

SIMULTANEOUS BUFFER INSERTION AND NON-HANAN OPTIMIZATION FOR VLSI INTERCONNECT UNDER A HIGHER ORDER AWE MODEL

Jiang Hu

Sachin S. Sapatnekar

Department of Electrical and Computer Engineering
University of Minnesota, Minneapolis, MN 55455
jhu, sachin@mountains.ee.umn.edu

ABSTRACT

We present a simultaneous Buffer Insertion and Non-Hanan Optimization (BINO) algorithm to improve the performance of VLSI interconnect. This algorithm aims to address the realistic situation where both the interconnect resources and timing constraints are stringent and the wire topology is to be optimized using available spaces for buffer insertions after cell placement. These spaces are fixed relative to the changing routing tree during non-Hanan optimization. The objective here is to minimize weighted sum of wire and buffer cost subject to timing constraints. In BINO, buffer insertion and non-Hanan optimization are conducted simultaneously and iteratively in a greedy fashion till the improvements are exhausted. To assure the accuracy of timing evaluation, the fourth order AWE model is employed. Experimental results on both $.18\mu\text{m}$ IC and MCM technology showed significant cost reductions.

1. INTRODUCTION

As the VLSI technology develops rapidly into deep sub-micron era, interconnect performance becomes one of the key points for the overall performance of a VLSI system. In the early stages, most research was focused on optimizing the interconnect topology under geometric criteria such as minimizing the total wire length or the tree radius. Ever since it was noticed that geometric criteria may have large discrepancies from actual timing criteria, Elmore delay [1] based routing algorithms, such as SERT and P-tree [7, 9], have constituted the mainstream. Recently, the drawbacks of Elmore model have been addressed and higher order delay models have been applied to interconnect routing and optimization [8, 10, 11] to improve the accuracy of timing evaluation.

Additional efforts have been made to enhance in the formulation of the optimization problem and the solution search space. Minimizing cost subject to timing constraints is an appealing objective that is widely used [8, 9, 10, 12, 13]. One important advancement related to the solution search space is the extension to non-Hanan points in MVERT algorithm [13]. This extension brings significant benefit that is coherent with the objective. We will show that more cost reduction can be achieved than [13] if higher order delay models are employed.

Under deep sub-micron technology, usually topology optimization itself is not sufficient to meet the requirement to the interconnect performance. Buffer insertion is a powerful tool that can

be used to augment topology optimization in interconnect routing and optimization algorithms [15, 16, 17, 18, 19]. Most previous buffer insertion approaches have been implemented through dynamic programming in a bottom-up fashion and have provided remarkable improvements. However, all of these methods have been restricted to only Hanan routing.

The BINO algorithm introduced in this work combines buffer insertion with non-Hanan optimization under the fourth order AWE [3] delay model. Our algorithm is especially aimed toward the realistic situation where both the interconnect resource and timing constraint are stringent while there are still available spaces for buffers after cell placement. The positions of these spaces are fixed relative to the changing routing tree during optimization. This and the non-Hanan property distinguish the environment of our algorithm from previous works.

Since the routing tree is subject to change during non-Hanan optimization, there is no clear bottom-up structure to be exploited by dynamic programming. In addition, some candidate buffer positions on the routing tree paths may move away from the paths while some formerly off-path spaces may be traversed by a path and become candidate buffer positions. To deal with the increased complexity caused by this fact and the non-Hanan property, we guide each move in the optimization in a greedy fashion with the objective of minimizing the weighted sum of wire cost and buffer cost subject to timing constraints. The non-Hanan optimization and buffer insertion are conducted simultaneously and iteratively until the improvements are exhausted.

Our algorithm was tested and compared with SART and MVART (AWE versions of SERT[7] and MVERT[13]) on both $.18\mu\text{m}$ IC and MCM technology. Significant wire cost reduction is obtained as expected.

