Sādhanā Vol. 35, Part 4, August 2010, pp. 407–418. © Indian Academy of Sciences

An efficient reduction algorithm for computation of interconnect delay variability for statistical timing analysis in clock tree planning

SIVAKUMAR BONDADA[a,*], SOUMYENDU RAHA[b] and SANTANU MAHAPATRA[a]

[a]Nano Scale Device Research Laboratory, Centre for Electronics Design and Technology, Indian Institute of Science, Bangalore 560 012
[b]Supercomputer Education and Research Centre, Indian Institute of Science, Bangalore 560 012
e-mail: bsiva@cedt.iisc.ernet.in; raha@serc.iisc.ernet.in; santanu@cedt.iisc.ernet.in

MS received 27 February 2009; revised 9 February 2010; accepted 17 February 2010

Abstract. In this paper, we propose a novel and efficient algorithm for modelling sub-65 nm clock interconnect-networks in the presence of process variation. We develop a method for delay analysis of interconnects considering the impact of Gaussian metal process variations. The resistance and capacitance of a distributed RC line are expressed as correlated Gaussian random variables which are then used to compute the standard deviation of delay Probability Distribution Function (PDF) at all nodes in the interconnect network. Main objective is to find delay PDF at a cheaper cost. Convergence of this approach is in probability distribution but not in mean of delay. We validate our approach against SPICE based Monte Carlo simulations while the current method entails significantly lower computational cost.

Keywords. Statistical timing analysis; VLSI clock interconnects; delay variability; PDF; process variation; Gaussian random variation; computational cost.

1. Introduction

Interconnects constitute a dominant source of circuit delay for modern chip designs. As the CMOS technology scales further, chip interconnects with lower width-to-height aspect ratio are dominating the physical design. The variations of the critical dimensions in modern VLSI technologies lead to variability in interconnect performance that must be fully accounted in timing verification in sub-65 nm technology nodes. Variations in the line width affect the resistance and the inter-layer capacitance. Variations in the inter wire spacing may cause

[*]For correspondence

a significant degradation in the signal integrity. Layout pattern dependent variations within the inter-layer oxide and the chip multiprocessing process also have a significant impact on the interconnect parasitics. These disparate sources of variations in the VLSI CMOS fabrication process lead to both random and systematic effects on circuit performance (Nassif 2001; Boning & Nassif). All of these make it increasingly difficult to accurately predict the performance of a circuit at the design stage, which ultimately translates to a parametric yield loss (Vishweswariah 2003). Thus, determining an accurate statistical description of the interconnect response is critical for designers.

Traditional static timing methodology is corner based. This requires an exponential number of timing runs as the number of correlated and significant sources of variation increases. Further timing verification can become pessimistic as is the nature of corner based methods that seek to deliver bounds on the arrival times. But it is intractable to analyse all possible corners, the missing corners may lead to failures detected after the manufacturing of the chip. The solution of this problem is statistical static timing analysis. It will reduce pessimism, improve verification time and provide means for increasing parametric yield.

In the presence of significant variations, interconnect model parameters such as wire resistance, capacitance, etc., need to be modelled as random variables or as spatial random processes. The conventional corner-based analysis is inadequate, and simulations based on sampling (Monte Carlo) require long computation times due to the large number of parameters and due to the need for generating large number of random variates corresponding to each parameter.

Therefore, it is required that an interconnect analysis framework be developed that considers random variations in physical dimensions and estimates the probability distribution function (PDF) of the interconnect delay. However, handling inter-die/intra-die variations and assessing their impacts on circuit performance can dramatically increase the cost of the statistical timing analysis. This underscores the need for fast and efficient methods that reduce the computational cost but with minimum error in computing the statistical description of the response (Agarwal *et al* 2003).

In this paper, a practical interconnect delay variation analysis technique is developed to facilitate the efficient computation of PDF of clock interconnect delay for Statistical Timing Analysis. The interconnect delay is a function of the process variables, which are typically modelled as Gaussian random variables. The ITRS (International Technology Road map for Semiconductors) (Interconnect 2006) suggests that total Metal-1 resistance variability is 28% in 65 nm node and capacitance variability is 20%. We assume that the resistance and capacitance of distributed RC lines to be correlated Gaussian random variables having the variability as given in ITRS. Based on this, the new algorithm proposed here reduces the overall network to small manageable RC distributed lines, which, in turn needs smaller computational effort. The new reduction algorithm proposed here provides significant computational speed improvement of up to four orders of magnitude over the Monte Carlo SPICE simulations.

