A New Layout-Driven Timing Model for Incremental Layout Optimization

Fang-Jou Liu, John Lillis, Chung-Kuan Cheng

Department of Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093
TEL: 619-534-8819
{fjliu, jlillis, kuan}@cs.ucsd.edu

Abstract

In this paper we present a new layout-driven timing model based on Asymptotic Waveform Evaluation (AWE) for improved timing analysis during routing. Our model enables the bottom-up computation of interconnect tree moments, and can be easily integrated with such a global router. Such an integration achieves *incremental layout optimization*, i.e., timing analysis and routing are tightly coupled, with feedback between them. This achieved incremental layout optimization, through our innovative timing model, is the main contribution of this work.

1. INTRODUCTION

With the rapid advances in VLSI technology, the speed bottleneck has shifted from gate delay to delay caused by on-chip interconnections. (In this paper we focus on chip-level interconnections.) Thus, reducing interconnection delay has become one of the most important goals in deep-submicron IC design.

In this paper we present a new approach for performance-driven routing by taking advantage of a new higher-order layout-driven timing model. This model, put simply, achieves the same goal as *Asymptotic Waveform Evaluation* (AWE) [8], that is, to obtain a more accurate timing estimation than afforded by Elmore delay. However, AWE, in its current form, can only be used for circuit *analysis*, not *routing*. That is, we still need to route first, *then* apply AWE to analyze the resulting topology, and if it does not conform to the electrical and timing requirements, do ripup and re-route, and so on. This not only limits the applicability of AWE to only circuit analysis, it still does not break the traditional route-extraction-simulation loop, which is often the bottleneck in design cycle time.

Our new timing model extends the applicability of AWE to the *routing domain*. That is, while routing is being performed, timing analysis is done at the same time, and subsequent routing is performed in accordance with the timing analysis results *at this moment*. Since routing is adjusted dynamically with the feedback from timing analysis, routing and timing analysis can be integrated and truly become one, achieving *incremental layout optimization*.

This work tries to address the shortcoming of [6]. [6] uses Elmore delay, which can produce a deviant timing estimation of as much as 20% (with respect to the real physical delay) under certain circumstances. Thus our objective is to derive a new, bottom-up timing model, based on AWE, that can use an arbitrary number of moments, and that can be easily integrated into a bottom-up global router such as [6]. As will be shown, our new timing model produces timing estimations of significantly higher fidelity (to physical delay) than afforded by Elmore delay, and this incremental, layout-driven timing model, along with its integration with our sink permutation-based global router, are the main contributions of this paper.

This paper is organized as follows. Section 2 gives an example to illustrate the inaccuracy of Elmore delay when some or all of the sinks have a nonzero initial required arrival time. Section 3 gives the structures of our new layout-driven timing models, and Section 4 presents experimental results. Section 5 concludes the paper and describes future work. Finally in Section 6 we survey some previous work.

2. AN EXAMPLE TO ILLUSTRATE THE PROBLEM OF ELMORE DELAY

In order to illustrate the problem of Elmore delay when some or all of the sinks have a nonzero initial required arrival time, here is a simple example with three points.

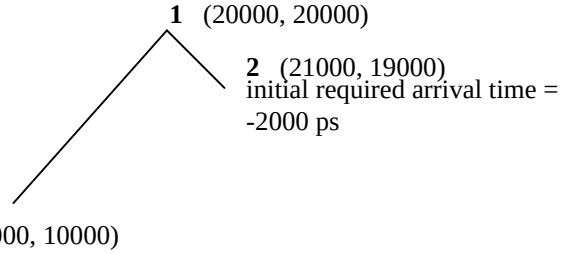

Figure 1 A simple example to illustrate the problem of Elmore delay

This work was supported in part by grants from the NSF MIP-9529077 as well as UC MICRO program.

ASP-DAC '97
0-89791-851-7$5.00

In this example, point one is the source, while points two and three are sinks. The coordinates are in microns (the figure is not in scale), with point two having a nonzero initial required arrival time of -2000 picosecond. (In our scheme, the more negative the initial required arrival time, the more critical the sink is.) Point three has an initial required arrival time of zero. The RC parameters are 0.008 ohm/micron and 0.00006 pf/micron, respectively. An imaginary point, point zero, is the driver, with a driver resistance of 25 ohm between it and point one. Please note point two, the closer sink to the source, is given a more stringent initial required arrival time, making it the critical sink instead of the usual point three were both sinks to have a zero initial required arrival time. This is to test the fidelity of Elmore delay under this situation. After routing, several Steiner points were added, which are not shown in this figure.

