

Figure 4: Tradeoff between delay and power dissipation for (a) a 9-pin net on a IC chip, and (b) a 9-pin net on a MCM substrate

words, for the same delay requirement, our solutions consume less power. For example, for the MCM 9-pin net, a delay of $\approx 20ns$ can be achieved with a 5-stage cascaded drivers and a power dissipation of $31mW$ if wiresizing is used. However, for minimum-width interconnect, a 6-stage cascaded drivers and a power dissipation of $> 35mW$ is required to meet the similar requirement.

- Our simultaneous driving sizing and wiresizing approach can reduce the delay by up to 40% when compared to the conventional method of sizing drivers only with same amount of power dissipation. For example, in Figure 3(b), with the same amount of power dissipation of $15mW$, our DWSA solution achieves a delay of only $3.8ns$ whereas the MIN solution has a delay of $> 6ns$.
- As indicated by the steep slope of the curve when the delay is small, there is a lower limit of delay achievable by driver sizing approach or simultaneous driver sizing and wiresizing approach. Near this limit, a substantial amount of power is required for an insignificant amount of reduction of signal delay. The advantage of our approach is that the limiting delay by DWSA algorithm is up to 44% lower than that of the conventional method of driver sizing only.

5 Conclusion and Future Work

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

In this work, driver sizing and wiresizing are done in two separate steps. We are in the process of developing an optimal or near-optimal algorithm to perform driver sizing and wiresizing simultaneously for both power and performance optimization. We are also developing an accurate analytical formulation of the relationship between power and delay. 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.

As mentioned in Section 1, 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.

Acknowledgments

This work is partially supported by ARPA/CSTO under Contract J-FBI-93-112, the National Science Foundation Young Investigator Award, and by a grant from Intel Corporation. We would like to thank Dr. Dian Zhou for providing us the technology parameters for CAZM $0.5\mu m$ CMOS model.

References

- 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.
- H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*, Addison-Wesley, 1990.
- 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.
- K. D. Boese, A. B. Kahng, Bernard A. McCoy, and G. Robins, "Toward Optimal Routing Trees", *4th ACM/SIGDA Physical Design Workshop*, 1993.
- 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.
- J. P. Cohoon and L. J. Randall, "Critical Net Routing", *Proc. IEEE Int'l. Conf. on Computer Design*, 1991, pp. 174-177.
- 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.
- 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.
- 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 in submission to *IEEE Trans. on CAD*).
- W. E. Donath, R. J. Norman, B. K. Agrawal, and S. E. Bello, "Timing Driven Placement Using Complete Path Delays", *Proc. ACM/IEEE Design Automation Conf.*, 1990, pp. 84-89.
- W. C. Elmore, "The Transient Response of Damped Linear Network with Particular Regard to Wideband Amplifier", *J. Applied Physics*, 19(1948), pp. 55-63.
- 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.
- MCNC Designers' Manual*, MCNC.
- S. Sutanthavibul and E. Shragowitz, "An Adaptive Timing-Driven Layout for High Speed VLSI", *Proc. ACM/IEEE Design Automation Conf.*, 1990, pp. 90-95.

model. We assume in the experiments that drivers in MCM are cascaded CMOS drivers with the same IC technology parameters. However, the drivers have additional capacitance due to the pad which contributes to the $1000fF$ loading capacitance. Note that the fringing effect is not accounted for in our delay formulation (7). The wiresizing solutions are obtained by lumping the fringing capacitance with the load capacitance. Hence, the wiresizing solutions are not exactly “optimal.”²

In our experiments, we compare our DWSA wiresizing solutions with the minimum-width solution (MIN). The set of wire width allowed is $\{W_1, 2W_1, 3W_1, 4W_1\}$, where W_1 is the minimum width ($0.95\mu m$ in IC and $10\mu m$ in MCM10). Hence, every wire segment in MIN has width W_1 .

