarrow w^*_E)$ and deduce the following:

$$\begin{aligned}&\Delta t(T, R_d, \mathcal{W}^*/E{:}\, w_E^* \rightarrow w_E'^*)\\&\quad + \Delta t(T, R_d', \mathcal{W}'^*/E{:}\, w_E'^* \rightarrow w_E^*)\\&\quad = (R_d - R_d') \cdot c_0 \cdot (w_E^* - w_E'^*)\\&\quad + \mathcal{K}_2 \cdot \sum_{E' \in \mathrm{Ans}(E)} F(E', E) \cdot (w_E^* - w_E'^*)\\&\quad \times \left(\frac{1}{w_{E'}^*} - \frac{1}{w_{E'}'^*}\right)\\&\quad + \mathcal{K}_2 \cdot \sum_{E' \in \mathrm{Des}(E)} F(E, E') \cdot (w_{E'}^* - w_{E'}'^*)\\&\quad \times \left(\frac{1}{w_E^*} - \frac{1}{w_E'^*}\right)\\&\quad \leq 0\end{aligned}$$

Since $w_E^* - w_E'^* < 0$ and $F(E_1, E_2) = H(E_1)$ if $E_2 \in \mathrm{Des}(E_1)$,

$$\begin{aligned}&(R_d - R_d') \cdot c_0 + \mathcal{K}_2 \cdot \sum_{E' \in \mathrm{Ans}(E)} H(E') \cdot \left(\frac{1}{w_{E'}^*} - \frac{1}{w_{E'}'^*}\right)\\&\quad + \mathcal{K}_2 \cdot \sum_{E' \in \mathrm{Des}(E)} H(E) \cdot (w_{E'}'^* - w_{E'}^*) \cdot \left(\frac{1}{w_E^* \cdot w_E'^*}\right)\\&\quad \geq 0\end{aligned}$$

Since $R_d - R_d' < 0$ and $\mathcal{K}_2 > 0$, the result follows. ☐

As defined in [11] and [12], a *single-stem tree* is a tree with only one segment (called the *stem segment* of that tree) incident on its root.

Lemma 3: Consider a single stem tree T. Let E be the stem segment of tree T. If $R_d < R_d'$, then $w_E^* \geq w_E'^*$.

Proof: Suppose $w_E^* < w_E'^*$, we obtain the following from (26) in Lemma 2:

$$\frac{H(E)}{w_E^* \cdot w_E'^*} \cdot \sum_{E' \in \mathrm{Des}(E)} (w_{E'}'^* - w_{E'}^*) > 0$$

This implies that $\sum_{E' \in \mathrm{Des}(E)} w_{E'}'^* > \sum_{E' \in \mathrm{Des}(E)} w_{E'}^*$. Hence, area$(\mathcal{W}'^*)$ > area$(\mathcal{W}^*)$. However, this contradicts Lemma 1. ☐

Lemma 4: If $w_E^* < w_E'^*$ and $w_{E'}^* \geq w_{E'}'^*$ for all $E' \in \mathrm{Ans}(E)$, then

$$\sum_{E' \in \mathrm{Des}(E)} (w_{E'}'^* - w_{E'}^*) > 0 \tag{27}$$

Proof: Since $\sum_{E' \in \mathrm{Ans}(E)} H(E') \cdot (\frac{1}{w_{E'}^*} - \frac{1}{w_{E'}'^*}) \leq 0$, it follows directly from (26) in Lemma 2 that the following is true:

$$H(E) \cdot \sum_{E' \in \mathrm{Des}(E)} (w_{E'}'^* - w_{E'}^*) > 0$$

Since $H(E) \geq 0$ (λ_i's are non-negative), it implies that $H(E) > 0$ and $\sum_{E' \in \mathrm{Des}(E)} (w_{E'}'^* - w_{E'}^*) > 0$. ☐

Lemma 5: Let E be an edge in the routing tree T. Let P_E denote the edges $E' \in \mathrm{Ans}(E)$ and T_E denote the single-stem subtree rooted at E (including E). Given R_d and the wire widths of edges in P_E, optimizing the wire width assignment of edges in $T - P_E$ implies that the following objective function is minimized:

$$\begin{aligned}&t_{P_E}(T_E, R_d, \mathcal{W})\\&\quad = R_d \cdot c_0 \cdot \sum_{E' \in T_E} w_{E'} + \mathcal{K}_2 \cdot \sum_{E' \in P_E, E'' \in T_E} F(E', E'') \cdot \frac{w_{E''}}{w_{E'}}\\&\quad + \mathcal{K}_2 \cdot \sum_{E', E'' \in T_E} F(E', E'') \cdot \frac{w_{E''}}{w_{E'}}\\&\quad + \mathcal{K}_3 \cdot \sum_{E', E'' \in T_E} F(E', E'') \cdot \frac{1}{w_{E'}}\\&\quad + \mathcal{K}_4 \cdot \sum_{E' \in T_E} G(E') \cdot \frac{1}{w_{E'}} + \mathcal{K}_5 \cdot \sum_{E' \in T_E} H(E') \cdot \frac{1}{w_{E'}}\end{aligned} \tag{28}$$

Proof: Let $\overline{T_E \cup P_E}$ denote the edges not in $T_E \cup P_E$. We define:

$$\begin{aligned}&t_{P_E}(P_E, R_d, \mathcal{W})\\&\quad = \sum_{E \in P_E} \mathcal{K}_1 \cdot w_E + \mathcal{K}_2 \cdot \sum_{E, E' \in P_E} F(E, E') \cdot \frac{w_{E'}}{w_E}\\&\quad + \mathcal{K}_3 \cdot \sum_{E \in P_E, E' \in T} F(E, E') \cdot \frac{1}{w_E}\\&\quad + \mathcal{K}_4 \cdot \sum_{E \in P_E} G(E) \cdot \frac{1}{w_E}\\&\quad + \mathcal{K}_5 \cdot \sum_{E \in P_E} H(E) \cdot \frac{1}{w_E}\end{aligned}$$

$$\begin{aligned}&t_{P_E}(\overline{T_E \cup P_E}, R_d, \mathcal{W})\\&\quad = \mathcal{K}_1 \cdot \sum_{E' \in \overline{T_E \cup P_E}} w_{E'}\\&\quad + \mathcal{K}_2 \times \sum_{E' \in P_E, E'' \in \overline{T_E \cup P_E}} F(E, E') \cdot \frac{w_{E''}}{w_{E'}}\\&\quad + \mathcal{K}_2 \cdot \sum_{E', E'' \in \overline{T_E \cup P_E}} F(E', E'') \cdot \frac{w_{E''}}{w_{E'}}\\&\quad + \mathcal{K}_3 \cdot \sum_{E', E'' \in \overline{T_E \cup P_E}} F(E', E'') \cdot \frac{1}{w_E'}\\&\quad + \mathcal{K}_4 \cdot \sum_{E' \in \overline{T_E \cup P_E}} G(E') \cdot \frac{1}{w_{E'}}\\&\quad + \mathcal{K}_5 \cdot \sum_{E' \in \overline{T_E \cup P_E}} H(E') \cdot \frac{1}{w_{E'}}\end{aligned}$$

Hence, we can rewrite objectives function specified by (12) as

$$\begin{aligned}t(T, R_d, \mathcal{W}) &= \mathcal{K}_0 + t_{P_E}(P_E, R_d, \mathcal{W})\\&\quad + t_{P_E}(\overline{T_E \cup P_E}, R_d, \mathcal{W}) + t_{P_E}(T_E, R_d, \mathcal{W})\end{aligned}$$

Given the wire width assignment of $P_E, \mathcal{K}_0$ and $t_{P_E}(P_E, R_d, \mathcal{W})$ are both constants. Since the variables (widths of edges in T_E) in $t_{P_E}(T_E, R_d, \mathcal{W})$ do not appear in $t_{P_E}(\overline{T_E \cup P_E}, R_d, \mathcal{W})$, optimizing $t(T, R_d, \mathcal{W})$ implies optimizing $t_{P_E}(T_E, R_d, \mathcal{W})$. (Note that the proof of Lemma 5 essentially proves the separability property in [9] and [13]). □

Lemma 6: If T is a single stem tree, then $R_d < R'_d$ implies $\mathcal{W}^*$ dominates $\mathcal{W}'^*$.

Proof: Sort all edges in T according to a topological order staring from the stem. Let E be the first edge in this order such that $w^*_E < w'^*_E$. Then, for all $E' \in \mathrm{Ans}(E), w^*_{E'} \geq w'^*_{E'}$. We use P_E to denote the edges $E' \in \mathrm{Ans}(E)$. By Lemma 3, P_E is not an empty set. Let T_E denote the subtree rooted at E (including E). Lemma 5 implies that the wire width assignment $\mathcal{W}^*$ of edges in T_E, denoted as $\mathcal{W}^*(T_E)$, is an optimal wire width assignment of T_E under the objective function (28), assuming that the wire widths of edges in P_E are fixed as specified by $\mathcal{W}^*$.

