# Simultaneous Driver and Wire Sizing for Performance and Power Optimization[*]

Jason Cong and Cheng-Kok Koh
Department of Computer Science
University of California, Los Angeles, CA 90024

## 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 3) 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 11%–47% and power dissipation by 26%–63% compared with existing design methods.*

## 1 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. 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 [6], AHHK trees [1], maximum performance trees [5], A-trees [9], low-delay trees [3], and IDW/CFD trees [12]. Moreover, the wiresizing algorithm in [9, 10] can further minimize interconnect delay by optimally assigning different wire width to each wire segment in the interconnect design.

Very recently, Cong, Koh and Leung [7] 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 sub-optimal solutions since driver 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. In Section 2, we present the general formulation of the simultaneous driver sizing and wiresizing problems under the two objective functions. In Section 3, we present results on optimal SDWS solutions for delay minimization. In Section 4, we present results on optimal SDWS solutions under the combined delay and power minimization objective. Section 5 shows the experimental results obtained by our SDWS algorithms and the comparative study with other existing methods. Section 6 concludes the paper. Due to page limitation, proofs of the results are omitted. Details are available in technical report [8]. The reader is also strongly recommended to be familiar with the results in [10], which are referred to several times in this paper.

## 2 Problem Formulation

### 2.1 Performance Optimization

Assume that we are given a routing tree $T$ implementing a signal net which consists of a source $N_+$, and a set of $m$ sinks $\{N_1, N_2, \cdots, N_m\}$. 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 $\{E_1, E_2, \cdots, E_n\}$ 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 $\{W_1, W_2, \cdots, W_r\}$ ($W_1 < W_2 < \cdots < W_r$). We use $w_{E_i}$ to denote the width of the wire segment $E_i$, $i = 1..n$.

Furthermore, we assume that the signal net is driven by a chain of cascaded drivers of $k$ stages at the source as shown in Figure 1. We use $\mathcal{D} = \{d_1, d_2, \cdots, d_k\}$ to denote a driver sizing solution, where $d_i$ denotes the size of driver $D_i$ at $i$-th stage ($i = 1..k$). 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.*

[*] This work is partially supported by ARPA/CSTO under Contract J-FBI-93-112, the National Science Foundation Young Investigator Award MIP9357582, and by a grant from Intel Corporation.

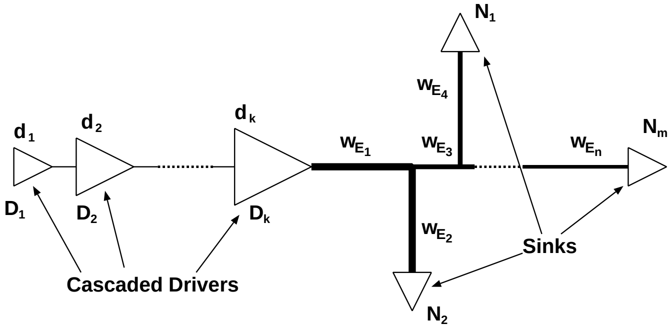

Figure 1: A $k$-stage cascaded drivers driving an interconnect tree $T$ with sinks $\{N_1, N_2, \cdots, N_m\}$. $w_{E_i}$ denotes the width of the wire segment $E_i$, $i = 1..n$ and $d_i$ denotes the size of driver $D_i$ at $i$-th stage, $i = 1..k$.

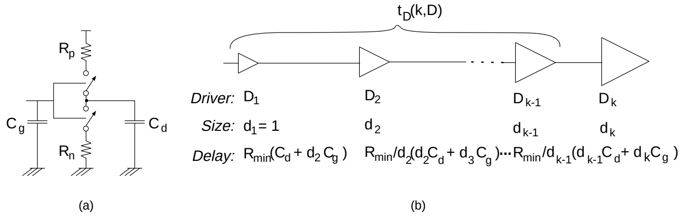

Figure 2: (a) A switch-level RC model of a driver (b) Inter-stage delay of a $k$-stage cascaded drivers until the gate of the $k$-th driver.

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:

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

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 [11].

### 2.1.1 RC Delay Model for Drivers

We estimate the delay of a driver based on a switch-level RC model of driver. Figure 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.