2. Algorithm

The reduction algorithm takes advantage of quickly solving in a straight forward manner a small segment of interconnect for its delay variability. The algorithm computes reasonably accurate delay variability at any node in much less time compared to solving the entire interconnect network repeatedly for delay variability in a Monte Carlo scheme. The algorithm applies to all short and medium length interconnects.

Figure 1a shows a typical interconnect network containing four internal nodes and five output nodes terminated with different capacitive loads. Here each interconnect segment is considered as a distributed RC line. Depending on the way process variability affects the variability of resistance and capacitance of the interconnect, we have to choose the granularity (i.e. the number of simplified segments) in the distributed RC line. The granularity of the RC line network must be determined empirically from process variability parameters so that the computation can capture the delay variability adequately. For example, based on ITRS data for 65 nm, we choose 10 μm length line segments as single wire segments i.e. every 10 μm segment is treated as a single RC element where R is a Gaussian random variable representing bulk resistance of the segment and C is another Gaussian random variable representing the capacitance of that segment. The resistance random variable and corresponding capacitance random variable are taken to have a correlation coefficient of 0·8 according to (Usha Narasimha *et al* 2006). Normally we can solve this kind of network in Monte Carlo fashion and get the delay variability at each node, but this is computationally expensive. We reduce this cost by collapsing the segments into equivalent RC loads for delay variability calculation purpose.

2.1 *Reduction procedure*

Initially all the leaf segments (i.e. segments which have no further branches) are identified and modified (collapsed) into equivalent RC loads so that these loads can preserve the delay properties of the network. Figure 2 shows different types of loads namely L-type load, 3-stage π-load and 100-stage π-load. If there are more than one leaf segments at a node, they are merged. This merging is not just simple calculation of equivalent resistances and capacitances for a set of parallel resistances and capacitances, as it does not take the electrical properties into account. One can assume two segments to be parallel (OR short circuit at their end points) only when the voltage at their end points is the same. We use delay instead of voltage to decide on which segments are in parallel or which two nodes are to be short-circuited. So the equivalent value of resistances and capacitances that are in between the nodes whose delays are the same are calculated. Figure 3 explains this procedure. There are two non-uniform distributed RC lines in the figure each containing N (3 for 3-stage π-load and 100 for 100-stage π-load) segments. We can calculate the delay at each of the N nodes approximately using the simple Elmore delay metric for both the lines. If the two lines are identical, then the resultant line will have half the resistance and double the capacitance of individual lines in each segment. Otherwise we look for the node (say $N1$) in the line having the largest delay, at which the delay is same as the output node delay of the line having the smallest delay. $N1$ and the output node of the line having the smallest delay are shorted as shown in figure 3. In the same manner we short the nodes having the same delay from node $N1$ onwards. Hence as shown in figure 3, the nodes from the two lines are selectively shorted and merged to form a single RC line. However, some portion of the line will remain unmerged. After this step the reduced interconnect network looks like the one in figure 1b.

We continue the collapsing on the newly formed leaf branches. And thus, figure 1b reduces to figure 1c. During this step a distributed line with output load is collapsed into its equivalent load and then merged with another load according to the Elmore delays as explained earlier. This collapsing is done by taking all segment lengths into consideration i.e. the number of RC elements allotted in equivalent load for a segment are in proportional to the segment length. These steps are further executed till the reduced interconnect network contains a single distributed line with its output loaded as shown in figure 1e.