For this simple example, the three-pole solver using our new timing models predicts a delay from point zero (driver) to point two (critical sink) of 46.17ps, while Elmore delay's prediction is 99.96ps. The actual delay computed by SPICE is 44ps. The three-pole solver using our new models produces a timing prediction far more accurate than that of Elmore delay, which has an error of over 100%. This example illustrates clearly the need for a more sophisticated timing model, particularly with nonzero initial sink required arrival time. This is the focus of our work.

3. STRUCTURES OF THE LAYOUT-DRIVEN TIMING MODELS

Now we give the structures of our layout-driven timing models. First we will, as an example, list the structures of the second- and third-order models, and then we will give a formula for the general structure of models of an arbitrary order.

Here are some notations used in the models below. In our RC routing tree (using the lumped RC model):

- n_i and n_t denote the source and critical sink of the routing tree, respectively.
- n_{i+1} is the unique node one wire segment downstream from n_i on the path from n_i to n_t.
- $P_{i,i+1}$ denotes the unique path from n_i to n_{i+1}, while $P_{i+1,t}$ denotes the unique path from n_{i+1} to n_t.
- T_j is the subtree containing all the nodes rooted at n_j, including n_j itself and all its descendants.
- $f_{i,i+1}$ and $f_{i+1,t}$ are the values of *f*, the third-order delay model (the third moment), corresponding to $P_{i,i+1}$ and $P_{i+1,t}$, respectively, and $C_{i,t}{}^{f}$ represents the third-order mutual coupling between $P_{i,i+1}$ and $P_{i+1,t}$.
- $e_{i,i+1}$ and $e_{i+1,t}$ are the values of *e*, the second-order delay model (the second moment), corresponding to $P_{i,i+1}$ and $P_{i+1,t}$, respectively, and $C_{i,t}{}^{e}$ represents the second-order mutual coupling between $P_{i,i+1}$ and $P_{i+1,t}$.
- $d_{i,i+1}$ and $d_{i+1,t}$ are the values of *d*, the first-order delay model (the first moment, or Elmore delay), corresponding to $P_{i,i+1}$ and $P_{i+1,t}$, respectively.
- $R_{i,i+1}$ is the resistance between n_i and n_{i+1}; C_k is the capacitance to ground at node n_k *only* (not the lumped capacitance of the subtree rooted at n_k).
- $C_{T_{i+1}}$ is the lumped capacitance of the subtree rooted at n_{i+1}.
- $DD(n_{i+1})$ is the set of *direct descendants* of n_{i+1} (their common direct ancestor is n_{i+1}), with $|P_{i+1,j}| = 1$ for $\forall j \in DD(n_{i+1})$.
- $X_{i,i+1}$ and $X_{i+1,j}$ are the tree-recursive terms corresponding to $P_{i,i+1}$ and $P_{i+1,j}$, respectively, used in the computation of $e_{i,i+1}$.
- $Y_{i,i+1}$ and $Y_{i+1,j}$ are the tree-recursive terms corresponding to $P_{i,i+1}$ and $P_{i+1,j}$, respectively, used in the computation of $f_{i,i+1}$.

3.1 e, the second-order delay model

Theorem 1: $e_{i,t}$, the second-order delay model from source n_i to sink n_t, can be computed bottom-up as:

$$e_{i,t} = e_{i,i+1} + e_{i+1,t} + C_{i,t}{}^{e}$$

where

$$e_{i,i+1} = R_{i,i+1} X_{i,i+1}$$

$$X_{i,i+1} = d_{i,i+1} C_{T_{i+1}} + \sum_{j \in DD(n_{i+1})} X_{i+1,j}$$

$$C_{i,t}{}^{e} = d_{i,i+1} d_{i+1,t}$$

3.2 f, the third-order delay model

Theorem 2: $f_{i,t}$, the third-order delay model from source n_i to sink n_t, can be computed bottom-up as:

$$f_{i,t} = f_{i,i+1} + f_{i+1,t} + C_{i,t}{}^{f}$$

where

$$f_{i,i+1} = R_{i,i+1} Y_{i,i+1}$$

$$Y_{i,i+1} = e_{i,i+1} C_{T_{i+1}} + \sum_{j \in DD(n_{i+1})} Y_{i+1,j} + d_{i,i+1} \sum_{j \in DD(n_{i+1})} X_{i+1,j}$$

$$C_{i,t}{}^{f} = e_{i,i+1} d_{i+1,t} + d_{i,i+1} e_{i+1,t}$$

3.3 A general method for constructing layout-driven timing models

From Section 3.1 and Section 3.2 we can clearly see that there is a general pattern in the forms of the higher-order delay models. Here $m^{(k)}$ denotes the *k*th-order model ($k \geq 2$), corresponding to the *k*th moment.