4.1 Single-Sink Net

In this experiment, we assume that the cascaded drivers are driving a sink through a $2cm$ long interconnect³. 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 $1ns$ and a period of $40ns$ ($25MHz$).

A stage ratio of ϵ is used to size the chain of drivers. We tested our algorithm on two cases: (i) when the sink capacitance is small (10 times the minimum transistor size), and (ii) when the sink capacitance is large ($1pF$), which may correspond to an I/O pad. The signal delay (time taken to reach 90% of its final value) and power dissipation is computed using HSPICE. Table 3 and 4 summarize the signal delay and power dissipation.

Number of Drivers	MIN		DWSA	
	Delay	Power	Delay	Power
1,2	>20	-	> 20	-
3	9.19	4.5	9.19	4.5
4	5.45	7.0	4.42	8.2
5	4.35	9.9	2.74	13.9
6	4.28	14.8	2.41	21.1
7	4.16	30.6	2.34	36.5
8	4.37	73.6	2.48	78.6

Table 3: Signal delay (ns) and power dissipation (mW) for a single pin net with a loading capacitance of $26.8fF$.

Number of Drivers	MIN		DWSA	
	Delay	Power	Delay	Power
1,2	> 20	-	>20	-
3	11.91	5.0	11.70	5.3
4	7.42	7.6	5.87	9.0
5	6.36	10.4	3.81	14.5
6	6.19	15.1	3.65	21.5
7	6.04	30.8	3.48	36.7
8	6.25	73.7	3.70	78.8

Table 4: Signal delay (ns) and power dissipation (mW) for a single pin net with a loading capacitance of $1pF$.

We can see from Table 3 and 4 that our driver sizing and wiresizing solutions cut down the number of cascaded drivers required to achieve similar delay. This effectively leads to a

²Our recent analysis showed that the three properties of optimal wiresizing solutions for performance optimization still hold when we include fringing effect in the Elmore delay model. We are in the process of incorporating the results in our DWSA algorithm.

³Assuming a chip size of $1cm \times 1cm$. A $2cm$ wire is the longest possible point-to-point interconnection.

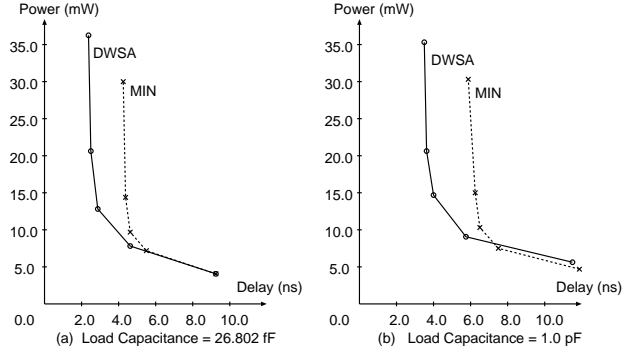


Figure 3: Tradeoff between delay and power dissipation for (a) a small load capacitance of $26.8fF$, and (b) a large load capacitance of $1.0pF$.

reduction in power dissipation. We show the tradeoff between power dissipation and signal delay in Figure 3.

4.2 Multiple-Sinks Net

We replace the single sink net with a multiple-sinks net connected by a tree topology. We tested our algorithm on two cases: (i) a 9-pin net on a $0.5cm \times 0.5cm$ IC chip, and (ii) a 9-pin net on a $10cm \times 10cm$ MCM substrate. In either case, the loading capacitance at a sink is the minimum loading capacitance in the corresponding technology. For the first case, the input signal is identical to the input signal in the single sink net experiment ($25MHz$). In the latter case, the input signal is a square wave with rise and fall time of $1ns$ and a period of $100ns$ ($10MHz$). In both cases, five 9-pin nets were generated.

The experimental results are summarized in Table 5 and 6, and illustrated graphically in Figure 4.