Let $\mathcal{W}^*/T_E : \mathcal{W}^* \to \mathcal{W}'^*$ denote the wire width assignment obtained by replacing the wire width assignment $\mathcal{W}^*(T_E)$ of T_E with $\mathcal{W}'^*(T_E)$, i.e., $(\mathcal{W}^* - \mathcal{W}^*(T_E)) \cup \mathcal{W}'^*(T_E)$. Similarly, let $\mathcal{W}'^*/T_E : \mathcal{W}'^* \to \mathcal{W}^*$ denote the wire width assignment obtained by replacing the wire width assignment $\mathcal{W}'^*(T_E)$ of T_E with $\mathcal{W}^*(T_E)$, i.e., $(\mathcal{W}'^* - \mathcal{W}'^*(T_E)) \cup \mathcal{W}^*(T_E)$. Comparing $t_{P_E}(T_E, R_d, \mathcal{W}^*)$ with $t_{P_E}(T_E, R_d, \mathcal{W}^*/T_E : \mathcal{W}^* \to \mathcal{W}'^*)$, we obtain the following inequality:

$$\begin{aligned}
&t_{P_E}(T_E, R_d, \mathcal{W}^*) - t_{P_E}(T_E, R_d, \mathcal{W}^*/T_E : \mathcal{W}^* \to \mathcal{W}'^*)\\
&\quad = R_d \cdot c_0 \cdot \sum_{E' \in T_E} (w^*_{E'} - w'^*_{E'})\\
&\qquad + \mathcal{K}_2 \cdot \sum_{E' \in P_E, E'' \in T_E} F(E', E'') \cdot \left(\frac{w^*_{E''}}{w^*_{E'}} - \frac{w'^*_{E''}}{w^*_{E'}}\right)\\
&\qquad + \mathcal{K}_2 \cdot \sum_{E', E'' \in T_E} F(E', E'') \cdot \left(\frac{w^*_{E''}}{w^*_{E'}} - \frac{w'^*_{E''}}{w'^*_{E'}}\right)\\
&\qquad + \mathcal{K}_3 \cdot \sum_{E', E'' \in T_E} F(E', E'') \cdot \left(\frac{1}{w^*_{E'}} - \frac{1}{w'^*_{E'}}\right)\\
&\qquad + \mathcal{K}_4 \cdot \sum_{E' \in T_E} G(E') \cdot \left(\frac{1}{w^*_{E'}} - \frac{1}{w'^*_{E'}}\right)\\
&\qquad + \mathcal{K}_5 \cdot \sum_{E' \in T_E} H(E') \cdot \left(\frac{1}{w^*_{E'}} - \frac{1}{w'^*_{E'}}\right)\\
&\quad \leq 0
\end{aligned}$$

Similarly, comparing $t_{P_E}(T_E, R'_d, \mathcal{W}'^*)$ with $t_{P_E}(T_E, R'_d, \mathcal{W}'^*/T_E : \mathcal{W}'^* \to \mathcal{W}^*)$, we obtain the following inequality:

$$\begin{aligned}
&t_{P_E}(T_E, R'_d, \mathcal{W}'^*) - t_{P_E}(T_E, R'_d, \mathcal{W}'^*/T_E : \mathcal{W}'^* \to \mathcal{W}^*)\\
&\quad = R'_d \cdot c_0 \cdot \sum_{E' \in T_E} (w'^*_{E'} - w^*_{E'})\\
&\qquad + \mathcal{K}_2 \cdot \sum_{E' \in P_E, E'' \in T_E} F(E', E'') \cdot \left(\frac{w'^*_{E''}}{w'^*_{E'}} - \frac{w^*_{E''}}{w'^*_{E'}}\right)\\
&\qquad + \mathcal{K}_2 \cdot \sum_{E', E'' \in T_E} F(E', E'') \cdot \left(\frac{w'^*_{E''}}{w'^*_{E'}} - \frac{w^*_{E''}}{w^*_{E'}}\right)\\
&\qquad + \mathcal{K}_3 \cdot \sum_{E', E'' \in T_E} F(E', E'') \cdot \left(\frac{1}{w'^*_{E'}} - \frac{1}{w^*_{E'}}\right)\\
&\qquad + \mathcal{K}_4 \cdot \sum_{E' \in T_E} G(E') \cdot \left(\frac{1}{w'^*_{E'}} - \frac{1}{w^*_{E'}}\right)\\
&\qquad + \mathcal{K}_5 \cdot \sum_{E' \in T_E} H(E') \cdot \left(\frac{1}{w'^*_{E'}} - \frac{1}{w^*_{E'}}\right)\\
&\quad \leq 0
\end{aligned}$$

Summing up the above two inequalities, we have:

$$\begin{aligned}
&(R_d - R'_d) \cdot c_0 \cdot \sum_{E' \in T_E} (w^*_{E'} - w'^*_{E'})\\
&\quad + \mathcal{K}_2 \cdot \sum_{E'' \in T_E} \left\{ (w^*_{E''} - w'^*_{E''}) \cdot \sum_{E' \in P_E} F(E', E'') \right.\\
&\qquad \left. \times \left(\frac{1}{w^*_{E'}} - \frac{1}{w'^*_{E'}}\right) \right\}\\
&\quad \leq 0
\end{aligned}$$

Since $F(E', E'') = H(E')$ for $E' \in P_E$ and $E'' \in T_E$, we have:

$$\begin{aligned}
&(R_d - R'_d) \cdot c_0 \cdot \sum_{E' \in T_E} (w^*_{E'} - w'^*_{E'})\\
&\quad + \mathcal{K}_2 \cdot \sum_{E' \in P_E} H(E') \cdot \left(\frac{1}{w^*_{E'}} - \frac{1}{w'^*_{E'}}\right)\\
&\quad \times \sum_{E'' \in T_E} (w^*_{E''} - w'^*_{E''}) \leq 0
\end{aligned}$$

From Lemma 4, $\sum_{E' \in T_E} (w'^*_{E'}) < 0$ [negation of (27)]. In addition, we know that $R_d - R'_d < 0$ and $\mathcal{K}_2 > 0$. Therefore, we can deduce that

$$\sum_{E' \in P_E} H(E') \cdot \left(\frac{1}{w^*_{E'}} - \frac{1}{w'^*_{E'}}\right) > 0$$

This contradicts the assumption that $w^*_{E'} \geq w'^*_{E'}$ for all $E' \in P_E$. □

According to the separability, we can decompose T into several single-stem trees and optimize each single-stem tree independently. Applying Lemma 6 to each single-stem tree, we have:

Theorem 5: (DS/WS Relation) For any tree T with one or more critical sinks, $R_d < R'_d$ implies $\mathcal{W}^*$ dominates $\mathcal{W}'^*$. □

This result reveals the relation between driver sizing and wiresizing, and it plays as important role in determining the lower and upper bounds of the optimal k-SDWS-D solutions in next subsection.

B. Lower and Upper Bound Computation for Optimal k-SDWS-D Solutions

We first present an algorithm called the *k-SDWS/D LU-Bound* algorithm for computing the upper and lower bounds of an optimal k-SDWS-D solution for a given number of stage k: Starting with an initial wire width assignment (say, all segments have the minimum width), we compute the

```
k-SDWS/D LU-Bound Algorithm

Function k-SDWS/D LU-Bound($T, R_{min}, k$)

    /* Given a tree $T$, the minimum width transistor resistance $R_{min}$, and the
    stage number $k$, return the upper and lower bounds of the optimal k-SDWS-D
    solution. */

    $\mathcal{W}_l \leftarrow$ Min_Wire_Width;
    while true
        Compute capacitive load: $C_{IL}(T, \mathcal{W}_l)$;
        Compute optimal driver stage ratio: $s_l \leftarrow (\frac{C_{IL}(T,\mathcal{W}_l)}{C_g})^{1/k}$;
        Compute the $k$-th driver resistance: $R_d \leftarrow \frac{R_{min}}{s_l^{(k-1)}}$;
        $\mathcal{W} \leftarrow$ Delay_Optimal_Wiresizing($R_d, T$);
        if $\mathcal{W} > \mathcal{W}_l$ then
            $\mathcal{W}_l \leftarrow \mathcal{W}$
        else break;
    end while
    Output $s_l$ as the stage ratio of the lower bound of driver sizing solution
        and $\mathcal{W}_l$ as the lower bound of wire sizing solution;
    $\mathcal{W}_u \leftarrow$ Max_Wire_Width;
    while true
        Compute capacitive load: $C_{IL}(T, \mathcal{W}_u)$;
        Compute optimal driver stage ratio: $s_u \leftarrow (\frac{C_{IL}(T,\mathcal{W}_u)}{C_g})^{1/k}$;
        Compute the $k$-th driver resistance: $R_d \leftarrow \frac{R_{min}}{s_u^{(k-1)}}$;
        $\mathcal{W} \leftarrow$ Delay_Optimal_Wiresizing($R_d, T$);
        if $\mathcal{W} < \mathcal{W}_u$ then
            $\mathcal{W}_u \leftarrow \mathcal{W}$
        else break;
    end while
    Output $s_u$ as the stage ratio of the upper bound of driver sizing solution
        and $\mathcal{W}_u$ as the upper bound of wire sizing solution;
end Function;
```

Fig. 4. The k-SDWS/D LU-Bound algorithm for computing lower and upper bounds of optimal SDWS solution for a given stage number k.

capacitive load of the routing tree. Based on Theorem 1, the optimal stage ratio, denoted by s, of the k-stage cascaded drivers can be computed and a driver sizing solution is obtained. Now, we perform delay optimal wiresizing [12] on the routing tree T based on the resistance of the k-th driver. If the wire width assignment changes, the capacitive load changes. A new driver stage ratio is computed to yield a new optimal driver sizing solution. Then, the optimal wiresizing solution will be computed again based on the new driver size of the k-th driver. The process is repeated until the wire width assignment does not change in consecutive iterations. The algorithm is described formally in Fig. 4.

Let $\mathcal{W}_0$ be the initial minimum width assignment and $\mathcal{W}_i$ denote the optimal wire width assignment obtained in iteration i. Similarly, let R_d^i denote the resistance of the k-th driver at the beginning of iteration i. During iteration i $(i \geq 1)$, R_d^i is computed based on the capacitive load $C_{\text{IL}}(T, \mathcal{W}_{i-1})$ and $\mathcal{W}_i$ is obtained based on R_d^i. It is obvious that in the first iteration, the wire widths in $\mathcal{W}_1$ is an upward revision of the wire width in $\mathcal{W}_0$ (since $\mathcal{W}_0$ is the minimum-width assignment). Hence, the capacitive load can only increase which implies an increase in the stage ratio and a decrease in the resistance of the k-th driver, i.e., $R_d^2 < R_d^1$. According to Theorem 5, $\mathcal{W}_2$ dominates $\mathcal{W}_1$. In general, we can show by mathematical induction that $R_d^{i+1} \leq R_d^i$ (since $\mathcal{W}_i$ dominates $\mathcal{W}_{i-1}$) and in turn $\mathcal{W}_{i+1}$ dominates $\mathcal{W}_i$ (according to Theorem 5). Hence, the algorithm will terminate (since there is an upper bound on the maximum wire width) and indeed computes the lower bound of driver and wire sizes of the optimal k-SDWS-D solution. Similarly, we start with a maximum wire width assignment and compute the upper bound of driver and wire sizes of the optimal k-SDWS-D solution. Therefore, we have the following result:

Theorem 6: Given a chain of k drivers, the k-SDWS/D LU-Bound algorithm computes the lower and upper bounds of an optimal k-SDWS-D solution.