2. PRELIMINARIES

2.1. The problem environment

Our algorithm is applied in a post-placement scenario where buffer insertion is possible, but it is preferable to do so in regions that are left unoccupied by any cells, so as not to disturb the placement. The input to BINO then includes a set of pre-defined available buffer spaces scattered in the routing region, as demonstrated by the shaded area in Figure 1. Each buffer space is defined by its center position and radius. A more detailed depiction of a buffer space is shown in Figure 1(a). The *critical zone* of a buffer space is a region such that if a buffer is centered within it, the area the buffer occupied will not exceed the border of the buffer space. Only when the critical zone is traversed by a routing path, can a buffer be inserted in the space; and this buffer should be centered within the critical zone. Therefore, the critical zone is of more interest than buffer space itself. Note that when we say that a path passes through a buffer space, it means that the path passes through the critical zone of this buffer space. For simplicity, we use spaces

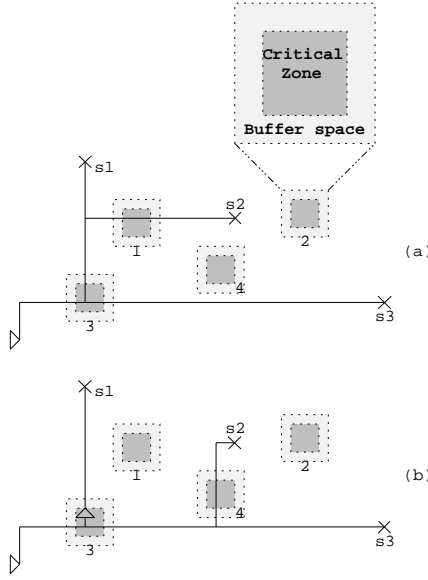


Figure 1. The fixed buffer space positions and the routing tree that changed from (a) to (b) during the optimization.

of equal size and only one buffer is allowed to be inserted in a space. Larger spaces can be easily expressed as a union of small spaces. Each buffer size is same as that of the source driver.

During the optimization, the positions of buffer spaces are fixed while the routing tree keeps changing. The relationship between a buffer space and the routing tree is also changed accordingly and the only allowable buffer insertions correspond to buffer spaces through which the routing tree passes. The philosophy behind this is that when interconnect resources are scarce, it is undesirable to introduce large detours to reach a buffer location. In the example of Figure 1, there are four fixed buffer spaces and the routing tree is changed from (a) to (b). Space 4 was not on any routing path in (a) and it is passed through by a routing path after the update, then it will be referred as a candidate buffer position in (b). On the other hand, space 1 was a candidate buffer position in (a), but is not on any routing path after the update and is therefore no longer a candidate in (b). Space 3 was also a candidate buffer position in (a) and a buffer is inserted there during the optimization.

2.2. The motivation for combining buffer insertion with non-Hanan optimization

As observed in [13], the non-Hanan points can be used to reduce the wire cost while the timing constraints are satisfied. This is illustrated in Figure 2(a). We define a segment to be a contiguous set of straight edges that are either all horizontal or all vertical and without any buffers. Note that this definition is slightly different from that in [7, 13] as it incorporates the presence of buffers. A maximal segment is a segment that is not properly contained in any other segment. In Figure 2, pw is a maximal segment and p is the root of this maximal segment. Consider the connection from sink v to pw . Let x denote the distance from the connection point to p and let CC to express the closest connection [7] from v to pw . The work in [7] proved that under the Elmore model the *delay* of any sink in the routing tree is a concave function with respect to x . A straightforward extension to this conclusion is that the *delay violation* at any sink is also a concave function of x . Figure 4 shows an example of such a function curve. Though the Elmore delay may have large errors for specific points, its qualitative fidelity is still true and serves as good strategic guide. Our experimental results also support this assertion.

