

5

ITERATIVE RLC MODELS FOR INTERCONNECT DELAY OPTIMIZATION IN VLSI ROUTING ALGORITHMS

Zulkifli Md-Yusof
Mohamed Khalil-Hani
Nasir Shaikh-Husin
Muhammad Nadzir Marsono

5.1 INTRODUCTION

Buffer insertion (van Ginneken, 1990), and wire-sizing techniques (Lillis, Cheng and Lin, 1996) have been widely used to minimize global interconnect delay path between interconnect source and sink points. These techniques rely on delay models (Pileggi, 1995) to estimate buffer insertion points – from simple first order linear model (Elmore, 1948) to more complex moment matching techniques (Ismail, Friedman and Neves, 1999a). Thus, interconnect analysis and modeling is of paramount importance in realizing a successful global interconnect routing. For effective buffer insertion point estimation, both source-to-sink and sink-to-source delay estimation may be used (Shaikh-Husin and Khalil-Hani, 2007).

As VLSI fabrication technology scales to smaller feature sizes and larger layout areas, global interconnect delay increasingly dominates device delay (Bakoglu, 1990). In the nanometer range, the effect of inductance becomes much more significant and

therefore, needs to be considered in interconnect routing algorithms (Tores, 1995). This reverse-scaling phenomenon results in smaller interconnect dimensions, and hence, slower signal transmission. It has also been shown that inductance affects area and power consumption (Ismail, Friedman and Neves, 1999b). Therefore, interconnect delay models should now include inductance parameter.

This chapter proposes closed-form iterative interconnect delay models for delay optimization on global interconnects in deep-submicron VLSI layout designs. The complex moment-matching technique in (Ismail, Friedman and Neves, 1999a) is adapted to estimate the source-to-sink and sink-to-source delays for use in the determination of buffer insertion points. The models are targeted for application in a routing algorithm which inserts buffer simultaneously as the shortest delay path is searched (Shaikh-Husin and Khalil-Hani, 2007).

This chapter is organized as follows. Related work is discussed in Section 5.2. Section 5.3 presents the proposed delay models and explains the application of these models in a simultaneous routing and buffer insertion algorithm. Experimental work and results are presented in Section 5.4. Conclusion and future work are presented in Section 5.5.

5.2 RELATED WORKS

As VLSI design reaches deep submicron technology, the delay model used to estimate delay for interconnect routing has evolved from the simplistic lumped RC model (Rubinstein, Penfield and Horowitz, 1959) to the sophisticated high-order moment matching delay model (Ismail, Friedman and Neves, 1999a). In the lumped RC model, R refers to the resistance of the driver, and C refers to the sum of the total capacitance of an interconnect and the total gate capacitance at the sink/source. Elmore wire model offers fidelity for estimating delay in global interconnect routing

techniques (Elmore, 1948). However it does not include the inductance effects, which has been proven to show 35% delay overestimation in nanometer VLSI technology (Ismail, Friedman and Neves, 1999b). Ismail, Friedman and Neves (1999a) has successfully developed a closed form solution for delay, rise time, overshoots, and settling time estimations for signals in an RLC tree.

There are two distinct techniques to route global interconnects – sequential (Sherwani, 1999) and concurrent (Hu and Shing, 1985). These routing techniques can be further classified into two-terminal and multi-terminal routing problems. Maze routing algorithms are the most widely used for routing two-terminal nets on a grid graph (Moor, 1959). This approach relies on finding the shortest path between a source and a sink point as illustrated in the grid graph in Figure 5.1, before inserting buffers on selected points estimated by wire-only and buffer-terminated wire delay.

Recently, Shaikh-Husin and Khalil-Hani (2007) proposed a simultaneous routing and buffer insertion algorithm, called S-RABILA. The algorithm finds a buffered path such that the interconnect delay of the routed path is minimized. The execution time of S-RABILA is improved significantly by employing a novel look-ahead technique. The algorithm uses Elmore delay, without

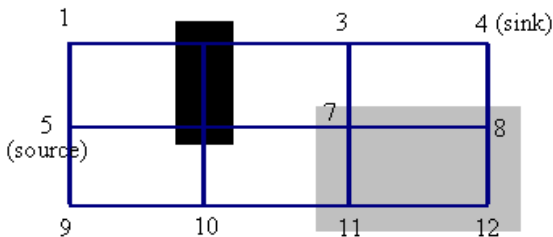


Figure 5.1 An example grid-graph. The dark area represents wire-obstacle area, whereas the gray area represents buffer-obstacle area.

taking into consideration the inductance effect. Our proposed delay models can be applied to improve the accuracy of S-RABILA.

