A Placement Methodology for Robust Clocking*

Ganesh Venkataraman, Jiang Hu
Department of Electrical and Computer Engineering
Texas A&M University
College Station, TX 77843
{ganesh, jianghu}@ece.tamu.edu

ABSTRACT

As the VLSI technology scales towards the nanometer regime, circuit performance is increasingly affected by variations. These variations need to be considered at an early stage in performance optimization. This work proposes a new placement methodology that facilitates low cost and robust clock network. It is based on the observation that bringing tightly constrained flip-flops close to each other can reduce the non-common paths between them in clock network. Such a reduction will in-turn improve the tolerance of the clock network towards variations in delay/skew. Monte Carlo experiments (based on spatial correlations) indicate that our methodology can reduce the maximum skew violation due to variations by up to 62% with less than 2.7% increase in total wire length.

1. INTRODUCTION

When the VLSI technology scales toward nanometer regime, circuit performance is increasingly affected by PVT variations which include Process variations, supply Voltage fluctuations and Temperature changes. Consequently, the PVT variations have to be considered during circuit design and performance optimization. The performance of synchronous circuits is essentially governed by the interaction between logic path signal delay and clock signal delay at flip-flops (or latches). Traditionally, placement is a design stage where logic path delay can be optimized [1] while clock skew is decided during skew optimization [2–4] and clock network layout [4, 5]. In this work, we will show a connection between placement and clock network routing. According to this observation, we propose a placement methodology that facilitates low cost and robust clock network. The basic idea is to move tightly constrained flip-flops close to each other during cell placement so that their non-common paths can be reduced and therefore their tolerance to variations in skew can be improved. To the best of our knowledge, the only previous work with similar objective is [6] which addresses the same issue during floor planning. However, neither clock skew nor clock tree wirelength is reported in [6]. Compared to floor planning, placement stage allows more reliable timing estimation and more detailed control on flip-flop locations. In our methodology, traditional placement objectives, such as signal net wirelength, are considered together with the tolerance of the clock network towards variations in skew. Monte Carlo simulations considering spatial correlations were performed on the resulting circuits. These experiments indicate that our proposed methodology can reduce the maximum skew violation due to variations by up to 62% with less than 2.7% increase in wire length.

*This work was supported in part by Semiconductor Research Corporation under contract number 2004-TJ-1205.

2. REVIEW AND MOTIVATION

Timing performance of synchronous circuits is essentially determined by long path and short path constraints. Consider two flip-flops (i, j) with clock signal arrival times (t_i, t_j). Let D^{ij}_{max} (D^{ij}_{min}) denote the maximum (minimum) combinational logic delay between the output of i and input of j. The following timing constraints have to be satisfied [2]:

$$t_i - t_j \leq T - D^{ij}_{max} - t_{setup} \qquad \text{long path constraint}$$
$$t_i - t_j \geq t_{hold} - D^{ij}_{min} \qquad \text{short path constraint}$$

where T is the clock period. Therefore, circuit performance depends on both logic path delay and correct delivery of clock signals. Let $U_{ij} = T - D^{ij}_{max} - t_{setup}$ and $L_{ij} = t_{hold} - D^{ij}_{min}$(U_{ij} and L_{ij} are referred to as the lower and upper permissible range respectively). Then, the *clock skew* $q_{ij} = t_i - t_j$ has to be within the *permissible range* of $[L_{ij}, U_{ij}]$. The above restriction will be referred to as skew constraint. Traditionally, the input to clock tree routing is the position (or placement) of a set of flip-flops/latches (referred to as sinks) along with their load capacitance and the position of the clock source. Clock routing aims at creating a network connecting all the sinks to the source while maintaining the skew constraints. The end objective of creating a clock network could be to minimize wire length [7], increase tolerance to variations [4, 8] or minimize power consumption [9]. One of the classical works on clock tree construction is the referred to as Deferred Merge Embedding (DME) [7]. DME consists of a bottom-up phase and a top-down phase. The bottom-up phase recursively merges two sub-trees and finds the locus of the merging points such that zero skew is maintained. The top-down phase finds the exact location of the merged nodes. UST/DME [10] attempts to find a routing