Number of Drivers	MIN		DWSA	
	Delay	Power	Delay	Power
1,2	> 20	-	> 20	-
3	10.62	5.2	10.62	5.2
4	4.76	8.7	4.74	8.8
5	3.08	13.2	2.44	14.9
6	2.76	18.8	1.61	24.7
7	2.77	33.0	1.51	41.5

Table 5: Average Delay (ns) and power (mW) for a 9-pin net on an IC chip

Number of Drivers	MIN		DWSA	
	Delay	Power	Delay	Power
1-3	> 50	-	> 50	-
4	46.95	16.7	44.10	17.3
5	26.53	27.7	21.62	31.0
6	21.71	35.8	13.55	47.9
7	20.85	43.4	9.91	73.0
8	20.43	65.2	8.56	104.4
9	20.57	142.4	8.69	171.3

Table 6: Average Delay (ns) and power (mW) for a 9-pin net on a MCM substrate

We can observe the following results in both experiments:

1. The curve corresponding to conventional method (MIN) dominates the curve by our DWSA solution. In other

Theorem 2 For any given tree T , there exists a monotone optimal width assignment \mathcal{W}^* . \square

3.2.3 Dominance Property

Definition 3 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 4 Given a routing tree T , a wire width assignment \mathcal{W} on 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 (7), subject to the fixed assignment of \mathcal{W} on the other segments.

Theorem 3 Let \mathcal{W}^* be an optimal width assignment. If a width assignment \mathcal{W} dominates \mathcal{W}^* , then any local refinement of \mathcal{W} still dominates \mathcal{W}^* . Similarly, if a width assignment \mathcal{W} is dominated by \mathcal{W}^* , then any local refinement of \mathcal{W} is dominated by \mathcal{W}^* . \square

The separability, the monotone property, and the dominance property played an important role in the performance optimal wiresizing algorithm presented in next subsection.

3.3 Optimal Wiresizing Algorithm for Performance Optimization

We first introduce the notion of a *single-stem tree* used in the following discussion. A single-stem tree is a tree with only one segment (called the *stem segment* of that tree) incident on its root. We use $SST(E)$ to denote the single-stem tree with stem E .

According to the separability, once E and every segment in $Ans(E)$ are assigned the appropriate widths, the optimal wire width assignment for the single-stem subtrees $SST(E_{c1})$, $SST(E_{c2})$, \dots , $SST(E_{cb})$ of the tree $SST(E)$ (with respect to the width assignment of E and segments in $Ans(E)$) can be *independently* determined, where the segments E_{c1}, \dots, E_{cb} are the children of E .

Assume we are given a single-stem tree with stem E , and a set of possible widths $\{W_1, W_2, \dots, W_r\}$, we can determine the optimal assignment \mathcal{W}^* on $T(E)$ by enumerating all the possible width assignments of E . For each of the possible width assignment W_k of E ($1 \leq k \leq r$), we determine the optimal assignment for each single-stem subtree $SST(E_{ci})$ ($1 \leq i \leq b$) of $SST(E)$ independently by recursively applying the same procedure to each $SST(E_{ci})$ with $\{W_1, W_2, \dots, W_k\}$ as the set of possible widths (to guarantee the monotone property). The optimal assignment for E is the one which gives the smallest total delay.

If the original routing tree T is not a single-stem tree, however, we can decompose T into b single-stem trees, where b is the degree of the root of T , and apply the algorithm to each individual single-stem tree separately. This is called the Optimal Wiresizing Algorithm with the Elmore Delay Model (OWSA/ED).

Theorem 4 Given a routing tree with n segments and r possible wire widths, the worst case time complexity of OWSA/ED is $O(n^r)$. \square

In essence, OWSA/ED enumerates *all* the possible combinations of monotone wire width assignments along every source-to-leaf path in the routing tree. The complexity indeed can grow exponentially with respect to r (which is usually a small constant in practice). This is the case when

the tree is simply a chain of segments, where the total number of possible assignments evaluated by OWSA/ED equals to $\binom{n+r-1}{r-1} = \Omega(n^{r-1})$.