Proof: We give the proof for the lower bound computation only. The proof for upper bound can be derived similarly. Let $(\mathcal{D}^*, \mathcal{W}^*)$ be the optimal SDWS-D solution for a chain of k drivers and d_k^* be the optimal size of driver k. Let $(\mathcal{D}_i, \mathcal{W}_i)$ denote the driver and wiresizing solution computed in iteration i and d_k^i be the size of driver k in iteration i. We first show that $(\mathcal{D}_1, \mathcal{W}_1)$ is dominated by $(\mathcal{D}^*, \mathcal{W}^*)$.

Since $\mathcal{W}_0$ is dominated by $\mathcal{W}^*$, we have $C_{\text{IL}}(T, \mathcal{W}_0) \leq C_{\text{IL}}(T, \mathcal{W}^*)$. By Theorem 1,

$$d_k^1 = \left(\frac{C_{\text{IL}}(T, \mathcal{W}_0)}{C_g}\right)^{(k-1)/k} \leq \left(\frac{C_{\text{IL}}(T, \mathcal{W}^*)}{C_g}\right)^{(k-1)/k} = d_k^*$$

By Theorem 5, we conclude that $\mathcal{W}_1$ (which is computed based on R_d^1) is dominated by $\mathcal{W}^*$ since $R_d^1 = R_{\min}/d_k^1 \geq R_{\min}/d_k^* = R_d^*$ where R_d^* is the resistance of driver k in the optimal solution.

Assuming that $(\mathcal{D}_i, \mathcal{W}_i)$ is dominated by $(\mathcal{D}^*, \mathcal{W}^*)$, we can follow the same arguments and deduce that $(\mathcal{D}_{i+1}, \mathcal{W}_{i+1})$ is also dominated by $(\mathcal{D}^*, \mathcal{W}^*)$. Hence, the solution computed is a lower bound of the optimal solution for a given stage number k. ☐

Experimental results show that the algorithm terminates after three or four iterations in most cases. In addition, the upper and lower bounds meet for most instances, which implies that the optimal k-SDWS-D solution is obtained. Note that the upper and lower bounds include both the driver and wire sizes.

C. Optimal Algorithm for the SDWS-D Problem

For a given stage number k, we compute the optimal driver and wire sizing solution as follows: we compute the upper and lower bounds of the k-SDWS-D optimal solutions using the k-SDWS/D LU-Bound algorithm. In the case where the bounds do not meet, we can obtain a set of discrete driver sizes defined by the bounds computed by the k-SDWS/D LU-Bound algorithm for the k-th driver. For each driver size, we can compute the optimal wiresizing solution using the algorithm in [12] and use Theorem 1 to compute the sizes of driver $D_2 \cdots D_{k-1}$ using $d_k \cdot C_g$ as the capacitive load that $(k-1)$-stage drivers have to drive. The *k-SDWS/D Optimal* algorithm is given in Fig. 5.

Theorem 7: Given a chain of k drivers and p possible sizes for the last driver and a routing tree with n segments and r possible wire widths, the worst case time complexity to compute the optimal driver and wire sizes for the k-SDWS-D problem is $O(p \cdot n^r)$. ☐

Note that p is usually very small since the lower and upper bounds computed in Section III-B are very tight (in fact, they meet for almost all test cases). The factor $O(n^r)$ is the worst case complexity of optimal wiresizing algorithm which in fact

```
k-SDWS/D Optimal Algorithm

Function k-SDWS/D Optimal($T, R_{min}, k$)

    /* Given a tree $T$, the minimum width transistor resistance $R_{min}$, and a
    stage number $k$, return the optimal stage ratio $s$, and the optimal wire width
    assignment $\mathcal{W}$. */

    Compute the upper and lower bounds:
        $\{s_u, s_l, \mathcal{W}_u, \mathcal{W}_l\} \leftarrow$ k-SDWS/D LU-Bound($T, R_{min}, k$);
    if $\mathcal{W}_u > \mathcal{W}_l$ (upper and lower bounds do not meet) then
        Best_Delay $\leftarrow \infty$;
        Discretize possible driver sizes of the $k$-th driver in the interval
            $[(s_l)^{k-1}, (s_u)^{k-1}]$;
        for each driver size $d_p$ in $[(s_l)^{k-1}, (s_u)^{k-1}]$ do
            $d_k \leftarrow d_p$;
            Compute $k$-th driver resistance: $R_d \leftarrow \frac{R_{min}}{d_p}$
            $\mathcal{W} \leftarrow$ Delay_Optimal_Wiresizing($R_d, T$);
            Compute driver sizes for $D_{1..k-1}$ according to Theorem 1;
            Current_Delay $\leftarrow t(T, k, \mathcal{D}, \mathcal{W})$;
            if Current_Delay < Best_Delay then
                Solution $\leftarrow \{\mathcal{D}, \mathcal{W}\}$;
                Best_Delay $\leftarrow$ Current_Delay;
        end for
    else /* $\mathcal{W}_u = \mathcal{W}_l$ and $s_u = s_l$ */
        Construct Solution from $\{s_u, s_l, \mathcal{W}_u, \mathcal{W}_l\}$;
        Best_Delay $\leftarrow t(T, k, \mathcal{D}, \mathcal{W})$;
    end if
    return Solution;
end Function;
```

Fig. 5. The k-SDWS/D Optimal algorithm for computing the optimal SDWS solution for a given stage number k.

```
SDWS/D Optimal Algorithm

Function SDWS/D Optimal($T, R_{min}$)

    /* Given a tree $T$, and the minimum width transistor resistance $R_{min}$, return
    the optimal number of stages $k$, the optimal stage ratio $s$, and the optimal
    wire width assignment $\mathcal{W}$. */

    Compute $k^D_{MAX}$;
    Best_Delay $\leftarrow \infty$;
    for $k \leftarrow 1$ to $k^D_{MAX}$ do
        Compute optimal driver and wire sizing solution for $k$:
            $\{\mathcal{D}, \mathcal{W}\} \leftarrow$ k-SDWS/D Optimal($T, R_{min}, k$);
        Current_Delay $\leftarrow t(T, k, \mathcal{D}, \mathcal{W})$;
        if Current_Delay < Best_Delay then
            Best_Solution $\leftarrow \{k, \mathcal{D}, \mathcal{W}\}$;
    end for
    return Best_Solution;
end Function;
```

Fig. 6. The SDWS/D Optimal algorithm.

run in $O(n^3 \cdot r)$ time based on effective lower and upper bound computation using the dominance property (Details of the wiresizing algorithm can be found in [12]). Therefore, the k-SDWS/D Optimal algorithm runs in $O(n^3 \cdot r)$ time in practice.

To compute the optimal stage number $k^*_{\mathcal{D}}$ for the SDWS-D problem, our *SDWS/D Optimal* algorithm performs a linear search for $1 \leq k \leq k^D_{\mathrm{MAX}} = \left\lceil \frac{\ln C_{\mathrm{IL}}(T, \mathcal{W}_{\mathrm{MAX}})/C_g}{\ln s^*} \right\rceil$ where s^* is the optimal stage ratio to be defined later in Lemma 8. The algorithm is given in Fig. 6. Optimality of the SDWS/D Optimal algorithm is justified by the following results:

Lemma 7: Given a routing tree T with a fixed wire width assignment $\mathcal{W}$, the performance measure $t(T, k, \mathcal{D}, \mathcal{W})$ in (15) is a concave function with respect to k for $k > 0$ (assuming k is continuous).

Proof: We can rewrite the performance measure $t(T, k, \mathcal{D}, \mathcal{W})$ in (15) as

$$t(T, k, \mathcal{D}, \mathcal{W}) = k \cdot R_{\min} \cdot C_d + R_{\min} \cdot C_g \cdot \sum_{i=1}^{k-1} \frac{d_{i+1}}{d_i} + R_{\min} \cdot \frac{C_{\mathrm{IL}}(T, \mathcal{W})}{d_k} + t(T, \mathcal{W}) \qquad (29)$$

where

$$\begin{aligned} t(T, \mathcal{W}) = & \sum_{N_i \in \mathrm{sink}(T)} \lambda_i \cdot \sum_{E \in P(N_+, N_i)} \frac{r_0 \cdot c_0}{2} \\ & + \mathcal{K}_2 \cdot \sum_{E, E' \in T} F(E, E') \cdot \frac{w_{E'}}{w_E} \\ & + \mathcal{K}_3 \cdot \sum_{E, E' \in T} F(E, E') \cdot \frac{1}{w_E} + \mathcal{K}_4 \cdot \sum_{E \in T} G(E) \cdot \frac{1}{w_E} \\ & + \mathcal{K}_5 \cdot \sum_{E \in T} H(E) \cdot \frac{1}{w_E} \end{aligned}$$

Note that $t(T, \mathcal{W})$ is a constant for a fixed wire width assignment.

Given a fixed wire width assignment $\mathcal{W}$, we can compute $C_{\mathrm{IL}}(T, \mathcal{W})$. Using Theorem 1, $t(T, k, \mathcal{D}, \mathcal{W})$ is minimized when the stage ratio is $s = (\frac{C_{\mathrm{IL}}(T, \mathcal{W})}{C_g})^{1/k}$ for a given k. Let $M = (\frac{C_{\mathrm{IL}}(T, \mathcal{W})}{C_g})$, we can rewrite the minimized performance measure as

$$t(T, k, \mathcal{D}, \mathcal{W}) = k \cdot R_{\min} \cdot C_d + k \cdot R_{\min} \cdot C_g \cdot M^{1/k} + t(T, \mathcal{W})$$

Taking the second derivative of $t(T, k, \mathcal{D}, \mathcal{W})$ with respect to k, we obtain

$$\frac{d^2 t(T, k, \mathcal{D}, \mathcal{W})}{dk^2} = R_{\min} \cdot C_g \cdot \frac{1}{k^3} \cdot M^{1/k} \cdot \ln M > 0$$

Note that $M > 1$ since $C_{\mathrm{IL}}(T, \mathcal{W})$ is assumed to be larger than C_g. Hence, $t(T, k, \mathcal{D}, \mathcal{W})$ is a concave function. □

Since $t(T, \mathcal{W})$ is a constant in (29) for a fixed wire width assignment, the following result holds:

Lemma 8: [21] Given a routing tree with a fixed wire width assignment $\mathcal{W}$, the optimal stage ratio s^* is given by the expression

$$s^* = e^{1 + a/s^*}$$

where $a = C_d/C_g$ and e is the base of the natural logarithm. The corresponding optimal stage number $k^*_{\mathcal{W}}$, which may not be an integer, is

$$k^*_{\mathcal{W}} = \frac{\ln C_{\mathrm{IL}}(T, \mathcal{W})/C_g}{\ln s^*}$$

Note that s^* depends only on the process parameters and not the loading capacitance whereas $k^*_{\mathcal{W}}$ depends on both the process parameters and wire width assignment. Moreover, in our SDWS/D Optimal algorithm, $k^D_{\mathrm{MAX}} = \lceil k^*_{\mathcal{W}_{\mathrm{MAX}}} \rceil$.