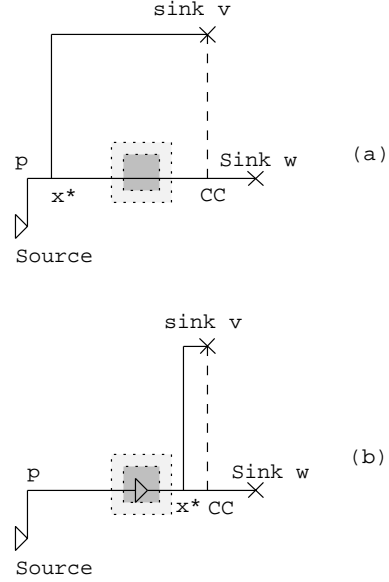


Figure 2. An example that buffer insertion can reduce wire cost further in non-Hanan optimization.

The objective of minimizing wire cost subject to timing constraints can be translated to a set of local optimizations that search for a connection point as close to CC as possible while keeping the maximum delay violation to be non-positive. Sometimes, this optimal connection is a non-Hanan point, as in the case of x^* in Figure 2(a) or the AWE curve in Figure 4.

In order to reduce wire cost, it is desired to move the connection point as close to CC as possible, i.e., to maximize x . However, the value of x may be capped by the constraint of non-positive delay violation. The utility of buffer insertion is to relax this timing constraint, if possible, so as to achieve further wire cost reduction as illustrated in Figure 2(b).

As in [20], a buffer may be inserted to achieve delay reduction in one of two ways: (1) by providing improved drive strength on a critical path, or (2) isolating paths to non-critical sinks by inserting a buffer at a multifanout point to reduce the load on the critical path.

2.3. The motivation for using fourth order AWE in non-Hanan optimization

As interconnect wires become increasingly thinner and longer, the interconnect resistance may overshadow the driver resistance. Consequently, the net capacitance of sinks and downstream capacitance are shielded to the driver resistance by the interconnect resistance. This effect is called resistive shielding [5]. The Elmore delay does not correctly take the resistive shielding effect into account and tends to overestimate the delay. This error can be remarkably large, especially for the stub situation (i.e., when a sink that is close to the source co-exists with a much longer wire), where the Elmore delay can be several times larger than the actual delay.

Table 1 shows an example of a net with five sinks to illustrate the inaccuracy of the Elmore delay. Its routing topology is illustrated in Figure 3. The load capacitance is the same for each sink. The delays on all sinks are computed using the Elmore formula, fourth order AWE and a SPICE transmission line model, and the percentage errors relative to SPICE are calculated. The Manhattan distance from each sink to the source are also listed for reference. We can see that the error of Elmore delay can be over 300% and the delay from fourth AWE is clearly superior. In fact, as the wire

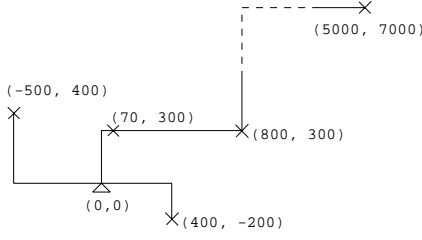


Figure 3. A routing tree on which Elmore delay gives large errors.

size shrinks, this trend will be more and more severe.

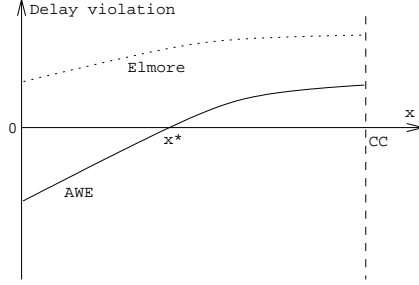


Figure 4. An example that Elmore delay and higher order AWE delay may result in different connection choice.

To see how this will affect the non-Hanan routing, consider the graph in Figure 4. The graph plots the delay violation function against the location of the connection point, x , as pictured in Figure 2. The dotted curve indicates the Elmore delay while the solid curve represents the fourth order AWE result. The solution corresponds to the point closest to CC where the delay violation function is negative or zero. For the Elmore delay, which overestimates the delay near the source, no solution is found, whereas an actual solution exists and corresponds to x^* .