- For source n_i and sink n_t:

$$m_{i,t}{}^{(k)} = m_{i,i+1}{}^{(k)} + m_{i+1,t}{}^{(k)} + C_{i,t}{}^{(k)}$$

  where n_{i+1} is the unique node one segment downstream from n_i on the path from n_i to n_t. $m_{i,i+1}{}^{(k)}$ is the value of the *k*th-order model corresponding to $P_{i,i+1}$, $m_{i+1,t}{}^{(k)}$ to $P_{i+1,t}$, and $C_{i,t}{}^{(k)}$ is the *k*th-order coupling between $P_{i,i+1}$ and $P_{i+1,t}$.

- For the single wire segment $P_{i,i+1}$ with $|P_{i,i+1}| = 1$:

$$m_{i,i+1}{}^{(k)} = R_{i,i+1} X_{i,i+1}{}^{(k)}$$

  where $R_{i,i+1}$ is the resistance between n_i and n_{i+1}, and

$$X_{i,i+1}{}^{(k)} = \sum_{j \in DD(n_{i+1})} X_{i+1,j}{}^{(k)} + m_{i,i+1}{}^{(k-1)} C_{T_{i+1}} + \sum_{l=1 \ldots k-2} \left(m_{i,i+1}{}^{(l)} \sum_{j \in DD(n_{i+1})} X_{i+1,j}{}^{(k-l)} \right)$$

  where $DD(n_{i+1})$ is the set of direct descendants of n_{i+1}, and $C_{T_{i+1}}$ is the total lumped capacitance of the subtree rooted at n_{i+1}.

- For the *k*th-order coupling term

$$C_{i,t}{}^{(k)} = \sum_{l=1 \ldots k-1} m_{i,i+1}{}^{(l)} m_{i+1,t}{}^{(k-l)}$$

4. EXPERIMENTAL RESULTS

We have integrated the new timing models into our sink permutation-based global router [6]. The timing engine uses three poles, requiring the first- through fifth-order delay models (corresponding to the first five moments). The reason for using three instead of the usual two poles is that, according to our experience, two poles are generally not enough to achieve the accuracy we want, and one more pole is required for reasonably accurate delay modeling.

Two kinds of experiments were performed. In the first kind of experiment, we set the initial sink required arrival time to zero, to demonstrate that with zero initial required arrival time three-pole and Elmore delay produce results of comparable quality, in terms of actual required arrival time, as well as area, with three-pole having a much smaller discrepancy between predicted and actual required arrival time at the root. In the second kind of experiment we set the initial required arrival time of some sinks to be nonzero, to show that under this situation the three-pole model produces routing topologies significantly better than those of Elmore delay, both in terms of the required arrival time at the root, and the area of the routing tree. The test cases were randomly generated.

4.1 Zero Initial Sink Required Arrival Time

In this experiment the initial sink required arrival time of each sink is set to zero. Table 1 summarizes the results.

For a specific sink number, we applied our approach to three test cases. The rows titled "Predicted" denote the required arrival time (in picoseconds) of the root of the routing tree predicted by the three-pole and Elmore models, respectively, and the rows titled "Actual" denote the actual required arrival time (in picoseconds) of the root as computed by SPICE. "Area" denotes the area costs (in micron^2) of the routing trees produced by three-pole and Elmore delay, respectively. As we can see, with zero initial sink required arrival time, Elmore delay and the three-pole model produce results of comparable quality (the actual required arrival time

Zero Initial Sink Required Arrival Time		6 Sinks			9 Sinks		
		Case 1	Case 2	Case 3	Case 1	Case 2	Case 3
Three-Pole	Predicted	3003	2020.8	1844.4	3295.5	3973.9	3978.6
	Actual	3027	2041	1865	3300	3976.3	3967.7
	Area	319903	258494	184755	473132	340966	332016
Elmore	Predicted	2773	1820.6	1676.5	2986.47	3573.2	3567.4
	Actual	3027	2041	1865	3300	3910.7	3967.7
	Area	319903	258494	184755	473132	337510	332016

Table 1 Comparison of the three-pole model with Elmore delay. The predicted and actual required arrival time values are in picoseconds, while the area's unit is micron^2.