Theorem 8: (Optimality of SDWS/D Optimal Algorithm): Given a routing tree T, the optimal stage number $k^*_{\mathcal{D}}$ is less than or equal to k^D_{MAX}.

Proof: Suppose not, i.e., $k_D^* > k_{\mathrm{MAX}}^D$. Let $\mathcal{W}^*$ denote the wiresizing solution of the optimal SDWS-D problem. Since $\mathcal{W}^*$ is dominated by $\mathcal{W}_{\mathrm{MAX}}$, $C_{\mathrm{IL}}(T, \mathcal{W}_{\mathrm{MAX}}) \geq C_{\mathrm{IL}}(T, \mathcal{W}^*)$. Therefore,

$$k_D^* > k_{\mathrm{MAX}}^D = \lceil k^*_{\mathcal{W}_{\mathrm{MAX}}} \rceil \geq \lceil k^*_{\mathcal{W}^*} \rceil \geq k^*_{\mathcal{W}^*}.$$

From Lemmas 7 and 8, $t(T, k, \mathcal{D}, \mathcal{W})$ is a concave function and it is strictly monotonically increasing for $k > k^*_{\mathcal{W}}$. This contradicts the assumption that k_D^* is the optimal stage number since $k_D^* > \lceil k^*_{\mathcal{W}^*} \rceil \geq k^*_{\mathcal{W}^*}$. □

IV. SIMULTANEOUS DRIVER AND WIRE SIZING FOR BOTH DELAY AND POWER OPTIMIZATION (SDWS-DP PROBLEM)

In the SDWS-DP problem formulated in Section II-B, the drivers and wires are sized to minimize the following objective function (22):

$$\mathrm{obj}(T, k, \mathcal{D}, \mathcal{W}) = \alpha \cdot \mathrm{Power}(T, k, \mathcal{D}, \mathcal{W}) + \gamma \cdot t(T, k, \mathcal{D}, \mathcal{W})$$

We can adjust parameters α and γ to achieve smooth trade-off between performance and power. By choosing the parameters carefully, we can compute a driver and wire sizing solution to meet a delay constraint while minimizing the power dissipation.

In this section, we shall first study the properties of the optimal SDWS-DP solutions and then present an efficient algorithm for computing the optimal SDWS-DP solution.

A. Properties of Optimal SDWS Solutions for Both Delay and Power Optimization

The lemmas and theorems in this subsection are used to prove the correctness and optimality of the bound computation and optimal algorithm to the SDWS-DP problem. If the reader is only interested to learn the SDWS/DP algorithms, one may proceed directly to Section IV-B.

First, we define the monotone property and the dominance relation[4] for the driver sizing solutions.

Definition 4: Given a chain of k cascaded drivers D_i's, a driver sizing solution $\mathcal{D}$ is a monotone assignment if $d_{i+1} > d_i$ for all $i = 1, 2, \ldots, k-1$.

Definition 5: Given two driver sizing solutions $\mathcal{D} = \{d_1, d_2, \ldots, d_k\}$ and $\mathcal{D}' = \{d_1', d_2', \ldots, d_k'\}$, $\mathcal{D}$ dominates $\mathcal{D}'$ if $d_i \geq d_i'$ for $1 \leq i \leq k$.

The following theorem defines a set of equations which will be used for optimal driver sizing solution computation for a fixed loading capacitance $C_{\mathrm{IL}}(T, \mathcal{W})$ in Section IV-B.

Theorem 9: Given a fixed loading capacitance $C_{\mathrm{IL}}(T, \mathcal{W})$, the optimal driver size assignment $\mathcal{D}$ for a chain of k drivers satisfies the following equations when $\gamma > 0$:

$$\begin{aligned} &\alpha \cdot (\mathcal{L}_1 + \mathcal{L}_2) + \gamma \cdot \mathcal{J}_2 \cdot \left(\frac{1}{d_{i-1}} - \frac{d_{i+1}}{d_i^2} \right) = 0 \\ &\qquad \text{for all } i = 2 \ldots k-1 \\ &\alpha \cdot (\mathcal{L}_1 + \mathcal{L}_2) + \gamma \cdot \mathcal{J}_2 \cdot \left(\frac{1}{d_{k-1}} - \frac{C_{\mathrm{IL}}(T, \mathcal{W})}{d_k^2 \cdot C_g} \right) = 0 \end{aligned} \tag{30}$$

Proof: The optimal driver sizing solution minimizes the objective function specified by (23):

$$\begin{aligned} \mathrm{obj}_{T,k,\mathcal{W}}(\mathcal{D}) &= \alpha \cdot (\mathcal{L}_1 + \mathcal{L}_2) \cdot \sum_{i=2}^{k} d_i \\ &\quad + \gamma \cdot \mathcal{J}_2 \cdot \left(\frac{C_{\mathrm{IL}}(T, \mathcal{W})}{d_k \cdot C_g} + \sum_{i=1}^{k-1} \frac{d_{i+1}}{d_i} \right) \end{aligned}$$

Differentiating $\mathrm{obj}_{T,k,\mathcal{W}}(\mathcal{D})$ with respect to d_i for $2 \leq i \leq k$, we obtain

$$\begin{aligned} \frac{\partial \mathrm{obj}_{T,k,\mathcal{W}}(\mathcal{D})}{\partial d_i} &= \alpha \cdot (\mathcal{L}_1 + \mathcal{L}_2) \\ &\quad + \gamma \cdot \mathcal{J}_2 \cdot \left(\frac{1}{d_{i-1}} - \frac{d_{i+1}}{d_i^2} \right) \\ &\quad \text{for all } i = 2 \cdots k-1 \\ \frac{\partial \mathrm{obj}_{T,k,\mathcal{W}}(\mathcal{D})}{\partial d_k} &= \alpha \cdot (\mathcal{L}_1 + \mathcal{L}_2) \\ &\quad + \gamma \cdot \mathcal{J}_2 \cdot \left(\frac{1}{d_{k-1}} - \frac{C_{\mathrm{IL}}(T, \mathcal{W})}{d_k^2 \cdot C_g} \right) \end{aligned}$$

By setting $\frac{\partial \mathrm{obj}_{T,k,\mathcal{W}}(\mathcal{D})}{\partial d_i} = 0$ for $2 \leq i \leq k$, we obtain the system of equations given in (30). □

Note that there are many solutions to the set of nonlinear equations specified by (30) and they may assign drivers with sizes smaller than the minimum driver size. The following result allows us to consider only monotone drive sizing solution.

Theorem 10: (Monotone Property) For any given capacitive load $C_{\mathrm{IL}}(T, \mathcal{W})$, any optimal driver sizing solution to the SDWS-DP problem under the combined delay and power optimization objective function specified by (23) is monotone.

Proof: Let $\mathcal{D}$ be the nonmonotone optimal driver size assignment. Suppose j is the smallest index such that $d_j \leq d_{j-1}$. Let $\mathcal{D} - \{d_j\}$ be the assignment obtained after discarding driver D_j in the nonmonotone optimal driver chain. For convenience, we use d_{k+1} to denote $\frac{C_{\mathrm{IL}}(T, \mathcal{W})}{C_g}$. According to objective function defined in (23), we have:

$$\begin{aligned} \mathrm{obj}_{T,k,\mathcal{W}}(\mathcal{D}) &= \alpha \cdot (\mathcal{L}_1 + \mathcal{L}_2) \cdot \sum_{i=2}^{k} d_i \\ &\quad + \gamma \cdot \mathcal{J}_2 \cdot \sum_{i=1}^{k} \frac{d_{i+1}}{d_i} \\ \mathrm{obj}_{T,k-1,\mathcal{W}}(\mathcal{D} - \{d_i\}) &= \alpha \cdot (\mathcal{L}_1 + \mathcal{L}_2) \\ &\quad \cdot \sum_{i=2, i \neq j}^{k} d_i + \gamma \cdot \mathcal{J}_2 \\ &\quad \times \left(\sum_{i=1}^{j-2} \frac{d_{i+1}}{d_i} + \frac{d_{j+1}}{d_{j-1}} \right. \\ &\quad \left. + \sum_{i=j+1}^{k} \frac{d_{i+1}}{d_i} \right) \end{aligned}$$

---

[4] It has been shown in [9] that the dominance property for driver sizing solutions, which can be defined in a similar manner as the dominance property for wiresizing solution in Theorem 4, holds.

Therefore,

$$\begin{aligned}\Delta &= \mathrm{obj}_{T,k,\mathcal{W}}(\mathcal{D}) - \mathrm{obj}_{T,k-1,\mathcal{W}}(\mathcal{D} - \{d_i\})\\ &= \alpha \cdot (\mathcal{L}_1 + \mathcal{L}_2) \cdot d_j + \gamma \cdot \mathcal{J}_2 \cdot \left(\frac{d_j}{d_{j-1}} + \frac{d_{j+1}}{d_j} - \frac{d_{j+1}}{d_{j-1}}\right)\end{aligned}$$

Since $\frac{1}{d_j} - \frac{1}{d_{j-1}} \geq 0$, we have $\Delta > 0$, which contradicts the assumption that $\mathcal{D}$ is optimal. □

We shall show the following result which allows us to apply a similar approach as that in Section III to solve the SDWS-DP problem:

Theorem 11: (WS/DS Relation) Given two wire width assignments $\mathcal{W}$ and $\mathcal{W}'$ for a routing tree T driven by a chain of k drivers. Let $\mathcal{D}$ and $\mathcal{D}'$ be the optimal driver sizing solutions for $\mathcal{W}$ and $\mathcal{W}'$, respectively. If $C_{\mathrm{IL}}(T, \mathcal{W}) \geq C_{\mathrm{IL}}(T, \mathcal{W}')$, then $\mathcal{D}$ dominates $\mathcal{D}'$.