5.3 PROPOSED INTERCONNECT DELAY MODEL

In this work, we adapt and improve the RLC interconnect model in (Ismail, Friedman and Neves, 1999a) for application in simultaneous routing and buffer insertion algorithm proposed in S-RABILA (Shaikh-Husin and Khalil-Hani, 2007).

In (Ismail, Friedman and Neves, 1999a), the RLC delay model at any node i is calculated as

$$t_{Di} = \frac{1.047 e^{(-\zeta_i/0.885)} + 1.39\zeta_i}{\omega_i} \quad (5.1)$$

where

$$\omega_i = \frac{1}{\sqrt{T_{LCi}}} \quad (5.2)$$

$$\zeta_i = \frac{1}{2} \frac{T_{RCi}}{\sqrt{T_{LCi}}} \quad (5.3)$$

$$T_{RCi} = \sum_k C_k R_{ik} \quad (5.4)$$

$$T_{LCi} = \sum_k C_k L_{ik} \quad (5.5)$$

In Equations 5.4 and 5.5, C_k refers to a capacitance component at any segment k . R_{ik} is the common resistance from input to nodes i and k , whereas L_{ik} is the common inductance from input to the nodes i and k .

The two summations in Equations 5.4 and 5.5 can be rewritten as

$$T_{RCi} = \sum_k C_{Tk} R_k \quad (5.6)$$

$$T_{LCi} = \sum_k C_{Tk} L_k \quad (5.7)$$

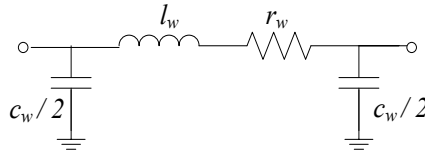
where R_k and L_k is the resistance and inductance of segment k , respectively. C_{Tk} is total load capacitance seen by R_k and L_k . T_{RCi} and T_{LCi} have to be obtained to estimate the delay at any node i . We now propose the iterative form of these equations, suitable for interconnect delay computation needed in S-RABILA algorithm.

Four forms of the model are proposed for wire-only and buffer-terminated interconnect types.

(i) *Wire-only source-to-sink interconnect delay estimation*

$$(r, l, T_{RCi}, T_{LCi})$$

$$(r', l', T'_{RCi}, T'_{LCi})$$



$$r' = r_w + r$$

$$l' = l_w + l$$

$$T'_{RCi} = (r + r_w/2)c_w + T_{RCi}$$

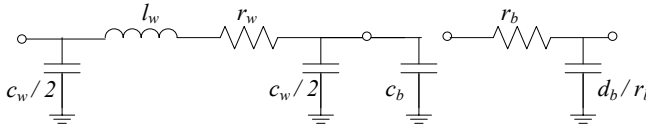
$$T'_{LCi} = (l + l_w/2)c_w + T_{LCi}$$

The calculation of elements in tuple $(r', l', T'_{RCi}, T'_{LCi})$ for the next node depends on the current segment wire parameters (c_w, r_w, l_w) and tuple (r, l, T_{RCi}, T_{LCi}) of the previous node.