Nonzero Initial Sink Required Arrival Time	6 Sinks			9 Sinks		
	1/3	2/3	All	1/3	2/3	All
AT (%)	12.4	12.5	15.7	27	3.4	19.5
Req. (%)	4.2	0.76	0.73	2.87	3.19	0.5
Area (%)	8.5	11.8	15.1	25.1	0.23	19.1
Elmore Error (%)	21.5	3.8	5.4	8.7	7.3	5.3
Three-Pole Error (%)	1.8	0.071	0.27	0.1	1.7	0.02

Table 2 Improvements of the three-pole model over Elmore delay for different metrics

values computed by SPICE are mostly the same for the two models, and the area costs are mostly the same). However, please note there is a 10% to 20% improvement of three-pole over Elmore delay in terms of the proximity of the predicted and actual required arrival time. Thus with zero initial required arrival time our router's advantage is that it significantly closes the gap between predicted and actual required arrival time, while at the same time producing routing topologies of comparable quality to those of Elmore delay.

4.2 Nonzero Initial Sink Required Arrival Time

Table 2 summarizes the results with nonzero initial sink required arrival time. For a specific sink number, we set 1/3, 2/3, and all of the sinks' initial required arrival time to be nonzero, and measured the improvements of the three-pole model over Elmore delay, using several metrics. "AT" is area*(actual required arrival time at the root of the routing tree as computed by SPICE); "Req." is the actual required arrival time at the root of the routing tree; "Area" is the area of the routing tree; the last two rows represent the discrepancy between the predicted and actual required arrival time values at the root, as a percentage of the actual required arrival time, for Elmore delay and three-pole, respectively. As can be seen, our new three-pole timing model has two significant advantages over Elmore delay:

1. Our three-pole model produces a routing tree that is better than that produced by Elmore delay, both in terms of the required arrival time at the root, and the area taken by the tree. Particularly the area metric is improved significantly compared to that of Elmore delay, with an average improvement of well over 10%, and as high as 25%. Similarly the AT metric is improved substantially.

We observed a 20% improvement in the **delay** from the root to the critical sink. However, because of the nonzeroness of the initial required arrival time, we cannot simply use the delay as the timing metric; instead we chose to use as the timing metric the required arrival time at the root of the routing tree, resulting in a much smaller improvement percentage (because of the "weight" of the initial required arrival time added in). However, the observation of a 20% improvement in delay is indeed consistent with the literature [5]. Also, with the increasing emphasis on consumer electronics, chip size has become a major concern, and our approach has the potential to make a significant contribution here.

1. Our three-pole timing model substantially reduces the discrepancy (error percentage, as represented in the last two rows of Table 2) between the predicted and actual required arrival time. As can be seen, there is a discrepancy of as high as 21.5% between the predicted and actual required arrival time given by Elmore delay, while for our three-pole model this discrepancy is reduced to less than 2%, a significant improvement over Elmore delay.

From Section 4.1 and Section 4.2 we can see that our new models produce results of comparable quality to those produced by Elmore delay, when all sinks have a zero initial required arrival time, with significantly reduced discrepancy between predicted and actual required arrival time. Furthermore, our new models produce timing predictions of significantly higher quality with nonzero initial required arrival time, with an average improvement over Elmore delay of well over 10%. These two qualities make our router suitable not only for cases with nonzero sink initial required arrival time, but for **general use** as well.

5. CONCLUSION AND FUTURE WORK

In this paper we have presented a new model for recursive, bottom-up timing analysis based on AWE. This model is based on higher-order moments, and can use an arbitrary number of moments in computation. This model is structured to facilitate a global router that proceeds from the leaves (sinks) of the routing tree to the root. We have integrated our new model with our sink permutation-based global router and shown that it produces routing trees of significantly higher quality than produced by Elmore delay, particularly when some or all of the sinks have a nonzero initial required arrival time. This integration is, to the best of our knowledge, the first attempt to adapt AWE to the routing domain, achieving tight integration in routing and timing analysis.

For future work we intend to improve the execution speed of our model, and to investigate more closely the relationship of the critical sink pinpointed by the new model and Elmore delay (sometimes they pinpoint the same sink as critical; sometimes they don't). A close investigation of this problem may result in even more accurate timing predictions in the future.