Figure 2(b) illustrates the delay due to a sequence of $k$ cascaded drivers $\mathcal{D}$. 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:

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

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

$$t_D(k, \mathcal{D}) \quad = \quad \mathcal{J}_1 + \mathcal{J}_2 \cdot \sum_{i=1}^{k-1} \frac{d_{i+1}}{d_i} \qquad (2)$$

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 2.1.2.

### 2.1.2 Distributed Elmore Delay Model for Interconnect

We use the distributed Elmore delay model [11] for interconnect delay measure. The formulations used in this section are based on those in [10]. The reader is strongly recommended to be familiar with the notations defined in [10]. In order to model a routing tree as a distributed RC circuit accurately, each wire segment in the routing plane is divided into a sequence of wires of unit length. In this case, $T$ consists of a set of unit-length wire segments, each may have a different width.

Given an grid edge $E$, we use $w_E$, $r_E$ and $c_E$ to denote the width, the interconnect resistance and capacitance, respectively, of the grid edge $E$. Assume that a unit-width unit-grid-length wire has wire resistance $r_0$, wire area capacitance $c_0$ and wire fringing capacitance $c_1$, then $r_E = \frac{r_0}{w_E}$ and $c_E = c_0 \cdot w_E + c_1$ for any grid edge $E$.

Let $R_d$ and $C_{DR}$ be the resistance and diffusion capacitance of the last driver in the driver chain respectively, i.e. $R_d = R_{min}/d_k$, $C_{DR} = C_d \cdot d_k$. Following a similar derivations in [10], the distributed Elmore delay $t_i(T, R_d, \mathcal{W})$ at $N_i$ under a given wiresizing solution $\mathcal{W}$ is:

$$\begin{aligned} & t_i(T, R_d, \mathcal{W}) \\ = \quad & \mathcal{K}_0^i + \mathcal{K}_1 \cdot \sum_{E \in T} w_E + \mathcal{K}_2 \cdot \sum_{E, E' \in T} f_i(E, E') \cdot \frac{w_{E'}}{w_E} + \\ & \mathcal{K}_3 \cdot \sum_{E, E' \in T} f_i(E, E') \cdot \frac{1}{w_E} + \mathcal{K}_4 \cdot \sum_{E \in T} g_i(E) \cdot \frac{1}{w_E} + \\ & \mathcal{K}_5 \cdot \sum_{E \in T} h_i(E) \cdot \frac{1}{w_E} \qquad (3) \end{aligned}$$

where, $\mathcal{K}_0^i = R_d \cdot C_{DR} + R_d \cdot \sum_{u \in sink(T)} c_u^s + R_d \cdot \sum_{E \in T} c_1 + \sum_{E \in P(N_+, N_i)} \frac{r_0 \cdot c_0}{2}$, $\mathcal{K}_1 = R_d \cdot c_0$, $\mathcal{K}_2 = r_0 \cdot c_0$, $\mathcal{K}_3 = r_0 \cdot c_1$, $\mathcal{K}_4 = r_0$, $\mathcal{K}_5 = \frac{r_0 \cdot c_1}{2}$, and the functions $f_i(E, E')$, $g_i(E)$ and $h_i(E)$ are defined as follows:

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

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

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

Let $sink(T)$ 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:

$$t(T, R_d, \mathcal{W}) = \sum_{N_i \in sink(T)} \lambda_i \cdot t_i(T, R_d, \mathcal{W}) \qquad (4)$$

where $\lambda_i$ is the weight of the delay penalty to sink $N_i$ [3, 10]. The larger $\lambda_i$ is, the more critical sink $N_i$ is. We normalize $\lambda_i$'s such that

$\sum_{N_i \in sink(T)} \lambda_i = 1$. We can rewrite Eqn. (4) as follows:

$$\begin{aligned} & t(T, R_d, \mathcal{W}) \\ = \; & \mathcal{K}_0 + \mathcal{K}_1 \cdot \sum_{E \in T} w_E + \mathcal{K}_2 \cdot \sum_{E,E' \in T} F(E,E') \cdot \frac{w_{E'}}{w_E} + \\ & \mathcal{K}_3 \cdot \sum_{E,E' \in T} F(E,E') \cdot \frac{1}{w_E} + \mathcal{K}_4 \cdot \sum_{E \in T} G(E) \cdot \frac{1}{w_E} + \\ & \mathcal{K}_5 \cdot \sum_{E \in T} H(E) \cdot \frac{1}{w_E} \qquad (5) \end{aligned}$$