(ii) *Buffer-terminated source-to-sink interconnect delay estimation*

$$(r, l, T_{RCi}, T_{LCi})$$

$$(r', l', T'_{RCi}, T'_{LCi})$$



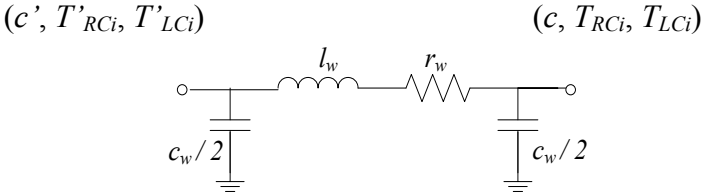
$$r' = r_b, l' = 0$$

$$T'_{RCi} = r(c_w + c_b) + r_w(c_w/2 + c_b) + d_b + T_{RCi}$$

$$T'_{LCi} = l(c_w + c_b) + l_w(c_w/2 + c_b) + T_{LCi}$$

Similar to the first model, tuple $(r', l', T'_{RCi}, T'_{LCi})$ for the next node depends on the current segment wire parameters (c_w, r_w, l_w) , buffer parameters (c_b, d_b) , and tuple (r, l, T_{RCi}, T_{LCi}) of the previous node.

(iii) *Wire-only sink-to-source interconnect delay estimation*



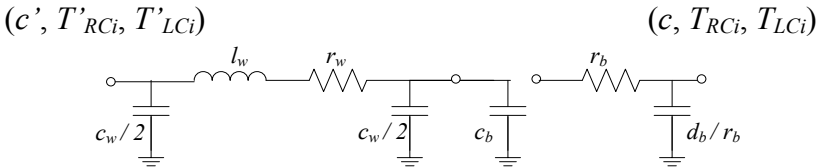
$$c' = c_w + c$$

$$T'_{RCi} = r_w(c_w/2 + c) + T_{RCi}$$

$$T'_{LCi} = l_w(c_w/2 + c) + T_{LCi}$$

Likewise, tuple (c', T'_{RCi}, T'_{LCi}) for the next node depends on the current segment wire parameters (c_w, r_w, l_w) and tuple (c, T_{RCi}, T_{LCi}) of the previous node.

(iv) *Buffer-terminated sink-to-source interconnect delay estimation*



$$c' = c_w + c_b$$

$$T'_{RCi} = r_w(c_w/2 + c_b) + d_b + r_b c + T_{RCi}$$

$$T'_{LCi} = l_w(c_w/2 + c_b) + T_{LCi}$$

The tuple (c', T'_{RCi}, T'_{LCi}) for the next node depends on the current segment wire parameters (c_w, r_w, l_w) , buffer parameters (c_b, d_b) , and element values in tuple (c, T_{RCi}, T_{LCi}) of the previous node.

With the proposed RLC models, four-tuple (r, l, T_{RCi}, T_{LCi}) or three-tuple (c, T_{RCi}, T_{LCi}) is needed for source-to-sink and for sink-to-source delay estimations, respectively. As elaborated in the previous chapter, we can also calculate delay in both directions simultaneously. In other words, we can start delay calculation beginning from both the source and from the destination. Somewhere in the middle, at node n , when partial delay from source to node n and partial delay from the sink to node n are known, the total delay from source to sink can be determined. This bidirectional method could shorten the time taken to complete the delay calculation. The total delay between two vertices could be estimated by using the value of T_{RCi} and value of T_{LCi} as given in Equations 5.8 and 5.9 below:

$$T_{RCi} = \sum_k C_k R_{ik} + \sum_k C_{Tk} R_k + R_{ik} * C_{Tk} \quad (5.8)$$

$$T_{LCi} = \sum_k C_k L_{ik} + \sum_k C_{Tk} L_k + L_{ik} * C_{Tk} \quad (5.9)$$

5.4 EXPERIMENTAL WORK

To confirm our forms of model with finding in (Pileggi, 1995), we use Predictive Technology Model (PTM) circuit parameters for an identical grid graph. The PTM parameters are compiled by the Nanoscale Integration and Modeling Group at Arizona State University and are available for download from the PTM website at <http://www.eas.asu.edu/~ptm>. Note that instead of (c, t) and (r, t) pairs for 1-D grid-graph as in Figure 5.2 for S-RABILA routing algorithm using Elmore model in (Shaikh-Husin and Khalil-Hani, 2007), the implementation with the proposed RLC models requires additional tuple elements, l and T_{LCi} . The resulting 1-D grid graph using the proposed RLC model is given in Figure 5.3.

