RC Interconnect Synthesis—A Moment Fitting Approach[*]

Noel Menezes, Satyamurthy Pullela, Florentin Dartu, and Lawrence T. Pillage
Department of Electrical and Computer Engineering
The University of Texas at Austin
Austin, Texas 78712-1084

Abstract

Presently, delays due to the physical interconnect between logic gates account for large portions of the overall path delays. For this reason, synthesis of the logic gate fanout structure is of paramount importance during performance optimization. This paper presents a methodology for on-chip RC interconnect synthesis. Moment sensitivities are used to vary the wire widths of the branches in an RC interconnect tree to achieve performance targets. In this paper, signal slopes and delays at critical fanout nodes are the targets, and the impact on total metal area is considered. An $O(MN^2)$ *procedure for computing the exact moment sensitivities in an RC tree is described.*

1.0 Introduction

As process technologies are advanced, and integrated circuit feature sizes are reduced, the portion of the path delays attributable to the gates decreases, while the percentage of the delay due to the RC interconnect between gates increases. Most existing algorithms for synthesis and delay optimization, however, do not consider RC interconnect effects, and gate sizing is used to reduce path delays. Such approaches are suboptimal since they ignore the delay reduction possible by interconnect sizing. An optimal design must concurrently size gates and interconnect to optimize delay, signal edge rates, silicon area, metal area, and combinations thereof.

Interconnect sizing, which determines the RC delay, has a significant impact on the delay of the gate which drives the interconnect being optimized. Interconnect sizing also affects the delay of subsequent stages since the signal edge rates at the fan-out nodes are strongly dependent on the RC interconnect behavior. The subsequent gate delays are in turn dependent on the slope and shape of the input signals which drive them. For this reason, it is important to optimize the delays as well as the signal slopes at critical fan-out nodes.

Signal edge rates also impact the total power dissipation since they control, in part, the rush-through power dissipation of the logic gates. Power is even more dependent on the sizing of the gates (which specifies the "resistance" of the transistor path between power and ground) and the total metal area (which specifies the total capacitance, hence the load's energy dissipation per transition). Interconnect, which degrades the signal edge rates between gates, must therefore be considered in the early phases of design from both a delay and a power point of view.

In this paper we describe a methodology for RC *interconnect synthesis*. In contrast to manually designing an RC interconnect by calculating its moments [10], analyzing its delay, and modifying the interconnect repeatedly (a trial-and-error procedure) to achieve the target slopes and delays at the fanout nodes of interest, we generate the moments which must yield the required slopes and delays, and then modify the interconnect to fit these moments. Moment sensitivities are used to guide the search for the interconnect wire widths which achieve these objectives. A direct optimization method is used to fine tune the results.

1.1 Background

In [5], a brute-force approach for calculating the delay and slope sensitivities of linear interconnect was proposed. The approach was based on AWE [11] and the AWE time domain sensitivities [5]. For each critical fanout node, an AWE-based adjoint analysis technique was used to compute the time-domain voltage sensitivities with respect to the circuit element values. However, no consideration was given to actually modifying the physical interconnect to achieve the objectives, and no consideration was given to the use of these sensitivities to head toward a design objective.

Recently, an algorithm for optimal wiresizing of the branches in an interconnect tree using an Elmore delay approximation [1, 2] was presented in [7]. The properties of monotonicity, separability, and dominance which apply to the Elmore delay result in an elegant wiresizing algorithm. However, the algorithm's optimality is predicated on the assumption that the wire resistance and capacitance per unit area remain fixed or scale linearly with the width. This restricts the approach to single layer routes and precludes the non-linear fringing and coupling effects observed in present technologies. Noting that monotonicity and separability do not apply to the interconnect tree for the Elmore delay under certain conditions, a sensitivity-based wiresiz-

[*] 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.

ing algorithm is presented in [17]. Both approaches model the driver by a resistor. The error resulting from using the Elmore delay in conjunction with a single-resistor gate model has been shown to be typically 20-30% [8] and can be higher when the gate and RC interconnect interaction is not considered.

2.0 Interconnect synthesis

We introduce our interconnect synthesis methodology in terms of a simple circuit consisting of a driver and a fanout tree structure as shown in Figure 1. We assume that the delay and slope requirements at the fanout nodes of interest (*critical sinks*) are specified, and that the original driver size is selected from a set of precharacterized gates. Delays and slopes are manipulated by changing the wire widths of the branches in the tree (wiresizing).