where $\mathcal{K}_0 = \sum_{N_i \in sink(T)} \lambda_i \cdot \mathcal{K}_0^i$, $F(E,E') = \sum_{N_i \in sink(T)} \lambda_i \cdot f_i(E,E')$, $G(E) = \sum_{N_i \in sink(T)} \lambda_i \cdot g_i(E)$, and $H(E) = \sum_{N_i \in sink(T)} \lambda_i \cdot h_i(E)$.

### 2.1.3 SDWS-D Performance Measure

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 Eqn. (1) can be written as

$$\begin{aligned} & t(T,k,\mathcal{D},\mathcal{W}) \\ = \; & \left\{ \mathcal{J}_1 + \mathcal{J}_2 \cdot \sum_{i=1}^{k-1} \frac{d_{i+1}}{d_i} \right\} + \left\{ \mathcal{K}_0 + \mathcal{K}_1 \cdot \sum_{E \in T} w_E + \right. \\ & \mathcal{K}_2 \cdot \sum_{E,E' \in T} F(E,E') \cdot \frac{w_{E'}}{w_E} + \\ & \mathcal{K}_3 \cdot \sum_{E,E' \in T} F(E,E') \cdot \frac{1}{w_E} + \mathcal{K}_4 \cdot \sum_{E \in T} G(E) \cdot \frac{1}{w_E} + \\ & \left. \mathcal{K}_5 \cdot \sum_{E \in T} H(E) \cdot \frac{1}{w_E} \right\} \qquad (6) \end{aligned}$$

## 2.2 Performance and Power Optimization

Driver and wire sizing are effective approaches to reduce interconnect delay. However, larger driver size and additional routing capacitance also increases the power dissipation. In practice, high-speed circuit design requires a careful tradeoff 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 performance measure $obj(T,k,\mathcal{D},\mathcal{W})$ defined below is minimized:*

$$obj(T,k,\mathcal{D},\mathcal{W}) = \alpha \cdot Power(T,k,\mathcal{D},\mathcal{W}) + \gamma \cdot t(T,k,\mathcal{D},\mathcal{W}) \quad (7)$$

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

The *k-SDWS-DP* problem is similarly defined (as the k-SDWS-D problem) if we fix the number of stages $k$.

### 2.2.1 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) [14]. 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 [14]:

$$\begin{aligned} Power_{cap} &= f \cdot C_L \cdot V_{dd}^2 \\ Power_{sc} &= f \cdot \frac{\beta}{12} \cdot (V_{dd} - 2V_t)^3 \cdot t_{rf} \end{aligned}$$

where $f$ is the switching frequency of the input signal, $\beta$ 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 $D_i$ ($i = 1..k-1$), 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 $C_d \cdot d_k + C_{IL}(T,\mathcal{W})$ where $C_{IL}(T,\mathcal{W})$ is the total capacitance of interconnect tree $T$ (including the loading capacitance at the sinks):

$$C_{IL}(T,\mathcal{W}) = \sum_{u \in sink(T)} c_u^s + c_0 \cdot \sum_{E \in T} w_E + \sum_{E \in T} c_1$$

Hence the capacitive power of the cascaded drivers is

$$Power_{cap}(k,\mathcal{D},C_{IL}(T,\mathcal{W})) = \mathcal{L}_0 + \mathcal{L}_1 \cdot \left( \sum_{i=2}^{k} d_i + \frac{C_{IL}(T,\mathcal{W})}{C_g + C_d} \right)$$

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

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 [14]:

$$Power_{sc}(k,\mathcal{D},C_{IL}(T,\mathcal{W})) \;=\; \mathcal{L}_2 + \mathcal{L}_2 \cdot \sum_{i=2}^{k} d_i$$

where $\mathcal{L}_2 = f \cdot \frac{\beta_{min}}{12} \cdot (V_{dd} - V_t)^3 \cdot t_{rf}$.