The results in [9] showed that significant speedup of the optimal wiresizing algorithm can be achieved when we use a greedy wiresizing algorithm to compute the lower and upper bound of each segment in the optimal wiresizing solution. The Greedy Wiresizing Algorithm (GWSA/ED) works as follows: Starting with an initial wire width assignment (say, all segments have the minimum width), we traverse the routing tree and perform a local refinement on each segment whenever possible. This process is repeated until no improvement can be achieved on any segment in the last round of traversal. The GWSA/ED algorithm has a worst case complexity of $O(n^3 \cdot r)$.

According to the dominance property, if we start with the minimum-width assignment where each segment has the minimum wire width (which is dominated by the optimal solution \mathcal{W}^*), the resulting assignment computed by the GWSA/ED algorithm gives a lower bound of the optimal width for each segment. Similarly, the algorithm gives an upper bound of the optimal width for each segment if we start with the maximum-width assignment.

The GWSA/ED and OWSA/ED algorithms can therefore be combined as follows: First, we obtain the lower and upper bounds $W_{L(E)}$ and $W_{U(E)}$ of each wire segment E using the greedy algorithm. Then, we run the OWSA/ED algorithm with lower and upper bounds on each segment E such that $W_{L(E)} \leq w_E \leq W_{U(E)}$. Since the lower and upper bounds of each segment obtained from the greedy algorithm are very close or even identical in most cases, the total number of candidate assignments to be examined by the combined algorithm is much smaller than that by the OWSA/ED algorithm alone. The readers may refer to [9] for more details.

4 Experimental Results

We have implemented the optimal OWSA/ED and DWSA algorithms in ANSI C for the Sun SPARC station environment. The algorithm is tested on a simple circuit 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. In each case, the interconnect is divided into wire segments, each of length $25\mu\text{m}$ and modeled by a π -type circuit, in order to model the distributed nature of interconnect. HSPICE is used to simulate the circuit using the technology parameters summarized in Table 2.

Parameters	IC	MCM
Min Driver Resistance (Ω):	13598	13598
Min Loading Capacitance (fF):	2.6802	1000
Wire Resistance (Ω/\square):	0.044	0.02
Wire Capacitance (area) ($aF/\mu\text{m}^2$):	41.3	3.46
Fringing Capacitance (2 sides) ($aF/\mu\text{m}$):	150	50.4

Table 2: Technology parameters based on (a) IC technology CAZM $0.5\mu\text{m}$ CMOS model [13] (b) MCM technology [5].

The IC technology parameters are based on the CAZM $0.5\mu\text{m}$ CMOS process model. The minimum driver resistance is obtained for a minimum size transistor (width = $1.0\mu\text{m}$, length = $0.5\mu\text{m}$) 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 loading capacitance is the gate capacitance of a minimum size n-transistor. The MCM parameters are obtained from MCM10

$$\begin{aligned} & \mathcal{K}_4 \cdot \sum_{E, E' \in T, E \neq E'} f_i(E, E') \cdot \frac{w_{E'}}{w_E} + \\ & \mathcal{K}_5 \cdot \sum_{E \in T} g_i(E) \cdot \frac{1}{w_E} \end{aligned} \quad (4)$$

where $\mathcal{K}_{1i} = \sum_{E \in P(N_+, N_i)} \frac{r_0 \cdot c_0}{2}$, $\mathcal{K}_2 = R_d \cdot \sum_{E \in T} c_E^s$, $\mathcal{K}_3 = R_d \cdot c_0$, $\mathcal{K}_4 = r_0 \cdot c_0$, $\mathcal{K}_5 = r_0$, and the functions $f_i(E, E')$ and $g_i(E)$ are defined as follows:

$$\begin{aligned} f_i(E, E') &= \begin{cases} 1 & \text{if } E \in P(N_+, N_i) \text{ and } E' \in Des(E) \\ 0 & \text{otherwise} \end{cases} \\ g_i(E) &= \begin{cases} \sum_{v \in sink(E)} c_v^s & \text{if } E \in P(N_+, N_i) \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

where $Des(E)$ is the set of segments in the subtree “rooted” at E (excluding E), and $Ans(E)$ is the set $\{E' | E \in Des(E')\}$ (again, excluding E). A careful study of f_i 's and g_i 's reveals that:

$$\begin{aligned} f_i(E_1, E_2) &\geq f_i(E_1, E'_2) && \text{if } E_2 \in Des(E'_2) \\ f_i(E_1, E_2) &\geq f_i(E'_1, E_2) && \text{if } E_1 \in Ans(E'_1) \\ g_i(E_1) &\geq g_i(E'_1) && \text{if } E_1 \in Ans(E'_1) \end{aligned} \quad (5)$$

2.3 Multiple Critical Sinks Formulation

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:

$$t(\mathcal{W}) = \sum_{N_i \in sink(T)} \lambda_i \cdot t_i(\mathcal{W}) \quad (6)$$

where λ_i is the weight of the delay penalty to sink N_i . The larger λ_i is, the more critical sink N_i is. We normalize λ_i 's such that $\sum_{N_i \in sink(T)} \lambda_i = 1$. We can rewrite (6) as follows:

$$\begin{aligned} t(\mathcal{W}) &= \sum_{N_i \in sink(T)} \lambda_i \cdot \mathcal{K}_{1i} + \mathcal{K}_2 + \mathcal{K}_3 \cdot \sum_{E \in T} w_E + \\ & \mathcal{K}_4 \cdot \sum_{E, E' \in T, E \neq E'} F(E, E') \cdot \frac{w_{E'}}{w_E} + \\ & \mathcal{K}_5 \cdot \sum_{E \in T} G(E) \cdot \frac{1}{w_E} \end{aligned} \quad (7)$$

where $F(E, E') = \sum_{N_i \in sink(T)} \lambda_i \cdot f_i(E, E')$, and $G(E) = \sum_{N_i \in sink(T)} \lambda_i \cdot g_i(E)$. We can show that F 's and G 's have *exactly* the same properties as f_i 's and g_i 's in (5). Note that the first two terms of (7) are constant given a routing topology. In the remaining sections, we shall use the multiple-critical-sink formulation (7) without the first two terms in our wiresizing algorithm for performance optimization.

3 The DWSA Heuristic Algorithm

3.1 Algorithm Overview

In this section, we present a simple heuristic algorithm which performs both driver sizing and wiresizing for performance and power optimization. We assume that the first driver in the chain is a minimum size transistor of resistance R_{min} . In addition, we assume that the sequence of drivers increases in size gradually with a constant stage ratio s . Note that such a driver sizing solution is optimal when $s = e$, the base of the natural logarithm [2], if we do not consider wiresizing at the same time.

Driver Sizing and Wiresizing Algorithm

```

Function DWSA( $T, B, R_{min}, s$ )
 $k \leftarrow 1$ ;
Previous_Delay  $\leftarrow \infty$ ;
while true
   $R_d \leftarrow \frac{R_{min}}{s^{(k-1)}}$ ;
   $\mathcal{W} \leftarrow \text{PerformanceOptimalWiresizing}(R_d, T)$ ;
  Current_Delay  $\leftarrow t(k, \mathcal{D}, \mathcal{W})$ ;
  if Current_Delay  $\leq B$  then
    return  $k, \mathcal{W}$ 
  end if
  if Current_Delay  $\geq$  Previous_Delay then
    return no feasible solution
  end if
   $k \leftarrow k + 1$ ;
  Previous_Delay  $\leftarrow$  Current_Delay;
end while
end Function;

```

Table 1: The heuristic driver sizing and wiresizing algorithm (DWSA).