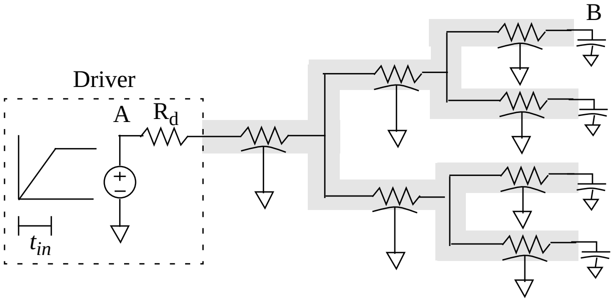


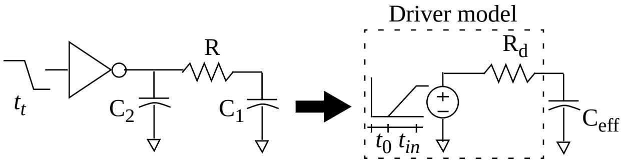
Figure 1. A driver and fanout tree.

2.1 The gate delay model

We assume that the driver has been precharacterized by a linear, time varying Thevenin equivalent as described in [14]. Each gate is modeled by a single fixed-value resistor, R_d, and a driving voltage ramp (Figure 2). The offset, t_0, and the transition time, t_{in}, of the voltage ramp are precharacterized for different values of the gate-input slope, t_t, and output capacitance, C_L, by:

$$t_0 = f(t_t, C_L)$$
$$t_{in} = g(t_t, C_L) \tag{1}$$

For a purely capacitive load, the gate delay and output slope are determined by solving for the waveform at the output of the gate model. For large fanout nets, however, the total capacitance of the interconnect is not a valid model of the load [12, 13]. In [13], the 2nd-order driving point admittance of the load, which is modeled by a π-circuit [9], is shown to be adequately accurate for such nets. Equation (1), however, applies for capacitive loads only and, therefore, an effective capacitance for the π-circuit is determined by averaging the current drawn by the π-circuit and the effective capacitance, C_{eff}, during the ramp transition time, t_{in} [14] (Figure 2). The effective capacitance captures the effect of the RC interconnect load on the driver. (For clarity, we do not consider the offset time, t_0, of the voltage source in the following subsections.)



$t_0 = f(t_t, \mathrm{C_{eff}})$
$t_{in} = g(t_t, \mathrm{C_{eff}})$

Figure 2. Thevenin equivalent voltage source of the driver and π-load.

2.2 Moment fitting

Given a precharacterized gate, the lengths of the branches of the fanout interconnect tree (Figure 1) and the target delays and slopes for the critical sinks, we vary the widths of the tree branches so as to meet the delay and slope specifications. For each critical sink, we first generate a target waveform which has the specified delay and slope. We then match the waveform at the critical sink to this target waveform by *moment fitting*.

If the desired output waveform at a critical sink, e.g. node *B* in Figure 1, is a ramp with a 50% delay, t_d, and a transition time, t_{out} (Figure 3), the widths of the interconnect tree should be varied so as to match the transfer function, $H(s)$ (= $V_B(s)/V_A(s)$), to:

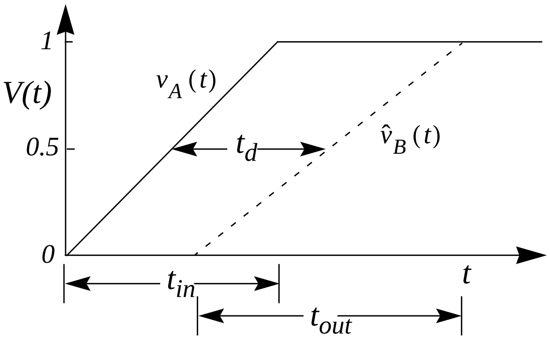


Figure 3. A ramp waveform at a fanout node.

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

$$= \hat{m}_0 + \hat{m}_1 s^2 + \hat{m}_2 s^2 + \ldots + \hat{m}_n s^n + \ldots$$

Ideally one can realize a transfer function as in (2) by forcing the *moments*, m_i, (like [11], we refer to the coefficients of *s* in the Taylor series expansion as moments although the two are related by $1/N!$) 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 [11], however, that due to the low-pass nature of RC interconnect trees, the voltage response at any node can be accurately modeled by its lower-order moments. Hence, matching the lower-order moments of $H(s)$ to the lower order moments of $\hat{H}(s)$ which are obtained from (2):

$$\hat{m}_0 = 1 \tag{3}$$

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

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

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

should yield a circuit with the targeted delay and slope. (We only show the first four moments of $\hat{H}(s)$ in (3). The actual order of this moment fitting depends on the required accuracy.)

Observe from (3) that for a ramp waveshape assumption at the critical sink, the first moment of $\hat{H}(s)$ evaluates to the specified delay, t_d. This implies that if the output were indeed perfectly modeled by a ramp, then by matching the first moment of $H(s)$—the familiar Elmore delay [1, 2]—to the first moment of $\hat{H}(s)$ the delay requirement could be met. However, it is apparent that the signal transition time, t_{out}, is a function only of the higher-order moments. Matching just m_1, the Elmore delay, is obviously insufficient!

