Simultaneous Gate and Interconnect Sizing for Circuit-Level Delay Optimization*

Noel Menezes, Satyamurthy Pullela, and Lawrence T. Pileggi†
Department of Electrical and Computer Engineering
The University of Texas at Austin
Austin, TX 78712-1084

Abstract—With delays due to the physical interconnect dominating the overall logic path delays, circuit-level delay optimization must take interconnect effects into account. Instead of sizing only the gates along the critical paths for delay reduction, the trade-off possible by simultaneously sizing gate and interconnect must also be considered. We show that for optimal gate and interconnect sizing, it is imperative that the interaction between the driver and the RC interconnect load be taken into account. We present an iterative sensitivity-based approach to simultaneous gate and interconnect sizing in terms of a gate delay model which captures this interaction. During each iteration, the path delay sensitivities are efficiently calculated and used to size the components along a path.

I. Introduction

With the interconnect delays dominating the overall path delays for today's increasingly dense integrated circuits, algorithms for synthesis and delay optimization must take RC interconnect effects into account. Moreover, since routing, which determines interconnect sizes, usually follows synthesis, which determines gate sizes in traditional design flows, it has been noted that for current integrated circuits the delay estimates predicted by synthesis tools are not consistent with the actual path delays observed after routing. This is mainly due to the inherent inability of synthesis tools to predict interconnect delays. Hence, there is often a need to size gates even after synthesis to meet delay and/or power requirements. Also, a substantial delay reduction is possible by sizing the branches of the interconnect tree between gates [1].

A. Background

Gate and transistor sizing have been studied in great detail in the past [2, 3, 4] for technologies in which gate delays dominate the overall path delay. With the increasing role of interconnect, timing-driven placement approaches [5] have been proposed with a view toward minimizing the interconnect delay. Several algorithms for optimal interconnect tree design have also been proposed in the literature with the more recent ones attempting to minimize the delay at the sinks [6] instead of minimizing the length of the interconnect branches in the tree [7]. Recently, recognizing that large reductions in interconnect delay can be achieved by selectively widening the branches of the interconnect tree (wiresizing), algorithms for optimal wiresizing have been proposed in [1, 8, 9].

The approach in [1, 8] uses the properties of monotonicity, separability, and dominance which apply to the Elmore delay [10] to determine the optimal wiresizing solution. Recognizing that monotonicity and separability do not apply to the Elmore delay under certain conditions, a sensitivity-based wiresizing algorithm is presented in [9]. Both approaches model the driver by a resistor with a value that stays constant during wiresizing. However, the error resulting from a fixed-resistor driver model in conjunction with the Elmore delay can be as high as 20-30% [11]. During wiresizing, the load on the driver changes significantly which should be reflected in a load-dependent gate resistance value. Moreover, a fixed-resistor gate delay model when used in conjunction with the Elmore delay approximation implicitly assumes that the load on the gate is accurately modeled by the total interconnect capacitance, an assumption that is especially invalid for RC-interconnect loads [12]. Furthermore, input transition time effects must be considered for accuracy. This calls for a built-in timing analysis capability. Most importantly, since the load presented by the interconnect tree changes significantly during wiresizing, sizing the driver should be an integral part of the wiresizing approach itself. This allows the trade-off between driver and interconnect delay to be exploited.

In a gate-delay dominated environment, it is well known [3, 4] that while increasing the size of a gate may reduce its delay, it increases the delay of the previous stage because of the increased capacitive load. However, in an interconnect-delay dominated circuit the additional circuit-level effects described above need to be considered too.

In this paper, we present a sensitivity-based approach to the simultaneous gate and interconnect sizing problem. For accurate path delay calculation we use an efficient timing analyzer that takes the input transition time effects, the complex gate-RC-load interaction, as well as the interconnect delays into account (Section II). We first demonstrate our approach to gate and interconnect sizing for a single logic stage (Section III). In Section IV, we extend these techniques to path delay reduction.

* This work was supported in part by the Semiconductor Research Corporation under contract 94-DJ-343, the National Science Foundation under contract MIP-9157263, and IBM Corp.

