Buffered Clock Tree Sizing for Skew Minimization Under Power and Thermal Budgets

Krit Athikulwongse, Xin Zhao, and Sung Kyu Lim
School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, Georgia, USA
e-mail: {krit, xinzhao, limsk}@ece.gatech.edu

Abstract— **In this paper, we study the clock tree sizing problem for thermal-aware skew minimization under power and thermal budgets. Clock wire/buffer sizing affects not only the delay/skew, but also the power dissipation of the clock tree. This effect in turn triggers changes in thermal distribution, making re-computation of the delay/skew necessary. Thus, the interaction among skew, power, and temperature is highly complicated if tied with clock wire/buffer sizing. In order to efficiently combat the time-varying nature of underlying thermal profile, we focus on two kinds of skew, depending on the number of thermal profiles given: skew value and skew range. The former refers to the skew value computed under a single steady-state thermal profile, whereas the latter refers to the skew range computed based on multiple thermal profiles. Our thermal-aware sequential-linear-programming approach maintains near-zero skew value and narrow skew range while keeping the power dissipation and temperature under the given budgets.**

I. Introduction

Due to high switching activity and high capacitive load, clock network accounts for a significant fraction of the total power dissipation in synchronous VLSI systems. Low-power clock tree design has become a crucial step in determining the performance and reliability of VLSI design. Meanwhile, the non-uniform chip thermal distribution has been identified as one of the major sources of the performance and reliability degradation. This thermal variation induces a substantial amount of skew in the clock tree, which has adverse impacts on the performance and reliability. Thus, modern clock tree designs should consider the interaction of the power and thermal issues, and tackle both issues simultaneously.

Clock wire width sizing affects the delay (and thus skew) in a non-linear fashion due to the wire resistance and capacitance changes. Buffer sizing affects the driving resistance and input capacitance of the buffer, which in turn changes clock skew. This wire/buffer sizing also affects the power dissipation of the clock tree, and triggers change on thermal distribution, which in turn brings back the delay/skew changes. Thus, the interaction among skew, power, and temperature is highly complicated if tied with clock wire/buffer sizing. Therefore, wire/buffer sizing should well balance the skew minimization, temperature optimization, and low-power design requirements.

Several works have been proposed to address low-power clock tree optimization. In [10], the wire and buffer sizing was formulated as a sequential-linear-programming (SLP) problem so that the power was minimized under the general skew constraints. The authors in [7] solved the clock tree sizing problem with the consideration of process variations. They minimized the process-variation-aware skew under given power budget. The authors in [5] constructed a tree that balanced the skew under two given static thermal profiles (uniform and worst-case). Lastly, the authors in [11] built a thermal-aware clock tree by computing bottom-up merging points based on thermal sensitivity.

None of these works, however, considers the power and thermal issues simultaneously. In addition, clock tree construction and sizing are not done together. More importantly, most of these works assume that thermal profiles are not changed during optimization. However, the impact of sizing on thermal variations can not be ignored. Without considering the temperature changes caused by sizing, the expected minimized skew in clock tree optimization is not realistic any longer. The contributions of this paper are as follows:

- For the first time, we solve the buffered clock tree sizing problem for thermal-aware skew minimization under the power and thermal budgets. Our thermal-aware SLP (TA-SLP) algorithm addresses the impact of sizing on skew and thermal distribution and vice versa. We also present a speedup scheme named loosely thermal-aware SLP (LTA-SLP) that efficiently performs updates of the delay, thermal, and power sensitivity values to be used in the sequential linear programming (SLP) formulation.
- In order to efficiently combat the time-varying nature of underlying thermal profile, we focus on two kinds of skew, depending on the number of thermal profiles given: skew value and skew range. The former refers to the skew value computed under a single steady-state thermal profile, whereas the latter refers to the skew range computed based on multiple thermal profiles. Our TA-SLP approach maintains near-zero skew value and narrow skew range while keeping the power dissipation and temperature under the given budgets.
- Related experiments show that our approaches outperform a well-known existing work [7] by 70% and 64% in terms of the average and maximum skew, respectively, measured on multiple thermal profiles.

The remainder of this paper is organized as follows. Section II presents the problem formulation. Section III presents our thermal model. Section IV presents our TA-SLP-based clock tree sizing algorithm. Section V presents experimental results, and we conclude in Section VI.

II. Problem Formulation

The buffered clock tree sizing problem for thermal-aware skew minimization under the power and thermal budgets is defined as follows: The inputs include a non-sized buffered clock tree, underlying thermal profile (single or multiple) that does not include temperature contribution from clock tree,[1] and the power and thermal budgets. The output is a sized clock tree in which the wire widths and buffer sizes are determined. The objectives include (a) the thermal-aware clock skew (either skew value or skew range, depending on the number of profiles given) is minimized, (b) the total power dissipation is kept under the given budget, and (c) the temperature of all the thermal tiles are kept under the given budget.

III. Thermal-Aware Delay Model

A. Wire and Buffer Sizing Impact on Thermal-aware Delay

Copper resistivity ρ increases linearly with temperature, and can be simply described by $\rho(T) = \rho_0 + \alpha \cdot T$, where ρ_0 is the resistivity at $0\,^\circ C$, α is the temperature coefficient of resistivity, and T is the

This material is based upon the work supported by the National Science Foundation under CAREER Grant No. CCF-0546382, the Center for Circuit and System Solutions (C2S2), and the Interconnect Focus Center (IFC).