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

$$\begin{aligned} & Power(T,k,\mathcal{D},\mathcal{W}) \\ = \; & \mathcal{L}_0 + \mathcal{L}_1 \cdot \frac{C_{IL}(T,\mathcal{W})}{C_g + C_d} + \mathcal{L}_2 + (\mathcal{L}_1 + \mathcal{L}_2) \cdot \sum_{i=2}^{k} d_i \quad (8) \end{aligned}$$

### 2.2.2 Trade-Off Between Performance and Power

Eqn. (7) formulates the trade-off between performance and power. Substituting the terms $Power(T,k,\mathcal{D},\mathcal{W})$ and $t(T,k,\mathcal{D},\mathcal{W})$ in Eqn. (7) with Eqns. (8) and (6), respectively, we can observe that 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 $C_{IL}(T,\mathcal{W})$. Suppose a wire-width assignment $\mathcal{W}$ is given for a routing tree $T$, the capacitive load $C_{IL}(T,\mathcal{W})$ can be computed. Eliminating the constant terms, the drivers are sized to minimize the following:

$$\begin{aligned} obj_{T,k,\mathcal{W}}(\mathcal{D}) \quad &= \quad \alpha \cdot (\mathcal{L}_1 + \mathcal{L}_2) \cdot \sum_{i=2}^{k} d_i + \\ & \gamma \cdot \mathcal{J}_2 \cdot \left( \frac{C_{IL}(T, \mathcal{W})}{d_k \cdot C_g} + \sum_{i=1}^{k-1} \frac{d_{i+1}}{d_i} \right) \end{aligned} \quad (9)$$

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:

$$\begin{aligned} & obj_{T,k,\mathcal{D}}(\mathcal{W}) \\ = \; & (\alpha \cdot \mathcal{L}_1 \cdot \frac{1}{C_g + C_d} \cdot c_0 + \gamma \cdot \mathcal{K}_1) \cdot \sum_{E \in T} w_E + \\ & \gamma \cdot \mathcal{K}_2 \cdot \sum_{E, E' \in T} F(E, E') \cdot \frac{w_{E'}}{w_E} + \\ & \gamma \cdot \mathcal{K}_3 \cdot \sum_{E, E' \in T} F(E, E') \cdot \frac{1}{w_E} + \\ & \gamma \cdot \mathcal{K}_4 \cdot \sum_{E \in T} G(E) \cdot \frac{1}{w_E} + \gamma \cdot \mathcal{K}_5 \cdot \sum_{E \in T} H(E) \cdot \frac{1}{w_E} \end{aligned} \quad (10)$$

This formulation is similar to Eqn. (5) 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 $\alpha$ and $\gamma$ in Eqn. (7) might be difficult. We find in our study that it is convenient to optimize the following objective:

$$\begin{aligned} & obj(T, k, \mathcal{D}, \mathcal{W}) \\ = \; & \alpha \cdot \frac{Power(T, k, \mathcal{D}, \mathcal{W})}{Power_{min}(T)} + (1 - \alpha) \cdot \frac{t(T, k, \mathcal{D}, \mathcal{W})}{t_{min}(T, \mathcal{D}, \mathcal{W})} \end{aligned} \quad (11)$$

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 3. Therefore, they can be treated as constants. In this case, we can easily adjust parameter $\alpha$ between 0 and 1 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.

# 3 Simultaneous Driver and Wire Sizing for Performance Optimization

## 3.1 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 Eqn. (6).

**Theorem 1** *[14] Given the loading capacitance $C_L$, the number of stages $k$ and the minimum gate capacitance $C_g$, the optimal stage ratio is $s = (\frac{C_L}{C_g})^{1/k}$.* □

**Theorem 2** *[10] Given a routing tree, the optimal wiresizing solution satisfies the separability, the monotone property and the dominance property.* □

The three properties still hold after we model the driver capacitance and fringing capacitance in interconnect delay formulation.

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 have shown in this work the following result that

**Theorem 3 (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 an important role in determining the lower and upper bounds of the optimal k-SDWS-D solutions in next subsection.

## 3.2 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 capacitive load of the routing tree. Based on Theorem 1, the optimal stage ratio, denoted by $s$, of the cascaded driver of $k$ stage can be computed and a driver sizing solution is obtained. Now, we perform delay optimal wiresizing [10] 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 Table 1. We have shown that

**Theorem 4** *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.* □

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.