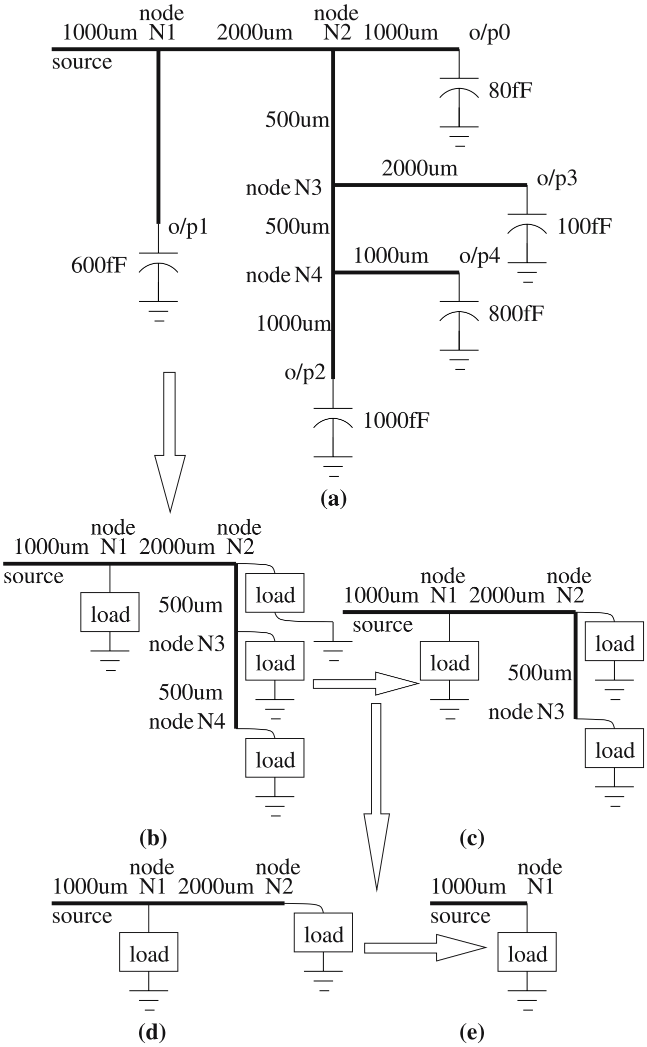

Figure 1. **(a)** Original interconnect network, **(b)** to **(e)** are reduced networks following the reduction algorithm. In figure **(b)** the segment $N2 - o/p0$ is collapsed into an equivalent load. Similarly, the segments $N3 - o/p3$ and $N1 - o/p1$ are collapsed into their equivalent loads. At node 4, two leaf segments $N4{-}o/p4$ and $N4{-}o/p2$ are collapsed individually and merged using algorithm proposed. In figure **(c)** the segment $N3$–$N4$ is collapsed and merged with load already existed at node $N3$. In figure **(d)** the segment $N2$–$N3$ is collapsed and merged with load already existed at node $N2$. In figure **(e)** the segment $N1$–$N2$ is collapsed and merged with load already existed at node $N1$.

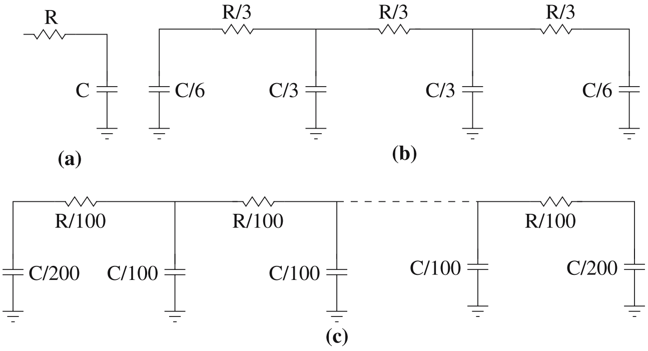

Figure 2. Different types of loads for mocking merged part of network, **(a)** L-type load, **(b)** 3-stage π load and **(c)** 100-stage π load.

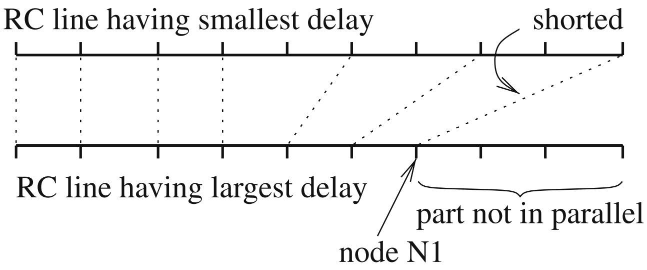

Figure 3. Merging two loads; for convenience 10 node RC lines are taken. Dashed lines indicate shorting the nodes of two RC lines. The part not in parallel remains unchanged in the resultant RC line.

