

# Thermal Aware Clocktree Optimization in Nanometer VLSI Systems Considering Temperature Variations

Chunchen Liu<sup>¶</sup>, Jichang Tan<sup>^</sup>, Ruei-Xi Chen<sup>^</sup>, Guanglei Liu<sup>\*</sup>, Jeffrey Fan<sup>\*</sup>,

<sup>¶</sup> Department of Electrical and Computer Engineering, University of California, San Diego

<sup>^</sup> Department of Computer Science and Information Engineering, St John's University, Taipei, Taiwan

<sup>\*</sup> Department of Electrical and Computer Engineering, Florida International University, Miami, Florida

## Abstract

This paper proposed an accurate yet efficient clocktree synthesis algorithm that tolerates the temperature variation in the nanometer Very Large Scale Integration (VLSI) systems. Observing that there exists the correlation between input thermal power and temperature, we propose a clustered perturbation based parameterization to compactly generate a clocktree model considering the temperature variation. Moreover, using a structured and parameterized model reduction, an accurate worst case skew can be efficiently obtained from a transient simulation, which provides both nominal temperatures at those sinks and their sensitivities with respect to the changes of merging points. With the use of the sensitivities from the macromodel, a thermal aware routing based clocktree optimization is performed level by level. The experimental results show that the proposed algorithm reduces up to 1.7X- 5X worst case skew in comparison to the existing approaches.

## 1. Introduction

Due to miniaturization in modern VLSI systems, thermal issues were caused by the increased power density. The existing power optimization algorithms were focused on reducing the overall power dissipation, but local hot spots and thermal gradients can cause serious problems to high performance clock design in VLSI design. Nevertheless, uniform temperature is assumed in most of existing work on clock optimization [3][4]. Considering the non-uniform on-chip temperature distribution, [2] and [1] are the pioneer work to attack the thermal-aware clock tree optimization. The concept of merging diamond was introduced to help to find the skew balance region between uniform and non-uniform thermal profiles. However, the temporal variant of on-chip temperature was not considered in [2] and [1]. These effects cannot be ignored, especially when different applications or data sets are applied to the same chip. Meanwhile, the merging point perturbation will increase wire length overflow in those two proposed methods.

In this paper, we propose a clock tree optimization algorithm to minimize the worst case temperature induced skew consider the on-chip temperature variations. In addition, accurate delay model is assumed by performing transient time analysis at sinks of the impulse respond. A clustering based sampling technique and a structured parameterized model order reduction are performed to increase the efficiency of the algorithm. Without using merging point perturbation, we use thermal aware routing to balance the skew which will not cause too much serious wire-length overflow. High order delay model is used in this paper in order to evaluate and balance the real skews.

## 2. Thermal Model

The time variance of on-chip temperature cannot be ignored. In order to model it, we impose a grid onto the chip and each grid is assigned by a temperature range. This temperature range can be obtained by measurement or thermal simulation. This paper optimizes the worst-case temperature induced skew.

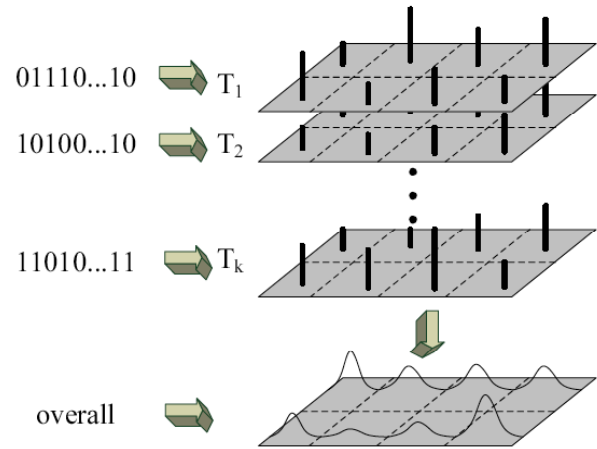


Figure 1: On-chip temperature variation during runtime

A complete instruction set is tested and the corresponding K temperature profiles are obtained (see Figure 1). The overall temperature variation can then be extracted based on the K temperature profile. The range of possible temperature in each sub-region can be obtained and expressed as