2.3 Target moment generation

The moments in (3) are derived assuming a ramp at the critical sink, which is unrealistic for RC trees and meshes. There are no realizable RC circuits that can produce these moments due to the ideal waveshape assumption. Therefore, we instead generate target moments of the transfer function at a critical sink i in the RC interconnect tree in terms of a q-pole transfer function [11]:

$$\hat{H}^i(s) \approx \sum_{i=1}^{q} \frac{k_i}{s + p_i} \tag{4}$$

Equation (4) is uniquely specified in terms of the first 2q moments [11]

$$\hat{m}_j^i = \sum_{i=1}^{q} \frac{k_i}{p_i}\left(-\frac{1}{p_i}\right)^j \qquad 0 \le j \le 2q-1 \tag{5}$$

For generating the target moments at a critical sink with a specified target delay and slope, we determine a q-pole transfer function that would yield the specified delay and slope. The response of a q-pole system to a ramp input with input transition time, t_{in}, is given by:

$$v_{out}(t) = \frac{1}{t_{in}}\left\{t - \sum_{i=1}^{q} \frac{k_i}{p_i^2}(1 - e^{-p_i t})\right\} \qquad t \le t_{in} \tag{6}$$

$$= 1 - \frac{1}{t_{in}}\sum_{i=1}^{q} \frac{k_i}{p_i^2} e^{-p_i t}(e^{p_i t_{in}} - 1) \qquad t > t_{in}$$

We determine the k_i's and p_i's such that the ramp response, $v_{out}(t)$, exhibits the specified slope and delay. For example, if for a critical sink a 50% delay, t_d, and a slope, $1/t_{out}$, at the 50% point are specified, we determine k_i's and p_i's to minimize

$$\left\{v'_{out}\left(t_d + \frac{t_{in}}{2}\right) - \frac{1}{t_{out}}\right\}^2 + \left\{v_{out}\left(t_d + \frac{t_{in}}{2}\right) - 0.5\right\}^2 \tag{7}$$

subject to the constraint $v_{out}(0) = 0$. We note that any delay and slope definition—for example, the 50% delay and the 10-90% slope—can be used to determine the poles and residues of a q-pole transfer function with the specified ramp-response delay and slope. Given the target poles and residues, the target moments are computed from (5).

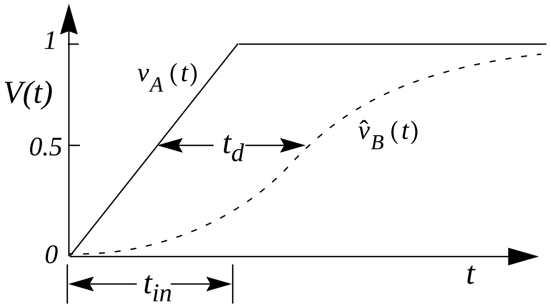


Figure 4. A q-pole output response.

The transfer-function moments at the critical sinks depend on the resistance and capacitance of the wires, which in turn depend on the widths and lengths. After determining the target moments at each critical sink, the transfer-function moments are matched to the target moments by sizing the wires in the interconnect tree. Moment sensitivity information (the Jacobian) is used to guide an iterative search to determine the necessary change in the wire widths at each iteration.

The next subsection describes an efficient algorithm for constructing the moment sensitivity matrix. We then describe the optimization technique used to match the transfer-function moments to the target moments. Though we present the optimization assuming wire width variation, this approach can be easily extended to handle length variation too.

3.0 Moments and sensitivity computation

Each branch of the RC-tree is a distributed resistance-capacitance (RC) line which can be modeled by one or more equivalent L-, T-, or π-circuits [13]. (To simplify the terminology, the equations in the following subsections are presented assuming an L-model.) The branch connecting node *i* to its parent node in the RC tree is labeled *i*. The length of a wire *i* is represented by l_i and its width by w_i. The resistance of branch *i* is denoted by R_i and the capacitance at node *i* by C_i. *(*R_i and C_i depend on the length, width, and layer of wire *i*.) *D(i)* denotes the set of nodes downstream of node *i* while the set of branches along the path from node *i* to the root of the tree is denoted by *P*(*n*). We also assume that the tree has *N* branches. We optimize for the first *2q* moments at each of the *M* specified critical sinks in the tree.

3.1 Moment Computation

It can be shown [11] that the k^{th} moment of the impulse response at a node *n*, m_k^n, in an RC tree is obtained by replacing the capacitances at all nodes *j* in the circuit by current sources of value $C_j(-1)^{k-1}m_{k-1}^j$ and computing the voltage at node *n*. That is,

$$m_k^n = (-1)^k \sum_{i \in P(n)} R_i M_{k-1}^i \qquad (8)$$

where

$$M_{k-1}^i = \sum_{j \in D(i)} C_j \, (-1)^{k-1} m_{k-1}^j \qquad (9)$$

The zero-th moment of the impulse response for all nodes is unity and the first moment is the well-known Elmore delay [1, 2]. From (8) and (9), it is obvious that moment computation takes *O*(*N*) time.

