SACTA: A Self-Adjusting Clock Tree Architecture for Adapting to Thermal-Induced Delay Variation

Jieyi Long, *Student Member, IEEE*, Ja Chun Ku, Seda Ogrenci Memik, *Senior Member, IEEE*, and Yehea Ismail, *Member, IEEE*

***Abstract*—Aggressive technology scaling down and low-power design techniques lead to uneven distributed power density, which translates into heat flow in the chips, causing significant temperature variations in both spatial and temporal terms. In order to mitigate the negative impacts of temperature variations on circuit timing, we propose SACTA, a self-adjusting clock tree architecture, which performs temperature-dependent dynamic clock skew scheduling to prevent timing violations in a pipelined circuit. The dynamic and adaptive features of SACTA are enabled by our proposed automatic temperature-adjustable skew buffers and temperature-insensitive skew buffers. These special delay elements are carefully tuned to ensure resilience of the entire circuit against temperature variation. To determine their configurations, we proposed an efficient and general clock tree design and optimization framework. Furthermore, we show that SACTA is applicable across a wide spectrum of circuits, including multi-$V_{\mathrm{dd}}/V_{\mathrm{th}}$ designs. Experimental results show that a pipeline supported by SACTA is able to prevent thermal-induced timing violations within a significantly larger range of operating temperatures (on average, the violation-free range can be enhanced by over 15 °C).**

***Index Terms*—Clock tree architecture, design automation, self-adjusting, variation-tolerance.**

I. Introduction

POWER density of modern VLSI designs increases rapidly as technology advances, creating severe performance and reliability threats. Although supply voltage levels and effective switched capacitance values are decreasing with scaling, the rate of increase in the total number of devices on a chip largely surpasses this. As a result, power consumption is increasing at a higher rate than chip performance. Increasing leakage power due to scaling is one important factor contributing to this phenomenon. New packaging and cooling technologies are being developed to control high operating temperatures. However, the high cost of such enhancements often counteracts against other design concerns. Also, sophisticated cooling solutions may not be feasible for systems with stringent size constraints. Consequently, on-chip temperatures are rising steadily as technology is scaling down.

Thermal effects need to be taken into account at the design time due to their impact on various design metrics, such as leakage power, device lifetime, and circuit timing. Temperature variation affects timing in multiple ways. Interconnect resistance and cell delay are both dependent on temperature. Furthermore, thermal gradients on power lines create voltage drops leading to supply voltage fluctuations. This may cause significant changes in switching speed of gates and result in timing violations.

Making matters worse, environmental factors may lead to drastic variations of chip temperatures. Power management techniques such as clock gating, voltage islands, and power gating contribute to significant temporal and spatial variation. In addition, a chip may be deployed in diverse environments where the level of cooling support as well as the nominal temperature could not be accurately predicted at the design time. For example, an embedded processor chip could be part of different systems with different cost and size constraints leading to vastly different cooling available in the field. Finally, power consumption of a chip, therefore, temperature, can be highly input dependent, leading to variations in chip temperature throughout the execution of an application.

The combined effects of all the aforementioned phenomena further amplify the challenges of chip design considering thermal effects. The idea of designing circuits with guaranteed performance bounds while exhibiting resilience against environmental variations arises as an attractive option.

In this paper, we introduce SACTA, a Self-Adjusting Clock Tree Architecture, to address this problem. SACTA guarantees correct timing behavior in pipelined circuits within a large range of thermal conditions through a self-adjusting, temperature-sensitive skew distribution mechanism. SACTA exploits clock skews to "steal" time from adjacent pipeline stages to maintain the performance in the presence of delay variations within pipeline stages. Adaptability is achieved by utilizing a set of special skew buffers. These elements are designed to exhibit carefully tuned, temperature-dependent delay behavior in synchronization with the temperature levels prevalent in the logic of the pipeline. Thereby, they generate a self-adjusting skew tailored to the temperature-dependent timing behavior of each pipeline stage. We also developed a systematic design method to determine the physical specifications of these skew buffers (i.e., parameters that define their delay behavior) for a given pipelined circuit and a range of operating temperatures.