[1] More explanation on this point is provided in Section III-C.

978-1-4244-5767-0/10/$26.00 2010 IEEE

temperature in $^\circ C$. The equation can be rewritten as $\rho(T) = \rho_0 \cdot (1 + \beta_0 \cdot T)$, where $\beta_0 = \alpha/\rho_0$. Given a wire with length l, width w, and thickness t, its temperature-dependent resistance $R(T)$ can be expressed as

$$R(T) = \frac{\rho_0 \cdot l}{w \cdot t} \times (1 + \beta_0 \cdot T).$$

The wire capacitance is $C = c_0 lw$, where c_0 is the capacitance per area. For a buffer with gate width w_b, the driving resistance is $r_\mathrm{b}/w_\mathrm{b}$, and the input capacitance is $c_\mathrm{b} w_\mathrm{b}$, where r_b and c_b are unit-width driving resistance and input capacitance, respectively. In our model, the buffer driving resistance also varies linearly with temperature within the small region of temperature change.[2] As suggested in [2], we assume that the wire capacitance and buffer input capacitance do not change with temperature.

From the SPICE simulations on buffers with different sizes and temperatures, we observe that the buffer intrinsic delay, t_d, slightly increases with buffer size, and increases linearly with temperature. Given a buffer size S and temperature T, t_d can be modeled as

$$\begin{aligned} t_d(T,S) &= t_{d_0}(S) \times (1 + \beta_{in}(S) \cdot (T - T_0)), \\ \beta_{in}(S) &= \alpha - S \cdot \tau, \end{aligned}$$

where t_{d_0} is the buffer intrinsic delay at a reference temperature T_0, and β_{in} is the temperature coefficient of buffer intrinsic delay. For $10\times$ to $70\times$ minimum-size buffers at 60 to $110^\circ C$, with the constants $\alpha = 0.005$ and $\tau = 0.000013$, the mean absolute error of our model from SPICE results is within $0.04ps$.

B. Thermal Calculation

The chip temperature can be estimated by the compact substrate thermal model proposed in [8]. Referring to the electrical-thermal duality, the thermal resistance matrix $\mathbf{R}$ is introduced to calculate the temperature: $\mathbf{t} = \mathbf{R} \times \mathbf{p}$, where $\mathbf{t}$ and $\mathbf{p}$ are the temperature and power vectors. The chip is divided into $n \times n$ thermal tiles. Clock tree components belonging to the same thermal tile have uniform temperature. If the temperature of tile i is T_i, and power dissipation of tile j is p_j, then thermal resistance R_{ij} represents the temperature change in tile i caused by the power change in tile j: $R_{ij} = \partial T_i / \partial p_j$. In addition to substrate devices, clock buffers also contribute to the power dissipation of each tile, which is used to calculate temperature. Our algorithm can work at different sizes of thermal tiles. We use 20×20 thermal tiles in this work because it gives results comparable to 40×40 thermal tiles, and reduces thermal analysis time by $6\times$.

C. Multiple Thermal Profiles

Many existing works on thermal-aware physical design tackle a single "steady-state" thermal profile. However, thermal profiles may vary significantly, depending on the workload and switching activities of the underlying application. It is possible that an application generates numerous distinct thermal profiles during its execution. It is, therefore, computationally infeasible to construct layouts that can tolerate this time-varying thermal profile. In our work, we limit our scope to k-given thermal profiles, and size the wires and buffers so that the skew variation is minimized over these profiles while meeting the power and thermal budgets.

We use the following thermal profiles in this work: (1) **single uniform** thermal profile: the temperature at all thermal tiles is set to $80^\circ C$ as used in [5]. (2) **multiple non-uniform** thermal profiles: for each benchmark, we start with 10 different non-uniform "device power profiles", where the value at each tile corresponds to the power dissipated by the devices covered by the tile. The power values at the same tile in these profiles can vary as high as 10% from their average.[3] We then convert these power profiles into thermal profiles using a thermal analyzer [8]. Note that the power and temperature contribution from clock tree and buffers is not included in these profiles because these profiles represent the underlying power/thermal profiles, on top of which the clock tree is to be added. Our final reported chip temperature values do include both devices and clock wires/buffers. Section V-A provides details on these profiles. (3) **single average** thermal profile: this is the average thermal profile among the multiple non-uniform profiles mentioned above.

Our strategy is to optimize the clock trees under the average thermal profiles, and then evaluate them using multiple profiles. Related experiments show that this approach is very practical, and generates stable solutions. More details are provided in Section V-E.

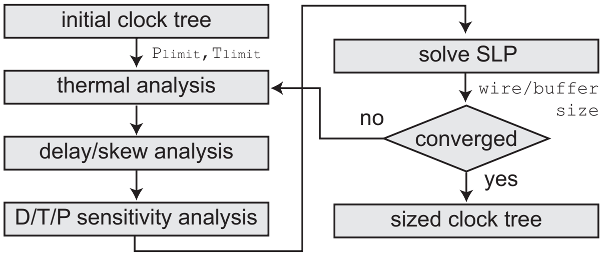

Fig. 1. Overview of our TA-SLP sizing algorithm.