† Formerly Lawrence T. Pillage

32nd ACM/IEEE Design Automation Conference ®

II. Timing Analysis

For efficiency, most timing analyzers precharacterize gate delays and gate output transition times as functions of load capacitance, C_L, and input transition time, t_t [12]:

$$\begin{aligned} t_{50} &= k_1 t_t + k_2 C_L t_t + k_3 C_L^2 + k_4 C_L + k_5 \\ t_{10-50} &= k_1' t_t + k_2' C_L t_t + k_3' C_L^2 + k_4' C_L + k_5' \\ t_{10-90} &= k_1'' t_t + k_2'' C_L t_t + k_3'' C_L^2 + k_4'' C_L + k_5'' . \end{aligned} \quad (1)$$

where t_{50}, $t_{10\text{-}50}$, and $t_{10\text{-}90}$ are the 50% gate delay, and 10-50% and 10-90% gate output transition times respectively. Unlike most timing analyzers, we precharacterize the 10-50% transition time to model the effective capacitance loading. We use the "effective capacitance" model presented in [12, 13] to obtain an accurate estimate of the gate output waveform for highly-resistive interconnect loads. The second-order driving point admittance of the load, which is modeled by a π-circuit, is shown to be adequately accurate for on-chip RC interconnect. Essentially, the π-load is mapped to an effective capacitance, C_{eff}, which is then used to iteratively compute the parameters of a single-resistor voltage-ramp model (Fig. 1) [13]. This model, with the π-load, gives the necessary gate output waveform.

Computing the driver resistance, R_{dr}, is critical to the gate sizing problem. Recognizing that the gate behaves like a linear resistor discharging a capacitor for the tail portion of the response waveform [12], the value of R_{dr} is calculated by

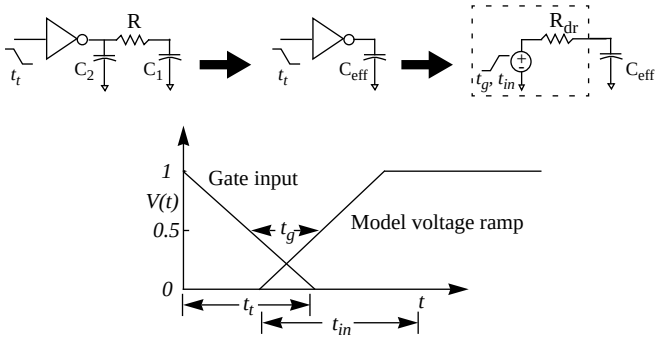

Figure 1. The voltage ramp delay model.

$$R_{dr} = \frac{t_{90} - t_{50}}{C_{eff} \ln 5} . \quad (2)$$

Since the driver resistance is computed from the effective capacitance, its dynamic load dependence is implicit in this model.

The timing analysis flow for a single stage along a path is shown in Fig. 2. Since the entire stage (driver as well as interconnect) has been reduced to a linear circuit driven by a ramp, the delay from the gate input to the output nodes of interest (*critical sinks*) can be calculated with linear complexity using RICE [14], an application-specific implementation of AWE [15]. This is applied to every stage along a path.

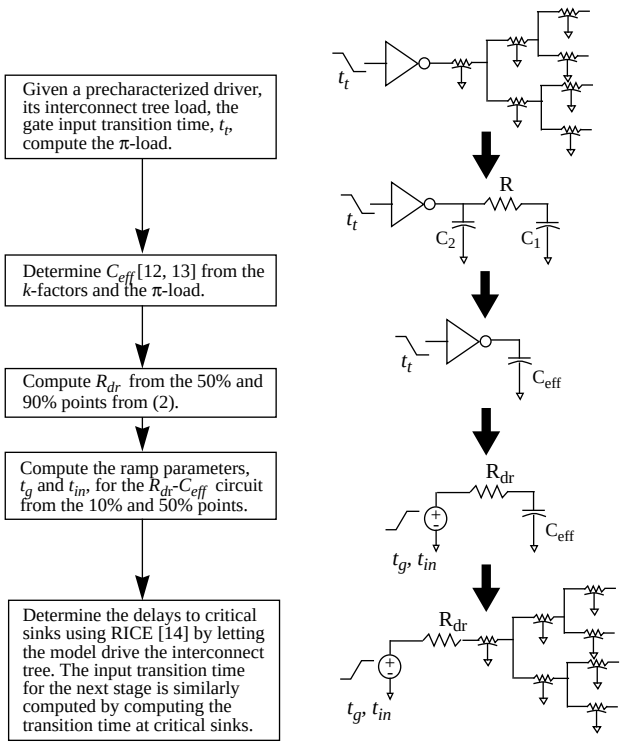

Figure 2. Timing analysis flow for a single stage.