## 3.3 Optimal SDWS-D Algorithm

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 [10] 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 the $(k-1)$-stage drivers have to drive. Based on this algorithm, we have shown that

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

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

  $\mathcal{W}_l \leftarrow$ 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 lower bound of the stage ratio 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 upper bound of the stage ratio of
    driver sizing solution and $\mathcal{W}_u$ as the
    upper bound of wire sizing solution;
end Function;
```

Table 1: The k-SDWS/D LU-Bound algorithm for computing lower and upper bounds of optimal SDWS solution for a given stage number $k$.

**Theorem 5** *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 3.2 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 run in $O(n^3 \cdot r)$ time based on effective lower and upper bound computation using the dominance property (Details of the optimal wiresizing algorithm can be found in [10]). Therefore, the *k-SDWS/D Optimal* algorithm runs in $O(n^3 \cdot r)$ time in practice.

Our *SDWS/D Optimal* algorithm for SDWS-D problem performs a linear search for the number of stages $k$ required, starting with $k = 1$. The process terminates when stage $k$ do not perform better than stage $k - 1$ (i.e. when adding an additional driver actually slows down the circuit).

# 4 Simultaneous Driver and Wire Sizing for Both Delay and Power Optimization

## 4.1 Optimal SDWS-DP Solutions

Differentiating $obj_{T,k,\mathcal{W}}(\mathcal{D})$ specified by specified by Eqn. (9) with respect to $d_i$, and setting $\frac{\partial obj_{T,k,\mathcal{W}}(\mathcal{D})}{\partial d_i} = 0$ for $2 \leq i \leq k$, we obtain a system of equations as follows:

$$\begin{aligned} \alpha \cdot (\mathcal{L}_1 + \mathcal{L}_2) + \gamma \cdot \mathcal{J}_2 \cdot \left( \frac{1}{d_{i-1}} - \frac{d_{i+1}}{d_i^2} \right) &= 0 \\ \text{for all } i = 2..k-1 & \qquad (12) \\ \alpha \cdot (\mathcal{L}_1 + \mathcal{L}_2) + \gamma \cdot \mathcal{J}_2 \cdot \left( \frac{1}{d_{k-1}} - \frac{C_{IL}(T, \mathcal{W})}{d_k^2 \cdot C_g} \right) &= 0 \end{aligned}$$

The set of equations which will be used for optimal driver sizing solution computation for a fixed loading capacitance $C_{IL}(T, \mathcal{W})$ when $\gamma > 0$ in Section 4.2.

In this paper, we have proved the following result:

**Theorem 6 (Monotone Property):** *For any given capacitive load $C_{IL}(T, \mathcal{W})$, any optimal driver sizing solution to the SDWS-DP problem under the combined delay and power optimization objective function specified by Eqn. (9) is monotone, i.e. $d_{i+1} > d_i$ for all $i = 1, 2, \cdots k - 1$.* □

The following result allows us to apply a similar approach as that in Section 3 to solve the SDWS-DP problem. First, we define the dominance property for the driver sizing solutions.

**Definition 1** *Given two driver sizing solutions $\mathcal{D} = \{d_1, d_2, \cdots, d_k\}$ and $\mathcal{D}' = \{d'_1, d'_2, \cdots, d'_k\}$, $\mathcal{D}$* dominates *$\mathcal{D}'$ if $d_i \geq d'_i$ for $1 \leq i \leq k$.*

**Theorem 7 (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 $C_{IL}(T, \mathcal{W}) \geq C_{IL}(T, \mathcal{W}')$, then $\mathcal{D}$ dominates $\mathcal{D}'$.* □

Also, one can see that Formulation (10) implies that Theorem 3 in Section 3.1 still apply to the optimal wiresizing solution under the combined delay and power optimization objective as defined in Eqn. (10). Therefore, the same optimal wiresizing solution algorithm in [10] can be used to optimize $obj_{T,k,\mathcal{D}}(\mathcal{W})$ in Eqn. (10).

## 4.2 Bound Computation and Optimal Algorithm for SDWS-DP Problem

For a given stage number $k$, we take 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 Eqn. (12). In our implementation, these equations are solved using MAPLE, a mathematical software for symbolic computation developed by University of Waterloo. Then, the optimal wiresizing solution based on the current driver sizing solution is computed using the algorithm in [10]. 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 (to differentiate it from the k-SDWS/D LU-Bound algorithm in which the driver sizing solution is computed differently). We have the following result (similar to the SDWS-D problem):