IV. Clock Tree Sizing Algorithm

A. Overview of the Algorithm

Figure 1 shows an overview of our TA-SLP-based sizing algorithm. The input to our sizing algorithm is a zero-skew buffered clock tree generated by BST-DME [6] under a uniform thermal profile. The power and thermal budgets are also given. First, the thermal analyzer computes the temperature based on the current clock tree (= clock wires and clock buffers) plus the underlying device thermal profile. The delay and skew calculator computes the thermal-aware delay and skew values. Then, delay sensitivity matrix, thermal sensitivity matrix, and power sensitivity vector are computed. These matrices represent the variations of delay, temperature, and power corresponding to an attempted wire/buffer size change. Once these matrices and vector are computed, a sequential linear sub-problem is constructed. The SLP is formulated and solved to obtain new wire/buffer sizes that minimize the thermal-aware skew while meeting the power and thermal budgets. The algorithm terminates if either the current tree meets the thermal and power budgets, or the skew update is small enough; otherwise, TA-SLP is repeated with the updated skew, power, and thermal analysis results.

B. Sequential Linear Programming

To solve the non-linear sizing problem for thermal-aware clock skew minimization, SLP is used to iteratively linearize the original non-linear problem into linear sub-problems, and then solve the linear sub-problems using LP. In each iteration, the sub-problem is assumed to be linear within a certain range of variables. Choosing a proper linear range for each variable is the key to guarantee the

[2] Junction thermal variation is an important source of driving resistance variation. Transistor parameters including threshold voltage, mobility, and silicon energy band gap are sensitive to thermal variations [3]. Addressing this issue in thermal-aware clock tree sizing is outside the scope of this paper.

[3] Higher power variation in extreme cases, for example clock gating, is out of scope of this work. We target our work on normal chip operation.

TABLE I
NOTATIONS USED IN OUR TA-SLP FORMULATION.

symbol	meaning
T_{limit}	thermal budget
P_{limit}	power budget
d_i	delay of sink i
t_i	temperature of tile i
$\mathbf{t}$	vector of tile temperatures t_i
w_j	size of wire/buffer j
p	power dissipation
d_i^0, d_i^1	current and estimated new delays of sink i
t_i^0, t_i^1	current and estimated new temperatures of tile i
$\mathbf{d}^0$, $\mathbf{d}^1$	vectors of sink delays d_i^0 and d_i^1
$\mathbf{t}^0$, $\mathbf{t}^1$	vectors of tile temperatures t_i^0 and t_i^1
w_j^0, w_j^1	current and new sizes of wire/buffer j
$\mathbf{w}^0$, $\mathbf{w}^1$	vectors of wire/buffer sizes w_j^0 and w_j^1
p^0	current power dissipation
$\mathbf{G}$	delay sensitivity matrix
$\mathbf{\Gamma}$	thermal sensitivity matrix
$\boldsymbol{\beta}$	power sensitivity vector
δ_j	size change of wire/buffer j
$\boldsymbol{\delta}$	vector of wire/buffer size changes δ_j
ϵ_j	size perturbation for wire/buffer j
$\widehat{\mathbf{w}}_j^0$	vector $\mathbf{w}^0$ with one wire/buffer size w_j^0 perturbed
$\hat{\mathbf{t}}_j^0$	vector of tile temperatures after a w_j^0 perturbed

convergence of SLP and to obtain high quality solutions. In each iteration of SLP, the linear range of variables is determined by the quality of the previous solution. Depending on this linear range, the delay, power, and thermal sensitivity values are computed, and the sub-problem is constructed and solved. Our TA-SLP-based sizing algorithm effectively determines the size of wires and buffers so that the thermal-aware skew is minimized. The notations used in our TA-SLP formulation are shown in Table I.

Given the current wire/buffer size w_j^0, the linear sub-problem used in our TA-SLP is to decide the size change δ_j for each wire/buffer so as to minimize the thermal-aware skew under power and thermal budgets. The linear sub-problem of our TA-SLP is formulated as follows:

Minimize

$$d_{\max} - d_{\min} \tag{1}$$

Subject to

$$\begin{aligned}
d_i^1 &\geq d_{\min}, \quad \forall i \in \text{sinks} && (2)\\
d_i^1 &\leq d_{\max}, \quad \forall i \in \text{sinks} && (3)\\
\mathbf{d}^1 &= \mathbf{d}^0 + \mathbf{G}\cdot\boldsymbol{\delta} && (4)\\
\mathbf{t}^1 &= \mathbf{t}^0 + \mathbf{\Gamma}\cdot\boldsymbol{\delta} && (5)\\
t_i^1 &\leq T_{\mathrm{limit}}, \quad \forall i \in \text{tiles} && (6)\\
p^0 + \boldsymbol{\beta}^{\mathrm{T}}\cdot\boldsymbol{\delta} &\leq P_{\mathrm{limit}} && (7)
\end{aligned}$$

The delay and skew are affected by two factors. The first factor is the changes of resistance and capacitance from wire/buffer sizing. The second factor is the thermal variations caused by the changes of wire/buffer sizes. The vector $\boldsymbol{\delta} = [\delta_1\ \delta_2\ \cdots\ \delta_m]^{\mathrm{T}}$ contains the wire/buffer size changes, where m is the total number of wires/buffers. Equation (1) is the objective function, which aims to minimize the skew under the given thermal profile.