20th International Conference on VLSI Design (VLSID'07)
0-7695-2762-0/07 $20.00 © 2007 IEEE

tree subject to skew constraints (not necessarily maintaining zero skew). With the dominance of interconnect delays, clock skew started becoming increasingly sensitive to process variations [11, 12]. Subsequently, some of the later works on clock tree routing focused on increasing the tolerance to process variations [4, 5, 8]. The *safety margin* of the skew is defined as $\min(U_{ij} - q_{ij}, q_{ij} - L_{ij})$. When PVT variations are considered, a large safety margin implies a large probability of satisfying the above constraints. In order to increase the safety margins, optimization can be performed on either logic signal or clock signal. The skew safety margins can be increased by optimizing logic path delay to reduce D^{ij}_{max} and increase D^{ij}_{min}. For clock signals, skew optimization [2–4], a procedure specifying clock signal delay target at every flip-flop, is capable of increasing the safety margins as well. In addition, the probability of satisfying the timing constraints is improved if the variations of logic path delay and clock skew are reduced. All the above mentioned works assume

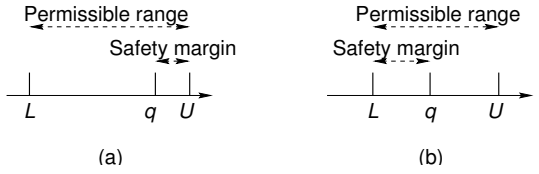

Figure 1: A skew q has a lower bound L and an upper bound U. A skew with larger permissible range as in (a) may have a smaller safety margin than (b).

that the placement information is given. In other words, they assume that the permissible ranges are *fixed*. Conventionally, cell placement is often augmented with timing objective [1] to reduce logic path delay and clock skew variation is usually handled during clock network layout [4]. In this work, we explore a new approach towards improving tolerance to clock skew variations through placement. Consider the routing paths from the clock source to two clock sinks (flip-flops) in a clock tree. The common portion of the two paths does not contribute to the skew between the two sinks [5]. It was suggested in [5] that a pair of clock sinks with tight permissible range are preferred to be merged earlier in clock tree construction so that the non-common portion of their paths is minimized. In this work, we suggest to use safety margin instead of permissible range to characterize the constraint for a skew. The probability of satisfying the timing constraints is eventually decided by the safety margin rather than the permissible range. Moreover, a skew with large permissible range may have small safety margin if the skew target is not at the center of the permissible range. This is illustrated in Figure 1.

The observation of [5] provides an approach to improve clock network robustness. However, this approach does not work for certain flip-flop placement. If two tightly constrained clock sinks (with small safety margin) are far apart, there are

Figure 2: Flip-flops with the same (different) gray scale have tight (loose) permissible range. For placement of (a), clock tree construction like in [5] results in large wirelength. The placement of (b) leads to both small wirelength and less skew variation between tightly constrained flip-flops.

always long non-common paths between them irrespective of when they are merged in the clock tree. Moreover, merging sinks far apart may cause large clock network wirelength overhead. For the example in Figure 2, sinks with same (different) gray scale have small (large) safety margin. For the placement of Figure 2(a), two dark (light) sinks are first merged at node j (k) and then node j and k are merged to the clock source. Obviously, long non-common paths still exist and the total wirelength is large in (a). If the placement is changed to Figure 2(b), the same merging order results in short non-common path and small total wirelength. This observation motivates us to move tightly constrained flip-flops close to each other during placement. The flip-flop