$$RT^i = [T_{min}^i, T_{max}^i], \forall \text{ sub-region } R_i \quad (1)$$

Based on [5], in an interconnect experiencing temperature profile  $T(x, y, t)$  along its length in time  $t$ , resistance will change linearly with temperature and the unit length resistance  $r_{unit}$  of the clock wire segment can be calculated as

$$r_{unit}(x, y, t) = \rho_0 \cdot [1 + \beta \cdot T(x, y, t)] \quad (2)$$

Where  $\rho_0$  is the unit length resistance at  $0^\circ\text{C}$ , and  $\beta$  is the temperature coefficient of resistance ( $1/^\circ\text{C}$ ). Also, note that unit length capacitance does not change with temperature variations along the interconnect length [5].

Definition 1. *Temperature Variation Aware Clocktree Optimization: Given source Src, sinks  $s_1 \dots s_n$ , abstract topology*

of the clock tree and temperature variation profile, find an embedded clock tree which minimizes the worse case skew under the temperature variation.

### 3. Routing Based Clock Optimization

Routing is a crucial step in clock tree synthesis design. Most of existing synthesis uses Manhattan distance to simulate and evaluate the delay between each two nodes [2], [1]. In real case, Manhattan distance is not exactable when considering time-variant temperature maps if our objective function is with weight of temperature and distance parameters. We use *thermal aware routing topology optimization* with considering both distance and temperature parameters in this paper, which is not only accurate but also can reduce the skews at the same time.

High temperature variance greatly impacts the delay and skew of zero-skew clock tree. Based on routing topology optimization, we try to find the routing path with smooth temperature gradient. We firstly construct an initial zero-skew clock tree by ZST/DME algorithm under the uniform temperature profile to minimize the wire length. We then mark some subareas with high possibility of higher temperature variation and avoid route thorough those area. After that, we can locally adjust the routing path between two merging points to lower the temperature variance effect and keep short wire length.

#### 3.1 Overall Algorithms

An initial zero-skew clock tree [6] constructed under uniform temperature profile is a good start point of our algorithm. Given an initial tree, we locally adjust the routing path by a bottom-up fashion. A levelization is first performed on the abstract topology of the initial tree. In each level  $N_i$  of the clocktree (except for the botton level with all sinks), we try to change the location of the non-sink nodes in  $N_i$  within a range.

**Definition 2.** Level  $i$  ( $i \geq 0$ ) in a given abstract topology  $T$  with source node  $nsrc$  is a node set  $N_i$ , in which the length (number of edges in the path) of the path ( $n \rightarrow nsrc$ ) is  $i$ ,  $\forall n \in N_i$ .

**Definition 3.** A routing path (RP) of node pair ( $s_1; s_2$ ) is a new path, which locates at the bound of the cycle with radius  $r$  centering at the initial merging point of this node pair.

**Definition 4.** An Embedding of Perturbation (EOP) is a pair of embedded paths ( $M, s_1$ ) and ( $M, s_2$ ) for each MPP  $M$  of node pair ( $s_1, s_2$ ).

**Definition 5.** A Routing Path Configuration (PC) in level  $i+1$  is a vector  $[\Phi_1, \dots, \Phi_m]$ , where  $\Phi_i = (M_i, P_i^1, P_i^2)$  is MPP  $M_i$  and corresponding EOP,  $P_i^1, P_i^2$ , of subnode pair ( $s_1^j, s_2^j$ ) in level  $i$ .