3.2 Moment Sensitivity Computation

The *first-order* sensitivities of the moments with respect to various elements in the RC-tree can be computed by the AWE-based adjoint analysis technique described in [4]. For RC-trees, however, the *exact* moment sensitivities can be directly computed in terms of the partial derivatives of the moments with respect to resistance, capacitance, and the lower order moments. That is,

$$\frac{\partial m_k^n}{\partial w_l} = \frac{\partial m_k^n}{\partial R_l}\frac{\partial R_l}{\partial w_l} + \frac{\partial m_k^n}{\partial C_l}\frac{\partial C_l}{\partial w_l} + \sum_{\forall i} \frac{\partial m_k^n}{\partial m_{k-1}^i}\frac{\partial m_{k-1}^i}{\partial w_l} \qquad (10)$$

The summation in (10) is over all nodes *i* in the tree. Differentiating (8) with respect to the appropriate quantities gives the partial derivatives of the k^{th} moment with respect to the resistance, capacitance and the lower moments:

$$\frac{\partial m_k^n}{\partial R_l} = (-1)^k M_{k-1}^l \qquad n \in D(l) \qquad (11)$$

$$= 0 \qquad \mathit{Otherwise}$$

The quantity $\partial m_k^n / \partial R_l$ is referred to as the *resistive sensitivity* of the k^{th} moment. Analogously, the *capacitive sensitivity* of the k^{th} moment is given by:

$$\frac{\partial m_k^n}{\partial C_l} = -\sum_{i \in P(n) \cap P(l)} R_i m_{k-1}^i \qquad (12)$$

The sensitivity of the k^{th} moment with respect to the $(k-1)^{th}$ moment at node *i* is given by:

$$\frac{\partial m_k^n}{\partial m_{k-1}^i} = -\sum_{u \in P(n) \cap P(i)} R_u C_i \qquad (13)$$

The partial derivatives of the resistances and the capacitances with respect to the widths are readily determined. If the resistances and capacitances scale with the widths ($R_l \propto 1/w_l$, $C_l \propto w_l$), the derivatives are given by:

$$\frac{\partial R_l}{\partial w_l} = -\frac{R_l}{w_l} \qquad \frac{\partial C_l}{\partial w_l} = \frac{C_l}{w_l} \qquad (14)$$

For notational simplicity, (14) assumes linear scaling of the resistance and capacitance with width. In present technologies, however, the coupling capacitances to adjacent conductors can be significant which results in non-linear fringing effects. That is, the fringe capacitance does not stay constant with width. In this case, the capacitance per unit length is modeled by an arbitrary (not necessarily linear) function, $c_l(w_l)$ (that is, $C_l = l_l c_l(w_l)$). Our approach can handle these nonlinear effects without loss of generality by computing $\partial C_l / \partial w_l$ from this function.

All of the quantities required for evaluating the Jacobian $\left[\frac{\partial m_k^n}{\partial w_i}\right]$ are given by (11)-(14). Exact moment sensitivity computation by this method is an $O(MN^2)$ operation. The approximate first-order sensitivity computation for an RC tree by the method described in [5] requires *O*(*MN*) time, where *M* is the number of critical sinks.

Note that for $k = 1$, (11) and (12) reduce to the familiar Elmore delay sensitivities presented in [15].

3.3 The Levenburgh-Marquardt algorithm

The moment sensitivities with respect to various wire widths are used to guide a search procedure to determine the best possible wire widths to meet the given specifications of delay and slope. In other words, this is cast as an optimization problem—determine the vector of widths ***W*** such that the mean square error between the 2*qM* target moments (since we optimize for the first 2*q* moments at each of the *M* critical sinks) $\hat{m}_j^i$, $1 \le i \le M, 0 \le j \le 2q-1$, and $m_j^i(w_1, w_2, \ldots, w_N)$, $1 \le i \le M, 0 \le j \le 2q-1$, is minimized.

The mean square error is defined as

$$\Phi = \sum_{i=1}^{M} \sum_{j=0}^{2q-1} \{ \hat{m}_j^i - m_j^i(\boldsymbol{W}) \}^2 \qquad (15)$$