Constraints (2) and (3) are the minimum and maximum delay constraints for each sink delay d_i^1, which are used to estimate skew. Constraint (4) is used to estimate delay values using the delay sensitivity matrix. Here, $\mathbf{d}^0 = [d_1^0\ d_2^0\ \cdots\ d_n^0]^{\mathrm{T}}$ and $\mathbf{d}^1 = [d_1^1\ d_2^1\ \cdots\ d_n^1]^{\mathrm{T}}$ are the delay vectors before and after the sizing, where n is the total number of sinks, and $\mathbf{G}$ is the $n \times m$ delay sensitivity matrix. For each sink i, the delay should satisfy

$$d_i^1 = d_i^0 + \sum_{\forall j} G_{ij}\cdot\delta_j, \quad j \in \text{wires/buffers},$$

where G_{ij} is the delay sensitivity of sink i with respect to wire/buffer j. The newly estimated delay d_i^1 after sizing is the sum of the current delay value d_i^0 and the additional delay caused by the size changes from *all* of the wire/buffer δ_j. Section IV-C discusses how to compute G_{ij} in detail.

Constraint (5) is used to estimate temperature values using the thermal sensitivity matrix. Here, $\mathbf{t}^0 = [t_1^0\ t_2^0\ \cdots\ t_k^0]^{\mathrm{T}}$ and $\mathbf{t}^1 = [t_1^1\ t_2^1\ \cdots\ t_k^1]^{\mathrm{T}}$ are the temperature vectors before and after sizing, where k is the total number of the thermal tiles, and $\mathbf{\Gamma}$ is the $k \times m$ thermal sensitivity matrix. For each thermal tile i, the temperature should satisfy

$$t_i^1 = t_i^0 + \sum_{\forall j} \mathit{\Gamma}_{ij}\cdot\delta_j, \quad j \in \text{wires/buffers},$$

where $\mathit{\Gamma}_{ij}$ is the thermal sensitivity of tile i with respect to wire/buffer j. The size change from *all* of the wires/buffers contribute to the temperature change from t_i^0 to t_i^1. Constraint (6) guarantees that the temperature at all thermal tiles should be below the given thermal budget T_{limit}. Constraint (7) is the power constraint, which guarantees that the total power dissipation should not exceed the power budget P_{limit}. The power sensitivity vector $\boldsymbol{\beta} = [\beta_1\ \beta_2\ \cdots\ \beta_m]^{\mathrm{T}}$, where m is the total number of wires/buffers. Section IV-C discusses how to compute $\mathit{\Gamma}_{ij}$ and β_j in detail.

In each linear sub-problem, the variable δ_j for each wire/buffer is restricted within a small range, which provides the approximated linear range for the delay gradient. The new size after sizing, $w_j^1 = w_j^0 + \delta_j$, is restricted to

$$w_j^1 \in [\max(L_j, w_j^0 - \epsilon_j), \min(U_j, w_j^0 + \epsilon_j)],$$

where L_j and U_j are the lower and upper bound for the size of wire/buffer j. Here, ϵ_j is the size perturbation for each δ_j, which determines the solution space for each sizing iteration. We improve the adaptive confidence scheme used in [7] in several ways, making it suitable to determine ϵ_j for thermal skew optimization.

C. Sensitivity Computation

In each iteration of TA-SLP sizing, the linear sub-problem assumes that the delay, temperature, and power dissipation are linearly dependent on the wire/buffer size (w_j) within a certain solution range of variables (δ_j). Since wire/buffer sizing has a global impact on delay/skew, temperature, and power, the three sensitivity matrices/vector should be computed before solving the linear sub-problem in each iteration.

Delay sensitivity matrix is computed to account for the delay variations caused by sizing. As discussed above, the delay of each sink is sensitive not only to the size changes δ_j, but also to the thermal profile. For a given wire/buffer j, if its size were to change from w_j^0 to $w_j^0 + \epsilon_j$, the thermal profile over all the thermal tiles would change from $\mathbf{t}^0$ to $\hat{\mathbf{t}}_j^0 = \mathbf{t}(\widehat{\mathbf{w}}_j^0)$, where $\widehat{\mathbf{w}}_j^0 = [w_1^0\, w_2^0 \cdots w_j^0 + \epsilon_j \cdots w_m^0]^{\mathrm{T}}$, and m is the total number of wires/buffers. Sizing wire/buffer j may change the delay of the sinks. The new delay for sink i after sizing wire/buffer j is determined by both the new thermal profile $\hat{\mathbf{t}}_j^0$ and size changes ϵ_j of wire/buffer j. The sink i delay sensitivity to wire/buffer j, denoted G_{ij}, is approximated as

$$G_{ij} = \left.\frac{\partial d_i}{\partial w_j}\right|_{w_j = w_j^0} \approx [d_i(\hat{\mathbf{t}}_j^0, \widehat{\mathbf{w}}_j^0) - d_i(\mathbf{t}^0, \mathbf{w}^0)]/\epsilon_j,$$

where $d_i(\mathbf{t}^0, \mathbf{w}^0)$ is the delay of sink i when wire/buffer sizes are set to $\mathbf{w}^0$, and the thermal profile is $\mathbf{t}^0$. Here, $d_i(\hat{\mathbf{t}}_j^0, \widehat{\mathbf{w}}_j^0)$ is the delay of sink i when wire/buffer j changes its size to $w_j^0 + \epsilon_j$, and the thermal profile is changed to $\hat{\mathbf{t}}_j^0$. It is important to note that, while constructing the $\mathbf{G}$ matrix, we consider not only the delay variations caused by RC parameter perturbations, but also the impact of thermal variations.