On the other hand, we have observed that the Elmore model tends to under-estimate delay at sinks far from the source¹. This may lead to the opposite error, as can be seen in the last row of Table 1. This under-estimation may result in over-reduction of cost while the timing constraints have not been satisfied yet. On the whole, a higher order model is greatly superior to the Elmore model in handling non-Hanan points.

The reason that we choose fourth order instead of a second or third order model is that second order gives less accuracy and for many examples that we tried, we found that the third order model induces positive poles more often.

In the computation of fourth order AWE delay, we first use the RICE algorithm [4] to obtain the moments. We solve the denom-

¹ The Elmore delay is theoretically proven to be an upper bound on the delay of an RC network in [6]. However, in practice, greater accuracies are obtainable by multiplying the Elmore delay formula of [2] by a factor of $\ln 2$, and we refer this quantity as the “Elmore delay” in our discussion, and this may be either optimistic or pessimistic.

Table 1. Elmore vs. AWE

Dist.	SPICE	Elmore	Error	4th AWE	Error
370	13.6	52.5	286%	12.8	-6%
600	9.5	39.8	319%	8.9	-6%
900	10.7	40.5	279%	10.5	-2%
1100	26.2	77.4	195%	25.5	-3%
12000	283.2	257.5	-9%	282.4	-0.3%

inator of Padé approximation result, which is a fourth order polynomial, using a closed form formula to obtain the poles. After an inverse Laplace transformation, the time domain exponential functions are expanded about the Elmore delay to fourth order Taylor series polynomials. A closed form solution to a fourth order polynomial exist and may be used to calculate the delay value. The additional computation cost of fourth order AWE as compared to a second order model is minor. This process is iterated until convergence, and we found that we always converged within 3 iterations. This method is related to the Newton-Raphson root-finding method: the Newton-Raphson method uses a first order Taylor series in each iteration, and our method uses a fourth order expansion instead.

2.4. Problem formulation

A list of notational terms used in this work is as follows:

- Q_i : required arrival time for sink i .
- T_{di} : the calculated delay for sink i in the routing tree.
- T_{vi} : delay violation of sink i , given by $T_{vi} = T_{di} - Q_i$.
- W : total wire length for a routing tree.
- C_l : load capacitance for a sink or a buffer.
- A_j : an available buffer space that can be a candidate buffer position, j is the index.
- α : weighting factor for buffer cost.
- c : capacitance per unit length for interconnect.
- n : number of sinks.
- m : number of initial available buffer spaces.
- k : number of buffers inserted in the routing tree.

We state the problem formulation as follows:

Given a source s_0 , a set of sinks $S = \{s_1, s_2 \dots s_n\}$, timing specifications $Q = \{Q_1, Q_2, \dots, Q_n\}$ for all sinks and a set of available buffer spaces $A = \{A_1, A_2, \dots, A_m\}$, construct a Steiner routing tree and choose a subset from A on which buffers are inserted such that the following problem is solved:

$$\begin{aligned} & \text{minimize} && (1 - \alpha)cW + \alpha C_l k \\ & \text{subject to:} && \max_{i \in S} (T_{vi}) \leq 0 \\ & && 0 \leq \alpha \leq 1 \end{aligned} \quad (1)$$

Here the weighting factor for the wire cost is $1 - \alpha$. The purpose of including c and C_l in the objective function is to normalize the wire and the buffer cost into comparable quantities.

3. ALGORITHM

The algorithm consists of two phases. Phase I, called SART (Steiner AWE Routing Tree), is similar to SERT except that the Elmore model is replaced by fourth order AWE. The output is a routing tree T' . The Phase II is the simultaneous buffer insertion and non-Hanan optimization.