---

#### Algorithm 1 Thermal Aware Clocktree Optimization (TCO)

---

**Input:** Source  $Src$ , sinks  $s_1, \dots, s_n$ , embedded tree  $T$

**Output:** A modified embedded clock tree  $T'$

1: ( $Levels, N_{levels}$ )  $\leftarrow$  Levelize  $T$

{Bottom up embedding from the second last level to the second level}

2: for  $i = N_{levels} - 1$  to 1 **do**

3:  $P C_i$  Thermal aware routing in level  $i$

4: **for**  $\forall p_i^j \in P C_i$  **do**

5:  $R_i^j, C_i^j \leftarrow$  Extract R/C of the updated tree for  $p_i^j$

6:  $Volt_{sk}, \forall sink_{sk} \leftarrow$  Solve the system based on (6)

7:  $Dly_{src \rightarrow sk}, \forall sink_{sk} \leftarrow$  Calculate Source-Sink Delay based on Back-Eular method

8: Calculate worse case skew under  $P C_i$  as follows

$Skew_j = \max_{\forall sink_{sk}} Dly(src \rightarrow sk) - \min_{\forall sink_{sk}} Dly(src \rightarrow sk)$

9: **end for**

10: Embedding in level  $i \leftarrow (PC_i^* = \min_{\forall PC_i} Skew_j)$

11: **end for**

12: **return**  $T'$

---

### 3.2 Obtain “Bottom Up” Path

The feasible routing path of sub node pairs  $S_i, S_j$  can be any point on the cycle centering at  $M$ . To reduce complexity, we sample the points in the four Manhattan directions (up, down, left, right) from  $M$ . The radius of each direction  $D_i$  in this MPP cycle is decided based on the local temperature gradient in  $D_i$ . We use the larger radius for the sharper temperature gradient. This strategy allows more searching space for bigger temperature variations. Basically, the following linear estimation is used to decide the feasible area radius  $r_p$  of  $M$  in direction  $D_i$ .

$$r_p(M) = \alpha \cdot grad(M, D_i) \quad (3)$$

where  $\alpha$  is a constant and  $grad(M, D_i)$  is the temperature gradient of  $M$  in direction  $D_i$ , which is calculated based on the given temperature profiles. In experiments, we find that it achieves the best runtime-performance tradeoff to choose the corner points  $M_1, M_4$  in the routing path candidate set to generate MPP.

Given an MPP  $M'$ , we then decide the EOP of  $M'$ . Due to the combinatorial natural of paths, there exist factorial different EOPs from an MMP  $M'$  and a level  $i$  node  $s_i$ . To consider the on-chip temperature un-uniformity, we seek an EP which has least sensitivity to temperature variation among all possible paths  $M' \rightarrow s_i$ . Since the standard deviations  $Std_j$  of each sub-region  $j$  can be obtained by the thermal simulation. The desired EOP,  $EOP_d(M', s_i)$ , is the shortest path in terms of the sum of square of standard deviations of the sub-regions that it passes as expressed by follows.

$$EOP_d(M', s_i) = \min_{\forall P \in M' \rightarrow s_i} \sum_{\forall e \in P} std(e)^2 \quad (4)$$

When the EOP  $EOP_d(M', s_i)$  of MPP  $M'$  is fixed, we calculate the new resistance based on (5).

$$R(M', s_i) = \sum_{\forall e \in EOP_d(M', s_i)} E[r_{unit}(e)] \cdot len(e) \quad (5)$$

where  $E[r_{unit}(x, y)]$  is the mean value of temperature in edge  $e$  and  $len(e)$  is the length of this edge.

### 3.3 Skew Calculation

#### 3.3.1 System State Function Generation

In subsection subsec.mop, we have generated a set of routing path configurations in level  $i$ . Now we need to calculate the worst case skew for each configuration  $P_i^j$  (line 4 to line 9 in Algorithm 1). We calculate source to sinks delay based on their impulse voltage responds, which can be obtained as follows.