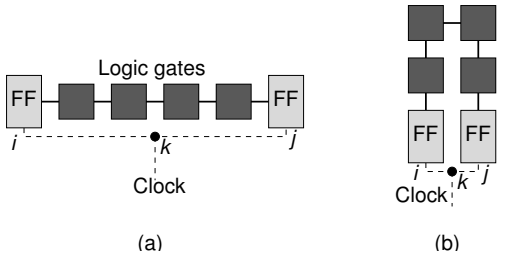

Figure 3: Placement in (b) results in the same combinational logic path delay as in (a) but smaller skew variation between flip-flop i and j than in (a).

placement cannot be performed without considering logic cells due to the connections between them. When flip-flops are moved for clock network robustness during placement, the adverse impact to the traditional objectives of placement needs to be minimized. For example, we may need to limit the increase on total wirelength of signal nets. Figure 3 shows a small and ideal case where the distance between two flip-flops is reduced without affecting wire length and path delay of combinational logic. Of course, practical cases are generally much more complicated. The purpose of this example is to demonstrate the direction of our efforts. If a pair of flip-flops cannot be moved close to each other due to constraints on signal wirelength, skew optimization can be performed to maximize the safety margin of the skew between them as in [4].

3. PRELIMINARIES

Let $S = \{s_1, s_2, \ldots s_N\}$ denote the N standard cells that need to be placed. Note that some of these cells need to be placed in fixed positions. Let $E = \{e_1, e_2, \ldots e_k\}$ denote that k nets in the circuit. Note that each net $e_i \subset S$ is composed by a set of cells. Let $C = \{c_1, c_2, \ldots c_n\}$ denote the clock sinks ($C \subset S$) and c_0 denote the clock source. A placement P of S is an assignment of coordinates (x_i, y_i) to each movable cell ($s_i \in S$). A clock tree T is a tree connecting all clock pins to the source c_0. The total wire length refers to the wire length of the nets (measured as half-perimeter of the bounding box) in E and the length of the tree T.

Placement for Robust Clocking. Find a placement P for all movable cells in S and a clock tree T such that the resulting clock network is tolerant to variations in clock skew and the total wire length is minimized.

We will employ the Elmore delay model [14, 15] for our delay computations and simulations. In this work, we intend to demonstrate the merits in a placement methodology that improves the tolerance of the clock network to process variations. Hence our results should be viewed in a relative sense - that is the advantage of a variation aware placement to the traditional wire length driven placement. Though the Elmore delay model is sometimes inaccurate, it has a high fidelity for guiding combinatorial optimizations. Hence it is a good measure of the relative reduction in skew due to process variations.

4. METHODOLOGY

The overview of our methodology is presented in Figure 4. The first step in the algorithm is to obtain an initial placement. One can use any standard cell placer for this purpose. But, for purposes detailed later, it would be better to use a force based stable placer like mPL [13]. This is followed by timing analysis on the placed net list. The objective of timing analysis is to obtain the minimum (D^{ij}_{min}) and maximum (D^{ij}_{max}) delays between every sequentially adjacent flip-flop pair (i, j). A pair of flip-flops are said to be sequentially adjacent if there is a pure combinational logic path between them. Any static timing analyzer capable of computing the minimum and maximum delays can be modified to compute the D^{ij}_{max} (D^{ij}_{min}) [2]. D^{ij}_{max} is the arrival time at j when the arrival time at i is zero and the arrival time at all other inputs (of this combinational block) is $-\infty$. Similar procedure can be used to compute D^{ij}_{min}. From D^{ij}_{max} and D^{ij}_{min}, we can compute the permissible ranges using the long path and short path constraints. The next step is the construction of the Zero Skew Tree (ZST). We employ the DME algorithm [7] for tree construction. Edahiro's nearest neighbor algorithm [16] is used to generate the tree topology (merging order). But, we also ensure that the resulting tree has a balanced structure like the one shown in Figure 5. Such a balanced structure gives the tree a higher tolerance to inter-die process variations [17]. Once the tree is constructed, we need to select certain critical