Proof: Let $\mathcal{D} = \{d_1, d_2, \ldots, d_k\}$ and $\mathcal{D}' = \{d'_1, d'_2, \ldots, d'_k\}$. First we show that $d_k \geq d'_k$.

$$\begin{aligned}&\mathrm{obj}_{T,k,\mathcal{W}}(\mathcal{D}) - \mathrm{obj}_{T,k,\mathcal{W}}(\mathcal{D}')\\ &\quad= \alpha \cdot (\mathcal{L}_1 + \mathcal{L}_2) \cdot \sum_{i=2}^{k}(d_i - d'_i) + \gamma \cdot \mathcal{J}_2\\ &\quad\times \left(\frac{C_{\mathrm{IL}}(T, \mathcal{W})}{C_g} \cdot \left(\frac{1}{d_k} - \frac{1}{d'_k}\right) + \sum_{i=1}^{k-1}\left(\frac{d_{i+1}}{d_i} - \frac{d'_{i+1}}{d'_i}\right)\right)\\ &\quad\leq 0\end{aligned}$$

$$\begin{aligned}&\mathrm{obj}_{T,k,\mathcal{W}'}(\mathcal{D}') - \mathrm{obj}_{T,k,\mathcal{W}'}(\mathcal{D})\\ &\quad= \alpha \cdot (\mathcal{L}_1 + \mathcal{L}_2) \cdot \sum_{i=2}^{k}(d'_i - d_i) + \gamma \cdot \mathcal{J}_2\\ &\quad\times \left(\frac{C_{\mathrm{IL}}(T, \mathcal{W}')}{C_g} \cdot \left(\frac{1}{d'_k} - \frac{1}{d_k}\right) + \sum_{i=1}^{k-1}\left(\frac{d'_{i+1}}{d'_i} - \frac{d_{i+1}}{d_i}\right)\right)\\ &\quad\leq 0\end{aligned}$$

Summing up the above two inequalities, we obtain:

$$(C_{\mathrm{IL}}(T, \mathcal{W}) - C_{\mathrm{IL}}(T, \mathcal{W}')) \cdot \left(\frac{1}{d_k} - \frac{1}{d'_k}\right) \leq 0$$

Since $C_{\mathrm{IL}}(T, \mathcal{W}) \geq C_{\mathrm{IL}}(T, \mathcal{W}')$, we conclude that $d_k \geq d'_k$.

Given $1 < j \leq k$, suppose $d_i \geq d'_i$ for $j \leq i \leq k$. We want to show that $d_{j-1} \geq d'_{j-1}$ and hence prove the dominance relation by induction. Let $\mathcal{D}/d_{1\cdots j-1} \rightarrow d'_{1\cdots j-1}$ denote the driver size assignment by replacing the sizes of drivers $D_{1\cdots j-1}$ in $\mathcal{D}$ by the corresponding sizes in $\mathcal{D}'$. Similarly, we define $\mathcal{D}'/d'_{1\cdots j-1} \rightarrow d_{1\cdots j-1}$ to be the driver size assignment by replacing the sizes of drivers $D_{1\cdots j-1}$ in $\mathcal{D}'$ by the corresponding sizes in $\mathcal{D}$.

$$\begin{aligned}&\mathrm{obj}_{T,k,\mathcal{W}}(\mathcal{D}) - \mathrm{obj}_{T,k,\mathcal{W}}(\mathcal{D}/d_{1\cdots j-1} \rightarrow d'_{1\cdots j-1})\\ &\quad= \alpha \cdot (\mathcal{L}_1 + \mathcal{L}_2) \cdot \sum_{i=2}^{j-1}(d_i - d'_i) + \gamma \cdot \mathcal{J}_2\\ &\quad\times \left\{\sum_{i=1}^{j-2}\left(\frac{d_{i+1}}{d_i} - \frac{d'_{i+1}}{d'_i}\right) + \left(\frac{d_j}{d_{j-1}} - \frac{d_j}{d'_{j-1}}\right)\right\} \leq 0\end{aligned}$$

$$\begin{aligned}&\mathrm{obj}_{T,k,\mathcal{W}'}(\mathcal{D}') - \mathrm{obj}_{T,k,\mathcal{W}'}(\mathcal{D}'/d'_{1\cdots j-1} \rightarrow d_{1\cdots j-1})\\ &\quad= \alpha \cdot (\mathcal{L}_1 + \mathcal{L}_2) \cdot \sum_{i=2}^{j-1}(d'_i - d_i) + \gamma \cdot \mathcal{J}_2\\ &\quad\times \left\{\sum_{i=1}^{j-2}\left(\frac{d'_{i+1}}{d'_i} - \frac{d_{i+1}}{d_i}\right) + \left(\frac{d'_j}{d'_{j-1}} - \frac{d'_j}{d_{j-1}}\right)\right\} \leq 0\end{aligned}$$

Summing up the above two inequalities, we obtain:

$$(d_j - d'_j) \cdot \left(\frac{1}{d_{j-1}} - \frac{1}{d'_{i-1}}\right) \leq 0$$

since $d_j - d'_j \geq 0$, we have $d_{i-1} \geq d'_{i-1}$. □

Furthermore, one can see that Formulation (24) implies that Theorems 2 through 5 in Sections III-A-2 and III-A-3 still apply to the optimal wiresizing solution under the combined delay and power optimization objective as defined in (24). Therefore, the same optimal wiresizing solution algorithm in [12] can be used to optimize $\mathrm{obj}_{T,k,\mathcal{D}}(\mathcal{W})$ in (24).

B. Bound Computation and Optimal Algorithm for SDWS-DP Problem

For a given stage number k, we taken the same approach as in solving the k-SDWS-D problem to compute the optimal solutions to the k-SDWS-DP problem: we first compute the upper and lower bounds of the optimal k-SDWS-DP solution. To compute the lower bound, we start with an initial wire width assignment in which all segments have minimum width wire. Based on the capacitive load of the routing tree, we compute a driver sizing solution using (30). Then, the optimal wiresizing solution based on the current driver sizing solution is computed using the algorithm in [12]. The process of alternative driver sizing and wiresizing is repeated until the wire sizing solutions do not change in consecutive iterations. The upper bound is computed similarly by starting with maximum wire width assignment. The algorithm outlined above is referred to as the *k-SDWS/DP LU-Bound* algorithm. Note that the driver sizing solution given by k-SDWS/DP LU-Bound algorithm is computed by solving (30) while the driver sizing solution given by k-SDWS/D LU-Bound algorithm is computed based on Theorem 1. We have the following result (similar to the SDWS-D problem):

Theorem 12: Given a chain of k drivers, the k-SDWS/DP LU-Bound algorithm computes the lower and upper bounds of an optimal k-SDWS-DP solution.

Proof: We use the same notations introduced in the proof of Theorem 6. Again, we prove the lower bound case only. We first show that $(\mathcal{D}_1, \mathcal{W}_1)$ is dominated by $(\mathcal{D}^*, \mathcal{W}^*)$.

Since $\mathcal{W}_0$ is dominated by $\mathcal{W}^*$, we have $C_{\mathrm{IL}}(T, \mathcal{W}_0) \leq C_{\mathrm{IL}}(T, \mathcal{W}^*)$. By Theorem 11 (instead of Theorem 1), $\mathcal{D}_1$ is dominated by $\mathcal{D}^*$, which implies that $d_k^1 \leq d_k^*$. According to Theorem 5, $\mathcal{W}_1$ (computed based on R_d^1) is dominated by $\mathcal{W}^*$ since $R_d^1 = R_{\min}/d_k^1 \geq R_{\min}/d_k^* = R_d^*$, where R_d^* is the resistance of driver k in the optimal solution.

Assuming that $(\mathcal{D}_i, \mathcal{W}_i)$ is dominated by $(\mathcal{D}^*, \mathcal{W}^*)$, we can show similarly that $(\mathcal{D}_{i+1}, \mathcal{W}_{i+1})$ is also dominated by $(\mathcal{D}^*, \mathcal{W}^*)$. Hence, according to mathematical induction, the solution computed is a lower bound of the optimal solution for a given drive stage number k. ☐

Given the lower and upper bounds $[d_k^l, d_k^u]$ of the k-th driver, the *k-SDWS/DP Optimal* algorithm tries every possible driver size in $[d_k^l, d_k^u]$ for the k-th driver size. Again, (30) (instead of Theroem 1) is used to compute the sizes of driver $D_2 \cdots D_{k-1}$ using $d_k \cdot C_g$ as the capacitive load that the $(k-1)$-stage drivers have to drive. For each possible d_k, the optimal wiresizing solution can still be computed using the algorithm in [12]. As in the SDWS-D problem, we have $d_k^l = d_k^u$ for most cases and a very small interval $[d_k^l, d_k^u]$ when $d_k^l \neq d_k^u$. We establish the following result which is similar to Theorem 7:

Theorem 13: Given a chain of k drivers and p possible sizes for the last driver and a routing tree with n segments and r possible wire widths, let $T(k)$ be the time taken to compute the sizes of drivers $D_2 \cdots D_{k-1}$ given the sizes of D_k by solving the systems of equations in (30). Then, the worst case time complexity to compute the optimal driver and wire sizes is $O(p \cdot (n^r + T(k)))$. ☐

Again, the factor $O(n^r)$ is the worst case time complexity for computing an optimal wiresizing solution. In practice, when dominance property is used to compute the lower and upper bounds of the optimal wiresizing solution, this term is reduced to $O(n^3 \cdot r)$ for almost all designs.

Due to the similarity between (15) and (24), optimal wiresizing solution under the combined objective of performance and power dissipation can be computed using the algorithm in [12]. However, computing an optimal driver sizing solution for a given capacitive load $C_{\mathrm{IL}}(T, \mathcal{W})$ is more difficult. In the SDWS-D algorithms, the driver sizing solution is obtained according to Theorem 1 by computing a fixed stage ratio. In the SDWS-DP algorithms, we have to solve the set of equations specified in (30). In our implementation, these equations are solved using MAPLE, a mathematical software for symbolic computation developed by University of Waterloo.