Thermal sensitivity is defined as the average thermal variation of thermal tiles caused by the size changes on wires/buffers. Temperature of thermal tiles are sensitive to wire/buffer sizes. For a thermal tile i with temperature t_i, if wire/buffer j changes its size from w_j^0 to $w_j^0+\epsilon_j$, the thermal sensitivity of this tile i with respect to wire/buffer j, denoted $\mathit{\Gamma}_{ij}$, is approximated as

$$\mathit{\Gamma}_{ij} = \left.\frac{\partial t_i}{\partial w_j}\right|_{w_j=w_j^0} \approx [t_i(\widehat{\mathbf{w}}_j^0) - t_i(\mathbf{w^0})]/\epsilon_j,$$

where $t_i(\mathbf{w}^0)$ is the temperature of tile i when wire/buffer sizes are set to $\mathbf{w}^0$, and $t_i(\widehat{\mathbf{w}}_j^0)$ is the temperature of tile i when wire/buffer j changes its size to $w_j^0 + \epsilon_j$.

Power sensitivity vector $\boldsymbol{\beta}$ is the average perturbation of the total power dissipation based on sizing. Once the size of wire/buffer j changes from w_j^0 to $w_j^0 + \epsilon_j$, the power sensitivity to wire/buffer j, denoted β_j, is defined as

$$\beta_j = \left.\frac{\partial p}{\partial w_j}\right|_{w_j=w_j^0} \approx [p(\widehat{\mathbf{w}}_j^0) - p(\mathbf{w}^0)]/\epsilon_j,$$

where $p(\mathbf{w}^0)$ is the power dissipation when wire/buffer sizes are set to $\mathbf{w}^0$, and $p(\widehat{\mathbf{w}}_j^0)$ is the power dissipation when wire/buffer j changes its size to $w_j^0 + \epsilon_j$.

If there are n sinks, m wires/buffers, and k thermal tiles, the size of $\mathbf{G}$ matrix is $n \times m$, $\mathbf{\Gamma}$ matrix is $k \times m$, and $\boldsymbol{\beta}$ is $m \times 1$. Realizing the facts that (i) sizing causes thermal variations, and (ii) delay is affected by both thermal variations and sizing, TA-SLP needs to know not only the thermal profile corresponding to the current wire/buffer sizes, but also the thermal variations caused by sizing. When constructing the $\mathbf{G}$ matrix in each sizing iteration, the power analyzer, thermal analyzer, and delay calculator are called m times, where m is the number of wires/buffers to be sized. During each call, the delay calculator computes the delay along *all* wires/buffers. Therefore, the complexity of the sensitivity computation is approximately O(m^2). Section IV-D discusses how to overcome this runtime overhead.

D. Runtime Improvement

Related experiment shown in Section V-F indicates that the runtime per iteration of our TA-SLP increases considerably when the thermal effect of sizing is considered. More specifically, the inclusion of thermal variation, which is caused by sizing when computing each G_{ij} as described in Section IV-C, makes the $\mathbf{G}$ matrix of TA-SLP much denser than the $\mathbf{G}$ matrix without the thermal effect of sizing. For benchmark r5, on average, the $\mathbf{G}$ matrix with and without the thermal effect of sizing has 23.6 and 2.0 million non-zero elements, respectively. Therefore, TA-SLP spends most of the time solving LP sub-problems.

Because the dense $\mathbf{G}$ matrix in TA-SLP causes long solving time for each linear sub-problem, making it sparser should reduce runtime of TA-SLP. To accomplish this goal, we replace the term $d_i(\hat{\mathbf{t}}_j^0, \widehat{\mathbf{w}}_j^0)$ with $d_i(\mathbf{t}^0, \widehat{\mathbf{w}}_j^0)$ when computing each G_{ij}, where d_i is the delay of sink i, $\mathbf{t}^0$ is the thermal profile before sizing in the *current* iteration, $\hat{\mathbf{t}}_j^0$ is the thermal profile after a wire/buffer j is perturbed, and $\widehat{\mathbf{w}}_j^0 = [w_1^0 \; w_2^0 \; \cdots \; w_j^0 + \epsilon_j \; \cdots \; w_m^0]^{\mathrm{T}}$ is the vector of current wire/buffer sizes with *only one* wire/buffer size w_j^0 perturbed. In other words, we use the thermal profile from the last iteration (= before sizing) instead of perturbed one when computing delay values. We observe that this scheme does not affect the quality of the solution too much because the temperature change is usually minor during late TA-SLP iterations. In addition, we still update $\mathbf{t}^0$ every iteration, so the delay values are computed based on relatively recent temperature. However, this scheme, named loosely thermal-aware SLP (LTA-SLP), improves runtime significantly while maintaining the solution quality. Section V-F presents the related experimental results.

TABLE II
IMPACT OF CLOCK WIRE/BUFFER SIZING ON POWER AND TEMPERATURE.

ckt	Min-size (0.24 μm, 12 ×)		Mid-size (0.6 μm, 38 ×)		Max-size (0.96 μm, 64 ×)	
	P_{clk} (W)	$T_{\max}$ (°C)	P_{clk} (W)	$T_{\max}$ (°C)	P_{clk} (W)	$T_{\max}$ (°C)
r1	0.46	67.92	1.01	72.55	1.63	78.08
r2	0.99	69.52	2.14	74.55	3.53	81.12
r3	1.38	70.47	3.05	75.06	4.95	81.01
r4	3.04	72.74	6.47	78.28	10.65	85.14
r5	4.73	74.28	10.10	79.55	16.68	87.21

TABLE V
IMPACT OF THE NUMBER OF THERMAL PROFILES ON SKEW RANGE (BASED ON TA-SLP WITH MID-SIZE INITIAL SOLUTION).