Our heuristic Driver- and Wire-Sizing Algorithm (DWSA) performs a linear search for the number of stages k required, starting with $k = 1$. At each iteration, we compute the optimal wiresizing solution for performance optimization (algorithm to be described in Section 3.3) based on the driver resistance of the last driver, which is reduced by a factor of s at each iteration. Based on the number of stages k and the wiresizing solution, we compute the delay of the current configuration using equation (1).

The algorithm terminates when the delay requirement is satisfied. This is sound since addition of a new stage will increase capacitance due to the new driver and possibly a larger interconnect capacitance due to the wiresizing, which results in larger power dissipation. The algorithm terminates without a solution when the delay at the current iteration is larger than the delay at the previous iteration. This occurs when the gain due to the increased driving capability cannot offset the delay due to the new driver. The DWSA algorithm is described formally in Table 1.

Since driver sizing is straightforward using the known optimal stage ratio, we concentrate in the rest of this section to present optimal wiresizing algorithm for performance optimization based on the results in [9].

3.2 Properties of Optimal Wiresizing Solutions for Performance Optimization

We consider the wiresizing problem for performance optimization which minimizes the multiple-critical-sink formulation (7) without taking into account power dissipation. In this case, we list the results obtained in [9], which showed several interesting properties of optimal wiresizing solutions for performance optimization, including the *separability*, the *monotone property*, and the *dominance property*.

3.2.1 Separability

Theorem 1 *If the width assignment of a path P originated from the source is given, the optimal width assignment for each subtree branching off P can be carried out independently.* \square

3.2.2 Monotone Property

Definition 2 *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 $E \in Ans(E')$.*

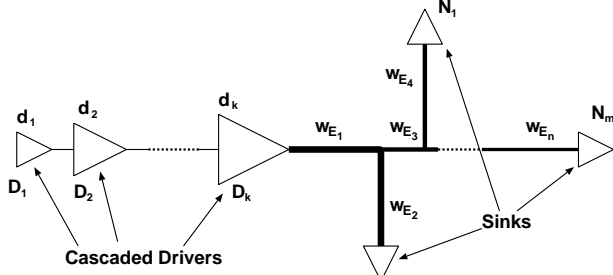


Figure 1: A k -stage cascaded drivers driving an interconnect tree T with sinks $\{N_1, N_2, \dots, N_m\}$. w_{E_i} denotes the width of the wire segment E_i , $i = 1..n$ and d_i denotes the size of driver D_i at i -th stage, $i = 1..k$.

the signal net from the source to one or several critical sinks, and it can be decomposed as follows:

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

where the first term measures the delay due to the drivers and the second term measures the interconnect delay. We estimate $t(k, \mathcal{D})$ by a simple RC formula. The interconnect delay is measured by the distributed Elmore delay model [11] as described below.

2.1 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 [9]. 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 Figure 2. In this case, T consists of a set of unit-length wire segments, each may have a different width¹. Since each grid point u in the tree T is uniquely identified with an incoming edge in the routing tree, we also use u to refer to that edge. Similarly, we can use an edge, say E to refer to the destination grid point of the edge.

For each edge ending at u in the tree T , we use a π -type RC circuit to model the interconnect, where r_u and c_u are the interconnect resistance and capacitance respectively. Given a node u , we use w_u to denote the width of the grid edge u . Assume that a unit-width unit-grid-length wire has wire resistance r_0 and wire capacitance c_0 , then $r_u = \frac{r_0}{w_u}$ and $c_u = c_0 \cdot w_u$ for any grid edge u .

We use c_u^s to denote the node capacitance at u . If u is a sink, c_u^s represents the loading capacitance at the sink. If there is a via or bend at node u , it can also be formulated by introducing a small capacitance at the node u . For simplicity, we assume that c_u^s is non-zero if u is a sink, and zero otherwise. 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 (the resistance of the last driver in the cascaded driver chain) in the later discussions.