III. Single-stage Gate and Interconnect Sizing

We size gates by assuming that the transistor widths within each gate scale isotropically. That is, the ratios of the transistor widths within a gate with respect to each other stay constant. This allows us to describe each gate in terms of its "width." For certain complex gates, or gates with buffer output stages (for example, domino logic gates), we recognize that only certain transistors within the gate will be sized isotropically. The area of each gate, a_g, is, therefore, described as a function of its width, w_g, by

$$a_g = a(w_g). \quad (3)$$

Our examples indicate that this function is reasonably linear for today's CMOS technologies. For a conventional CMOS library with discrete gate sizes the final gate size solution is mapped to the closest size available in the library.

A. Gate precharacterization for different widths

A fixed driver is precharacterized in terms of its input transition time and load capacitance by *k*-factor equations (1). With continuously-varying driver sizes, however, we need to precharacterize a gate in terms of its width, w_g. That is,

$$t_{50} = f(t_t, C_L, w_g). \quad (4)$$

Recognizing that the delay and output transition time of a gate is inversely proportional to its width and directly proportional to its load capacitance, we precharacterize gate delays and transition times as a function of C_L/w_g,

$$t_{50} = k_1 t_t + k_2 \frac{C_L}{w_g} t_t + k_3 \left(\frac{C_L}{w_g} \right)^2 + k_4 \frac{C_L}{w_g} + k_5 . \quad (5)$$

The 10-50% and 10-90% output transition times are also precharacterized by a similar empirical fit. The gate input capacitance is also precharacterized by

$$C_g(w_g) = c_1 + c_2 w_g + c_3 w_g^2 . \quad (6)$$

B. Single-stage delay optimization

We now describe how the target delays, $\hat{t}_d^i$, $1 \le i \le M$, specified at the fanout nodes of interest (*critical sinks*) can be achieved by sizing the gate and the branches of the RC net.

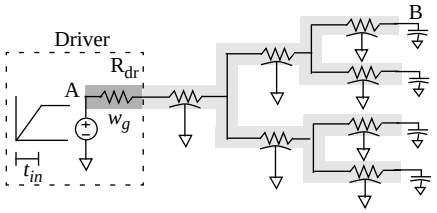

Figure 3. Single-stage gate and interconnect wire sizing.

The gate is replaced by its single-resistor voltage-ramp model with parameters R_{dr}, t_{in}, and t_g calculated from the gate width via the *k*-factor equations (Fig. 3). We then calculate the delay sensitivities to the interconnect wire widths as well as the gate width. These sensitivities are used to guide a gradient-based optimization technique to determine the gate and wire widths necessary to achieve the specified target delays.

C. Delay sensitivity computation

For an iterative sensitivity-based optimization, efficient sensitivity calculation is of paramount importance. The sensitivity computation approach described here is partially based on the theory described in [16]. In the following derivation, we ignore the gate offset time, t_g, (Fig. 1) for clarity.

If the desired output waveform at a critical sink, e.g. node *B* in Fig. 3, is a ramp with a 50% delay, t_d, and a transition time, t_{out}, (Fig. 4) the widths of the tree should be varied so as to match the transfer function, $H(s)$ (= $V_B(s)/V_A(s)$), to

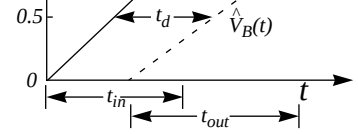

Figure 4. The desired ramp waveform at a fanout node.

$$\hat{H}(s) = \frac{\hat{V}_B(s)}{V_A(s)} = \frac{t_{in}}{t_{out}} \frac{(1 - e^{-s t_{out}})}{(1 - e^{-s t_{in}})} e^{-\left(t_d + \frac{t_{in}}{2} - \frac{t_{out}}{2} \right)}$$

$$= \hat{m}_0 + \hat{m}_1 s^2 + \hat{m}_2 s^2 + \ldots + \hat{m}_n s^n + \ldots \quad . \quad (7)$$

Ideally one can realize a transfer function as in (7) by forcing the *moments* [15], m_i, of the transfer function, $H(s)$, to match the moments, $\hat{m}_i$, of the desired transfer function, $\hat{H}(s)$, at the critical sink. It is shown in [15], however, that due to the low-pass nature of RC interconnect trees, the voltage response at any node can be accurately characterized by its lower-order moments. For example, matching the first four moments of $H(s)$ to the corresponding moments of $\hat{H}(s)$ from (7) yields

$$\hat{m}_0 = 1$$

$$\hat{m}_1 = -t_d$$

$$\hat{m}_2 = \frac{1}{2} t_d^2 - \frac{1}{24} t_{in}^2 + \frac{1}{24} t_{out}^2$$

$$\hat{m}_3 = -\frac{1}{24} t_d \left(-t_{in}^2 + t_{out}^2 + 4 t_d^2 \right). \quad (8)$$

Equation (8) shows the approximate relation between delay, transition time, and the circuit moments.