Existing techniques focus on the adverse effects of temperature on the delay of the clock tree and try to avoid thermal-induced timing violations by constructing zero skew clock trees [1]–[3]. They ignore the possibility that the combinational circuits may exhibit temperature-dependent delay violations, which may lead to failure even if a zero-skew

Manuscript received July 31, 2008; revised January 28, 2009 and March 27, 2009.

J. Long, S. Ogrenci Memik, and Y. Ismail are with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60208 USA (e-mail: jlo198@eecs.northwestern.edu; jieyi-long@northwestern.edu; seda@eecs.northwestern.edu; ismail@eecs.northwestern.edu).

J. C. Ku is with Samsung Electronics Co., Ltd., Kyungki-Do, 446-711 Korea (e-mail: jachun.ku@gmail.com).

Digital Object Identifier 10.1109/TVLSI.2009.2023992

clock is provided. Other techniques try to manage clock skew under thermal variations; however, they sacrifice performance to achieve immunity against variations [4]. Finally, dynamic adjustment techniques for microprocessor pipelines have been proposed, which incur significant overheads to enable timing violation detection and correction [5]. Our approach aims to provide an extremely low-cost solution that can effectively avoid thermal-induced delay violations considering the impact of temperature on both the clock tree and the datapath. Furthermore, we aim to preserve the target clock frequency at the same time.

We have evaluated the effectiveness of SACTA using a set of synchronous pipelined circuits. SACTA is able to prevent thermal-induced timing violations within a significantly larger range of operating temperatures (the violation-free range can by enhanced by over 15 °C). SACTA effectively enhances reliability while introducing negligible area and power overheads. The average area and power overheads are 1.53% and 0.92%, respectively. Even in the worst case, area and power overhead of SACTA are only 5.09% and 3.30%.

The remainder of this paper is organized as follows. Section II provides an overview of related work. In Section III, our model of temperature-dependent delay variation is presented, followed by the detailed discussion of the self-adjusting clock tree architecture. Design of our automatic temperature-adjustable (ATA)/insensitive (fixed) skew buffers is described in Section IV. We introduce our clock tree design framework in Section V. We extend our techniques to handle multi-$V_{\rm dd}/V_{\rm th}$ designs in Section VI. Our experimental evaluation is presented in Section VII. We conclude with a summary of our contributions and findings in Section VIII.

II. Related Work

The increasing impact of temperature variation on circuit timing, particularly clocked networks, has motivated several novel techniques. Some effort has been devoted to developing temperature-insensitive zero/bounded skew clock trees [1], [2]. The dual-supply-voltage clock tree [1] utilizes the fact that CMOS gate delay is insensitive to temperature at a specific supply voltage level $V_{\rm ZTC}$. By supplying the on-tree buffers with this voltage, the clock tree is made to have zero skew for any given thermal profile. It might at first appear as a viable idea to supply the entire design with this voltage level to achieve a system with absolute temperature independence. However, on one hand, technology trends, particularly the relationship between supply level and the necessary threshold voltage levels for transistors, indicate that supply levels are unlikely to scale beyond a certain level [6], In fact, from the 130-nm technology forward, $V_{\rm dd}$ has been scaling slowly, if at all [6]. On the other hand, it is shown that $V_{\rm ZTC}$ of a gate depends not only on technology but also on the operation conditions, especially the slope of the input signal [7]. Therefore, a circuit cannot be made to be temperature-insensitive by adjusting its supply voltage.

Another idea is to insert tunable delay buffers into the clock tree, which can be adjusted on-the-fly [2]. Buffer settings that guarantee bounded maximal skew for typical thermal profiles are precalculated and stored in an on-chip lookup table. During normal operation the actual thermal profile is captured by on-chip sensors, and a central control unit will reconfigure the clock tree to the appropriate tuning configuration.