Given a grid point u , we use $Des(u)$ to denote the set of grid points in the subtree rooted at u (excluding u), and $Ans(u)$ to denote the set of grid points $\{v | u \in Des(v)\}$ (again, excluding u). That is, $Des(u)$ is the set of “descendant” grid points

¹ In [8, 9], 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.

of u , and $Ans(u)$ is the set of “ancestor” grid points of u . Also, we use $sink(u)$ to denote the set of sinks in the subtree rooted at u , and C_u to denote the *total* capacitance in the subtree rooted at u (including both the wire capacitances and the sink capacitances). Furthermore, we use $P(u, v)$ to denote the unique path from u to v for any grid points u, v in the routing tree.

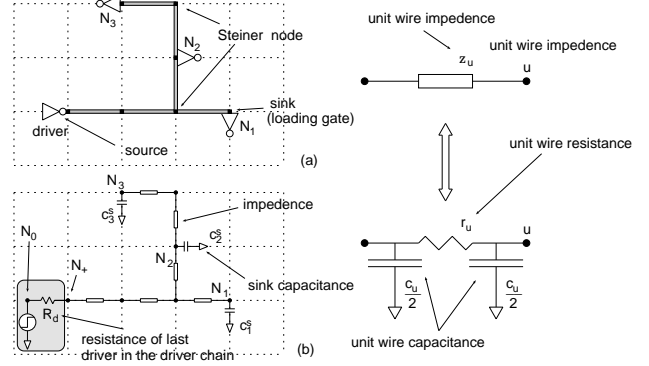


Figure 2: A uniform grid structure for the distributed RC interconnect model. (a) The layout of an interconnect tree T with 3 sinks N_1, N_2 , and N_3 . (b) The corresponding distributed RC interconnect model of T . Each grid edge in T connecting two adjacent nodes is modeled as a π -type RC circuit containing a resistor of r_u and two capacitors of $\frac{c_u}{2}$ each, where u is the farther end of the grid edge from the source. Each sink has an extra loading capacitance.

We use the Elmore delay model [11] 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:

$$t(N_i) = \sum_{u \in P(N_0, N_i)} r_u \cdot \left(\frac{c_u}{2} + C_u \right) \quad (2)$$

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

2.2 Single Critical Sink Formulation

We shall first study the case where there is only one critical sink N_i in the net. According to (2), the signal delay $t(N_i)$ at N_i under a given wiresizing solution \mathcal{W} is:

$$\begin{aligned} t_i(\mathcal{W}) &= \sum_{u \in P(N_0, N_i)} r_u \cdot \left(\frac{c_u}{2} + C_u \right) \\ &= \sum_{u \in P(N_+, N_i)} \frac{r_0 \cdot c_0}{2} + R_d \cdot \sum_{u \in T} c_u^s + \\ &\quad R_d \cdot c_0 \cdot \sum_{u \in T} w_u + r_0 \cdot c_0 \cdot \sum_{u \in P(N_+, N_i)} \sum_{v \in Des(u)} \frac{w_v}{w_u} \\ &\quad + r_0 \cdot \sum_{u \in P(N_+, N_i)} \sum_{v \in sink(u)} c_v^s \cdot \frac{1}{w_u} \end{aligned} \quad (3)$$

Since a grid point u can be identified by its incoming edge, say E_i , we can express (3) as follows:

$$t_i(\mathcal{W}) = \mathcal{K}_{1i} + \mathcal{K}_2 + \mathcal{K}_3 \cdot \sum_{E \in T} w_E +$$

Wiresizing with Driver Sizing for Performance and Power Optimization

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

Abstract

In this paper, we study the effect of wiresizing with driver sizing on performance and power optimization. Our study showed that wiresizing is an effective method to reduce interconnect delay. Optimal wiresizing solutions require additional routing area but do not increase the power dissipation significantly. We propose to combine the driver sizing and wiresizing techniques as a low-power solution to meet the high-speed interconnect requirement. Experimental results have shown that together with cascaded drivers, our wiresizing solution consumes less power while meeting the delay requirement when compared to the conventional methods of using cascaded drivers with uniform-width interconnection.