In general, for RC trees, the delay, t_d^i, at any node *i* is

$$t_d^i = f\left(t_{in}, m_1^i, m_2^i, m_3^i, \ldots \right). \quad (9)$$

In (9), the superscript *i* is used to indicate the value of a quantity at node *i* in the RC tree. Therefore, for a ramp waveshape assumption at node *i* and a fixed ramp input transition time, the sensitivity of the delay at node *i* to the width of wire *l* (gate or interconnect) in the tree can be calculated from (8) by

$$\frac{\partial t_d^i}{\partial w_l} = \frac{\partial t_d^i}{\partial m_1^i} \frac{\partial m_1^i}{\partial w_l} + \frac{\partial t_d^i}{\partial m_2^i} \frac{\partial m_2^i}{\partial w_l} + \frac{\partial t_d^i}{\partial m_3^i} \frac{\partial m_3^i}{\partial w_l}$$

$$= -\frac{\partial m_1^i}{\partial w_l} - \frac{1}{m_1^i} \frac{\partial m_2^i}{\partial w_l} - \frac{1}{m_2^i} \frac{\partial m_3^i}{\partial w_l} . \quad (10)$$

For sensitivity computation purposes, fitting the first four moments provides adequate accuracy.

Interconnect delay sensitivities

If the driver size is also considered a variable to be optimized, the sensitivity of the k^{th} moment at a node *n* with respect to the width of a wire *l* in the tree is given by slightly modifying (10) in [16]:

$$\frac{\partial m_k^n}{\partial w_l} = \frac{\partial m_k^n}{\partial R_l} \frac{\partial R_l}{\partial w_l} + \frac{\partial m_k^n}{\partial C_l} \frac{\partial C_l}{\partial w_l} + \sum_{\forall i} \frac{\partial m_k^n}{\partial m_{k-1}^i} \frac{\partial m_{k-1}^i}{\partial w_l}$$

$$\frac{\partial m_k^n}{\partial R_{dr}} \frac{\partial R_{dr}}{\partial w_l} . \quad (11)$$

The summation in (11) is over all nodes *i* in the tree. R_l and C_l refer to the resistance and capacitance of wire *l* [16]. All of the terms in (11) except for $\partial R_{dr}/\partial w_l$ are computed using the methods described in [16]. $\partial R_{dr}/\partial w_l$ represents the influence of widening wire *l* on the gate resistance value. As stated pre-

viously, widening a wire has an effect on the load seen by the gate which manifests itself as a change in the π-load used to calculate the effective capacitance. This change in the effective capacitance changes the driver resistance value. That is,

$$\frac{\partial R_{dr}}{\partial w_l} = \frac{\partial R_{dr}}{\partial C_{eff}}\frac{\partial C_{eff}}{\partial w_l} = \frac{\partial R_{dr}}{\partial C_{eff}}\left(\frac{\partial C_{eff}}{\partial R}\frac{\partial R}{\partial w_l} + \frac{\partial C_{eff}}{\partial C_2}\frac{\partial C_2}{\partial w_l} + \frac{\partial C_{eff}}{\partial C_1}\frac{\partial C_1}{\partial w_l}\right). \quad (12)$$

The sensitivities of the π-circuit parameters, R, C_2, C_1, with respect to w_l are readily calculated. Even though C_{eff} has no closed-form solution, the partials $\partial C_{eff}/\partial R$, $\partial C_{eff}/\partial C_2$, $\partial C_{eff}/\partial C_1$ are computed from the last iterative solution of C_{eff} [12, 13].