Table 1 gives a quantitative example of how accurately this reduction algorithm using 3-stage π-load preserves the delays at the internal nodes of the interconnect network shown in figure 1a. As shown in the table 1, there is some error in the delay of the nodes as predicted by the reduced networks. Node $N1$ has the largest error. However, this is acceptable as we are seeking convergence of the algorithm in variance only for variation analysis purpose. Thus the reduction algorithm has successfully reduced the original interconnect network into a very simple distributed line with equivalent load.

2.2 *Variation models*

The reduction procedure is conducted by considering mean values for resistances and capacitances. In this subsection, we consider the effect of their variability on delays. Two models

Table 1. Comparison of delays at different nodes among different networks.

Delay (ns)	$N1$	$N2$	$N3$	$N4$
Figure (1a)	0·919	5·23	6·01	6·36
Figure (1b)	0·916	5·26	6·03	6·38
Figure (1c)	0·916	5·26	6·03	
Figure (1d)	0·919	5·24		
Figure (1e)	1·4			

Table 2. Comparison of delay variability results (standard deviation of delay) from Monte Carlo simulations and our algorithm.

Node	Monte carlo	Our model	% error
$o/p0$	1·202e − 12	9·304e − 13	−22·62
$o/p1$	4·007e − 13	5·005e − 13	+19·93
$o/p2$	1·399e − 12	1·571e − 12	+12·16
$o/p3$	1·351e − 12	1·063e − 12	−21·34
$o/p4$	1·407e − 12	1·474e − 12	+04·72
$N1$	3·994e − 13	4·151e − 13	+03·93
$N2$	1·161e − 12	8·250e − 13	−28·94
$N3$	1·311e − 12	9·201e − 13	−29·86
$N4$	1·408e − 12	1·372e − 12	−02·61

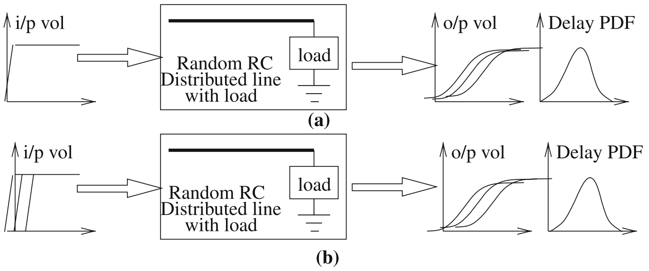
i/p vol
Random RC Distributed line with load
load
o/p vol
Delay PDF
(a)
i/p vol
Random RC Distributed line with load
load
o/p vol
Delay PDF
(b)

Figure 4. **(a)** Model-1, **(b)** model-2; System here is RC distributed line containing the variability. But load is just an equivalent of rest of the deterministic network that has been collapsed and merged.

shown in figure 4 are used for this purpose. Model-1 gives the delay variability of a uniformly distributed RC line with load. Here the distributed RC line's resistance and capacitance are correlated Gaussian random variables as said earlier. In Model-1 the load represents all the branches of the network collapsed with mean values of resistances and capacitances. As shown in figure 4a, for a deterministic voltage step input, the distributed RC random line's output voltage across the load rises with a delay that has a Gaussian distribution. Delay and voltage are positively correlated with respect to their voltage slope. This is in agreement with (Xiaoji Ye 2007). The Model-1 gives the standard deviation of the delay's Gaussian distribution.

Model-2 is same as the Model-1 except that in this case the input also has got variability. Input voltage step rises at different times with a certain Gaussian variability. Model-2 also gives the standard deviation of the delay of the output voltage for similar system as in Model-1.

There are different ways to implement Model-1. It can be implemented as pre-calculated model for L-type load and 3-stage π-load. This model is a function of deterministic resistance in the load, R; deterministic capacitance in the load, C; and the random resistance per unit length and the random capacitance per unit length of distributed RC line and the length of the distributed RC line. Alternatively, it can be implemented at run time for a 100-stage π-load, i.e. simulating the distributed RC random line with load for calculating the standard deviation of its delay. For Model-2 we consider input step voltage delay variability and internal delay variability (estimated by Model-1) as separate noise terms introduced in an ideal system. Since the variability at a node depends on both its predecessor and successor nodes, the internal delay random variable and input delay random variable in figure 4b are correlated. Moreover, resistance or capacitance changes caused by process variations are correlated. Assuming the correlation coefficient between these two terms as ρ, we can predict the total delay variability at the output of distributed RC random line with load in figure 4b as follows:

$$\mathrm{Delay}_{\mathrm{output}} = \mathrm{Delay}_{\mathrm{internal}} + \mathrm{Delay}_{\mathrm{input}}, \tag{1}$$

$$\sigma_{\mathrm{delay-output}} = \sqrt{\sigma^2_{\mathrm{delay-internal}} + \sigma^2_{\mathrm{delay-input}} + 2\rho\sigma_{\mathrm{delay-internal}}\sigma_{\mathrm{delay-input}}}. \tag{2}$$