In SART, starting with a single source, a partial routing tree grows in a greedy fashion. In each growing step, a previously unconnected sink is selected and connected to a certain node in the partial tree such that the maximum delay is minimized.

Algorithm: Buffer_candidate_update.

Input: A routing tree \tilde{T} .

B' = previous candidate buffer positions.

A' = all available spaces not in B' .

Output: B = candidate positions for updated \tilde{T} .

A = all available spaces not in B .

1. for each position B_i in B'
2. if B_i is not on any path of \tilde{T}

$$B' = B' - B_i;$$

$$A' = A' \cup B_i;$$
3. for each space A_i in A'
4. if any path in \tilde{T} passes through A_i

$$A' = A' - A_i;$$

$$B' = B' \cup A_i;$$
5. $A = A'$;
6. $B = B'$;

After each move in Phase II, each initial buffer space may result with one of the three possibilities: (i) a buffer is inserted in it, (ii) it is still on a path in the routing tree as a candidate buffer position, (iii) it is not on any path of the routing tree. Corresponding to these three results, we maintain three sets: set I for buffer nodes, set B for candidate buffer positions and set A for all the off-path spaces. Once it is decided that a buffer will be inserted in a space, this space becomes a buffer node and is assigned into set I . The function *Buffer_candidate_update* maintains the sets A and B to be updated accordingly.

Algorithm: BINO

Input: T' from SART,

A = available buffer spaces.

Output: BINO tree T ,

I = positions at which buffers inserted.

1. $B = \emptyset$;
2. $I = \emptyset$;
3. *Buffer_candidate_update*(T' , B , A);
4. while $B \neq \emptyset$ and there is cost improvement
5. for each candidate B_i in B
6. insert a buffer at B_i tentatively;
7. *Non-Hanan_optimization*(\tilde{T}):
8. for each sink s_j in descending order of dist to s_0
9. disconnect s_j and its downstream subtree T_{si} ;
10. for each max segment E_i in $T \setminus T_{si}$
11. find optimal connection between s_j and E_i ;
12. connect s_j to optimal location in $T \setminus T_{si}$;
13. if B_i is the buffer that causes the largest improvement
$$B = B - B_i;$$

$$I = I \cup B_i;$$
14. *Buffer_candidate_update*(\tilde{T} , B , A);

The above is the algorithm of Phase II. The input routing tree T' is the default global optimal solution. During Phase II, each candidate position is inserted a buffer tentatively and then the non-Hanan optimization is conducted. The non-Hanan optimization algorithm (MVART) here is similar to MVERT [13], but fourth order AWE is used for delay evaluation instead of the Elmore delay metric.

Sometimes a buffer space may cover a multifanout node of the routing tree, as shown in Figure 5, rather than a single wire segment. The choice of the branch that the buffer will be inserted in is according to the criticalities of sinks on each branch. This approach is similar to the work in [18, 20]. The sink with higher delay violation value has a higher criticality. If the sink criticalities for these branches are close to each other, the candidate buffer will be inserted to drive all of these branches simultaneously like the insertion in Figure 5(a). Otherwise, the buffer will be inserted in the branch of non-critical sinks so that the load from the non-critical sink and path are isolated to the path to the critical sink, which is illustrated in Figure 5(b).

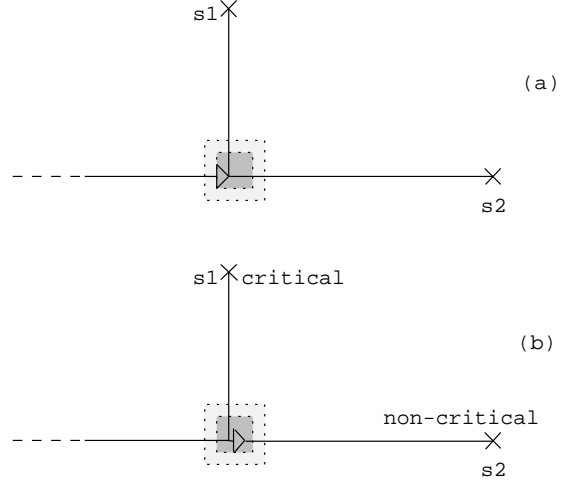


Figure 5. Buffer insertion at an intersection of multiple paths according to the sink criticalities.