Gate delay sensitivities

Changing the gate width affects parameters of the ramp delay model: R_{dr}, t_{in}, and t_g. The effect of changing the gate width on the delay to a critical sink i can, therefore, be expressed as

$$\frac{\partial t_d^i}{\partial w_g} = \frac{\partial t_d^i}{\partial t_{in}}\frac{\partial t_{in}}{\partial w_g} + \frac{\partial t_d^i}{\partial t_g}\frac{\partial t_g}{\partial w_g} + \frac{\partial t_d^i}{\partial R_{dr}}\frac{\partial R_{dr}}{\partial w_g}. \quad (13)$$

The quantities $\partial t_{in}/\partial w_g$, $\partial R_{dr}/\partial w_g$, and $\partial t_{in}/\partial w_g$ in (13) are calculated numerically by a finite-difference method. $\partial t_d^i/\partial R_{dr}$ is calculated from (10) and the moment sensitivities [16], while $\partial t_d^i/\partial t_g = 1$. Calculating $\partial t_d^i/\partial t_{in}$, the effect of the ramp transition on the delay, again requires a finite difference computation. This computation is performed efficiently by assuming that the response at node i is characterized by its Elmore delay pole.

D. Delay optimization by Levenberg-Marquardt

Following [16, 17], given a procedure to compute the delay sensitivities and the delays at the critical sinks, we use the Levenberg-Marquardt method [18] to find the gate and wire widths that will meet the target delays.

The importance of weighting wires during Levenberg-Marquardt optimization for an optimal solution is described in [16]. These weights are usually computed on the basis of the relative position of a wire in the tree, the routability of a wire, or a combination of such criteria.

For the most area-efficient solution, the wire that causes the maximum change in delay at a critical sink for the smallest change in circuit area is assigned the maximum weight. Hence, the weight for a wire j over all critical sinks should be

$$\alpha_j = \sum_{i=1}^{M} \frac{\partial t_d^i}{\partial a_j} \quad (14)$$

where a_j (= $w_j l_j$) is the area of wire j. Recognizing that the Elmore delay, m_1^i, is a good relative indicator of the overall delay [19], each wire j in the tree is assigned a weight

$$\alpha_j = \sum_{i=1}^{M} \frac{\partial m_1^i}{\partial a_j} = \frac{1}{l_j}\sum_{i=1}^{M} \frac{\partial m_1^i}{\partial w_j}. \quad (15)$$

While the above weighting scheme applies in a straightforward manner to the wires of the interconnect tree, the weighting factor for the gate is calculated differently. We employ a similar metric—compute the weighting factor for the gate based upon the sensitivity of the delay at a critical sink to the gate area. Also, it is possible that gate area may have a different (higher or lower) cost than interconnect area depending on the circuit design, technology, architecture [20] or power requirement. In this case, we have a predetermined factor γ_{GI} which indicates the relative cost of gate area as compared to interconnect area. Therefore, a weighting factor for the gate wire which is consistent with (15) is

$$\alpha_g = \frac{1}{\gamma_{GI}}\sum_{i=1}^{M} \frac{\partial t_d^i}{\partial a_g} = \frac{1}{\gamma_{GI}}\sum_{i=1}^{M}\left(\frac{\partial t_d^i}{\partial w_g} \Big/ \frac{\partial a_g}{\partial w_g}\right). \quad (16)$$

$\partial a_g/\partial w_g$ is calculated from (3) and $\partial t_d^i/\partial w_g$ from (13).

Equations (15) and (16) can be appropriately modified to avoid excessively wide wires or large gates or to restrict wire widening in heavily congested areas.

E. Results

To illustrate our approach, we show the gate and wire sizes necessary to achieve target delays at the critical sinks equal to 30% of the sink delays when the gate and the interconnect wires are at minimum size (w_{gmin} = 5, w_{imin} = 1.0 μm), of the stage in Fig. 5. A γ_{GI} of 10 is assumed. A sheet resistance of

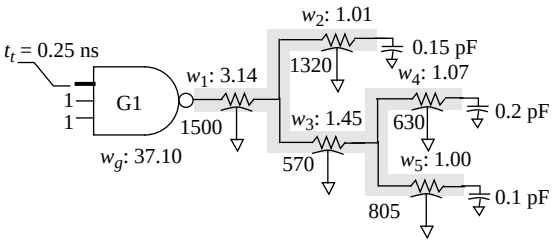

Figure 5. Gate and wire widths after single-stage optimization.