2.3 *Variation analysis*

We start the variation analysis by introducing RC variability in the fully reduced network in figure 1e. The distributed RC line is a line with random variable parameters but the load (as is assumed in the algorithm) is fixed (deterministic) at the mean values of its parameters. Using Model-1 we obtain the delay variability at node $N1$ as $\sigma_{\mathrm{delay-node}\ N1}$. This delay has to be scaled, because the mean delay of node $N1$ in figure 1e is not equal to the mean delay of node $N1$ in figure 1a as given in table 1. In fact, there is a difference in the slope of rising voltage at node $N1$ at half voltage point ($V = 0{\cdot}5$ v) between two networks. Voltage at node $N1$ in reduced network figure 1e rises slower than the voltage at node $N1$ in original network figure 1a. This is due to the sudden increase in capacitance presented by the equivalent load used at node $N1$ in reduced network. We know σ_{delay} is inversely proportional to slope of rising voltage at half voltage point (Xiaoji Ye *et al* 2007). And the slope is inversely proportional to the capacitance that is presented by the equivalent load at that node. This is the cause of the increase in delay. So we scale $\sigma_{\text{delay}-\text{node } N1}$ of the reduced network with respect to the capacitance ratio (or delay ratio) and obtain an approximate of $\sigma_{\text{delay}-\text{node } N1}$ of the original network. This calculated standard deviation of node $N1$ can further be used to estimate delay variation at successor nodes. After expanding (i.e. opposite to reduction, going backwards) the load, we come to figure 1d, in which the $\sigma_{\text{delay}-\text{node } N1}$ is already calculated. Model-1 is run on the segment from node $N1$ to node $N2$ with load, by incorporating R, C variability in the distributed line between node $N1$ and node $N2$. Here too, load is deterministic and is fixed at the mean value of its parameters. This gives $\sigma_{\text{delay}-\text{node } N2-\text{internal}}$ which should be scaled as explained previously. Now $\sigma_{\text{delay}-\text{node } N2}$ is obtained from $\sigma_{\text{delay}-\text{node } N1}$ and $\sigma_{\text{delay}-\text{node } N2-\text{internal}}$ using the equation 2. This procedure of expanding, using Model-1 to find the internal variation and using Model-2 to find total variation is continued till the original interconnect network and the standard deviation of the delay at all nodes is found.

3. Results

We take the original interconnect network shown in figure 1a having $1{\cdot}5\,\Omega/\mu$m and $0{\cdot}2\, fF/\mu$m, which is usual for medium interconnects, and conduct the Monte Carlo analysis with 1000 iterations. Delay standard deviation is calculated at every node. The same network is modified using our reduction algorithm and variation analysis is done. These delay variation results are compared with Monte Carlo analysis results in table 2. Results presented here are obtained using 3-stage π-load and with ρ of 0·5. The accuracy of delay standard deviation can be increased by using 100-stage π-load. Because, in the case of 3-stage π-load the load can not accurately preserve the distributive nature of RC network which is being merged. A 3-stage π-load presents sudden increase in capacitance and resistance at nodes, which causes delay to increase. But a 100-stage π-load provides a change in capacitance and resistance closer to the one that is presented in the original network, even after merging

Table 3. Results with input delay variation of 10^{-13}, caused by V_{DD} variation.

Node	Monte carlo	Our model	% error
$o/p0$	1·33e − 12	1·00e − 12	−24·56
$o/p1$	5·42e − 13	6·40e − 13	+18·12
$o/p2$	1·53e − 12	1·73e − 12	+13·28
$o/p3$	1·49e − 12	1·19e − 12	−19·89
$o/p4$	1·48e − 12	1·55e − 12	+04·67
$N1$	1·35e − 12	4·84e − 13	+03·20
$N2$	1·35e − 12	9·82e − 13	−27·23
$N3$	1·451e − 12	1·07e − 12	−25·76
$N4$	1·55e − 12	1·51e − 12	−02·61

the loads. Number of stages in the load can be chosen according to our desired accuracy and network complexity. In general, the accuracy of the algorithm increases with the number of stages in the load.