At each non-Hanan optimization, in the descending order of distance to source, each sink is reconnected to the routing tree to meet the objective. In reconnection for any sink s_i , this sink and its downstream subtree T_{si} is disconnected first. It is then connected to each maximal segment on the routing tree $T \setminus T_{si}$ tentatively to find the optimal solution. The best connection on a maximal segment is obtained by binary search.

There are three abstract layers of the optimal solution search procedure. The top layer is related to buffer insertion, which will give the so-called global optimum solution. The middle layer refers to the non-Hanan optimal solution search under a certain tentative buffer insertion configuration. The bottom layer is conducted for reconnection of a specific sink to a maximal segment through the binary search. The best of the optimal solutions in bottom layer is chosen as the optimal for the middle layer. In the same fashion, the global optimum is selected from the solutions in the middle layer.

At the beginning of each reconnection, the initial connection configuration is stored as default middle layer optimal solution. During the process of search, each bottom layer optimal obtained will be compared with the current middle layer solution. If it is better according to the objective of (1), it will be saved and the current middle layer optimal solution will be updated accordingly.

After all the candidate positions are tested, the solution that can make largest cost improvement subject to timing constraints is chosen as the final decision. Then all the three sets A , B and I are updated. This process is repeated iteratively till there is no cost improvement or no candidate position left.

4. COMPUTATION COMPLEXITY

From the estimation in [14], the computation cost for MVERT is $O(n^4 + n^4 \cdot \frac{L}{\epsilon})$. The first term comes from the Phase I in MVERT, which is a variation of SERT. The parameter L is the maximum length over all maximal segments and ϵ represents the resolution for the binary search in the Phase II of MVERT.

Although we use the fourth order AWE instead of Elmore in BINO, as the number of traversals or iterations is fixed, the complexity for each delay calculation is still $O(n)$. Thus the cost for Phase I (SART) in BINO is $O(n^4)$. In Phase II of BINO, there are two layers of iterations outside of each MVART, each of which is upper-bounded by the number of buffer spaces. The combination of the total cost is $O(m^2 \cdot n^4 \cdot \frac{L}{\epsilon})$. Practically, the multiplier is much less than m^2 , since the number of candidate buffer spaces

that the net passes through is often much smaller than the total number of available spaces.

5. EXPERIMENTAL RESULTS

The experimental results are shown in Table 2 and Table 3 for IC and MCM technology, respectively. The parameters for MCM are from [7] and the IC parameters are for $0.18\ \mu m$ technology which are also scaled from [7]. The sink and buffer space locations for each test are generated randomly, and the number of sinks for each net varies from 4 to 12. Since we consider the situation that the interconnect resource is more stringent than that of buffer, the weighting factor for wire cost is chosen to be 0.8 and the weighting factor for buffer cost α is 0.2. The area of each critical zone is chosen to be $200\ \mu m \times 200\ \mu m$ for IC and $400\ \mu m \times 400\ \mu m$ for MCM. There are approximately 50 buffer spaces for each test, thus the total area of the critical zones accounts for about 2% of the area of a routing region. According to our experiments, the variations of delay from the change of a buffer position within a critical zone is small and can be neglected.

The δW in Table 2 and 3 is the percentage wire cost reduction with respect to the SART tree. The last column corresponds to the number of input buffer spaces, and the next-to-last column shows the number of buffers finally inserted. The results from SART and MVART are also listed for comparison. Since the timing constraints are quite stringent, the maximum delay violations, V_{vMAX} , from most of SART results are positive. Sometimes even pure non-Hanan optimization cannot satisfy the timing specification. This hinders the ability of pure non-Hanan optimization to reduce the cost further and the BINO becomes a necessary step.