Methodology Overview
Input: Standard cells S, nets E, clock sinks C Output: Placement of standard cells and clock tree
1. Find an initial placement (P_1) using Force based Quadratic Placement 2. Run a timing analyzer to compute the skew permissible range for all sequentially adjacent flip-flops 3. Construct a Zero Skew Tree (ZST) using P_1 4. Find the set of critical sink pairs ($C_{critical}$) 5. Insert a pseudo net between all pairs in ($C_{critical}$) 6. Run the wire length driven placer with $E \leftarrow E + C_{critical}$ to obtain new placement P_2 7. Run ZST on P_2

Figure 4: Placement for Robust Clocking

pairs that are more prone to the effects of process variations. This is a key step in the algorithm and will be explained in greater detail in the following section. Let $C_{critical}$ denote the set of critical pairs selected. $C_{critical} = (p_1, p_2, \ldots p_\alpha)$, where p_i denotes a pair of flip-flops, say (c^1_i, c^2_i). α denotes the total number of critical pairs. This is a user-controlled parameter. Once the critical pairs are selected, the set of nets (E) is augmented with $C_{critical}$ and we run the placement engine again. We shall refer to the set $C_{critical}$ as **pseudo nets** since they do not exist in the circuit. Remaining nets (in E) will henceforth be refereed to as **regular nets**. The purpose of inserting pseudo nets is to bring the flip-flop pairs that are more susceptible to variations closer. This, in turn increases the safety margin and hence the tolerance to variations. The number of critical pairs selected (α) is far less than the number of nets in the circuit. Hence the extra load the placement engine is negligible.

5. SELECTION OF PSEUDO NETS

The crucial part of the algorithm is the selection of pseudo nets. The pseudo nets should be selected in such a way that the following objectives are met:

1. The flip-flop pairs that have poor safety margin should be brought close by
2. The increase in the signal net length should be minimal

The second point is especially important since a large increase in signal net length will translate into larger area and power. Such large increase (if it happens) will be a bad trade-off and will negate the benefits of higher variation tolerance. Further we also do not want to create a large change in the placement obtained from step 1. We perform the timing analysis (to compute the permissible ranges) based on the placement in step 1. The critical pairs are computed on the basis of this timing analysis. Hence, a large change in the placement at step 2 will reduce the effectiveness of our

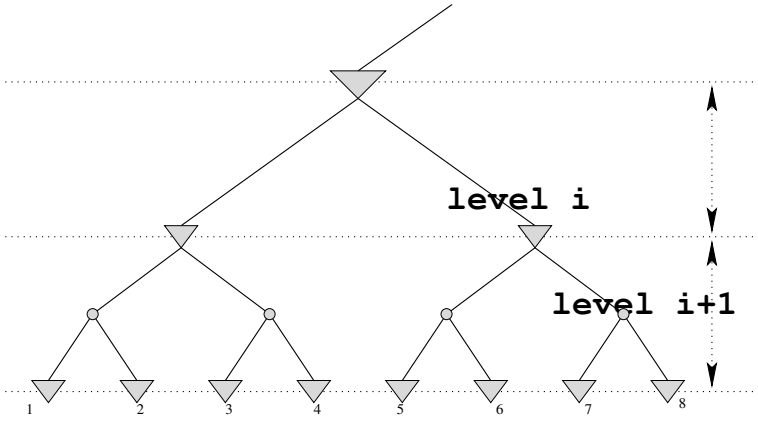

Figure 5: Balanced Zero Skew Tree

critical pair selection. For this reason, we need to use a stable placer like mPL [13]. When there is limited change on a circuit, the change of the result of a stable placer is also limited. Further in order to accomplish this, we introduce three parameters:

1. α: Denotes the total number of pseudo nets inserted. Limiting α also limits the extra effort on part of the placer.
2. β: Denotes the number of pseudo nets associated with a particular flip-flop i. Suppose we introduce pseudo net between node i and nodes i_1, i_2, i_3, i_4 and each of these nodes are far apart from each other, then it is likely that these pseudo nets will have a big increase in the length of the regular nets. Such a scenario is illustrated in figure (6) where the introduction of 4 pseudo nets could have a significant impact on the regular net length. This case is undesirable and could be avoided by setting β to a low value. For example, if β equals 1, only one of the four pseudo nets shown in figure (6) will be inserted.
3. γ: Prevents the introduction of pseudo nets between nodes that are already close to each other in the original placement. Let M_{max} denote the maximum distance between any two sinks in the original placement. Then we do NOT introduce a pseudo net between two nodes i and j if the distance between them is smaller than γ times M_{max}.