0.14 Ω/□, a per-unit area capacitance of 0.08 fF/μm^2, and a fringe capacitance of 0.03 fF/μm are assumed. The minimum and maximum allowable widths for the interconnect branches are 1.0 μm and 6.0 μm respectively. Plotted in Fig. 6 is the circuit area cost ($\gamma_{GI} area_{gate} + \sum w_i l_i$) required for different percentage target delays. From the plot, we can see that there is a clear trade-off between gate area and interconnect area for delay reduction – it is important to simultaneously size both.When delay reduction for this example was attempted by gate sizing alone, percentage target delay reductions of 50% and above *could not* be achieved by gate sizing alone (the gate size for 30% delay reduction is 8.29). This is because the interconnect delays account for more than 50% of the overall delay, especially as the gates are made wider. In cases like

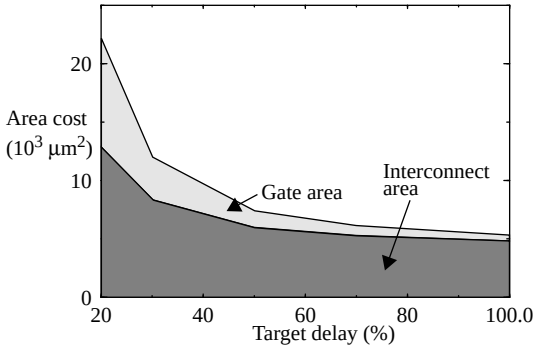

Figure 6. Circuit area cost as a function of percentage delays for the circuit of Figure 5.

these, interconnect sizing for delay reduction is essential. In

Net	No. of fanout nodes	No. of branches	Before optimization		After optimizaton	
			Gate size	Int. area (μm^2)	Gate size	Int. area (μm^2)
Steiner	5	9	10	5800	31.15	8749
16-pin binary	16	31	20	27020	63.23	34065
Line	1	10	5	10000	14.47	14603

Table 1. Results for single-stage optimization.

Table 1 the gate and interconnect area necessary for a 50% delay reduction for the examples of [16] are shown.

IV. Gate And Interconnect Sizing for Paths

Single stage sizing can be used in conjunction with a slack allocation algorithm [21] to reduce overall delay of a circuit. A more interesting problem, however, is that of reducing the delay along a path in a circuit. We again cast this problem as a sensitivity-based optimization. The path delays are evaluated using the timing analyzer described in section II.

A. Path delay sensitivities

We refer to the critical sink of a stage that lies along the path of interest as the path sink for that stage The path delay, t_P, is the sum of the successive path-sink to path-sink delays, t_p^n (= $t_d^n + t_g^n$), of each stage *n* along the path. Hence the path delay sensitivity to any component (gate or interconnect wire) is the sum of the sensitivities of the path-sink to path-sink delays to that component. That is, for *N* stages along a path,

$$\frac{\partial t_P}{\partial w_l} = \sum_{n=1}^{N} \frac{\partial t_p^n}{\partial w_l}. \tag{17}$$

Hence, computing the path delay sensitivity involves computing the individual stage delay sensitivities. The effect of sizing a component in stage *n* on the delay of stage *n*, t_p^n, is calculated from single-stage delay sensitivity equations, (10) - (13). The effect of sizing a component in stage *n* on *any other stage* along the path is captured by the following two observations:

Consider two successive stages, *n* and *n* + 1, along a path *P*,

Observation 1: Except for the gate of the succeeding stage, *n* + 1, no component of any following stage has an effect on the delay, t_p^n, of stage *n*.

Observation 2: Sizing any component *l* of stage *n* affects the path sink transition time, t_{out}^{n+1}, of stage *n* (which is the gate input transition time, t_t^{n+1}, for stage *n* + 1) which in turn affects the delay of stage *n* + 1 as well as the gate input transition time, t_t^{n+2}, of stage *n* + 2. The change in t_t^{n+2} in turn affects t_p^{n+2} and so on. However, this cascading effect is simplified by recognizing that for CMOS circuits the effect of the input transition time of a gate in a stage on the input transition time to any stage other than the succeeding stage is negligible. More clearly, while varying t_t^{n+1} affects t_t^{n+2}, its effect on t_t^{n+3} is negligible.

Observation 1 recapitulates the basic gate sizing problem for paths−while increasing the size of a gate along a path reduces the delay of its stage, it increases the delay of the previous stage because of the extra capacitive load it presents to that stage. Therefore, if w_g^n refers to the width of the gate of stage *n*, then

$$\frac{\partial t_P}{\partial w_g^{n+1}} = \frac{\partial t_p^n}{\partial w_g^{n+1}} + \frac{\partial t_p^{n+1}}{\partial w_g^{n+1}}. \tag{18}$$

The second term in (18) is calculated from (10) while the first term is calculated from

$$\frac{\partial t_p^n}{\partial w_g^{n+1}} = \frac{\partial t_p^n}{\partial C_g^{n+1}} \frac{\partial C_g^{n+1}}{\partial w_g^{n+1}}. \tag{19}$$

The product terms in (19) are calculated from (6) and (10).