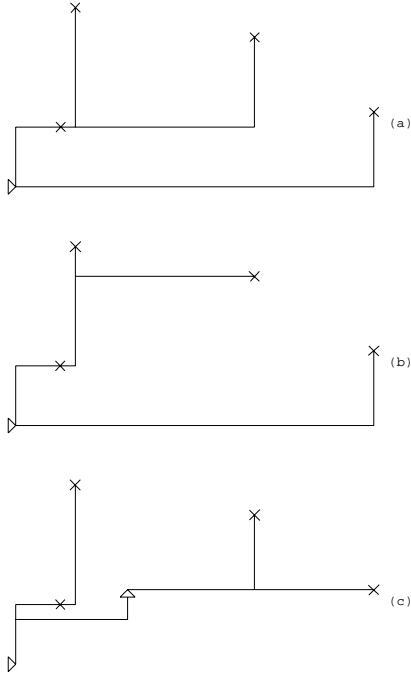


Figure 6. Routing tree results from (a) SART, (b) MVART, (c) BINO.

An actual set of results for a four sinks net is depicted in Figure 6 according to real scale. This clearly shows the expected cost improvement. The cost reduction from only non-Hanan optimization is 11%, while the reduction from BINO is 34%. In this example, the timing constraint is stringent. As a result, few shortest distance connections are made, and no non-Hanan points can be found to reduce the wire cost in the pure non-Hanan optimization.

The BINO can relax the constraints and take the advantage of non-Hanan point to reduce more cost.

From Table 2 and 3, we can see that BINO can reduce significantly more cost than pure non-Hanan optimization under these somewhat harsh conditions. The average wire cost improvement is 31% for $.18\ \mu m$ IC technology. For MCM, BINO provides an average wire cost improvement of 33%. The BINO algorithm can also satisfy the timing constraints that is impossible for SART and MVART.

In our experiments, the time cost of the computation is usually within one minute for nets of up to 12 sinks. In the worst case the run time can be a couple of minutes. On the whole, the computational cost of our algorithm is reasonable, since these optimizations are carried out only for global timing-critical nets.

Table 2. Experimental results on $.18\ \mu m$ IC

n	SART	MVART		BINO			
	T_{vMAX}	δW	T_{vMAX}	δW	T_{vMAX}	k	m
4	0.65	18%	0.24	41%	-0.24	2	40
4	0.45	11%	0.29	50%	-0.15	1	45
4	1.15	4%	0.54	25%	-0.36	1	35
8	3.03	11%	1.98	16%	-0.09	3	45
8	0.23	1%	0.09	45%	-0.37	2	55
8	3.04	7%	2.00	23%	-0.57	3	40
12	1.80	12%	1.06	27%	0	2	50
12	1.50	8%	-0.01	12%	0	3	45
12	0.48	1%	0.38	40%	-0.06	2	50

Table 3. Experimental results on MCM

n	SART	MVART		BINO			
	T_{vMAX}	δW	T_{vMAX}	δW	T_{vMAX}	k	m
4	-0.01	19%	-0.26	36%	-0.08	1	30
4	-0.79	10%	0	15%	-0.39	1	40
4	0.40	22%	0.06	29%	0	1	40
8	4.03	13%	0.33	29%	-0.07	1	60
8	1.57	20%	0.11	35%	-0.04	2	50
8	1.85	20%	1.01	23%	-0.20	3	70
12	-0.31	40%	-2.59	48%	-0.07	3	40
12	1.65	8%	1.10	21%	0	4	50
12	4.31	14%	0.72	59%	-0.06	3	50

6. CONCLUSION

In this paper, we present a post-placement simultaneous buffer insertion and non-Hanan optimization algorithm to improve the VLSI interconnect performance. This algorithm is especially effective when both the timing constraints and wire resources are stringent. Experiments showed it can reduce wire cost significantly for both $.18\ \mu m$ IC and MCM technology. The fourth order AWE model is applied to assure the quality of the results.