Our selection of pseudo nets follows a greedy procedure outlined below. Let S_{ij} denote the skew safety margin between flip-flop i and j. S_{ij} equals $min(-L_{ij}, U_{ij})$ since we are constructing a zero skew tree. For example, if the skew permissible range between two nodes is $(-150psec, 50psec)$, then the safety margin is 50 psec. In other words if the skew q_{ij} deviates by S_{ij} (in magnitude), there will be failure. Lower safety margin implies higher probability of failure. Let M_{ij} denote the distance between i and j. Let S_{min} and M_{max} denote the minimum safety margin and maximum

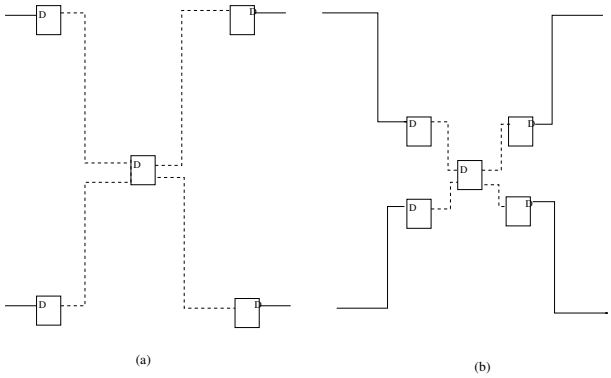

Figure 6: Dashed (continuous straight) lines denote pseudo (regular) nets. (a) shows the case with small regular and large pseudo net length. In (b) registers are moved closer at the expense of higher regular net length

distance between any two clock sinks respectively. Then the criticality of a pair of nodes is defined as:

$$Criticality = SW(\frac{S_{min}}{S_{ij}}) + (1 - SW)(\frac{M_{ij}}{M_{max}}) \quad (1)$$

In equation (1), SW denotes the weight given to safety margin $(0 \leq SW \leq 1)$. The intuition behind the above equation comes from the fact that sink pairs with lower safety margin (or) higher distance are more susceptible to variations [8]. Using the above criticality measure, we select the α nodes that have the highest criticality subject to the fact that each node pair is at least a distance γM_{max} from each other and the number of pseudo nets that each node is involved in is less than β. The selection process is summarized in figure (7).

6. EXPERIMENTAL RESULTS

We use the ISCAS89 benchmark circuits for our experiments. Technology mapping on the benchmark circuits is performed using SIS [18]. Standard cell placement is done

Procedure: Selection of Pseudo Nets
1. S_c = set of clock sink pairs sorted in non-increasing order of criticality N_p = set of pseudo Nets $S_1 = (C_1, C_2)$ most critical pair in S_c
2. $\forall\, i \in C$, β_i = 0
3. $N_p \leftarrow \emptyset$
4. while $\lvert N_p \rvert \leq \alpha$ and $S_c \neq \emptyset$
4.1 Select $S_1 = (C_1, C_2)$
4.2 If $distance(C_1, C_2) \leq \gamma * M_{max}$ and $\beta_1 < \beta$ and $\beta_2 < \beta$
4.2.1 $N_p \leftarrow N_p + S_1$
4.2.2 $\beta_1 \leftarrow \beta_1 + 1$
4.2.3 $\beta_2 \leftarrow \beta_2 + 1$
4.3 else
4.3.1 $S_c \leftarrow S_c \backslash S_1$
5. Output N_p

Figure 7: Selection of Pseudo Nets

Case	#Sinks	#Cells	#Nets
s13207	500	3502	3350
s15850	566	3939	3852
s35932	1728	14882	14562

Table 1: Benchmark Characteristics

using mPL [13]. Assuming a $180nm$ technology, wire capacitance values are obtained by using the SPACE 3D extraction tool [19]. The other wire parameters such as ILD (interlayer dielectric) dimensions and sheet resistances are taken from [20]. The buffer resistance and capacitance are assumed to be 129Ω and 14.27fF respectively.