Given the initial tree  $T_{init}$ , the system transfer function of Modified Nodal Analysis (MNA) can be expressed as follows

$$(G_0 + sC_0)x = Bu \quad (6)$$

$$y = L_0^T x \quad (7)$$

where  $G_0$  is the conductive matrix and  $C_0$  is the capacitive matrix. Suppose there are  $N$  nodes in the tree.  $G_0$  and  $C_0$  are both  $N \times N$  matrices.  $G_0$  can be obtained by the incident matrix of the tree and  $C_0$  is a diagonal matrix where the diagonal elements are the lumped capacitive values of the tree nodes. Since the source node of the tree is the only input of the system, we have

$$B = [1, \underbrace{0, \dots, 0}_{N-1}]^T \quad (8)$$

where  $B$  is a  $N \times 1$  matrix.  $u$  is a  $N \times N$  identity matrix and  $x$  is a length  $N$  vector corresponding to the voltage of the impulse respond of each tree node, including all sinks and Steiner points, which is can decrease the total length of connection. We organize vector  $x$  as the following way

$$x = [x_{s_1}, \dots, x_{s_n}, x_{S_1}, x_{S_2}, \dots, x_{S_t}] \quad (9)$$

where  $s_i, i = 1, \dots, n$  are sinks and  $S_i, i = 1, \dots, t$  are Steiner points. Since we are interested in the voltage responds at sinks, we have

$$L_0 = \begin{bmatrix} I_{n \times n} \\ 0_{(N-n) \times n} \end{bmatrix} \quad (10)$$

The selected output vector  $y$  is

$$y = [y_{s_1}, \dots, y_{s_n}] \quad (11)$$

For each PC,  $PC_i$ , in level  $i$ , (6) can be re-written as

$$[G_0 + \Delta G_i + s(C_0 + \Delta C_i)] \cdot (x + \Delta x_i) = B \quad (12)$$

Discard second order terms  $\Delta G_i \Delta x_i \Delta C_i \Delta x_i$  and obtain follows

$$(G_0 + sC_0)x = Bu \quad (13)$$

$$G_0 \Delta x_i + \Delta G_i x = 0 \quad (14)$$

Suppose we select  $m$  PCs,  $c_1, \dots, c_m$ , in level  $i$ , the system transfer function can be re-written as

$$(G_0 + sC_0)x = Bu \quad (15)$$

$$G_0 \Delta x_1 + \Delta G_1 x = 0 \quad (16)$$

$$\dots$$

$$G_0 \Delta x_m + \Delta G_m x = 0 \quad (17)$$

Re-write this new system by matrix, we have

$$(G_{PC} + sC_{PC}) \cdot x_{PC} = B_{PC} \quad (18)$$

$$y_{PC} = L_{PC}^T \cdot x_{PC} \quad (19)$$

where

$$G_{PC} = \begin{bmatrix} G_0 & 0 & \dots & 0 & 0 & 0 \\ \Delta G_1 & G_0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \Delta G_k & 0 & \dots & G_0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \Delta G_m & 0 & 0 & \dots & 0 & G_0 \end{bmatrix} \quad (20)$$

$$C_{PC} = \begin{bmatrix} C_0 & 0 & \dots & 0 & 0 & 0 \\ \Delta C_1 & C_0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \Delta C_k & 0 & \dots & C_0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \Delta C_m & 0 & 0 & \dots & 0 & C_0 \end{bmatrix} \quad (21)$$

$$x_{PC} = [x, \Delta x_1, \dots, \Delta x_m]^T \quad (22)$$

$$B_{PC} = [B, \underbrace{0, \dots, 0}_{m-1}]^T \quad (23)$$

The output variables  $y_{PC}$  can be divided into two parts, i.e. nominal value and first order sensitivity. Therefore  $y_{PC}$  and  $L_{PC}$  (block-wise diagonal matrix) can be written as follow

$$y_{PC} = [y, \Delta y_1, \dots, \Delta y_m]^T \quad (24)$$

$$L_{PC} = \text{Diag}(L_0, \underbrace{\dots, L_0}_m) \quad (25)$$

where each  $\Delta y_j$  is a length  $n$  vector, which represents the voltage responds change for each sink under routing path configuration  $PC_j$ .