3.1 V_{DD} *Variation*

An extension of our algorithm incorporates the driving voltage V_{DD} variation. With this the source point of each wire will also have variability in input delay, which is caused by V_{DD} variation of the driving CMOS-gate. As our model can calculate the output delay variability of an RC segment, as a function of input delay variability and RC parasitic variability, we can reuse our model to include the V_{DD} variation. Here we should note that the change in the voltage level of the driving cell's output signal due to the V_{DD} variation does not affect the way our algorithm works, because the variability at the output node of the RC segment is not a function of the absolute V_{DD} voltage, rather it is a function of the input delay variation and the RC parasitic variation. the time constant of the RC chain is independent of the voltage of its driver. The driving CMOS gate output delay variation caused by its V_{DD} rail variation comes under device modelling which is outside the scope of the present study. The results with input delay variation of 10^{-13}, caused by V_{DD} variation is shown in table 3.

4. Discussion

4.1 *Model-1*

The reliability of Model-1 i.e. representing the network with equivalent load accurately depends on two factors. Accuracy of equivalent loads to represent the original network increases with number of segments used in that equivalent load. Secondly, this accuracy inevitably depends on the complexity of the interconnect network. Equivalent loads show excellent tracking of original network for simple and well symmetric interconnect networks like on-chip clock tree networks, etc. And their ability to mock original network degrades as the asymmetric nature of the interconnect network grows.

Figure 5a shows a common H-tree used for skew free clock distribution which is simple and symmetric in nature. This network is fully reduced using our algorithm and accuracy of different loads are given in figure 6 and table 4. Though our aim in this paper is to estimate delay variability, it is good enough if the original network and fully reduced networks show same voltage response. We have chosen slope at 50% voltage point along with nominal 50% voltage delay for comparison, because delay variability depends on slope of the voltage waveform at 50% voltage point (Xiaoji Ye *et al* 2007). Comparison of voltage at node $N1$ is the worst-case among all nodes. The results demonstrate an excellent match of voltage response between H-tree interconnect network figure 5a and its fully reduced network with 100-stage π-load though 3-stage π-load and even 6-stage π-load fail to track the original interconnect network's voltage response accurately. This can be easily understood because merging all the nodes having same voltage response in H-tree figure 5a looks like a fully reduced network with ideal load figure 5b. This ideal load is a cascade of two different *uniformly distributed* RC lines. A N-stage π-load is nothing but discretization of this ideal load where the discretization causes the only possible error. In simple and symmetric interconnect networks like H-tree, etc., this ideal load is just a cascade of uniformly distributed RC lines, thus allowing large-stage loads to track original interconnect network closely.

In case of complex or asymmetric interconnect networks like the original interconnect network shown in figure 1a, there are other sources of errors along with discretization. Using

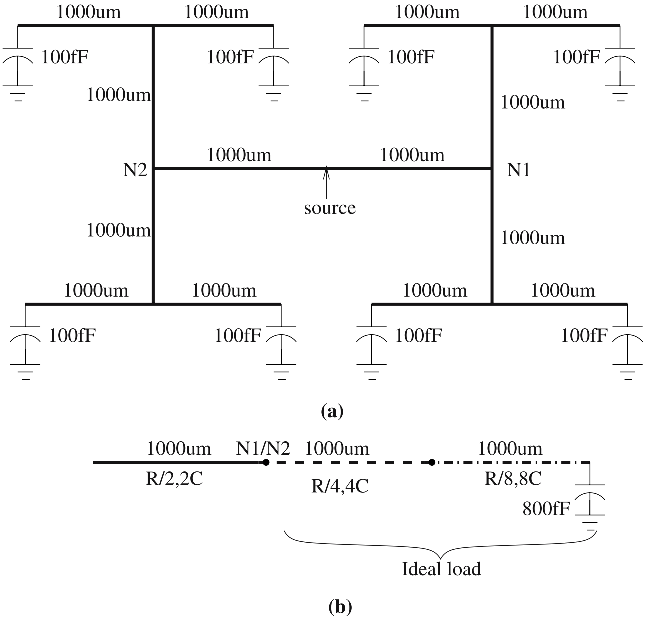

Figure 5. **(a)** A typical symmetric H-tree type interconnect network, **(b)** Fully reduced network of **(a)** with ideal load. Note: R,C are per unit length values of distributed RC lines of H-tree.