**Theorem 8** *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.* □

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, Eqn. (12) (instead of Theorem 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 [10]. 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 5:

**Theorem 9** *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 Eqn. (12). 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.

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

**Theorem 10** *Let $k_{MAX}$ be the smallest stage number such that capacitive load $C_{IL}(T, \mathcal{W}_{MAX})$ for a routing tree $T$ with maximum wire width assignment $\mathcal{W}_{MAX}$ does not have a monotone optimal solution for a chain of $k_{MAX}$ drivers. There exists no monotone optimal driver sizing solution for any $k \geq k_{MAX}$ cascaded drivers given any wire width assignments.* □

# 5 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 2 and 3.

The parameters for the driver are based on the CAZM 0.5$\mu m$ CMOS process model [13]. The minimum driver resistance is obtained for a minimum size transistor (width $= 1.0\mu m$, length $= 0.5\mu m$) through HSPICE simulation of the drain-to-source current available when the drain-to-source voltage for a n-device is $0.95 \times V_{dd}$ ($V_{dd} = 5.0V$). 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 [4] and CAZM 0.5$\mu m$ CMOS process model [13].

| | |
|---|---|
| Min Driver Resistance (Ω): | 13598 |
| Min Gate Capacitance ($fF$): | 2.6802 |
| Min Diffusion Capacitance ($fF$): | 1.0403 |

Table 2: Driver parameters for CAZM 0.5$\mu m$ CMOS [13].

| Parameters | MCM | IC |
|---|---|---|
| Min Loading Capacitance ($fF$): | 1000 | 2.6802 |
| Wire Resistance (Ω/□): | 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 |

Table 3: Interconnect parameters based on (a) MCM10 technology [4] and (b) CAZM 0.5$\mu m$ CMOS model [13].

## 5.1 Performance of SDWS/D Algorithm

In our experiments, we compare our SDWS/D wiresizing 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 [7]. The set of wire width allowed is $\{W_1, 2W_1, 3W_1, 4W_1\}$, where $W_1$ is the minimum width (10$\mu m$ in MCM10 technology and 0.95$\mu m$ in IC technology). Hence, every wire segment in CDSMIN and ODSMIN has width $W_1$.

For the experiment on MCM (IC) design examples, we assume that the cascaded drivers are driving (a) a sink through a 5$cm$ (1$cm$) long interconnect, (b) a 4-sink net randomly placed on 10$cm$ × 10$cm$ substrate (1$cm$ × 1$cm$ IC chip), and (c) a 8-sink net randomly placed on 10$cm$ × 10$cm$ substrate (1$cm$ × 1$cm$ IC chip). 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$MHz$). The sink capacitances for MCM and IC design are the minimum loading capacitance (1$pF$) and 10 times the minimum loading capacitance (26.8$fF$), respectively. In each case, the interconnect is divided into wire segments and modeled by a $\pi$-type RC circuit in order to model the distributed nature of interconnect. The grid edges are of length 100$\mu m$ and 10$\mu m$ for MCM and IC design, respectively. We randomly generate two critical sinks for 4-sink net and four critical sinks for 8-sinks net.

Table 4 summarizes the signal delays by the four design methods for MCM and IC design, respectively. 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 47% (45%), 47% (43%) and 11% (11%), respectively for MCM (IC) design.

## 5.2 Performance of SDWS/DP Algorithm

We have studied the trade-off between power dissipation and delay using our SDWS/DP algorithm. In Tables 5 and 6, 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 for MCM and IC design examples, respectively. Each table lists a set of target delays that the net is expected to achieve and the most power economical solution by each design

|  | MCM |  |  |  | IC |  |  |  |