Although these techniques can effectively eliminate delay fluctuations on the clock tree, *considering the clock tree alone is insufficient*. In a circuit where a zero-skew tree is used, to achieve maximal performance, $T_{\rm cp}$ is roughly set to the delay of the critical path. Since the delay of the critical path has positive dependence on temperature, as temperature increases, the delay of the critical path may exceed $T_{\rm cp}$ at a certain point. A zero-skew design will fail beyond this temperature.

Instead of trying to reduce thermal-induced clock skew, an alternative method is to use clock skew for enhancing system immunity to temperature variation. For example, a robust integer linear programming formulation for solving the clock skew scheduling problem in the presence of process and environment variations has been proposed [4]. However, improvement in reliability was achieved at the expense of system performance, making this approach less attractive to high-performance circuit designers.

Ernst *et al.* proposed a circuit-level timing error detection/correction scheme [5] called Razor. One technique employed here is called *dynamic retiming*. The idea is to create intentional clock skews such that those pipeline stages that require longer execution times are assigned longer intervals. The skews can be changed dynamically if the execution time of the pipeline stage changes due to environmental fluctuations. However, compared to our scheme, Razor requires significantly more hardware resources (such as shadow registers for error detection and a central control unit for error rate monitoring and skew creation) and is able to provide only a discrete set of skew values (versus the continuous scale generated by our scheme).

Finally, a large body of work addressed static clock skew scheduling for pipelined circuits [8]–[10]. These techniques do not attempt to solve the problem of delay variability under dynamic environmental conditions. Hence, our problem is fundamentally different

In our preliminary work, we have introduced SACTA, a self-adjusting pipeline architecture, to mitigate the negative impact of temperature variation to circuit timing [11]. In our preliminary work, it was assumed that all the gates in the circuit share the same threshold voltage and supply voltage. However, in most low-power designs, multi-$V_{\rm th}$ cell library and/or voltage island technique are/is used. For these multi-$V_{\rm th}/V_{\rm dd}$ designs, the dependency between the circuit delay and the temperature becomes more subtle [7]. In this paper, we extend our techniques to handle the multi-$V_{\rm th}/V_{\rm dd}$ designs. In addition, we present a more detailed discussion on the temperature-dependent circuit delay model and experimental results carried out on an extended set of benchmarks in the paper.

III. Self-Adjusting Clock Tree Architecture

Consider a local pipeline stage between two registers R_i and R_{i+1} as shown in Fig. 1. The following two constraints should be met to preserve correct circuit functionality [10]:

$$x_i + D_{i,i+1} \leq x_{i+1} + T_{\rm cp} \quad (1)$$

$$x_i + d_{i,+1} \geq x_{i+1}. \quad (2)$$

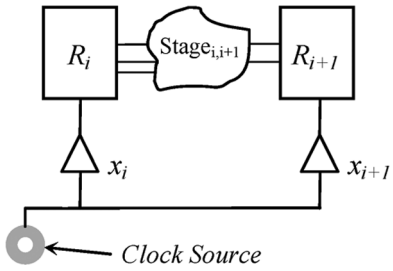


Fig. 1. Pipeline stage and associated clock signal.

Here x_i and x_{i+1} are defined as the clock signal delays from the clock source to the registers R_i and R_{i+1}, and $T_{\rm cp}$ is the clock period time. The difference between the arrival times of the clock signal at two successive registers is defined as the *clock skew*. This can be expressed as$(x_i - x_{i+1})$. The equations

$$D_{i,i+1} = T_{c-q} + T_{\rm logic}(\max) + T_{\rm setup}$$
$$d_{i,i+1} = T_{c-q} + T_{\rm logic}(\min) + T_{\rm hold}$$

signify the maximum and minimum delay between the two registers, respectively, where T_{c-q} is the clock-to-Q delay of register R_i, and $T_{\rm setup}/T_{\rm hold}$ are the setup/hold time of register $R_{\rm i}$. In other words, they denote the largest and shortest expected latencies for the pipeline stage located between these registers. In addition to constraints (1) and (2), in a fully synchronous system, the off-module skew is generally required to be zero. This means clock skew between the input registers and the output registers should be zero.