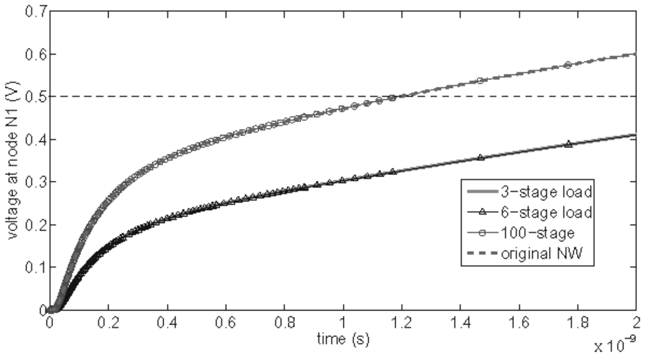

Figure 6. Comparison of different types of loads for their accuracy to preserve the delay at node $N1$ in a symmetric interconnect network like figure 5a.

Table 4. Comparison of 50% delay and slope at 50% voltage point of voltage waveform at node $N1$ in original symmetric H-tree network figure 5a to its fully reduced network with different equivalent loads.

Network	50% delay	Slope at 50%
H-tree figure 5a	1·18805e − 10 s	134·53 MV/s
Fully reduced NW (100-stage π-load)	1·18822e − 9 s	134·6 MV/s
Fully reduced NW (6-stage π-load)	2·98e − 9 s	81·8 MV/s
Fully reduced NW (3-stage π-load)	2·97523e − 9 s	81·6 MV/s
Fully reduced NW (L-type load)	2·6068e − 10 s	75·9 MV/s

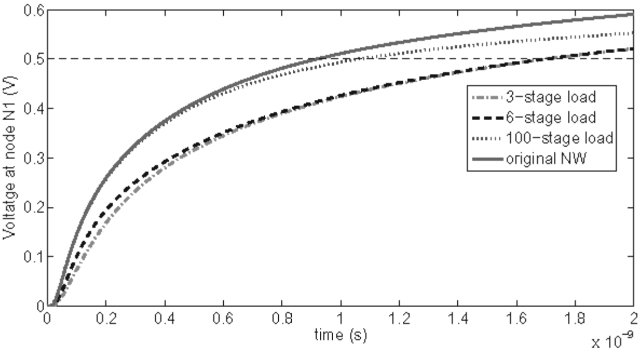

Figure 7. Comparison of different types of loads for their accuracy to preserve the delay at node $N1$ in an asymmetric interconnect network like figure 1a.

Elmore delay for identifying nodes having equal voltage response causes some error. It is not guaranteed that nodes which are being merged have absolutely same delay. In practice, this Elmore delay turns out to be crude approximation, as voltage need not be same throughout the full transition time for the two nodes which are deemed to behave electrically identical. So the error due to this approximation is purely dependent on complexity of interconnect network under consideration. Figure 7 and table 5 show how reduced interconnect network figure 1e having different loads approximate a complex original interconnect network figure 1a. Surely asymmetric or complex networks need more number of stages in the load compared to simple or symmetric networks to improve accuracy. Implementing larger number of stages in the equivalent load does produce accurate results but at the cost of more system memory and computational capacity.

4.2 *Model-2*

After accurately representing the original interconnect network by using equivalent load, it becomes computationally efficient to get the delay variability at all internal loads using Model-2. The Model-2 relies on the consideration that the output delay random variable of a random distributed RC line is summation of the input delay random variable and RC line's internal delay random variable. Here correlation between the input delay random variable and the internal delay random variable should be considered for accurate estimation of statistical properties of output delay random variable. Calculating this correlation coefficient numerically is computationally expensive and it can be estimated from the manufacturing variability data. The optimistic assumption of the correlation coefficient ρ to be 0·5 may cause some

Table 5. Comparison of 50% delay and slope at 50% voltage point of voltage waveform at node $N1$ in original complex network figure 1a to its fully reduced network figure 1e with different equivalent loads.