Before discussing the result of the experimental work, the look-ahead technique applied in S-RABILA need to be explained. The look-ahead concept applied in S-RABILA is illustrated using the example grid graph in Figure 5.1. The dark area represents area where wire is not allowed, and the gray area represents area where

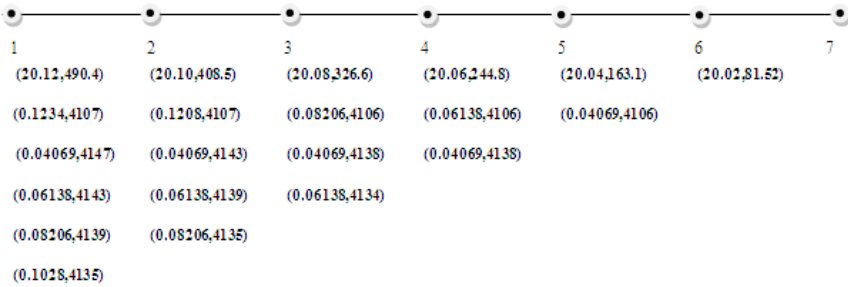


Figure 5.2 1-dimensional graph for determining look-ahead weight vectors. For each (c, t) pair, c is in pF and t is in ps.

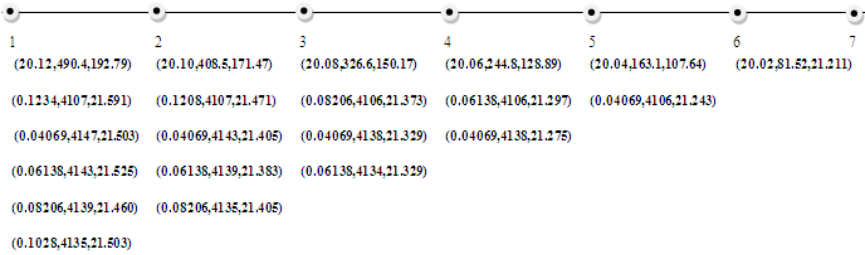


Figure 5.3 Sink-to-source 1-dimensional graph for determining look-ahead weight vectors. For each (c, T_{RCi}, T_{LCi}) tuple, c is in pF, T_{RCi} is in ps and T_{LCi} is in $\times 10^{-22}$ (Farad.Henry).

buffers are not allowed. Assume that vertex 5 is the source and 4 is the sink vertex. The topological distance between source and sink vertices must first be determined. The length of the shortest routing path from source-to-sink must avoid wire obstacles but could go through buffer obstacles. For the example in Figure 5.3, the source-to-sink topological distance is six.

A corresponding one-dimensional grid graph with length equal to the sink-to-source topological distance is created, as illustrated in Figure 5.2. The chosen wire parameters represent typical interconnect wires used in the PTM 65 nm fabrication process. Possible delay paths from each vertex to the sink vertex, for all

vertices with topological distance smaller than the source-to-sink topological distance, are computed. The capacitance-delay (c, t) pairs for each vertex are calculated using dynamic programming method, with only the non-dominated pairs stored for each vertex. The (c, t) pairs associated with each vertex provide the absolute lower bound of the delay from a vertex to the sink, since buffer can be inserted anywhere as necessary along the path. Hence, these (c, t) pairs can be viewed as look-ahead weights.

Shaikh-Husin and Khalil-Hani (2007) utilize the look-ahead pairs to predict the end-to-end delay, which can be calculated when both (r, t) and (c, t) pairs are available for a particular vertex. For a node M , the source-to-sink delay is given by:

$$d_E = t_M + t_m + r_M c_M \quad (5.10)$$

where (r_M, t_M) is the resistance-delay pair and (c_M, t_m) capacitance-delay pair are a look-ahead weights computed at node M . The use of look-ahead pairs speeds up the routing path construction, by guiding which vertex should be chosen next when sub-path expansion is carried out. Hence, both (r, t) and (c, t) pairs must be available for a particular vertex for buffer insertion algorithm using look-ahead technique.