We minimize Φ by the Levenburgh-Marquardt technique [6] which iteratively solves the equation

$$\left(\boldsymbol{S}^T \boldsymbol{S} + \lambda \boldsymbol{I} \right) \delta = \boldsymbol{S}^T (\hat{\boldsymbol{m}} - \boldsymbol{m}) \qquad (16)$$

where $\boldsymbol{S}$ is the $2qM \times N$ gradient matrix (Jacobian) with entries:

$$S_{2q(i-1)+j+1,\, l} = \frac{\partial m_j^i}{\partial w_l} \qquad (17)$$

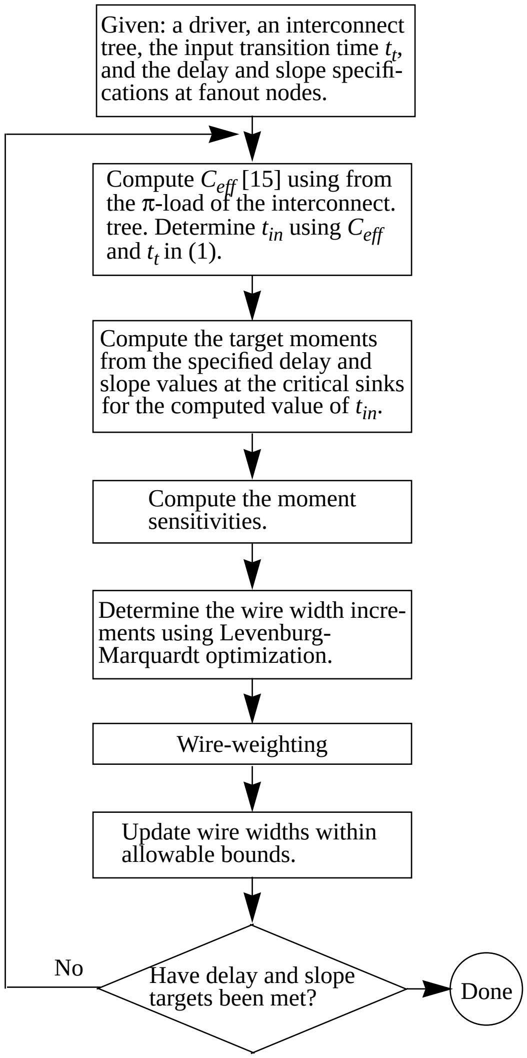
The solution δ of (16) is the vector of suggested wire width increments which is added to current width vector $\boldsymbol{W}$ before proceeding to the next iteration. The parameter λ in (16) is adjusted between iterations to enhance convergence to the final solution. The iterative method described above combines the benefits of the fast convergence of steepest descent methods in the initial stages to reach an approximate solution, and the convergence properties of Taylor series truncation methods in the final stages. Initially λ is set to a high value to effect a steepest descent method and as the solution converges, λ is reduced to a very small value to emulate the fast convergence properties of the Taylor series truncation method.

3.3.1 Wire-weighting during optimization

During Levenburgh-Marquardt optimization, we have assumed that every wire in the interconnect tree, regardless of its location on the chip or position in the tree, is equally important from a wire sizing point of view. This blind optimization will not yield the optimal solution since some wires are more important than others. For example, it is well known that widening the wires closest to the root of an RC tree is the most area-efficient way of reducing its delay [15]. Also, it is possible that some wires pass through congested areas on the chip and, therefore, they should be widened as little as possible. Hence, during optimization it is necessary to weight the wires to obtain an appropriate solution. These weights can be decided on the basis of the relative position of a wire in the tree, the routability of a wire, or a combination of such criteria.

It can be shown that during the early stages of optimization when value of λ is high (λ > 1 if (16) is normalized), weighting can be accomplished by multiplying the wire increment vector δ by a weight vector. When λ gets progressively small, this approach to weighting no longer applies. For small λ, since the suggested increments are small, weighting no longer has any effect on the overall solution quality.

Recognizing that the Elmore delay is a good indicator of the overall delay, for area efficiency during every iteration, each wire j in the tree is assigned a weight



Given: a driver, an interconnect tree, the input transition time t_t, and the delay and slope specifications at fanout nodes.
Compute C_{eff} [15] using from the π-load of the interconnect. tree. Determine t_{in} using C_{eff} and t_t in (1).
Compute the target moments from the specified delay and slope values at the critical sinks for the computed value of t_{in}.
Compute the moment sensitivities.
Determine the wire width increments using Levenburg-Marquardt optimization.
Wire-weighting
Update wire widths within allowable bounds.
Have delay and slope targets been met?
No
Done

Figure 5. Interconnect synthesis flow.

$$\alpha_j = \frac{1}{l_j} \sum_{i=1}^{M} \frac{\partial m_1^i}{\partial w_j} \left(\hat{m}_1^i - m_1^i \right) \qquad (18)$$

This weighting factor tends to favor wires with respect to which the critical sinks exhibit large Elmore delay sensitivities. Widening such wires has the maximum effect on delay with a minimal area penalty. Equation (18) also takes into account the difference in the current and target values of the Elmore delay at the critical sinks.

During each iteration, for every wire j the suggested width increment, δ_j, from (16) is multiplied by α_j/α_{max}, where α_{max} is the largest positive weight.