### 3.3.2 Skew Calculation

The expanded system expressed by (19) and (19) in time domain can be written as

$$G_{PC} + C_{PC} \cdot \frac{dX_{PC}(t)}{dt} = B_{PC} \quad (26)$$

$$Y_{PC}(t) = L_{PC} \cdot X_{PC}(t) \quad (27)$$

This time domain transient response can be solved by Back-Euler method. The system equation at time instant  $t$  with time step  $h$  is

$$G_{PC} + \frac{1}{h} \cdot C_{PC} \cdot X_{PC}(t) = \frac{1}{h} G_{PC} X_{PC}(t-h) + B_{PC} U \quad (28)$$

$$Y_{PC}(t) = L_{PC} \cdot X_{PC}(t) \quad (29)$$

Following the conventional definition for the propagation delay,  $Dly(\text{src} \rightarrow \text{si})$ , from source node  $\text{src}$  to sinks  $\text{si}$  in a net,  $Dly(\text{src} \rightarrow \text{si})$  is the time required for the node voltage to pass 80% of the peak voltage given the impulse excitation in the source node [7].

Starting from an initial solution, e.g  $X = 0$  when  $t = 0$ , the system expressed by (28) and (29) is solved iteratively. In each iteration  $i$ ,  $t = h \cdot i$ ,  $Y_{PC}^i$  is solved based on the results obtained in the previous iteration. Since the impulse respond increase monotonically before reach the peak voltage value, the time required to pass 80% of the peak voltage can be identified during the iterations.

After obtaining the source to sink delay of each routing path configuration  $PC_i^j$  in level  $i$ , we can calculate the worst case skew corresponding to  $PC_i^j$  (line 8 in Algorithm 1) as follows

$$Skew_j = \max_{\forall \text{sinks}_k} Dly(S_{rc} \rightarrow S_k) - \min_{\forall \text{sinks}_k} Dly(S_{rc} \rightarrow S_k) \quad (30)$$

## 4. Speedup Techniques

### 4.1 Clustering Based on Temperature Correlation

In level  $i+1$ , for each node pair, e.g.  $s_1$  and  $s_2$ , suppose there are  $m$  different merging point perturbations and  $h$  node pairs in level  $i$ . The full coverage of possible perturbations is  $m^h$ . The size of the expanded system (19) and (19) will increase exponentially

if we do not control the number of the combinations of the MPPs, which leads to un-solvability of the system in practice.

To reduce complexity, we may use “sampling” of routing path directions. Different from proximity based sampling in [3], our sampling technique to be presented in this section is based on the temperature variation sensitivity of the merging points in the same level. The routing path pattern can be the same for those nodes with similar sensitivity to temperature variation. We seek a “good” grouping (clustering) of the merging points in the same level. The temperature variations around the merging points within the same cluster are highly correlated, which leads to the similar sensitivity to the temperature variation.

Given cluster number  $k$  and the correlation strength between each two merging points, we use k-means classify algorithm [8] to partition  $m$  nodes in level  $i+1$  into  $k$  clusters. In each cluster, we use the same routing path direction pattern on all items belongs to it. Suppose we enumerate four directions (see subsection 3.2) in each cluster, the possible combination number of routing path direction of the clustered system in level  $i+1$  is  $4^k$  is much less than  $4^m$ , the one before clustering.

#### 4.1.1 Correlation Strength Calculation

For every two sub-regions  $Sub_x$ ;  $Sub_y$  for the temperature profile, we calculate their temperature correlation  $Cc(x, y)$  as follows.

$$Cc(x, y) = \frac{k \sum x_i y_i - \sum x_i \sum y_i}{\sqrt{k \sum x_i^2 - (\sum x_i)^2} \sqrt{k \sum y_i^2 - (\sum y_i)^2}} \quad (31)$$

where  $x_1, \dots, x_k$  and  $y_1, \dots, y_k$  are temperature values within sub-region  $Sub_x$  and  $Sub_y$ , respectively. Note that this process can be performed as the pre-process of the algorithm and the correlation coefficient  $Cc(x, y)$  can be built into a table.