Compared to the 1-dimensional graph in Figure 5.2, the proposed models result in better delay estimation. Figure 5.4 shows improvement to the delay estimation using S-RABILA when the RLC models are used instead of the Elmore RC model. The 35% delay overestimation using RC model confirms the findings in (Pileggi, 1995). Directly, for a targeted interconnect delay, the use of RLC models proposed here results in fewer buffers that could significantly reduce overall interconnect area and power dissipation.

5.5 CONCLUSION

In this paper, RLC interconnect delay models are proposed that are

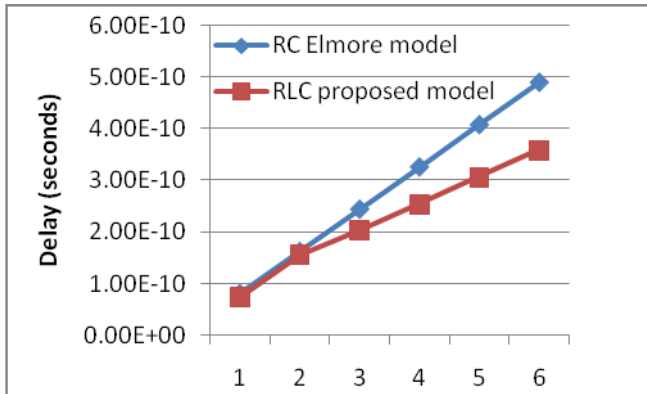


Figure 5.4 Delay calculation of six wire segments.

suitable for global interconnect routing as illustrated with the case study through S-RABILA algorithm. Simulated results showed that 35% delay overestimation could be avoided by using RLC models instead of Elmore model. The number of buffers needed to meet an interconnect delay target could be reduced. The work presented in this chapter can be extended to improve delay estimation overhead to meet multi-constraint global interconnect routing problems. For future work, we are interested to investigate computational overhead for interconnect delay estimation.

REFERENCES

- Bakoglu, H. B. (1990). *Circuits, Interconnections, and Packaging for VLSI*. Addison-Wesley.
- Elmore, W. C. (1948). The transient response of damped linear networks. *Journal of Applied Physics*. 19: 55 – 63.
- Hu, T. C. and M. T. Shing. (1985). A decomposition algorithm for circuit routing. In Hu, T. C. and E. S. Kuh (Ed.) *VLSI Circuit Layout* (pp. 144-152), IEEE Press.

- Ismail, Y. I., E. G. Friedman and J. L. Neves. (1999a). Equivalent Elmore delay for RLC trees. *Proc. 36th Design Automation Conference*. :715-720.
- Ismail, Y. I., E. G. Friedman and J. L. Neves. (1999b). Repeater insertion in tree structured inductive interconnect. *Proc. IEEE/ACM Int. Conf. on Computer-Aided Design*. :420-424.
- Lillis, J., C.-K. Cheng and T.-T. Y. Lin. (1996). Optimal wire sizing and buffer insertion for low power and a generalized delay model. *IEEE Journal of Solid-State Circuits*. 31:437-447.
- Moor, E. F. (1959). The shortest path through a maze. *Annals of the Harvard Computation Laboratory*. :185-292.
- Pileggi, L. (1995). Coping with RC(L) interconnect design headaches. *Proc. 1995 IEEE/ACM Int. Conf. on Computer-Aided Design*. :246-253.
- Rubinstein, R., P. Penfield and M. A. Horowitz. (1983). Signal delay in RC tree networks. *IEEE Trans. Computer-Aided Design*. 2(3):202–211.
- Shaikh-Husin, N. and M. Khalil-Hani. (2007). Optimal routing algorithm for minimizing interconnect delay in VLSI layout design. *Proc. Int. Conf. on Robotics, Vision, Information, and Signal Processing*. :345-349.
- Sherwani, N. A. (1999). *Algorithms for VLSI Physical Design Automation*. 3rd ed. Boston, MA: Kluwer Academic Press.
- Tores, J. (1995). Advanced copper interconnections for silicon CMOS technologies. *Applied Surface Science*. 91(1):112-123.
- van Ginneken, L. P. P. P. (1990). Buffer placement in distributed RC-tree networks for minimal Elmore delay. *Proc. Int. Symp. Circuits and Systems*. :865-868.