A *clock skew schedule* for a given pipeline is a set of delay values satisfying constraints (1) and (2). Existing techniques only address *static clock skew schedules*. However, these constraints (1) and (2) are in fact temperature-dependent. Therefore, a static clock skew schedule satisfying the constraints for some temperature profile may fail for some other profiles, even if for these profiles, a static clock skew schedule does exist. This has motivated us to develop a dynamic clock skew scheduling scheme and the self-adjusting clock tree architecture.

On circuit layout, the logic gates and the registers in the same pipeline stage are normally placed in close proximity, since this will help shorten the critical path and thereby enhance the system performance. This spatial correlation implies that during normal execution, the temperature profiles of the combinational logic and the associated pipeline registers are approximately same. We use $\theta_{i,i+1}$ to denote this *local temperature* for the pipeline stage between registers R_i and R_{i+1}. Then, the maximum and minimum stage latencies $D_{i,i+1}$ and $d_{i,i+1}$ can be expressed as functions of the local temperature as $D_{i,i+1}(\theta_{i,i+1})$ and $d_{i,i+1}(\theta_{i,i+1})$.

Our goal is to design a self-adjusting clock tree that is able to adapt to different thermal profiles. In other words, for a given thermal profile, if there exists a static clock skew schedule to guarantee correct operation, our clock tree should be able to configure itself to provide this clock skew. To solve this problem we need to first understand the relationship between temperature and delay for circuit elements. In the following we present the temperature-dependent delay model we have utilized to describe the behavior of logic.

A. Temperature-Dependent Delay Model and Its Validation

Circuit delay of a CMOS gate can be written as

$$\tau \propto \frac{CV_{\rm dd}}{I_d} \tag{3}$$

where C is the load capacitance driven by the gate, $V_{\rm dd}$ is the supply voltage (voltage swing), and I_d is the drain current of the transistors. The drain current stays mostly in the saturation region in deep submicron technologies due to velocity saturation [12]. Using the alpha-power law [12], the drain current in the saturation region is expressed as

$$I_d \propto \mu(\theta)\frac{W}{L}\left(V_{gs} - V_{\rm th}(\theta)\right)^{\alpha} \tag{4}$$

where θ denotes the gate temperature, μ is the carrier mobility, W and L are the channel width and length, respectively, V_{gs} is the gate-to-source voltage, $V_{\rm th}$ is the threshold voltage, and α is the velocity saturation index whose value is between 1 and 2 (closer to 1 in deep submicron technologies) [12].

Note that μ decreases as the temperature is raised. The dependency between μ and θ is usually modeled by

$$\mu(\theta) = \mu_0 \left(\frac{\theta}{\theta_0}\right)^m.$$

However, this relation does not hold true in saturation region due to velocity saturation [13]. The saturation velocity $v_{\rm sat}$ of the carrier, which is proportional to μ, decreases linearly with temperature. Hence, the temperature dependence of μ in the saturation region should be written as

$$\mu = \mu_0 - \eta(\theta - \theta_0) \tag{5}$$

where μ_0 is the mobility at the nominal temperature θ_0, which is typically 25 °C, and η is the temperature coefficient. $V_{\rm th}$ also decreases linearly as the temperature is raised, and is given by

$$V_{\rm th} = V_{\rm th0} - \kappa(\theta - \theta_0) \tag{6}$$

where $V_{\rm th0}$ is the threshold voltage at the nominal temperature, and κ is the temperature coefficient.

In deep submicron technologies, this temperature dependence of a CMOS gate delay can be simplified to a linear function of temperature [13]

$$\tau = \tau_0 + k(\theta - \theta_0) \tag{7}$$

where τ_0 is the delay at the nominal temperature.

Equation (7) can be produced by substituting (5) and (6) into (4), and taking a first-order approximation. This linear model was validated against HSPICE using a chain of inverters in the 65-nm PTM technology [14]. Fig. 2 illustrates the validation flow, and the result is shown in Fig. 3. It can be seen in Fig. 3 that there is an excellent agreement between the linear model and HSPICE. The maximum deviation of the measured delay values from the linear model is only 0.87% within a significant temperature range (from 25 °C to 125 °C).