From Observation 2 we have,

$$\frac{\partial t_p^{n+1}}{\partial w_l} = \frac{\partial t_p^{n+1}}{\partial t_t^{n+1}} \frac{\partial t_{out}^n}{\partial w_l} = \frac{\partial t_p^{n+1}}{\partial t_{in}^{n+1}} \frac{\partial t_{in}^{n+1}}{\partial t_t^{n+1}} \frac{\partial t_{out}^n}{\partial w_l}. \tag{20}$$

$\partial t_p^{n+1}/\partial t_{in}^{n+1}$ is calculated in a manner similar to $\partial t_d^i/\partial t_{in}$ in (13) while $\partial t_{in}^{n+1}/\partial t_t^{n+1}$ is calculated numerically. However, this numerical computation does not imply any additional computation overhead since it is done at the time when the gate delay parameters in (13) is computed. $\partial t_{out}^n/\partial w_l$ is computed in a manner similar to that of $\partial t_d^i/\partial w_l$ in (10).

Path delay optimization

Given the path delay sensitivities and an efficient timing analyzer to compute the path delay, we again use Levenberg-Marquardt optimization to size the path components to realize a target path delay. We use a weighting scheme similar to the one used for single stage optimization.

B. Example

We demonstrate our path delay reduction technique on the simple example of Fig. 7. We set the target delay for this path to half the original path delay (when all the gates and wires are set to their minimum sizes) of 5.44 ns. The widths necessary for the delay reduction are shown in Fig. 7. For a γ_{GI} of

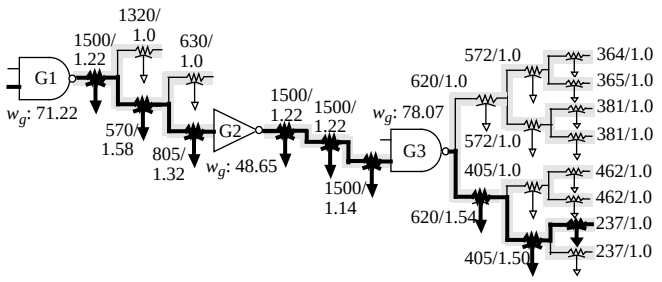

Figure 7. Gate and wire sizes for a 50% path delay reduction.

10, the circuit area cost ($\sum \gamma_{GI} area_{gates} + area_{int}$) for different percentage delay reductions for this path is shown in Table 2. As expected, we see that the gate sizes necessary for delay reduction by gate sizing only are higher than those required by simultaneous gate and interconnect sizing. In addition, we also see that the total circuit area costs resulting from simultaneous gate and interconnect sizing (Column 10) are smaller than those from gate sizing only (Column 9). In general, we observe that this reduction in circuit area cost increases with more aggressive delay reduction targets. We also see that a 50% delay reduction cannot be achieved for the example path by gate sizing only (indicated by a - in the last row) because of the dominant interconnect delay.

Delay reduction (%)	Final gate widths							Total area cost	
	Gate sizing only			Gate and wire sizing				Gate sizing	Gate & wire sizing
	G1	G2	G3	G1	G2	G3	Int. area		
0	5.00	5.00	5.00	5.00	5.00	5.00	15408	18408	18408
15	9.10	9.17	9.23	9.06	9.12	9.17	15922	20908	21392
30	17.07	17.97	20.88	15.63	15.99	16.75	16684	26726	26358
40	50.20	23.79	99.88	43.20	30.07	47.47	17486	50182	41634
50	-	-	-	71.22	48.65	78.07	17681	-	57269

Table 2. Results for delay reduction of the path in Figure 7.

V. Conclusions and Future Work

We have presented a path delay optimization technique which in addition to sizing gates also sizes interconnect wires along the path. Three important aspects necessary for delay optimization in interconnect dominated circuits are considered: 1.) The gate-RC-load interaction through the effective capacitance and the single-resistor voltage-ramp model. 2.) Accurate timing analysis for path delay computation. And 3.) Accurate interconnect delay calculation. We also show that in interconnect dominated circuits, interconnect and gate sizing should be performed simultaneously for optimal solutions.