To obtain the optimal stage number k^*_{DP}, the *SDWS/DP Optimal* algorithm performs a search for $1 \leq k < k^{\mathrm{DP}}_{\mathrm{MAX}}$ where $k^{\mathrm{DP}}_{\mathrm{MAX}}$ is the smallest stage number such that capacitive load $C_{\mathrm{IL}}(T, \mathcal{W}_{\mathrm{MAX}})$ for a routing tree T with maximum wire width assignment $\mathcal{W}_{\mathrm{MAX}}$ does not have a monotone optimal solution for a chain of $k^{\mathrm{DP}}_{\mathrm{MAX}}$ drivers. The following results prove the correctness of the SDWS-DP Optimal algorithm.

Lemma 9: For a given capacitive load C_{IL} driven by a chain of $k \geq 2$ drivers, if there exists a monotone optimal k-SDWS-DP driver sizing solution, then there exists a monotone optimal k-SDWS-DP driver sizing solution for any capacitive load $C'_{\mathrm{IL}} > C_{\mathrm{IL}}$. Equivalently, if there exists no monotone optimal k-stage driver sizing solution to the k-SDWS-DP problem for capacitive load C_{IL}, then there exists no monotone optimal k-SDWS-DP driver sizing solution for any capacitive load $C'_{\mathrm{IL}} > C_{\mathrm{IL}}$.

Proof: It is trivially true for $k = 2$ (Theorem 11). Suppose this is true for $k = l$. Now, consider $k = l + 1$. Let d_{l+1} and d'_{l+1} be the optimal driver size for driver D_{l+1} driving C_{IL} and C'_{IL} respectively. According to Theorem 11, $d_{l+1} \leq d'_{l+1}$. By hypothesis, there exists a monotone optimal driver solution for a chain of l drivers driving a capacitive load of $d'_{l+1} \cdot C_g$. We only have to show that $d'_{l+1} > d'_l$. By (30),

$$\begin{aligned} d_l'^2 &= (d'_{l-1} \cdot d'_{l+1}) \Big/ \left(\frac{\alpha \cdot (\mathcal{L}_1 + \mathcal{L}_2) \cdot d'_{l-1}}{\gamma \cdot \mathcal{J}_2} + 1 \right) \\ &< d'_{l-1} \cdot d'_{l+1} \end{aligned}$$

Since $d'_l > d'_{l-1}$, we can conclude that $d'_{l+1} > d'_l$. Hence, there exists a monotone optimal driver sizing solution for a chain of k drivers. The other result follows directly. ☐

Lemma 10: For a given capacitive load $C_{\mathrm{IL}}(T, \mathcal{W})$, if there exists no monotone optimal driver sizing solution for a chain of k drivers, then there exists no monotone optimal driver sizing solution for a chain of $k + 1$ drivers.

Proof: Suppose there exists a monotone optimal driver sizing solution for a chain of $k+1$ drivers. Let $\mathcal{D}$ and $\mathcal{D}'$ denote the optimal driver sizing solution for the chain of k drivers and $k+1$ drivers, respectively. Note that $\mathcal{D}$ is a nonmonotone solution whereas $\mathcal{D}'$ is a monotone solution. First we show that $d'_{k+1} < \frac{C_{\mathrm{IL}}(T,\mathcal{W})}{C_g}$. According to (30),

$$\begin{aligned} d'^2_{k+1} &= \left(d'_k \cdot \frac{C_{\mathrm{IL}}(T, \mathcal{W})}{C_g} \right) \Big/ \left(\frac{\alpha \cdot (\mathcal{L}_1 + \mathcal{L}_2) \cdot d'_k}{\gamma \cdot \mathcal{J}_2} + 1 \right) \\ &< d'_k \cdot \frac{C_{\mathrm{IL}}(T, \mathcal{W})}{C_g} \end{aligned}$$

Since $d'_k < d'_{k+1}$, we conclude that $d'_{k+1} < \frac{C_{\mathrm{IL}}(T,\mathcal{W})}{C_g}$. Since d'_i's for $1 \leq i \leq k+1$ are monotone, there exists a monotone optimal solution for a chain of k drivers driving any capacitive load larger than $d'_{k+1} \cdot C_g$ (according to Lemma 9). However, this contradicts the assumption that there exists no monotone optimal solution for a chain of k drivers driving capacitive load $C_{\mathrm{IL}}(T, \mathcal{W})$. ☐

Theorem 14: (Optimality of SDWS/DP Optimal Algorithm): Let $k^{\mathrm{DP}}_{\mathrm{MAX}}$ be the smallest stage number such that capacitive load $C_{\mathrm{IL}}(T, \mathcal{W}_{\mathrm{MAX}})$ for a routing tree T with maximum wire width assignment $\mathcal{W}_{\mathrm{MAX}}$ does not have a monotone optimal solution for a chain of $k^{\mathrm{DP}}_{\mathrm{MAX}}$ drivers. There exists no monotone optimal driver sizing solution for any $k \geq k^{\mathrm{DP}}_{\mathrm{MAX}}$ cascaded drivers given any wire width assignments.

Proof: By Lemma 10, there exists no monotone optimal k-stage driver sizing solution for capacitive load $C_{\mathrm{IL}}(T, \mathcal{W}_{\mathrm{MAX}})$ for any $k \geq K^{\mathrm{DP}}_{\mathrm{MAX}}$. Since $C_{\mathrm{IL}}(T, \mathcal{W}_{\mathrm{MAX}}) \geq C_{\mathrm{IL}}(T, \mathcal{W})$ for any wire width assignment $\mathcal{W}$, we can conclude from Lemma 9 that there exists no monotone optimal k-stage driver sizing solution for $k \geq k^{\mathrm{DP}}_{\mathrm{MAX}}$ given any wiresizing solution (including the optimal wiresizing solution). ☐

To compute $k^{\mathrm{DP}}_{\mathrm{MAX}}$, we perform a linear search starting with stage number $k = 2$. By solving the system of equations given in (30) at each stage, we can verify if a monotone solution exists for the current stage number.[5] We define $k^{\mathrm{DP}}_{\mathrm{MAX}}$ to be the first k such that the monotone property does not hold. By Lemma 10, there exists no monotone k-SDWS-DP optimal driver sizing solution for any $k > k^{\mathrm{DP}}_{\mathrm{MAX}}$ cascaded drivers driving the capacitive load $C_{\mathrm{IL}}(T, \mathcal{W}_{\mathrm{MAX}})$. Since all possible wire width assignments are dominated by $\mathcal{W}_{\mathrm{MAX}}$,

---

[5] By using the *k-monotone* property defined in [9], we can speed up the verification process.

SDWS/DP Optimal Algorithm

```
Function SDWS/DP Optimal($T, R_{min}, \alpha, \gamma$)

    /* Given a tree $T$, the minimum width transistor resistance $R_{min}$, and the
    parameters $\alpha$ and $\gamma$, return the optimal number of stages $k$, the optimal
    driver sizing solution $\mathcal{D}$, and the optimal wire width assignment $\mathcal{W}$. */

    if $\alpha = 0$ then
        return SDWS/D Optimal($T, R_{min}$); /* Delay Optimization */
    if $\gamma = 0$ then
        return $k = 1$, $d_1 = 1$, $\mathcal{W} = \mathcal{W}_{MIN}$; /* Minimum Power Dissipation */
    Compute $k_{MAX}^{DP}$;
    Best_Combined_Objective ← $\infty$;
    for $k \leftarrow 1$ to $k_{MAX}^{DP} - 1$ do
        Compute optimal driver and wire sizing solution for $k$:
            $\{\mathcal{D}, \mathcal{W}\}$ ← k-SDWS/DP Optimal($T, R_{min}, k, \alpha, \gamma$);
        Current_Combined_Objective ← $obj(T, k, \mathcal{D}, \mathcal{W})$;
        if Current_Combined_Objective < Best_Combined_Objective then
            Best_Solution ← $\{k, \mathcal{D}, \mathcal{W}\}$;
    end for
    return Best_Solution;
end Function;
```

Fig. 7. The SDWS/DP Optimal algorithm.

TABLE I

TECHNOLOGY PARAMETERS FOR CAZM 0.5 μm CMOS DRIVERS [18]

Min Driver Resistance (Ω)	13598
Min Gate Capacitance (fF)	2.6802
Min Diffusion Capacitance (fF)	1.0403

TABLE II

TECHNOLOGY PARAMETERS BASED ON (a) MCM10 TECHNOLOGY [5] AND (b) CAZM 0.5 μm CMOS MODE [18]

Parameters	MCM	IC
Min Loading Capacitance (fF)	1000	2.6802
Wire Resistance ($\Omega/\square$)	0.02	0.044
Wire Capacitance (area) ($aF/\mu m^2$)	3.46	41.3
Fringing Capacitance (2 sides) ($aF/\mu m$)	50.4	150

there exists no monotone optimal driver sizing solution for any $k > k_{\mathrm{MAX}}^{\mathrm{DP}}$ cascaded drivers given any wire width assignments. Therefore, it is sufficient for the SDWS/DP Optimal algorithm to compute, for each $1 \leq k < k_{\mathrm{MAX}}^{\mathrm{DP}}$, the optimal wire and driver sizing solution (using the k-SDWS/DP Optimal algorithm). The best among the $k_{\mathrm{MAX}}^{\mathrm{DP}} - 1$ solutions is the optimal SDWS-DP solution. The SDWS/DP Optimal algorithm is given in Fig. 7.

V. EXPERIMENTAL RESULTS

We have implemented the optimal SDWS/D and SDWS/DP algorithms in ANSI C for the Sun SPARC station environment. The algorithm is tested on a number of MCM and IC design examples consisting of a chain of cascaded drivers driving 1) a single sink net through a long interconnect and 2) a multiple-sink net through a tree-structure interconnect. HSPICE is used to simulate the circuit for accurate timing and power simulation to verify both our models and algorithms. The technology parameters used by HSPICE are summarized in Table I and II.