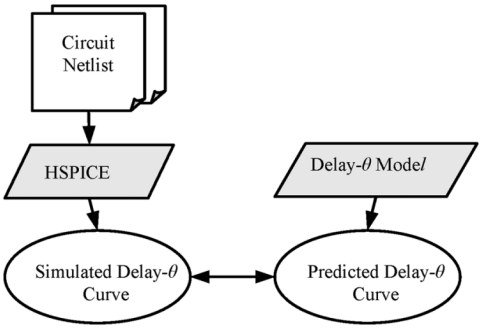


Fig. 2. Delay model validation flow.

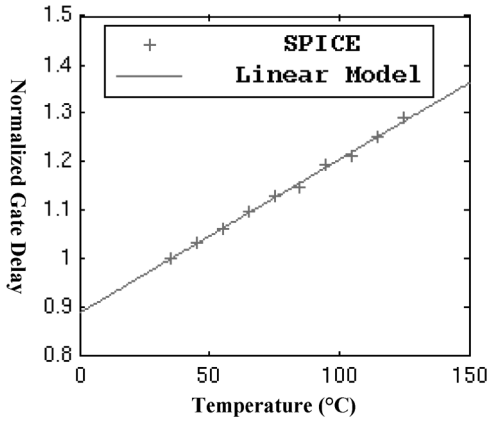


Fig. 3. Validation of the linear model for the temperature dependence of gate delay against HSPICE.

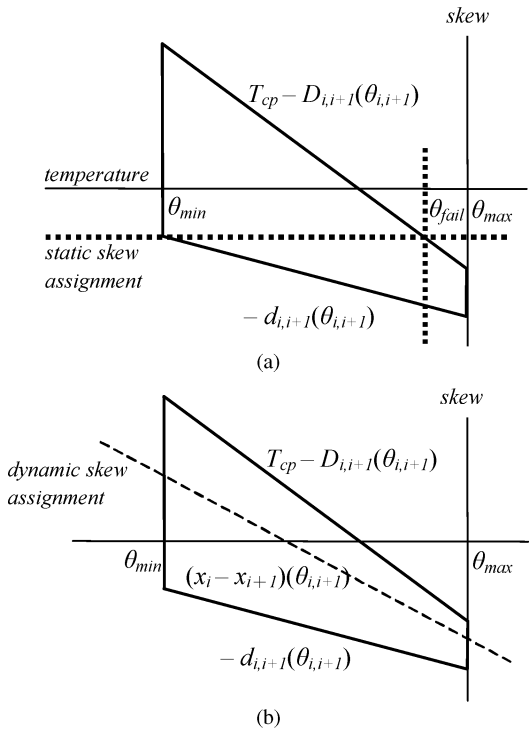


Fig. 4. (a) Failing temperature $\theta_{\rm fail}$. Timing violations may occur if $\theta_{\rm fail}$ is exceeded. (b) Utilizing temperature-sensitive skew to improve system immunity against temperature variation.

Note that this linear temperature dependence of the delay is equally applicable to the combinational logic and the skew buffers. In the local data path of pipeline stages, the RC delay of the signal is dominated by the gate capacitances. We can safely ignore the share of the interconnect delay. As a result, we can assume the same kind of temperature dependence between the combinational logic and the skew buffers.

B. Temperature-Dependent Dynamic Clock Skew Scheduling

Our methodology to design a self-adjusting clock tree architecture will utilize the linear model described in the previous section. Given this model, we formulate our problem as follows.