1 Introduction

As the VLSI fabrication technology reaches submicron device dimension and gigahertz frequency, interconnect delay has become the dominant factor in determining circuit speed [10, 14], and the *distributed* nature of the interconnect structure must be considered. Moreover, since a significant portion of system power in modern CMOS designs is consumed in charging and discharging interconnect structures, interconnect optimization will also lead to *low-power* VLSI designs, which are critical to today's portable and wireless computing and communications technologies. The objective of this paper is to study the impact of wiresizing on performance 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 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, 4], maximum performance trees [6], A-trees [8], low-delay trees [3], and IDW/CFD trees [12]. Moreover, the wiresizing algorithm in [8, 9] can further minimize interconnect delay by optimally assigning different wire width to each wire segment in the interconnect design. 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 [8].

In this paper, we study the effect of wiresizing with driver sizing on both performance and power optimization. We propose

to combine the driver sizing and wiresizing techniques together as a low-power solution to meet the high-speed interconnect requirement. We applied the optimal wiresizing algorithm under the distributed Elmore delay model and the optimal driving sizing techniques to a number of interconnect design cases. Our experimental results showed that although optimal wiresizing solutions require additional routing area, they do not increase the power dissipation significantly. In addition, our study has shown that together with cascaded drivers, our wiresizing solutions can meet delay requirement with less power dissipation when compared to the conventional methods of using cascaded drivers with uniform-width interconnection.

The remainder of this paper is organized as follows: In Section 2, we present the general formulation of the wiresizing problems under the distributed Elmore delay model. In Section 3, we present a heuristic algorithm for both driver sizing and wiresizing for performance and power optimization. Section 4 shows the experimental results obtained by our driver sizing and wiresizing algorithm and Section 5 concludes the paper with discussions of future work.

2 Problem Formulation

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 $\{N_1, N_2, \dots, N_m\}$. A node refers to the source, or a sink, or a Steiner node in T , and a segment connects two nodes in T . Assume that $\{E_1, E_2, \dots, E_n\}$ is the set of segments forming the tree T , where n is the total number of segments in the tree.

We assume that each wire segment has a set of discrete choices of wire widths $\{W_1, W_2, \dots, W_r\}$ ($W_1 < W_2 < \dots < W_r$), and the wire width within the same segment does not change. We use w_{E_i} to denote the width of the wire segment E_i , $i = 1..n$. For a given wiresizing solution \mathcal{W} to signal net T , let $t(\mathcal{W})$ be a measure of the performance of the signal net. We shall formulate the performance measure in subsequent subsections.

In general, we may 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 d_i to denote the size of the driver D_i at i -th stage, where $i = 1..k$. Let \mathcal{D} denote the set of driver sizes $\{d_1, \dots, d_k\}$. Given the above definitions, the problem of driver sizing and wiresizing for performance and power optimization can be defined as follows:

Definition 1 *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 $t(k, \mathcal{D}, \mathcal{W})$ is bounded by B , i.e. $t(k, \mathcal{D}, \mathcal{W}) \leq B$ and the power dissipation $Power(k, \mathcal{D}, \mathcal{W})$ is minimized.*

In this paper, we do not use an analytical formula for computing power dissipation. The power dissipation is measured using HSPICE through circuit simulation in our experiments. The performance measure $t(k, \mathcal{D}, \mathcal{W})$ approximates the delay of

*The current address of Kwok-Shing Leung is: Carnegie Mellon University, Pittsburgh, PA 15213.