ckt	10 profiles		100 profiles		1000 profiles	
	Ave.	Min./Max.	Ave.	Min./Max.	Ave.	Min./Max.
r1	0.26	0.12/0.40	0.24	0.09/0.53	0.23	0.04/0.62
r2	0.94	0.29/1.47	0.90	0.23/2.21	0.88	0.18/2.46
r3	1.21	0.47/1.99	1.17	0.47/2.80	1.14	0.29/3.44
r4	3.05	1.63/4.82	3.11	1.12/7.82	3.08	0.79/8.67
r5	4.34	2.43/6.36	4.46	1.62/9.28	4.36	1.02/10.50

V. Experimental Results

A. Experimental Setting

We implemented our sizing algorithm in C++/STL, and ran on Linux. The chip thermal profile is estimated by using a compact substrate thermal model [8]. We use Elmore delay model in this work because of its efficient computation and result fidelity. Other delay models can also be used without affecting our algorithm. The linear sub-problems in sequential linear programming (SLP) are solved by using MOSEK optimization solver. The IBM benchmarks for clock tree synthesis and routing, r1 to r5, are used [9].

We use parameters for 65-nm technology. The minimum wire width, wire thickness, and inter-level dielectric constant for Metal 6 are $0.24\mu m$, $0.43\mu m$, and 2.9, respectively [1]. We use PTM [4] to transform these values into the unit length resistance and unit length capacitance at $100°C$. The unit length resistance and capacitance at $0°C$ are $0.15\Omega/\mu m$ and $0.2fF/\mu m$, respectively. The driving resistance and input capacitance of the minimum-size buffer are $4.7k\Omega$ and $0.47fF$, respectively. The maximum output load capacitance of each buffer is set to $250fF$. The supply voltage is $1.2V$, and the clock frequency is $5GHz$. The skew values are reported in pico-seconds, clock source-to-sink delay in nano-seconds, power in Watts, and temperature in degree-Celsius.

We use three different initial wire widths and buffer sizes for SLP-based sizing: min-size, mid-size, and max-size. The wires are restricted to $1\times$ to $4\times$ minimum width, and the buffers are restricted to $12\times$ to $64\times$ minimum size as used in [7].

The 10 thermal profiles are converted from 10 device power profiles of total power dissipation similar to the average thermal profile.

TABLE III

IMPACT OF INITIAL SIZING SOLUTION ON THE FINAL OPTIMIZED SKEW VALUE/RANGE.

ckt	Skew value based on average thermal profile (ps)			Skew range based on 10 thermal profiles (ps)								
				Min-size			Mid-size			Max-size		
	Min-size	Mid-size	Max-size	Ave.	Min.	Max.	Ave.	Min.	Max.	Ave.	Min.	Max.
r1	0.00	0.00	0.77	0.36	0.10	0.57	0.26	0.12	0.40	1.08	0.88	1.24
r2	0.00	0.00	0.44	0.96	0.41	1.53	0.94	0.29	1.47	1.41	0.77	2.11
r3	0.01	0.00	0.31	1.79	0.76	3.03	1.21	0.47	1.99	1.51	0.77	2.12
r4	0.01	0.01	0.42	3.67	2.03	6.43	3.05	1.63	4.82	3.12	2.02	4.87
r5	0.01	0.01	0.44	6.31	3.99	9.46	4.34	2.43	6.36	5.17	3.15	8.12

TABLE IV

COMPARISON BETWEEN NON-THERMAL-AWARE SLP AND THERMAL-AWARE SLP BASED ON MID-SIZE INITIAL SOLUTION. WE REPORT THE SKEW RANGE, MAXIMUM TEMPERATURE, AVERAGE POWER, AND AVERAGE DELAY RESULTS BASED ON 10 DIFFERENT THERMAL PROFILES.

ckt	Budgets		Non-thermal-aware SLP (NTA-SLP)						Thermal-aware SLP (TA-SLP)						skew reduction (% improve)	
	P_{limit}	T_{limit}	Skew range (ps)			Ave. P_{total}	Ave. $T_{\max}$	Ave. Dly	Skew range (ps)			Ave. P_{total}	Ave. $T_{\max}$	Ave. Dly		
	(W)	(°C)	Ave.	Min.	Max.	(W)	(°C)	(ns)	Ave.	Min.	Max.	(W)	(°C)	(ns)	Ave.	Max.
r1	2.87	74.18	1.12	0.88	1.29	2.86	71.16	2.86	0.26	0.12	0.40	2.86	71.12	2.79	77.00	69.37
r2	6.07	77.07	3.51	2.77	4.52	6.05	72.98	4.47	0.94	0.29	1.47	6.04	72.91	4.45	73.18	67.42
r3	8.65	76.96	4.03	2.89	5.73	8.60	73.62	4.61	1.21	0.47	1.99	8.59	73.60	4.44	69.98	65.23
r4	18.32	80.88	10.21	7.57	12.94	18.25	76.59	6.31	3.05	1.63	4.82	18.23	76.59	7.25	70.08	62.71
r5	28.61	82.85	10.34	7.25	13.19	28.46	78.00	6.41	4.34	2.43	6.36	28.52	78.03	7.04	57.99	51.77
Average improvement (w.r.t. NTA-SLP)															69.65	63.30

TABLE VI

RUNTIME COMPARISON AMONG NTA-SLP, TA-SLP, AND LTA-SLP (= SPEEDUP VERSION OF TA-SLP).