REFERENCES

- [1] W. C. Elmore, "The transient response of damped linear network with particular regard to wideband amplifiers," *Journal of Applied Physics*, Vol. 19, pp. 55-63, 1948.
- [2] J. Rubinstein, P. Penfield and M. A. Horowitz, "Signal delay in RC tree networks," *IEEE Transactions on Computer-Aided Design*, Vol. CAD-2, No. 3, pp. 202-211, July 1983.

- [3] L. T. Pillage and R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 9, No. 4, pp. 352-366, Apr. 1990.
- [4] C. L. Ratzlaff, N. Gopal and L. T. Pillage, "RICE: rapid interconnect circuit evaluator," *Proc. 28th ACM/IEEE Design Automation Conference*, pp. 555-560, 1991.
- [5] J. Qian, S. Pullela and L. T. Pillage, "Modeling the effective capacitance for the RC interconnect of CMOS gates," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 13, No. 12, pp. 1526-35, Dec. 1994.
- [6] R. Gupta, B. Krauter, B. Tutuianu, J. Willis and L. T. Pillage, "The Elmore delay as a bound for RC trees with generalized input signals," *Proc. 33rd ACM/IEEE Design Automation Conference*, 1995.
- [7] K. D. Boese, A. B. Kahng, B. A. McCoy and G. Robins, "Near-optimal critical sink routing tree constructions," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 14, No. 12, pp. 1417-36, Dec. 1995.
- [8] J. Cong and C. K. Koh, "Interconnect layout optimization under higher-order RLC model," *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 713-720, 1997.
- [9] J. Lillis, C. K. Cheng, T. T. Lin and C. Y. Ho, "New performance driven routing techniques with explicit area/delay tradeoff and simultaneous wire sizing," *Proceedings of the 33rd ACM/IEEE Design Automation Conference*, pp. 395-400, Jun. 1996.
- [10] F. J. Liu, J. Lillis and C. K. Cheng, "Design and implementation of a global router based on a new layout-driven timing model with three poles," *Proceedings of the IEEE International Symposium on Circuits and Systems*, 1997.
- [11] J. Lillis and P. Buch, "Table-lookup methods for improved performance-driven routing," *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 368-373, 1998.
- [12] S. S. Sapatnekar, "RC interconnect optimization under the Elmore delay model," *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 392-396, 1994.
- [13] H. Hou and S. S. Sapatnekar, "Routing tree topology construction to meet interconnect timing constraints," *Proceedings of the International Symposium on Physical Design*, pp. 205-210, 1998.
- [14] H. Hou, J. Hu and S. S. Sapatnekar, "NonHanan routing", to be published on *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*.
- [15] L. P. V. Ginneken, "Buffer placement in distributed RC-tree networks for minimal Elmore delay", *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 865-868, 1990.
- [16] J. Lillis, C. K. Cheng and T. Y. Lin, "Simultaneous routing and buffer insertion for high performance interconnect", *Proceedings of the Sixth Great Lakes Symposium on VLSI*, pp. 148-153, 1996.
- [17] J. C. Shah and S. S. Sapatnekar, "Wiresizing with buffer placement and sizing for power-delay tradeoffs", *Proceedings of VLSI Design - 96*, pp. 346-351, 1996.
- [18] T. Okamoto and J. Cong, "Buffered Steiner tree construction with wire sizing for interconnect layout optimization", *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 44-49, 1996.
- [19] A. Salek, J. Lou and M. Pedram, "A simultaneous routing tree construction and fanout optimization algorithm", *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 625-630, 1998.
- [20] Y. Jiang, S. S. Sapatnekar, C. Bamji and J. Kim, "Combined transistor sizing with buffer insertion for timing optimization", *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 605-608, 1998.