For large nets, delay reduction can be achieved by gate sizing, net wiresizing, and buffer insertion. While this paper explores the first two possibilities, in the future we intend studying the power- and delay-reduction aspects of buffer insertion.

References

[1] J. Cong, and K.-S. Leung, "Optimal wiresizing under the distributed Elmore delay model," *Proc. IEEE/ACM Intl. Conf. on Computer-Aided Design*, November 1993.

[2] J. P. Fishburn and A. E. Dunlop, "TILOS: A posynomial programming approach to transistor sizing," *Proc. IEEE/ACM Intl. Conf. on Computer-Aided Design*, November 1985.

[3] D. Marple, "Transistor size optimization in the Tailor layout system," *Proc. ACM/IEEE 26th Design Automation Conferenc*e, June 1989.

[4] S. S. Sapatnekar, et al., "An exact solution to the transistor sizing problem for CMOS circuits using convex optimization," *IEEE Trans. Computer-Aided Design*, vol. 12, no. 11, pp. 1621- 1634, May 1992.

[5] M. Marek-Sadowska and S. P. Lin, "Timing-driven placement," *Proc. IEEE/ACM Intl. Conf. on Computer-Aided Design*, November 1989.

[6] K. D. Boese, A. B. Kahng, B. A. McCoy, and G. Robins, "Rectilinear Steiner Trees with minimum Elmore delay," *Proc. 31st ACM/IEEE Design Automation Conferenc*e, June 1994.

[7] J. Cong, et al., "Provably good performance-driven global routing," *IEEE Trans. Computer-Aided Design*, vol. 11, no. 6, pp. 739 - 752, June 1992.

[8] J. Cong and C.-K. Koh, "Simultaneous driver and wire sizing for performance and power optimization," *Proc. IEEE/ACM Intl. Conf. on Computer-Aided Design*, November 1994.

[9] S. S. Sapatnekar, "RC interconnect optimization under the Elmore delay model," *Proc. 31st ACM/IEEE Design Automation Conferenc*e, June 1994.

[10] J. Rubenstein, P. Penfield, Jr., and M. A. Horowitz, "Signal delay in RC tree networks," *IEEE Trans. Computer-Aided Design*, vol. CAD-2, pp. 202-211, July 1983.

[11] J. K. Ousterhout, "A switch-level timing verifier for digital MOS VLSI," *IEEE Trans. Computer-Aided Design*, vol. CAD-4, no. 3, pp. 336-349, July 1985.

[12] J. Qian, S. Pullela, and L. T. Pillage, "Modeling the *effective capacitance* for the RC interconnect of CMOS gates," *IEEE Trans. Computer-Aided Design*, vol. 13, no. 12, pp. 1526-1535, December 1994.

[13] F. Dartu, N. Menezes, J. Qian, and L.T. Pillage, "A gate-delay model for high-speed CMOS circuits," *Proc. 31st ACM/IEEE Design Automation Conferenc*e, June 1994.

[14] C.L. Ratzlaff, N. Gopal, and L.T. Pillage, "RICE: Rapid Interconnect Circuit Evaluator," *Proc. 28th ACM/IEEE Design Automation Conference*, June 1991.

[15] L.T. Pillage and R.A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. Computer-Aided Design*, vol. 9, no. 4, pp. 352-366, April 1990.

[16] N. Menezes, S. Pullela, F. Dartu, and L. T. Pillage, "RC interconnect synthesis−a moment fitting approach," *Proc. IEEE/ACM Intl. Conf. on Computer-Aided Design*, November 1994.

[17] Q. Zhu, W.-M. Dai, and J. G. Xi, "Optimal sizing of high-speed clock networks based on distributed RC and lossy transmission line models," *Proc. IEEE/ACM Intl. Conf. on Computer-Aided Design*, November 1993.

[18] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *J. Soc. Indust. App. Math.*, vol. 11, no. 2, pp. 431 - 441, June 1963.

[19] K. D. Boese, A. B. Kahng, B. A. McCoy, and G. Robins, "Fidelity and near-optimality of Elmore-based routing constructions," *Proc. IEEE/ACM Intl. Conf. Computer Design*, November 1993.

[20] D. K. Ferry, "Interconnection lengths and VLSI," *IEEE Circuits and Devices Magazine*, 1, July 1985.

[21] Wing Luk, "A fast physical constraint generator for timing driven layout," *Proc. 28th ACM/IEEE Design Automation Conferenc*e, June 1991.