6. Previous Work

6.1 Elmore Delay

Elmore delay [4] is the most popular delay metric used in performance-driven routing today. Elmore delay's advantage lies much in its simplicity—it's the only delay metric that does not require the repetitive transforms between the time- and frequency-domains to compute physical delay. However, it has been observed that under certain circumstances Elmore delay will produce large errors, and a timing metric based on at least two poles is required to produce a timing prediction reasonably close to physical delay.

6.2 SERT

Boese et. al.'s SERT [1] produces one of the best results among all routing algorithms. SERT is a greedy algorithm, and it routes using criteria in accordance with achieving the local optimum. Although SERT produces very good results, the area overhead is often unbearably large. It often produces star-shaped routing topologies with the source at the center and the sinks at the leaves, at the expense of the routing area. Because of this, the practicality of this algorithm is unclear.

6.3 A-Tree

The A-Tree approach of Cong et. al. [3] attempts to find a min-area routing tree, to be used as the minimum delay tree. A-Tree uses a linear delay model (area corresponds to delay). However in the real world area is not necessarily closely associated with delay, and this may cause A-Tree to produce timing estimations very different from SPICE simulations.

6.4 A Geometric Programming Approach

Sapatnekar [11] studied how to minimize the *maximum source-to-sink delay*. He noted that under this situation, the *separability* property of Cong's approach no longer holds. Sapatnekar proposed an approach based on geometric programming to handle the *continuous* wire sizing problem, i.e., the wire widths are not discretized. This is followed by a *mapping phase* to discretize the wire widths to make them compatible with IC process technologies.

6.5 Moment Matching

Moment matching, or *Asymptotic Waveform Evaluation* (AWE), in its current form originated from the work of Pillage [8]. The idea is to first obtain the *moments* (coefficients) of the transfer function, then match these moments to a Padé approximant. Once we have the Padé approximant, partial fraction expansion is performed to facilitate inverse Laplace transform, from which we can obtain the time-domain response of the waveform, with which we can solve for the desired output value.

Since [8], there has been much research on moment matching [7], [9], [10]. Recently an approach called Padé via Lanczos (PVL) [5] was proposed that has purportedly superior numerical stability to AWE's, and has the same computational complexity.

6.6 A Sink Permutation-based Global Router

The work of Lillis, et al. [6] harnesses the power of dynamic programming for global routing. [6] is based on the idea of *sink permutations*, that is, to permute the order of sinks to find one with the least delay, area, or to derive a *trade-off curve* between delay and area. To do this, a minimum spanning tree (MST) for the sinks is first found, then this MST is converted to a minimum tour-length tree, using the traveling salesman's heuristic, then sink permutations are performed on this tree. The algorithm outputs a trade-off curve relating delay and area, allowing the designer to explore the design space.

[6] produces very good results, but because of its use of Elmore delay, under certain circumstances large errors (up to 20%, with respect to delay calculated by SPICE) will still occur, especially when some (or all) of the sinks have nonzero initial required arrival time values. One of the objectives of this work is to address this shortcoming to obtain more robust timing estimations.

Bibliography

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

[2] J.J. Cong, K.S. Leung, "Optimal Wire Sizing Under Elmore Delay Model," *IEEE Trans. on CAD*, v.14 no.3 (1995) pp. 321-336.

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

[4] W.C. Elmore, "The Transient Response of Damped Linear Networks with Particular Regard to Wideband Amplifiers," *J. Appl. Phys.*, 19(1):55-63, 1948.

[5] P. Feldmann and R.W. Freund, "Efficient Linear Circuit Analysis by Padé Approximation via the Lanczos Process," *Proc. Of Intl. Conf. on Computer-Aided Design*, November 1994.

[6] J. Lillis, C. K. Cheng, T. T. Lin, "New Performance-Driven Routing Techniques with Explicit Area/Delay Tradeoff and Simultaneous Wire Sizing," *Proc. of Design Automation Conf.*, June 1996.

[7] 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.

[8] 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.

[9] J. Qian, S. Pullela, and L.T. Pillage, "Modelling the Effective Capacitance for the RC Interconnect of CMOS Gates," *IEEE Trans. Computer-Aided Design*, vol. 13, no. 12, pp. 1526-1535, December 1994.

[10] C.L. Ratzlaff, N. Gopal, and L.T. Pillage, "RICE: Rapid Interconnect Circuit Evaluator," *Proc. 28th ACM/IEEE Design Automation Conf.*, June 1994.

[11] S.S. Sapatnekar, "RC Interconnect Optimization under the Elmore Delay Model," *Proc. ACM/IEEE Design Automation Conf.*, 1994, pp. 387-391.