Network	50% delay	Slope at 50%
Figure 1a	9·1934e − 10 s	142·5 MV/s
Figure 1e with 100-stage π-load	1·06507e − 9 s	94·5 MV/s
Figure 1e with 6-stage π-load	1·2982e − 9 s	77·34 MV/s
Figure 1e with 3-stage π-load	1·40397e − 9 s	78·18 MV/s
Figure 1e with L-type load	2·0445e − 10 s	74·4 MV/s

error in the output delay variability, but it is acceptable in statistical timing optimization considering the cost benefits the algorithm provides.

5. Computational efficiency

The major advantage of our reduction algorithm is its computational efficiency compared to the Monte Carlo technique. In case of the reduction algorithm, the network that is solved is simple and also is smaller in size. So simulating this kind of interconnect is much quicker as outputs reach half voltage point faster and we don't need to simulate further beyond this point. Whereas a single simulation in Monte Carlo simulates the entire interconnect network at once and we need to continue this simulation till the longest delay node voltage reaches half voltage point. This enormously increases the required computing effort considering the fact that procedures such as matrix inversion, etc. are very expensive as the number of elements increase.

Usual computational cost that incurs in a 1000 iteration Monte Carlo simulation is $1000 * O(N_{\text{total}}^{x})$, where N_{total} is the number of RC elements in the entire network. x is around 4 and it is high because it also includes the voltage delay dependency on N_{total} along with spacial matrix computational non linearity. The computational cost for our reduction algorithm is given as $1000 * O(N_{\text{segment}}^{x}) * S + GT + LM * S$, where N_{segment} is number of RC elements in the single segment, S is the total number of segments in entire network, GT is cost for graph traversal which is a function of S, LM is cost for load merging which is a linear function of N_{segment}. Significant computational cost improvement is expected from our reduction algorithm over Monte Carlo technique because both the ratio $N_{\text{total}}/N_{\text{segment}}$ and x are high in practice making $(N_{\text{total}}/N_{\text{segment}})^{x}$ a very large quantity.

On the same computer, a single iteration for Monte Carlo took 1195 sec, where as it took 0·035 sec for calculating the standard deviation with Model-1 (with 3-stage π-load) which is computationally the most expensive model in our reduction algorithm. Together with Model-1 we have to use graph traversal techniques and load merging which are relatively inexpensive. But the cost grows nonlinearly for Monte Carlo simulations with number of elements. Overall, our algorithm gives a speed-up advantage of the order of 10^{4} compared to the usual Monte Carlo technique even with accurate 100-stage π-load.

6. Conclusion

We proposed a novel scheme for estimating the performance of clock interconnects in the presence of process variations. The variations are modelled as random variables of resistance and capacitance. We showed how the stochastic response of the interconnect network can be efficiently computed by our reduction algorithm. It also shows a weak convergence in variance to the actual estimates is sufficient and leads to inexpensive but reasonably accurate algorithm. We applied our algorithm on a typical interconnect network and the comparison of our results using the reduction algorithm against the classical Monte Carlo-based circuit simulations demonstrates a good match. Our algorithm provides a significant speed-up over Monte Carlo simulations. As the algorithm better suites for symmetric interconnects, we can use this for clock tree planning and also for early timing of asymmetric networks where accuracy is not of much concern.

References

Agarwal A, Blaauw D, Zolotov V 2003 Statistical timing analysis for intra-die process variations with spatial corrections. *Proc. Int. Conf. Computer-Aided Design*, San Jose, CA 900–907

Boning D, Nassif S 2001 Models of process variations in device and interconnect. Design of high-performance microprocessor circuits, 1st ed., A Chandrakasan, W J Bowhill, F Cox (eds). IEEE Press

ITRS Interconnect 2006 update

Nassif S R 2001 Modelling and analysis of manufacturing variations. *IEEE Conference on Custom Integrated Circuits* 223–228

Usha Narasimha, Binu Abraham, Nagaraj N S 2006 Statistical analysis of capacitance coupling effects on delay and noise. *Proc. International Symposium on Quality Electronic Design*

Vishweswariah C 2003 Death, taxes and failing chips. Proc. Design Automation Conf., Anaheim, CA 343–347

Xiaoji Ye, Frank (Ying) Liu, Peng 2007 Fast variational interconnect delay and slew computation using quadratic models. *IEEE Transactions on very large scale integration systems* 15(8): 913–926