3.4 Post-moment-fitting optimization

Even though the moment-fitting approach outlined in Figure 5 is optimal in a least-squares sense, it is not guaranteed to yield a solution with the exact delay and slope requirements for two reasons: Primarily, the *q*-pole transfer function synthesized during target moment generation does not yield the specified delay and slope exactly. Also the Levenburg-Marquardt optimization technique minimizes but does not eliminate the mean-square-error between the transfer-function moments and the actual moments. Since there may be an error at the end of moment fitting which may result in the delays and/or slopes lying on either side of the target value, a more accurate technique is used after moment fitting which explicitly calculates the delays instead of using moment fitting.

Again, we use the Levenburg-Marquardt optimization technique. This time, however, we use the exact delays instead of using the moments to guide the search. The exact delay and slope at each critical sink is computed using RICE [10]. The voltage waveform sensitivities can be approximated using adjoint methods [4] or numerically estimated using finite difference methods. Both are somewhat costly, since the delay and slope sensitivities must be evaluated from the voltage sensitivities using Newton-Raphson to solve the transcendental response equations. We instead employ a more efficient technique by assuming a ramp-like waveform at each critical sink. Under this assumption, from (3), the sensitivity of the delay at a critical sink i, t_d^i, to the width of wire l, w_l, in the tree can be approximated by:

$$
\begin{aligned}
\frac{\partial t_d^i}{\partial w_l} &= \frac{\partial t_d^i}{\partial m_1^i}\frac{\partial m_1^i}{\partial w_l} + \frac{\partial t_d^i}{\partial m_2^i}\frac{\partial m_2^i}{\partial w_l} + \frac{\partial t_d^i}{\partial m_3^i}\frac{\partial m_3^i}{\partial w_l} \\
&= -\frac{\partial m_1^i}{\partial w_l} - \frac{1}{m_1^i}\frac{\partial m_2^i}{\partial w_l} - \frac{1}{m_2^i}\frac{\partial m_3^i}{\partial w_l}
\end{aligned}
\qquad (19)
$$

Similarly, $\partial t_{out}^i / \partial w_l$ can be derived from (3). The solution of the moment-fitting technique serves as an initial solution for the more exact fine-grained optimization possible with exact delay computation. If efficiency is not a consideration, this technique using (19) can be used instead of moment fitting!

4.0 Driver selection

The interconnect synthesis procedure described above assumes that the gate used to drive the interconnect tree has already been selected from a set of gates of different driving strengths (sizes). However, selecting the optimal gate to drive an interconnect tree is a non-trivial part of the interconnect synthesis problem. The trade-off here is between area, power and delay. Larger drivers occupy more area and dissipate more power but they result in lower overall delays and *vice versa*.

The reduction in delay as a function of the area during optimization for a typical net and a fixed driver is shown in Figure 6. It is shown in [17] that the total area required to achieve the minimum delay by wiresizing only can be excessively large. For interconnect synthesis, where functionally identical gates of different driving capabilities are available, it may be more optimal to replace the gate by a stronger gate instead of adding more metal to the interconnect tree at the cost of silicon area. Hence, a technique for determining whether a driver is too weak for a given net is essential. That is, during optimization it should be possible to tell whether gate sizing is preferable to wiresizing. We use sensitivities to accomplish this.

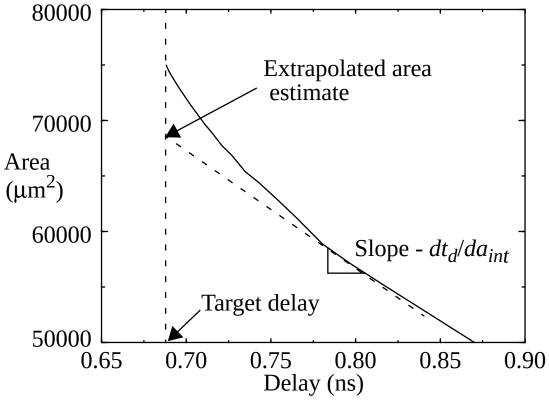


Figure 6. Variation in delay as a function of the interconnect area for a fixed driver.

The sensitivity of the delay at a critical sink i with respect to the interconnect area, a_{int} $(=\sum w_j l_j)$, can be approximated by:

$$
\frac{dt_d^i}{da_{int}} \approx -\frac{\sum_{j=1}^{N} \frac{\partial m_1^i}{\partial w_j}\Delta w_j}{\sum_{j=1}^{N} \Delta w_j l_j}
\qquad (20)
$$

where Δw_j $(= \alpha_j \delta_j / \delta_{max})$ is the width increment of wire j at each iteration. The sensitivity of the overall delay with respect to the driver resistance, dt_d^i/dR_d, can be approximated by $\partial m_1^i / \partial R_d$, which is calculated from (11). During each iteration of the optimization loop, the relative values of dt_d^i/da_{int} and dt_d^i/dR_d indicate the suitability of the driver to the interconnect net. Based on the technology, a critical ratio of dt_d^i/dR_d to dt_d^i/da_{int} can be empirically determined which when exceeded during optimization indicates that wiresizing should be abandoned in favor of driver sizing. If sizing the driver is based upon area constraints, the area required for a particular target delay can be optimistically approximated by extrapolating the current delay with dt_d^i/da_{int} as shown in Figure 6. (As the optimization proceeds this approximation becomes more accurate.) This solution area can then be compared to the area that would be required by the next larger driver.