ckt	Non-thermal-aware SLP (NTA-SLP)					Thermal-aware SLP (TA-SLP)							
								original TA-SLP			faster LTA-SLP		
	# var	# con	CPU	# itr	CPU/itr	# var	# con	CPU	# itr	CPU/itr	CPU	# itr	CPU/itr
r1	920	802	24	12	2.00	1,320	1,602	144	12	12.00	194	17	11.41
r2	2,031	1,795	105	12	8.75	2,431	2,595	569	13	43.77	586	20	29.30
r3	2,907	2,587	257	14	18.36	3,307	3,387	1,260	15	84.00	745	16	46.56
r4	6,358	5,710	1,138	15	75.87	6,758	6,510	8,999	13	692.23	1,959	14	139.93
r5	10,291	9,304	2,768	15	184.53	10,691	10,104	32,858	15	2,190.53	4,685	16	292.81

TABLE VII

IMPACT OF OUR SPEEDUP SCHEME LTA-SLP. THE % IMPROVEMENT IS WITH RESPECT TO THE NTA-SLP RESULT SHOWN IN TABLE IV.

ckt	Loosely thermal-aware SLP (LTA-SLP)						skew reduction % improve	
	Skew range (ps)			Ave. P_{total}	Ave. $T_{\max}$	Ave. Dly		
	Ave.	Min.	Max.	(W)	(°C)	(ns)	Ave.	Max.
r1	0.25	0.12	0.39	2.86	71.18	2.74	77.55	69.60
r2	0.94	0.29	1.49	6.04	72.91	4.51	73.18	67.12
r3	1.15	0.44	1.89	8.61	73.70	4.25	71.51	67.09
r4	2.91	1.56	4.55	18.26	76.61	6.39	71.46	64.85
r5	4.04	2.29	5.89	28.48	77.98	6.33	60.90	55.36
Average improvement (w.r.t. NTA-SLP)							70.92	64.80

These thermal profiles, however, have different thermal statistics, e.g., number of hotspots and tile temperature difference.

B. Impact on Power and Temperature

We show how the sizing affects the power dissipated by the clock tree/buffers (= P_{clk}) as well as the thermal distribution of the entire layout (= $T_{\max}$). Table II shows these power and temperature values based on three different sizes: min-size, mid-size, and max-size.[4] We observe that when the size of clock trees increases from min-size to max-size, clock power dissipation increases by 4 times, and temperature increases by 10 to $13°C$. This result clearly demonstrates the large impact of sizing on power and thermal variation.

C. Impact of Initial Sizing Solution

The impact of initial sizing solution (= initial wire width and initial buffer size used in SLP) on optimized skew is shown in Table III. The min-size solution uses $0.24\mu m$ for the initial wire width, and $12\times$ minimum for the initial buffer size. The mid-size initial solution uses $0.6\mu m$ and $38\times$, while the max-size initial solution uses $0.96\mu m$ and $64\times$. These initial sizes are then optimized with our TA-SLP-based sizing algorithm. We report two types of skew values: (1) the single skew value computed based on the average thermal profile, and (2) the skew range computed based on 10 thermal profiles as mentioned in Section III-C. We first observe that our TA-SLP algorithm reduces the thermal skew value to near zero for the average thermal profile regardless of the initial solution. When multiple thermal profiles are applied, we observe that the mid-size initial solutions result in slightly better skew ranges compared to the max-size. The final skew range for the min-size initial solution, however, is the worst because most wires/buffers in the final tree are narrow/small, making the tree more sensitive to thermal variation. Thus, our choice for the initial solution is the mid-size.

D. Comparison with Existing Works

In this section, we show sizing results for the following three approaches: (1) Pre-SLP: the initial buffered zero-skew tree generated from DME [6] under the uniform thermal profile. The wires and

[4] Note that the size of all clock wires and buffers is uniformly set to minimum, medium, or maximum.

buffers sizes are initially set to either the minimum, medium, or maximum values. (2) NTA-SLP: in this non-thermal-aware SLP-based sizing, we perform skew minimization under only power budget, which resembles an existing work presented in [7].[5] (3) TA-SLP: our thermal-aware SLP-based sizing.

We optimize clock trees based on the average thermal profile for NTA-SLP and TA-SLP, and evaluate the skew results based on 10 different thermal profiles. Detailed comparison between NTA-SLP and TA-SLP under the same power and thermal budgets is shown in Table IV. We set P_{limit} to 85% of the power when mid-size clock tree is used. We set T_{limit} to 95% of the temperature when max-size clock tree is used. We do not set themal budget too tight because NTA-SLP does not consider thermal during optimization. We report the skew range, average temperature, power, and delay results based on 10 different thermal profiles. The skew results for NTA-SLP are computed based on non-uniform thermal profiles. We observe from the skew improvement columns that our TA-SLP consistently outperforms NTA-SLP by 70% for the average skew and 63% for the maximum skew values. In addition, we note that the final skew range is narrow, e.g., $[2.43ps, 6.36ps]$ for r5 with our TA-SLP. This range becomes even smaller for smaller circuits. This small skew range indicates that our TA-SLP-based sizing methods are effective in making clock trees more thermal-variation-tolerant among multiple thermal profiles than NTA-SLP [7] while meeting the same given power and thermal budgets.