|---|---|---|---|---|---|---|---|---|
| Net | CDSMIN | ODSMIN | DWSA | SDWS/D | CDSMIN | ODSMIN | DWSA | SDWS/D |
| 1-sink | 1.31 | 1.17 | 1.15 | 1.02 | 1.54 | 1.48 | 1.05 | 0.95 |
| 4-sink | 4.35 | 4.33 | 2.45 | 2.31 | 2.14 | 2.05 | 1.28 | 1.17 |
| 8-sink | 4.05 | 3.63 | 2.24 | 2.14 | 1.91 | 1.83 | 1.22 | 1.09 |

Table 4: Average signal delay ($ns$) for (a) MCM10 technology and (b) CAZM $0.5\mu m$ CMOS technology.

| Delay | CDSMIN |  |  | ODSMIN |  |  | DWSA |  |  | SDWS-DP |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Constraint ($ns$) | 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 5: Comparisons of power requirement ($mW$) of various solutions meeting the delay specification ($ns$) for MCM design.

| Delay | CDSMIN |  |  | ODSMIN |  |  | DWSA |  |  | SDWS-DP |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Constraint ($ns$) | 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 |

Table 6: Comparisons of power requirement ($mW$) of various solutions meeting the delay specification ($ns$) for IC design.

method under the performance requirement. An empty entry implies that the solution cannot meet the delay specification. The SDWS solutions are obtained by choosing suitable $\alpha$ in objective function $obj(T, k, \mathcal{D}, \mathcal{W})$ specified by Eqn. (11) using binary search. 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% (26%), 48% (63%) and 19% (36%) reduction, respectively, when compared with the CDSMIN, ODSMIN and DWSA methods for MCM (IC) design. In addition, our SDWS solutions always require fewer stages of drivers (smaller $k$) which results in simpler layout design in practice.

## 6 Conclusions

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 effect on performance and power optimization. Our SDWS solutions provide a low power interconnect design solution to high performance circuits.

## 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", *Proc. IEEE Int'l Symp. on Circuits and Systems*, May 1993, pp. 1869-1872.

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

[3] K. D. Boese, A. B. Kahng, and G. Robins, "High-Performance Routing Trees With Identified Critical Sinks", *Proc. ACM/IEEE Design Automation Conf.*, 1993, pp. 182-187.

[4] J. B. Burr, J. R. Burnham, and A. M. Peterson, "System-wide Energy Optimization in the MCM Environment", *Proc. IEEE MCM Conf.*, 1991, pp. 66-83.

[5] J. P. Cohoon and L. J. Randall, "Critical Net Routing", *Proc. IEEE Int'l. Conf. on Computer Design*, 1991, pp. 174-177.

[6] J. Cong, A. B. Kahng, G. Robins, M. Sarrafzadeh, and C. K. Wong, "Provably Good Performance-Driven Global Routing", *IEEE Trans. on CAD*, 11(6), June 1992, pp. 739-752.

[7] J. Cong, C.-K. Koh, and K. S. Leung, "Wiresizing with Driver Sizing for Performance and Power Optimization", *Proc. 1994 Int'l Workshop on Lower Power Design*, 1994, pp. 81–86.

[8] J. Cong and C.-K. Koh, "Simultaneous Driver and Wire Sizing for Performance and Power Optimization", *UCLA Computer Science Department Tech. Report CSD-940020*, Los Angeles, CA 90024, May 1994.

[9] J. Cong, K. S. Leung, and D. Zhou, "Performance-Driven Interconnect Design Based on Distributed RC Delay Model", *Proc. ACM/IEEE Design Automation Conf.*, 1993, pp. 606-611.

[10] J. Cong and K. S. Leung, "Optimal Wiresizing Under the Distributed Elmore Delay Model", *Proc. IEEE Int'l. Conf. on Computer-Aided Design*, 1993, pp. 634-639 (full version accepted for publication in *IEEE Trans. on CAD* and available as *UCLA Computer Science Department Tech. Report CSD-930012*, Los Angeles, CA 90024, April 1993).

[11] W. C. Elmore, "The Transient Response of Damped Linear Network with Particular Regard to Wideband Amplifier", *J. Applied Physics*, 19(1948), pp. 55-63.

[12] X. Hong, T. Xue, E. S. Kuh, C. K. Cheng, and J. Huang, "Performance-Driven Steiner Tree Algorithms For Global Routing", *Proc. ACM/IEEE Design Automation Conf.*, 1993, pp. 177-181.

[13] *MCNC Designers' Manual*, MCNC.

[14] N. H. E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design: a Systems Perspective – 2nd ed*, Addison-Wesley, 1993.