Ideally, one would begin with the smallest driver, perform wiresizing, discarding the driver in favor of the next larger driver when its unsuitability becomes apparent during optimization.

Net	No. of fanout nodes	No. of bran-ches	No. of critical sinks	Target percentage reduction in delay							
				15%				50%			
				% delay deviation	% slope deviation	% delay dev. (PMF)	% area increase	% delay deviation	% slope deviation	% delay dev. (PMF)	% area increase
Steiner	5	9	1	1.90	0.95	0.20	10.36	4.56	0.31	1.72	51.72
			2	0.46	0.15	0.31	17.82	4.00	1.73	4.00	72.00
			5	3.40	4.80	1.14	9.01	9.71	3.31	1.17	73.62
16-pin binary	16	31	1	0.54	0.45	0.05	15.46	0.28	0.001	0.09	27.74
			4	2.14	8.66	0.67	17.67	3.90	9.48	1.04	46.46
			16	5.98	9.47	0.59	13.43	8.10	10.23	2.22	42.29
Line	1	10	1	1.52	6.84	0.02	22.03	4.81	0.22	0.04	183.36

Table 1. Results for driverless nets.

5.0 Results

The interconnect synthesis procedure described in the flowchart in Figure 5 has been implemented in 4500 lines of C code. In our implementation, we assume a 2-pole transfer function for each critical sink to compute the target moments i.e. $q = 2$ in (6). We optimize for the first four moments with λ initially set to 10. A scaling factor of 0.8 is applied to λ during each iteration. Interconnect nets of various topologies—single lines, Steiner tree routes, and binary clock trees—were used to test our approach. The resistance and capacitance per unit area of these nets is 0.03 $\Omega/\mu m^2$ and 0.02 $fF/\mu m^2$ with a fringe capacitance of 0.01 $fF/\mu m$. The minimum and maximum allowable widths for interconnect branches were 1.0 μm and 6.0 μm respectively. Load capacitances range from 10 fF to 50 fF. When necessary, long runs of interconnect were broken into electrically shorter [13] segments (for example, a single long line).

5.1 Driverless interconnect nets

We tested our moment-fitting approach on several driverless nets with total capacitances ranging from 1pF to 2 pF. An input ramp with a transition time of 0.1 ns was applied at the root of the tree. The results for a 15% and 50% target reduction in delay and a 15% improvement in the slope at the critical sinks for some typical nets are shown in Table 1. The nets were initially routed at minimum width. The final delays and slopes were computed using RICE [10]. The appropriate columns show the percentage standard deviation, calculated as $\left[\frac{1}{M}\sum\left[\frac{\text{target value - final value}}{\text{target value}}\right]^2\right]^{1/2}$, in the final delays and slopes. The final solutions do not meet their targets exactly because of the error incurred during target moment generation. That is, the q-pole system synthesized for a specified target delay and slope is not guaranteed to yield the specified delay and slope.

The final wiresizing solution from moment fitting was used as the initial solution for the post-moment-fitting (PMF) technique described in Section 3.4. The results of this experiment are shown in columns 6 and 10. As expected, since the PMF technique computes exact delays to guide the search, it provides more accurate results. However, computation of exact delays is more expensive. Solutions from the PMF technique typically take a second of CPU time on a SPARCstation 10 compared to the few tens of milliseconds required using direct moment fitting.

We also observe that the quality of the solution tends to worsen as the number of critical sinks is increased for a particular net. Specifying more critical sinks constrains the optimizer since it tries to satisfy several possibly conflicting constraints by sizing the same wires, leading to a poor solution. Also, with more critical sinks, the likelihood of the existence of a perfect wiresizing solution with delay and slope targets exactly satisfied at each critical sink decreases.

5.2 Interconnect trees with drivers

Since the interconnect load affects the driver delay, it needs to be taken into account during interconnect synthesis. Due to its location at the root of the interconnect tree, the driver output impedance which remains fixed (unless the driver is resized) plays a dominant role in determining the overall delay. Thus adding a driver constrains the optimization procedure significantly. An inverter with a precharacterized output impedance (R_d) of 82Ω was used to drive the nets in Table 1. This driver was sized to provide an adequate driving strength (output transition time) for the example nets. We compute the driver delay using the precharacterized equations for the driver's Thevenin voltage source in (1). We again tried to reduce the delays and increase the slopes of the same nets by 15% from their initial values. The results of this experiment are shown in Table 2.