The characteristics of the benchmark circuits are shown in table (1). Here the number of sinks and number of cells refer to the number of flip-flops and standard cells respectively. The number of nets refers to the number of regular nets in the circuit.

The proposed methodology was implemented in C++/PERL on a Linux Machine with 1GB RAM. Variations in skew were evaluated using 10,000 Monte Carlo simulations on the test circuits. Variations in wire width, sink load capacitance and buffer resistance were considered. These variations are assumed to follow Gaussian distribution with standard deviations (σ) equal to 6.67% (3σ equals 20%) of their nominal values. Spatial correlations among the variations are handled by the PCA (Principle Component Analysis) method as in [21]. The parameters β and γ were set to 1 and 0.25 respectively. The slack weight in equation (1) was set to 0.50. The value of α (number of pseudo nets) was set to 20 for the first two cases and 80 for the third case.

Table 2 gives the test case results for the base case. The base case is the zero skew tree constructed without any pseudo nets. SL, CNL and TNL denote the signal net length, clock net length and the total net length (signal + clock) in μm respectively. The CPU time is indicated in seconds. For each Monte Carlo run, the maximum skew violation over all violations is noted. MV, AV and STD denote the maximum, average and standard deviation of the maximum violation at each run (in psec). Skew violation is the deviation of the skew from the permissible range. Table 3 presents the results after inserting the pseudo nets. All parameter values (excepting CPU time) are indicated relative to their corresponding values in Table 2. The CPU time in Table 1 includes the time for placement, timing analysis and clock tree construction. The CPU time in Table 2 includes all the steps indicated in figure (4). The results lead to the following observations:

1. The number of pseudo nets introduced is a very small fraction of the number of regular nets (less than 1%)
2. Introducing pseudo nets could reduce the maximum skew violation by up to 62%.
3. The resulting increase in wire length is at most 2.7%.
4. Pseudo nets also lead to reduction in the wire length of the clock net.

Further the proposed methodology can be easily integrated with the existing placement and clock tree synthesis tools.

7. CONCLUSIONS

In this paper, we presented a placement methodology targeted at robust clocking. We studied the relation between the cell placement and resulting clock skew safety margins. A small number of pseudo nets were added to the existing regular nets. Such an addition leads to a clock network which has significantly better tolerance to process variations.

8. REFERENCES

[1] Y.-C. Chou and Y.-L. Lin. Effective enforcement of path-delay constraints in performance-driven placement. *IEEE Transactions on Computer-Aided Design*, 21(1):15–22, January 2002.

[2] J. P. Fishburn. Clock skew optimization. *IEEE Transactions on Computers*, C-39:945–951, July 1990.

[3] I. S. Kourtev and E. G. Friedman. Clock skew scheduling for improved reliability via quadratic programming. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 239–243, 1999.

[4] G. Venkataraman, C. N. Sze, and J. Hu. Skew scheduling and clock routing for improved tolerance to process variations. In *Proceedings of Asia and South Pacific Design Automation Conference*, pages 594–599, 2005.