The parameters for the driver are based on the CAZM 0.5 μm CMOS process model [18]. The minimum driver resistance is obtained for a minimum size transistor (width = 1.0 μm, length = 0.5 μm) through HSPICE simulation of the drain-to-source current available when the drain-to-source voltage for a n-device is 0.95 $\times V_{\mathrm{dd}}$ ($V_{\mathrm{dd}} = 5.0$ V). The minimum gate and diffusion capacitance is the gate and diffusion capacitance of a minimum size n-transistor. We assume in the experiments that drivers in all design examples (MCM and IC) are cascaded on-chip CMOS drivers. The minimum driver resistance, gate and diffusion capacitance are used to guide the design of drivers.

The minimum loading capacitance on a MCM substrate, which is the pad capacitance, is 1000 fF whereas the minimum loading capacitance on an IC chip is the minimum gate capacitance. The interconnect parameters are obtained from MCM10 model [5] and CAZM 0.5 μm CMOS process model [18].

A. Performance of SDWS/D Algorithm

In our experiments, we compare our SDWS/D solutions with the solutions obtained using a) driver sizing with constant stage ratio $s = e$ and minimum-wire-width (CDSMIN), b) optimal driver sizing based on Theorem 1 and minimum-wire-width (ODSMIN), and c) independent driver and wire sizing algorithm (DWSA), i.e., it performs driver sizing with constant stage ratio $s = e$ and optimal wiresizing [8]. The set of wire width allowed is $\{W_1, 2W_1, 3W_1, 4W_1\}$, where W_1 is the minimum width (10 μm in MCM10 technology and 0.95 μm in IC technology). Hence, every wire segment in CDSMIN and ODSMIN has width W_1.

For the experiment on MCM design examples, we assume that in each test net, the cascaded drivers are driving a) a sink through a 5 cm long interconnect, b) a 4-sink net randomly placed on 10 cm $\times$ 10 cm substrate, and c) a 8-sink net randomly placed on 10 cm $\times$ 10 cm substrate. In each case, the interconnect is divided into wire segments, each of length 100 μm and modeled by a π-type RC circuit, in order to model the distributed nature of interconnect. For each test circuit in b) and c), we performed two sets of experiments. In the first set of experiments, we let the farthest sink from the source be the critical sink. In the second set of experiments, we randomly generate two critical sinks for each 4-sink net and four critical sinks for each 8-sink net. The first driver in the chain is in turn driven by an ideal voltage source and the input signal is a square wave with rise and fall time of 1 ns and a period of 40 ns (25 MHz).

Table III summarizes the signal delays by the four algorithms. In both sets of experiments, our SDWS/D algorithm consistently outperforms the other three methods. Compared with the traditional methods of constant-ratio driver sizing or optimal driver sizing with uniform wire width (CDSMIN and ODSMIN), our SDWS/D solutions achieved an improvement of up to 49% in both cases. Compared with the recently reported DWSA method [8], our SDWS/D algorithm can reduce the delay by up to 11%.

We perform the same set of experiments on the following IC design examples: a) a sink through a 1 cm long interconnect, b) a 4-sink net randomly placed on 1 cm $\times$ 1 cm IC chip, and c) a 8-sink net randomly placed on 1 cm $\times$ 1 cm IC chip. In the IC

TABLE III

AVERAGE SIGNAL DELAY (ns) FOR (a) 1-CRITICAL-SINK NET AND (b) MULTICRITICAL-SINKS NET UNDER THE MCM10 TECHNOLOGY

Net	Single Critical Sink				Multiple Critical Sinks			
	CDSMIN	ODSMIN	DWSA	SDWS/D	CDSMIN	ODSMIN	DWSA	SDWS/D
1-sink	1.31	1.17	1.15	1.02	same as single critical sink			
4-sink	4.36	4.38	2.41	2.26	4.35	4.33	2.45	2.31
8-sink	5.54	5.53	2.82	2.80	4.05	3.63	2.24	2.14

TABLE IV

AVERAGE SIGNAL DELAY (ns) FOR (a) 1-CRITICAL-SINK NET AND (b) MULTICRITICAL-SINK NETS UNDER THE CAZM 0.5 μm CMOS TECHNOLOGY

Net	Single Critical Sink				Multiple Critical Sinks			
	CDSMIN	ODSMIN	DWSA	SDWS/D	CDSMIN	ODSMIN	DWSA	SDWS/D
1-sink	1.54	1.48	1.05	0.95	same as single critical sink			
4-sink	2.21	2.13	1.35	1.19	2.14	2.05	1.28	1.17
8-sink	2.53	2.47	1.44	1.30	1.91	1.83	1.22	1.09

TABLE V

POWER DISSIPATION (mW) AND DELAY (ns) FOR DIFFERENT α'S UNDER THE MCM10 TECHNOLOGY

α	$k = 2$		$k = 3$		$k = 4$		$k = 5$		$k = 6$	
	Power	Delay	Power	Delay	Power	Delay	Power	Delay	Power	Delay
0	31.42	9.075	81.08	3.787	128.5	2.785	174.9	2.488	211.0	2.411
0.1	28.89	9.076	69.56	3.863	100.3	2.808	109.2	2.525	113.2	2.417
0.2	28.07	9.077	51.66	4.156	74.31	3.010	95.02	2.648	97.09	2.545
0.3	27.79	9.121	47.90	4.448	65.18	3.212	76.79	2.772	78.14	2.761
0.4	27.02	9.352	43.43	4.750	58.11	3.444	65.87	3.148	66.92	3.186
0.5	26.00	9.756	39.49	5.298	50.71	3.870	59.70	3.278	59.32	3.449
0.6	25.03	10.085	36.30	5.751	45.25	4.439	51.19	3.852	51.39	3.938
0.7	23.29	10.929	33.44	6.407	40.09	5.100	45.39	4.256	45.82	4.498
0.8	20.22	13.178	29.08	7.724	36.70	5.478	39.81	5.190	40.05	5.218
0.9	16.64	16.696	25.63	9.103	33.63	6.244	33.63	6.505	34.74	6.601

design, we segment the interconnect into grid edges of length 10 μm and the sink capacitance is 10 times the minimum loading capacitance. The results are summarized in Table IV. Again, our SDWS/D algorithm consistently outperforms the other three methods. Compared with the CDSMIN, ODSMIN and DWSA methods, our SDWS/D solutions achieved an improvement of up to 49, 47, and 12%, respectively.

B. Performance of SDWS/DP Algorithm

We have studied the trade-off between power dissipation and delay using our SDWS/DP algorithm. For the same set of MCM 4-sink nets (with two critical sinks) in the previous experiment, we compute for each stage number $k =$ 2 to 6, a driver and wire sizing solution optimizing the $\text{obj}_\alpha(T, k, \mathcal{D}, \mathcal{W})$ for different α in (25). Table V summarizes the results for a set of α parameters. For each α, the table lists the power dissipation (mW) and delay (ns) of the driver and wire sizing solution for each stage number k = 2 to 6. As one can see, by adjusting the value of α in objective function (25), one can achieve smooth trade-off between power dissipation and performance.

In Fig. 8(a), the trade-off between delay and power dissipation for different k's is depicted by a set of curves. We can obtain a *contour* named *D/P-curve* which is dominated by the set of curves from above. Clearly, the D/P-curve represents best power and delay trade-off achieved by the optimal solutions of the SDWS/DP solutions. In Fig. 8(b), we compare the solutions obtained by CDSMIN, ODSMIN and DWSA with the D/P-curve by the SDWS/DP algorithm. We can observe that the D/P-curve is dominated by the curves representing the solution sets by CDSMIN, ODSMIN and DWSA. One can always select a SDWS solution from the D/P-curve to meet a delay target with minimum power dissipation or to meet a power budget with best performance.

TABLE VI

COMPARISONS OF POWER REQUIREMENT (mW) OF VARIOUS SOLUTIONS MEETING THE DELAY SPECIFICATION (ns) FOR MCM DESIGN

Delay Constraint (ns)	CDSMIN			ODSMIN			DWSA			SDWS-DP		
	Power	Delay	k	Power	Delay	k	Power	Delay	k	Power	Delay	k
10	29	7.67	5	29	10.00	2	30	7.31	5	26	9.10	3
5	44	4.99	7	83	4.63	4	54	3.86	6	43	4.74	3
4	-	-	-	-	-	-	54	3.86	6	50	3.87	4
3	-	-	-	-	-	-	86	2.69	7	77	2.77	5
2.5	-	-	-	-	-	-	139	2.45	8	113	2.41	6

TABLE VII

COMPARISONS OF POWER REQUIREMENT (mW) OF VARIOUS SOLUTIONS MEETING THE DELAY SPECIFICATION (ns) FOR IC DESIGN

Delay Constraint (ns)	CDSMIN			ODSMIN			DWSA			SDWS-DP		
	Power	Delay	k	Power	Delay	k	Power	Delay	k	Power	Delay	k
5	7.7	3.69	4	8.3	4.11	2	8.0	3.66	4	6.6	4.47	2
4	7.7	3.69	4	15.0	2.48	3	8.0	3.66	4	7.0	4.00	3
3	11.9	2.31	5	15.0	2.48	3	13.8	1.87	5	8.8	2.76	3
2	-	-	-	32.3	2.05	6	13.8	1.87	5	11.8	2.00	4
1.5	-	-	-	-	-	-	23.2	1.37	6	16.8	1.47	4