According to the subsection 3.2, we only choose the four corner points as the routing path candidates in the perturbation cycle of a merging point. We define the correlation strength,  $Ccs(u, v)$  of two merging points  $u$  and  $v$  in level  $i+1$  by considering the correlations between the corresponding routing path candidates of  $u$  and  $v$ .

$$Ccs(u, v) = \sum_{j=1}^4 Cc(Sub(u_j), Sub(v_j)) \quad (32)$$

where  $u_1, \dots, u_4$  and  $v_1, \dots, v_4$  are the routing path candidates of  $u$  and  $v$  on the four direction, respectively.  $Sub(u)$  is the ID of the sub-region that node  $u$  belongs to.

#### 4.1.2 Decide Cluster Number

Singular Value Decomposition (SVD) [9] provides the closest rank- $k$  approximation for a matrix  $X$  and it has been used in image processing for compression and noise reduction [10]. By setting the small singular values to zero, we can obtain very good matrix approximations, whose rank equals the number of remaining singular values, using only a small number of terms.

Motivated by [10], we perform SVD on the correlation strength matrix  $Ccs$  as follows.

$$Ccs = U \cdot \text{Diag}(D_1, \dots, D_i, 0, \dots, 0) \cdot V^T \quad (33)$$

where  $U^T U = I$  and  $V^T V = I$ . We try to find the number of dominant singular values,  $k$ , which indicates the principle correlation strength of the input system and discards the noise. The number of clusters should be decided based on  $k$ .

To obtain  $k$ , we order all non-zero singular values as  $D_1 \geq \dots \geq D_k$  and  $k$  is the first label which satisfies  $\sum_{i=1}^k D_i \geq 0.9 \cdot \sum_{i=1}^m D_i$ , where  $\sum_{i=1}^m D_i$  is the sum of all singular values.

Given the cluster number  $k$  and correlation matrix  $Ccs$  as the data attribution, we perform k-means clustering [8] algorithm to obtain a partition of merging points in the current level.

## 4.2 Parameterized Model Order Reduction

After clustering based sampling presented in the previous subsection, the size of the expanded system (19) and (19) is reduced. However, the size of each block, e.g.  $G_0$ , is large and cannot be reduced by the sampling process. Thereby, we seek help from the projection based model order reduction to further speedup the algorithm.

Since the expanded system is lower triangularized, a flat projection matrix  $V$  can be constructed recursively using Arnoldi method [11]. However, directly projecting (19) by  $V$  leads to a reduced macro-model losing the block structure for the state matrices and variables. As a result,  $y$  and  $\Delta y_i$  are coupled with each other. Instead of using the flat projection matrix  $V$ , we introduce a structured projection matrix

$$V = \text{diag}[V_0, V_1, \dots, V_m] \quad (34)$$

by partitioning  $V$  according to the dimension of  $x$  and  $\Delta x_i$ . As a result, the reduced state matrices are as follows

$$\begin{aligned} \tilde{G}_{PC} &= V^T G_{PC} V & \tilde{C}_{PC} &= V^T C_{PC} V & (\in R^{q \times q}) \\ \tilde{B}_{PC} &= V^T B_{PC} & \tilde{L}_{PC} &= V^T L_{PC} & (\in R^{1 \times q}) \end{aligned}$$

Because  $V \subseteq \gamma$ , a  $q$ -th ordered projection by  $V$  still preserves at least  $q$  moment matching according to [12]. After reduction, the time-domain transient response (28) and (29) for the original expanded system is re-expressed as

$$\tilde{G}_{PC} + \frac{1}{h} \cdot \tilde{C}_{PC} \cdot \tilde{X}_{PC}(t) = \frac{1}{h} \cdot \tilde{G}_{PC} \tilde{X}_{PC}(t-h) + \tilde{B} \quad (35)$$