Again we see that the final delays are close to the target delays. Slopes, however, are not so easy to control since the driver influences the shape of the waveform at the root of the interconnect tree. Therefore, the gate size would have to be increased to achieve the target slopes more accurately for this example.

Net	No. of critical sinks	% delay devn.	% slope devn.	% delay devn. (PMF)	% area increase
Steiner	1	1.90	5.61	0.01	33.25
	2	2.32	6.24	0.35	35.60
	5	3.41	7.48	1.91	38.36
16-pin binary	1	1.42	0.07	0.12	11.58
	4	2.71	7.17	0.84	13.98
	16	3.64	8.12	0.80	9.80
Line	1	2.31	3.24	0.01	31.63

Table 2. Results for 15% targeted improvement for nets with driver.

We note that since most of the delay for these nets is due to the driver, total logic stage delay reduction on the order of 50% is not possible since the gate delay is not reduced by interconnect sizing.

5.3 Clock skew minimization

Clock skew minimization involves reducing the delay difference from the root to all fanout nodes of an interconnect tree. Our synthesis procedure handles this by setting all target delays to a single value. Table 3 shows the efficacy of our optimizer in minimizing clock skew on the relatively large unbuffered benchmark examples from [16].

Net	No. of pins	Initial skew (ns)	Target delay (ns)	Final skew (ps)
r1	267	0.51	3.7	22.8
r2	598	2.66	6.5	37.5
r3	862	2.11	12.0	68.6
r4	1903	6.21	24.0	85.4

Table 3. Clock skew minimization

6.0 Conclusions

Given the increasing dominance of interconnect in path delays for present technologies, there is a need for interconnect synthesis methodologies. We have presented a moment fitting approach which varies the wire widths of the interconnect tree to meet performance targets. We have also presented a wire-sizing technique that uses exact delays to guide the iterative search. We have shown that using approximate delay sensitivities derived for a ramp approximation at the fanout nodes of an RC tree, in conjunction with exact delay computation, yields accurate results. The sensitivity-based synthesis approach allows for selecting an appropriate driver for a particular interconnect structure.

In the future, we plan to address higher-level aspects of interconnect synthesis such as driver sizing and interconnect-driven path optimization.

References

[1] 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.

[2] W. C. Elmore, “The transient response of damped linear networks with particular regard to wide-band amplifiers,” *J. Applied Physics*, vol. 19, no. 1, pp. 55-63, Jan. 1948.

[3] S. W. Director, and R. Rohrer, “The generalized adjoint network and network sensitivities,” *IEEE Trans. Circuit Theory*, vol. 16, pp. 318 - 323, August 1969.

[4] J. Y. Lee, X Huang, and R. Rohrer, “Pole and zero sensitivity calculation in asymptotic waveform evaluation,” *IEEE Trans. Computer-Aided Design*, vol. 11, no. 5, pp. 586 - 597, May 1992.

[5] A. Balivada, D. R. Holberg, and L. T. Pillage, “Calculation and application of time-domain waveform sensitivities in asymptotic waveform evaluation,” *Proc. IEEE Custom Integrated Circuits Conference*, May 1991.

[6] 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.

[7] J. Cong, and K.-S. Leung, “Optimal wiresizing under the distributed Elmore delay model,” *Proc. of the Intl. Conf. on Computer-Aided Design*, November 1993.

[8] 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.

[9] P. R. O’Brien and T. L. Savarino, “Modeling the driving-point characteristic of resistive interconnect for accurate delay estimation,” *Proc. IEEE Intl. Conf. Computer-Aided Design*, November 1989.

[10] C. L. Ratzlaff, N. Gopal, and L. T. Pillage, “RICE: Rapid Interconnect Circuit Evaluator,” *Proc. 28th Design Automation Conference*, June 1991.

[11] 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.

[12] C. L. Ratzlaff, S. Pullela, and L. T. Pillage, “Modeling the RC-interconnect effects in a hierarchical timing analyzer,” *IEEE Custom Integrated Circuits Conference*, May 1992.

[13] N. Gopal and L. T. Pillage, “Evaluation of on-chip interconnect using moment matching,” *Proc. of the Intl. Conf. on Computer-Aided Design*, November 1991.

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

[15] S. Pullela, N. Menezes, and L. T. Pillage, “Reliable *non-zero* skew clock trees using wire-width optimization,” *Proc. 30th Design Automation Conferenc*e, June 1993.

[16] R. S. Tsay, “An exact zero-skew clock routing algorithm,” *IEEE Trans. Computer-Aided Design*, vol. 12, no. 2, pp. 242-249, February 1993.

[17] S. S. Sapatnekar, “RC interconnect optimization under the Elmore delay model,” *Proc. 31st Design Automation Conferenc*e, June 1994.