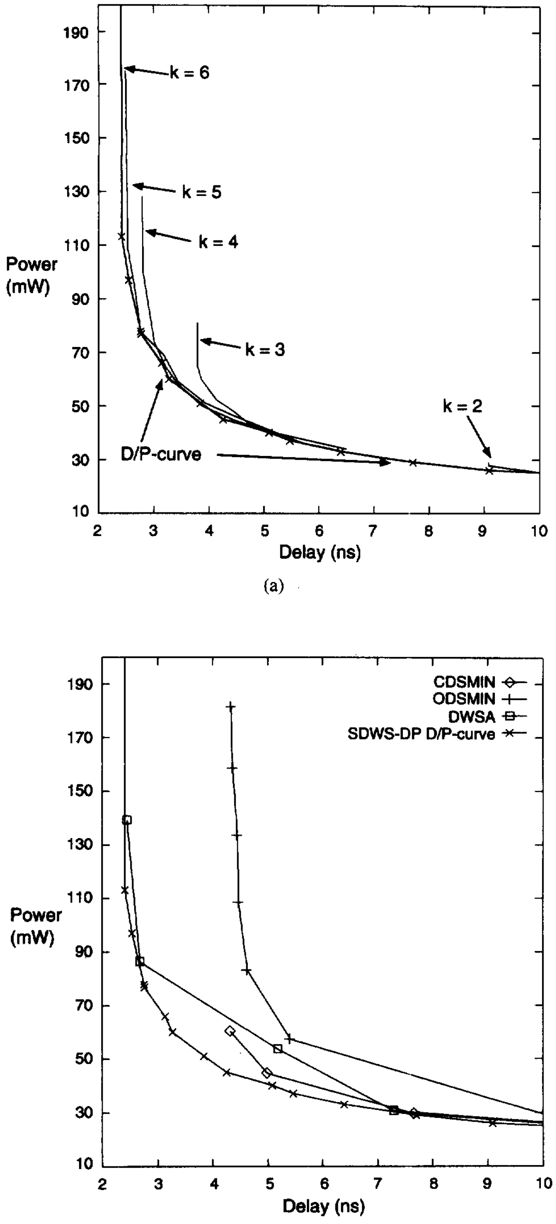
In Table VI, we compare the power requirement of the test circuit under different design methods (CDSMIN, ODSMIN, DWSA, and SDWS-DP) in order for the net to meet the delay specification. The table list a set of target delays that the net is expected to achieve and the most power economical solution by each design method under the performance requirement. An empty entry implies that the solution cannot meet the delay specification. We can observe that our SDWS solutions require least power while meeting the delay specification. Our SDWS solutions achieved an reduction in power dissipation by up to 10, 48, and 19% reduction, respectively, when compared with the CDSMIN, ODSMIN and DWSA methods. In addition, our SDWS solutions always require fewer stages of drivers (smaller k) which results in simpler layout design in practice.

We also evaluate the performance of our SDWS/DP algorithms on 4-sink nets (with two critical sinks) in IC routing used in the previous subsection. Table VII compares the power requirement of the test circuits under different design methods (CDSMIN, ODSMIN, DWSA and SDWS-DP) in order for the net to meet the delay specification. Our SDWS/DP algorithms can achieve up to 26, 63, and 36% reduction in power as compared to the CDSMIN, ODSMIN and DWSA methods, respectively.

We can observe that by selecting α carefully, we can minimize power dissipation while meeting the delay constraint. This suggests an approach to solve the following optimization problem which was raised in [8]: Given a routing tree T and a performance specification B, determine the number of stages k, the set of driver sizes $\mathcal{D}$, and a wiresizing solution $\mathcal{W}$ on T, such that the performance measure $t(T, k, \mathcal{D}, \mathcal{W})$ is bounded by B, i.e., $t(T, k, \mathcal{D}, \mathcal{W}) \leq B$ and the power dissipation $\text{Power}(T, k, \mathcal{D}, \mathcal{W})$ is minimized. We can solve the above problem by performing a binary search on α in (25) where $0 \leq \alpha \leq 1$.

VI. CONCLUSION AND FUTURE WORK

To the best of our knowledge, this is the first paper which presents in-depth study of the simultaneous driver and wire sizing problem and its effects on performance and power op-



(a)

(b)

Fig. 8. (a) D/P-curve containing the optimal SDWS/DP solutions. (b) Comparisons between CDSMIN, ODSMIN, DWSA and SDWS/DP solutions under the MCM10 technology.

timization. The results in this paper have shown convincingly that simultaneous driver and wire sizing can lead to significant reduction in the interconnect delay and power dissipation. In high performance systems where the delays due to long on-chip interconnect and chip-to-chip interconnects are critical, our optimal solutions allow smaller driver size and/or fewer number of cascaded drivers required to drive long interconnect with proper wiresizing. As a result, our SDWS solutions provide a low power interconnect design for high performance circuits.

One of the limitations of our formulation is the simplicity of the *RC* model for drivers. It has been shown that the *RC* delay model may have error up to 25% [19]. We would like to use a more accurate model, such as the slope model or the PR-slope model proposed in [19]. In addition, we would like to extend the short-circuit power formulation (20) to model the rise/fall time of the input signal to individual driver along the chain and investigate if the monotone and dominance properties, the WS/DS relation, and the optimality of the SDWS/DP Optimal algorithm still hold under such model.

We would also like to generalize the driver sizing and wiresizing problem such that long interconnect lines can be segmented and interleaved with optimal size repeaters. Also, as mentioned in Section I, interconnect topology optimization is another effective approach to reduce signal delay. A long term goal is to include wiresizing, driver sizing and also interconnect topology in the formulation of the power and performance optimization problem.

ACKNOWLEDGMENT

The authors would like to thank Dr. D. Zhou for providing the technology parameters for CAZM 0.5 μm CMOS model, and anonymous reviewers for their helpful comments.

REFERENCES

[1] C. J. Alpert, T. C. Hu, J. H. Huang and A. B. Kahng, "A direct combination of the Prim and Dijkstra constructions for improved performance-driven routing," in *Proc. IEEE Int. Symp. on Circuits and Syst.*, May 1993, pp. 1869–1872.

[2] H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*. Reading, MA: Addison-Wesley, 1990.

[3] K. D. Boese, A. B. Kahng, and G. Robins, "High-performance routing trees with identified critical sinks," in *Proc. ACM/IEEE Design Automation Conf.*, 1993, pp. 182–187.

[4] K. D. Boese, A. B. Kahng, B. A. McCoy, and G. Robins, "Rectilinear Steiner trees with minimum Elmore delay," in *Proc. ACM/IEEE Design Automation Conf.*, 1994, pp. 381–386.

[5] J. B. Burr, J. R. Burnham, and A. M. Peterson, "System-wide energy optimization in the MCM environment," in *Proc. IEEE MCM Conf.*, 1991, pp. 66–83.

[6] J. P. Cohoon and L. J. Randall, "Critical net routing," in *Proc. IEEE Int. Conf. on Computer Design*, 1991, pp. 174–177.

[7] J. Cong, A. B. Kahng, G. Robins, M. Sarrafzadeh, and C. K. Wong, "Provably good performance-driven global routing," *IEEE Trans. CAD*, vol. 11, no. 6, pp. 739–752, June 1992.

[8] J. Cong, C.-K. Koh, and K. S. Leung, "Wiresizing with driver sizing for performance and power optimization," in *Proc. 1994 Int. Workshop on Lower Power Design*, 1994, pp. 81–86.

[9] J. Cong and C.-K. Koh, "Simultaneous driver and wire sizing for performance and power optimization," UCLA Computer Science Dept. *Tech. Rep. CSD-940020*, Los Angeles, CA 90024, May 1994.

[10] ——, "Simultaneous driver and wire sizing for performance and power optimization," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, 1994, pp. 206–212.

[11] J. Cong, K. S. Leung, and D. Zhou, "Performance-driven interconnect design based on distributed RC delay model," in *Proc. ACM/IEEE Design Automation Conf.*, 1993, pp. 606–611.

[12] J. Cong and K. S. Leung, "Optimal wiresizing under the distributed Elmore delay model," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, 1993, pp. 634–639.

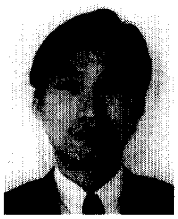
[13] ——, "Optimal wiresizing under Elmore delay model," to appear in *IEEE Trans. Computer-Aided Design*, Feb. 1995 (also available as UCLA Computer Science Dept. *Tech. Rep. CSD-930012*, Los Angeles, CA 90024, April 1993).

[14] W. C. Elmore, "The transient response of damped linear network with particular regard to wideband amplifier," *J. Appl. Phys.*, vol. 19, pp. 55–63, 1948.
[15] T. D. Hodes, B. A. McCoy, and G. Robins, "Dynamically-wiresized Elmore-based routing constructions," in *Proc. IEEE Int. Symp. Circuits and Syst.*, June 1994, pp. 463–466.
[16] X. Hong, T. Xue, E. S. Kuh, C. K. Cheng, and J. Huang, "Performance-driven Steiner tree algorithms for global routing," in *Proc. ACM/IEEE Design Automation Conf.*, 1993, pp. 177–181.
[17] H. C. Lin and L. W. Linholm, "An optimized output stage for MOS integrated circuits," *IEEE J. Solid-State Circ.*, vol. SC-10, no. 2, pp. 106–109, 1975.
[18] *MCNC Designers' Manual*, MCNC.
[19] J. K. Ousterhout, "Switch-level delay models for digital MOS VLSI," in *Proc. ACM/IEEE Design Automation Conf.*, 1984, pp. 542–548.
[20] S. S. Sapatnekar, "RC interconnect optimization under the Elmore delay model," in *Proc. ACM/IEEE Design Automation Conf.*, 1994, pp. 387–391.
[21] N. H. E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design: a Systems Perspective*, 2nd ed. Reading, MA: Addison-Wesley, 1993, pp. 229–231.

Jason Cong (M'90) received the B.S. degree in computer science from the Peking University in 1985, and the M.S. and Ph.D. degrees in computer science from the University of Illinois at Urbana-Champaign in 1987 and 1990, respectively.

Currently, he is an associate professor with the Computer Science Department of University of California, Los Angeles. From 1986 to 1990, he was a research assistant with the Computer Science Department, University of Illinois. He worked at the Xerox Palo Alto Research Center in the summer of 1987 and at the National Semiconductor Corporation in the summer of 1988. He has served on the program committees of a number of VLSI CAD Conferences, and chaired the Fourth ACM/SIGDA Physical Design Workshop. His research interests include computer-aided design of VLSI circuits, rapid system prototyping, and design and analysis of combinatorial and geometric algorithms.

Dr. Cong received the Best Graduate Award from the Peking University in 1985 and the Ross J. Martin Award for Excellence in Research from the University of Illinois at Urbana-Champaign in 1989. He also received the National Science Foundation Research Initiation Award in 1991 and the National Science Foundation Young Investigator Award in 1993. He is also the recipient of the Northrop Corporation Outstanding Junior Faculty Research Award from the UCLA in 1993.



Cheng-Kok Koh received the B.S. degree in computer science with first class honors from the National University of Singapore in 1992.

Currently, he is a research assistant with the Department of Computer Science of University of California, Los Angeles, where he is pursuing the Ph.D. degree. His research interests include computer-aided design of VLSI circuits, design and analysis of data structures and algorithms, and hypermedia systems.

Mr. Koh is a member of the Association for Computing Machinery.