$$\tilde{Y}_{PC}(t) = \tilde{L}_{PC} \cdot \tilde{X}_{PC}(t) \quad (36)$$

where

$$\tilde{G}_{PC} = \begin{bmatrix} \tilde{G}_0 & 0 & \dots & 0 & 0 & 0 \\ \Delta \tilde{G}_1 & \tilde{G}_0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \Delta \tilde{G}_k & 0 & \dots & \tilde{G}_0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \Delta \tilde{G}_m & 0 & 0 & \dots & 0 & \tilde{G}_0 \end{bmatrix} \quad (37)$$

and

$$\tilde{y}_{PC} = [\tilde{y}, \Delta \tilde{y}_1, \dots, \Delta \tilde{y}_m]^T \quad (38)$$

$$(39)$$

Note that  $\tilde{C}_{PC}$  the same structure as  $\tilde{G}_{PC}$

Because the reduction preserves the block structure, the reduced nominal value  $\tilde{y}$  and first-order sensitivity  $\Delta \tilde{y}_i$  can be solved independently. The voltage respond in each sink under routing path configuration  $PC_j$  is

$$\tilde{y}_{PC_j}(t) = \tilde{y}(t) + \Delta \tilde{y}_j(t) \quad (40)$$

Note that as the reduced system still has the lower-triangular structure, (35) and (36) can be especially solved using back substitution.

## 5. Experimental Results

We employ the standard clocktree benchmarks r1-r5 in our experiments which are the same as PECO and TECO [1][2]. The

initial zero-skew tree is constructed by the DME method under the Elmore delay model with no temperature variation. The interconnect has unit resistance,  $r_0 = 0.03\Omega/\mu m$ , unit capacitance  $c_0 = 2.0 \times 10^{-16} F/\mu m$ , and temperature sensitivity  $\beta = 0.0068$ . The above interconnect parameters are same as those in [2].

A chip is divided into a uniform grid with  $100 \times 100$  regions to obtain the distributed temperature map by a micro-architecture level cycle-accurate power/temperature simulator. Our experiments use six SPEC2000 applications (*art*, *ammp*, *compress*, *equake*, *gcc*, and *gzip*). We collect 100 temperature maps by simulating these applications in a sequence and recording temperature maps for every 10 million clock cycles after fast forwarding of 1 billion cycles. These applications lead to a temperature variation about  $50^\circ C$  over the 100 temperature maps. Table 1 compares our proposed TCO with the existing techniques DME, TECO [2], and perturbation based reembedding PECO from [1]. TCO obtains the smallest worst-case skews for all r1-r5 circuits, and reduces the worst-case skew by up to 5.01x (1422ps vs 283.5ps) compared to DME, by up to 2.1x (144.2ps vs 62.2ps) compared to TECO, and by up to 1.72x (107.3ps vs 62.2ps) compared with PECO.

We then plot the skew distributions over 100 temperature maps in Figure 2. All clock trees optimized by TCO have smaller skews than those obtained by DME. The averaged skew reduction by TCO is 6.4X, and the skew distribution by TCO is also much narrower.

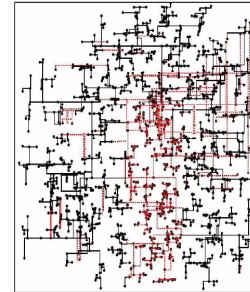
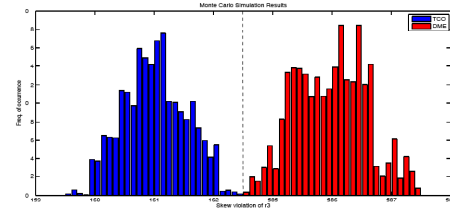
Table 1 also reports wirelength and runtime for different algorithms, where runtime includes the time to build macro-model. The three skew reduction algorithms all have small amount of wirelength increase compared to DME. TCO has almost the same wirelength as DME does. But more importantly, TCO has over 0.5% shorter wirelength than TECO and PECO. As to runtime, TCO has a runtime larger than DME using Elmore model, but has a smaller runtime compared with TECO and PECO. DME skew results here are quite different as classical one because we use high order delay model with time variant temperature maps here which will be more accurate in real case.