E. Impact of the Number of Thermal Profiles

We study the impact of the number of thermal profiles used to evaluate the skew range. Table V shows the final skew range of our TA-SLP based on 10, 100, and 1000 thermal profiles. We observe that the average skew value stays more or less the same while the skew range becomes larger as the number of profiles used increases. This observation agrees with the general statistical behavior that the probability of having bad skews increases as the number of samples used increases. The average value, however, tends to become stable as the number of samples used increases.[6]

F. Runtime Enhancement

Table VI shows the runtime for three algorithms: NTA-SLP, TA-SLP, and LTA-SLP. The last algorithm, named loosely-thermal-aware (LTA) SLP, refers to our runtime enhancement of TA-SLP discussed in Section IV-D. First, we see that the number of iterations for both NTA-SLP and TA-SLP are comparable. However, the runtime per iteration for our TA-SLP is about 6–12 times longer than that for NTA-SLP. Because the inclusion of thermal variation caused by sizing in computing G_{ij} causes the $\mathbf{G}$ matrix of our TA-SLP to be much denser than that of NTA-SLP as discussed in Section IV-D, MOSEK (our LP solver) spends most of the time solving linear sub-problems.

The runtime of our LTA-SLP is also shown in Table VI. Note that the number of variables and constraints between TA-SLP and LTA-SLP are identical. In addition, the number of iterations is comparable. The runtime per iteration for LTA-SLP is, however, much shorter than that for TA-SLP because of its sparse $\mathbf{G}$ matrix. Now the runtime per iteration between NTA-SLP and LTA-SLP is comparable. Moreover, the runtime difference decreases as the circuit size increases, making the runtime overhead for large circuits less significant. Table VII shows the impact of our speedup scheme on skew, power, temperature, and delay. Based on the comparison between Table IV and Table VII, we observe that the skew results in fact improved slightly while the power, temperature, and delay results are comparable. This result does not suggest that LTA-SLP outperforms TA-SLP in term of solution quality. The non-linear nature of the problem makes both TA-SLP and LTA-SLP give local optimal solution. The change of $\mathbf{G}$ matrix causes MOSEK to give different local optimal solution for LTA-SLP, which, in some cases, is better than the solution MOSEK gives for TA-SLP.

VI. Conclusions

In this paper, we solved the buffered clock tree sizing problem for thermal-aware skew minimization under the power and thermal budgets. Our thermal-aware sequential-linear-programming based sizing algorithm addresses the impact of sizing on thermal distribution and vice versa. We showed how to efficiently update the delay, thermal, and power sensitivity values to be used in the iterative SLP loops, which is the key to the success of our SLP method. Our algorithm generates near-zero-skew clock tree while keeping the power dissipation and temperature under the given budgets. In addition, we obtained very narrow skew range measured under multiple thermal profiles.

References

[1] P. Bai et al. A 65nm logic technology featuring 35nm gate lengths, enhanced channel strain, 8 Cu interconnect layers, low-k ILD and 0.57 μm^2 SRAM cell. In *IEEE Int. Electron Devices Meeting Tech. Dig.*, pages 657–660, San Francisco, CA, Dec. 13–15 2004.

[2] K. Banerjee, M. Pedram, and A. H. Ajami. Analysis and optimization of thermal issues in high-performance VLSI. In *Proc. Int. Symp. on Physical Design*, pages 230–237, Sonoma, CA, Apr. 1–4 2001.

[3] S. A. Bota, J. L. Rosselló, C. de Benito, A. Keshavarzi, and J. Segura. Impact of thermal gradients on clock skew and testing. *IEEE Design and Test of Computers*, 23(5):414–424, May 2006.

[4] Y. Cao, T. Sato, M. Orshansky, D. Sylvester, and C. Hu. New paradigm of predictive MOSFET and interconnect modeling for early circuit simulation. In *Proc. IEEE Custom Integrated Circuits Conf.*, pages 201–204, Orlando, FL, May 21–24 2000.

[5] M. Cho, S. Ahmed, and D. Z. Pan. TACO: Temperature aware clock-tree optimization. In *Proc. IEEE Int. Conf. on Computer-Aided Design*, pages 581–586, San Jose, CA, Nov. 6–10 2005.

[6] J. Cong, A. B. Kahng, C.-K. Koh, and C.-W. A. Tsao. Bounded-skew clock and Steiner routing. *ACM Trans. on Design Automation of Electronic Systems*, 3(3):341–388, July 1998.

[7] M. R. Guthaus, D. Sylvester, and R. B. Brown. Clock buffer and wire sizing using sequential programming. In *Proc. ACM Design Automation Conf.*, pages 1041–1046, San Francisco, CA, July 24–28 2006.

[8] C.-H. Tsai and S.-M. Kang. Cell-level placement for improving substrate thermal distribution. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 19(2):253–266, Feb. 2000.

[9] R.-S. Tsay. An exact zero-skew clock routing algorithm. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 12(2):242–249, Feb. 1993.

[10] K. Wang, Y. Ran, H. Jiang, and M. Marek-Sadowska. General skew constrained clock network sizing based on sequential linear programming. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 24(5):773–782, May 2005.

[11] H. Yu, Y. Hu, C. Liu, and L. He. Minimal skew clock embedding considering time variant temperature gradient. In *Proc. Int. Symp. on Physical Design*, pages 173–180, Austin, TX, Mar. 18–21 2007.

[5] We do not compare with a thermal-aware clock tree synthesis work in [5] since it performs routing topology optimization instead of sizing.

[6] Our current approach of handling multiple thermal profiles (= minimize the skew based on the average profile) can be extended by adopting various statistical methods. This is one of our future works.