[5] D. Velenis, E. G. Friedman, and M. C. Papaefthymiou. A clock tree topology extraction algorithm for improving the tolerance of clock distribution networks to delay uncertainty. In *Proceedings of the IEEE*

Case	BASE CASE						
	SL	CNL	TL	MV	AV	STD	CPU(s)
s13207	563690	128168	691858	82.88	13.86	12.56	72.8
s15850	669036	141432	810468	60.72	11.18	10.35	56.8
s35932	1542840	489569	2032409	216.75	39.20	32.48	186.6

Table 2: Results for the base case

Case	WITH PSEUDO NETS						
	SL	CNL	TL	MV	AV	STD	CPU(s)
s13207	1.04	0.94	1.027	0.38	0.581	0.499	163.1
s15850	1.02	0.935	1.009	0.384	0.585	0.515	116.4
s35932	1.026	0.98	1.015	0.61	0.683	0.652	391.9

Table 3: Results with Pseudo Net Addition (relative to Base Case, CPU time is absolute)

International Symposium on Circuits and Systems, pages 4.422–4.425, 2001.

[6] A. Jairath, B. Sivasubramanian, and D. Velenis. Block placement for reduced delay uncertainty in high performance clock distribution networks. In *Proceedings of the Midwest Symposium on Circuits and Systems*, 2005.

[7] T.-H. Chao, Y.-C. Hsu, J.-M.Ho, K. D. Boese and A. B. Kahng. Zero Skew Clock Routing with Minimum Wirelength. In *IEEE Transactions on Circuits and Systems-II*, vol. 39, no. 39, pp. 799–814, 1992.

[8] B. Lu, J. Hu, G. Ellis, and H. Su. Process variation aware clock treerouting. In*ISPD 2003* pp. 174–181

[9] J. Oh, M. Pedram. Gated clock routing for low-power microprocessor design. In *IEEE Transactions on Computer-Aided Design*, 20(6), pp. 715-722, June 2001.

[10] C.-W. A. Tsao and C.-K. Kohi. UST/DME: a clock tree router for general skew constraints. *ACM Transactions on Design Automation of Electronic Systems.*, vol. 7, no. 3, pp. 359–379, 2002.

[11] S. Zanella, A. Nardi, A. Neviani, M. Quarantelli, S. Saxena, and C. Guardiani. Analysis of the impact of process variations on clock skew. In *IEEE Transactions on Semiconductor Manufacturing*, vol. 13, no. 4, pp. 401–407, 2000.

[12] Y. Liu, S. R. Nassif, L. T. Pileggi, and A. J. Strojwas. Impact of interconnect variations on the clock skew of a gigahertz microprocessor. In *DAC 2000*, pp. 168–171.

[13] T. F. Chan, J. Cong, J. Shinnerl, and K. Sze. An enhanced multilevel algorithm for circuit placement. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 299–306, 2003.

[14] W. C. Elmore. The Transient Response of Damped Linear Networks with Particular Regard to Wideband Amplifiers In *Journal of Applied Physics*, Volume 19, Issue 1, pp. 55-63, Jan 1948.

[15] J. Rubinstein, P. Penfield, and M. A. Horowitz. Signal delay in RC tree networks. *IEEE Transactions on Computer-Aided Design*, CAD-2(3):202–211, July 1983.

[16] M. Edahiro. A clustering-based optimization algorithm in zero-skew routings. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 612–616, 1993.

[17] S. Pullela, N. Menezes, J. Omar, and L. T. Pillage. Skew and delay optimization for reliable buffered clock trees. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 556–562, 1993.

[18] E. M. Sentovich, K. J. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. R. Stephan, R. K. Brayton, and A. L. Sangiovanni-Vincentelli. SIS: a system for sequential circuit synthesis. Memorandum no. M92/41, ERL, University of California, Berkeley, May 1992.

[19] SPACE: VLSI physical design modeling and verification. http://space.tudelft.nl. Delft University of Technology in the Netherlands.

[20] S. P. Khatri, A. Mehrotra, R. K. Brayton, A. Sangiovanni-Vincentelli, and R. H. J. M. Otten. A novel VLSI layout fabric for deep sub-micron applications. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 491–496, 1999.

[21] H. Chang and S. S. Sapatnekar. Statistical timing analysis considering spatial correlations using a single PERT-like traversal. In *Proceedings of the IEEE/ACM Int Conference on Comp. Aided Design* , pages 621–625, 2003.