## 6. Conclusion

Existing clocktree synthesis optimization in VLSI design did not consider the extra skew caused by the temperature variations. In this paper, we have developed a minimal skew clock tree embedded with the time-variant temperature variations plus correlations. Thermal aware routing topology optimization is used to avoid hot-spots of temperature variation in the proposed approach. The experimental results show that the proposed algorithm reduces up to 5X worst-case skew compared to other existing methods.

## 7. References

- [1] H. Yu, Y. Hu, C. Liu, and L. He, "Minimal skew clock embedding considering temperature variation," in *Proc. Intl. Symp. Physical Design*, 2007.



**Figure 2:** (a) Given 100 temperature maps, compare skew distribution of TCO (blue) on benchmarks r3 with DME (red). Note that the scale for skew is non-uniform. (b) Initial clocktree (shown in black dash-line), and optimized clocktree (shown in red dot-line) after TCO of r3

- [2] M. Cho, S. Ahmed, and D. Z. Pan, "Taco: Temperature aware clocktree optimization," in *Proc. Intl. Conf. Computer Aided Design*, 2005.
- [3] J. Cong, A. B. Kahng, C.-K. Koh, and C.-W. A. "Tsao, Bounded-skew clock and Steiner routing," *ACM Trans. Design Automation of Electronics Systems*, 1997.
- [4] M. Edahiro, "Delay minimization for zero-skew routing," in *Proc. Intl. Conf. Computer-Aided Design*, pp. 563-566, 1993.
- [5] K. Banerjee, A. H. Ajami, and M. Pedram, "Analysis and optimization of thermal issues in high-performance vlsi," in *ISPD*, pp. 230-237, Apr. 2001.
- [6] A. B. Kahng and C.-W. A. Tsao, "More practical bounded-skew clock routing," in *Proc. Design Automation Conf.*, pp. 594-599, 1997.
- [7] D. S. Gao and D. Zhou, "Propagation delay in rlc interconnection networks," in *Proc. Int. Sym. On Circuits and Systems*, 1993.
- [8] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Berkeley Symposium on Mathematical Statistics and Probability*, 1967.
- [9] J. E. Gentle, *Singular Value Factorization*. Berlin: Springer-Verlag, 1998.
- [10] J. Richards, *Remote Sensing Digital Image Analysis*. New York: Springer-Verlag, 1993.
- [11] X. Li, P. Li, and L. Pileggi, "Parameterized interconnect order reduction for inter/intra-die variations," in *Proc. Intl. Conf. Computer-Aided Design*, 2005.
- [12] E.J.Grimme, *Krylov projection methods for model reduction (Ph. D Thesis)*. Univ. of Illinois at Urbana-Champaign, 1997.

**Table 1: Comparison between TCO and existing skew reduction techniques in our delay model**

Input(node)	wirelength(um)				w-skew (100-map)(ps)				runtime(s)			
	DME	TECO	PECO	TCO	DME	TECO	PECO	TCO	DME	TECO	PECO	TCO
r1(267)	1.32E+06	1.48E+06	1.37E+06	1.32E+06	144.2	132	107.3	62.2	0.5	1.8	1.5	1.2
r2(598)	2.56E+06	2.71E+06	2.61E+06	2.57E+06	535.3	338	162.8	132.9	1	9.8	7.9	4.3
r3(862)	3.38E+06	3.77E+06	3.44E+06	3.38E+06	587.5	327	200.2	163.2	1.4	24.5	19.1	11.7
r4(1903)	6.82E+06	6.99E+06	6.85E+06	6.83E+06	1422.2	999	138.2	283.5	2.1	88.5	73.8	48
r5(3101)	1.02E+07	1.22E+07	1.05E+07	1.03E+07	3521.3	911.6	2321	918.6	6.2	249.8	233.4	132.2