Problem 1. Thermal-Aware Dynamic Clock Skew Scheduling: Given a 1-D pipeline driven by a clock signal with period time $T_{\rm cp}$ and a thermal profile of the pipeline $\{\theta_{i,i+1} | \theta_{\min} \leq \theta_{i,i+1} \leq \theta_{\max}\}$, where $\theta_{i,i+1}$ is the temperature of the ith stage, if the set of constraints

$$x_i - x_{i+1} \geq -d_{i,i+1}(\theta_{i,i+1}) \tag{8}$$

$$x_i - x_{i+1} \leq T_{\rm cp} - D_{i,i+1}(\theta_{i,i+1}) \tag{9}$$

has a static solution $\{x_i\}_{\rm expect}$, then the self-adjusting clock tree should be able to adjust the actual arrival time $\{x_i\}$ to $\{x_i\}_{\rm expect}$ in order to avoid circuit malfunction.

Fig. 4 illustrates the relationships between the right-hand sides of inequalities (8) and (9) and temperature, where the temperature is plotted along the horizontal axis, while the vertical axis represents the magnitude of the clock skew. According to the analysis in Section IV-A, both $d_{i,i+1}(\theta_{i,i+1})$ and $D_{i,i+1}(\theta_{i,i+1})$ are linear functions of $\theta_{i,i+1}$, the temperature of the ith pipeline stage. Therefore, the right-hand sides of inequalities (8) and (9) can be represented by two lines as shown in Fig. 4(a). Furthermore, the upper bound $\theta_{\max}$ and the lower bound $\theta_{\min}$ of the target temperature range can be represented by two vertical lines. These four lines enclose a quadrangle whose boundary is depicted in bold in Fig. 4.

In Fig. 4, a static clock skew (i.e., the magnitude of the skew is independent of temperature) can be represented by a horizontal line. If static clock skew scheduling is used, in order to increase the tolerable temperature range, the best we can do to is to set $(x_i - x_{i+1})$ equal to $-d_{i,i+1}(\theta_{\min})$ [the dotted horizontal line shown in Fig. 4(a)]. However, as shown in Fig. 4(a), this horizontal line might intersect with the boundary of the quadrangle at a temperature $\theta_{\rm fail}$ that is below the maximum target operating temperature $\theta_{\max}$. This means that constraint (9) would be violated if the temperature is above $\theta_{\rm fail}$.

However, if we can couple $(x_i - x_{i+1})$ with the temperature $\theta_{i,i+1}$ with a linear function, then constraints (8) and (9) will never be violated as long as the local temperature $\theta_{i,i+1}$ remains between $\theta_{\min}$ and $\theta_{\max}$. An example of such a linear function is depicted by the dotted line in Fig. 4(b).

Now our problem becomes designing a clock tree that can supply dynamically changing skew values to pipeline registers, where the skew value of each pipeline stage must be a linear function of temperature. Note that we have established a linear relationship between the delay of a logic gate and temperature. We will take advantage of this result and employ special skew buffers in our clock tree architecture, which will render the temperature-dependent behavior we desire.

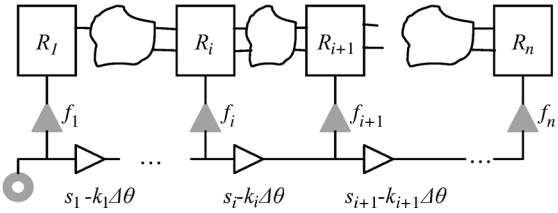


Fig. 5. SACTA: Self-adjusting clock tree architecture.

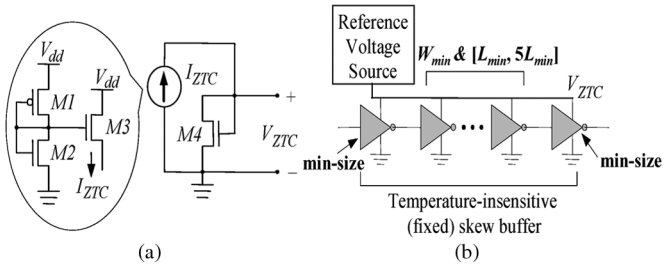


Fig. 6. (a) Schematic of the ZTC voltage reference circuit. (b) Design of the fixed skew buffer using the ZTC voltage reference.

C. Self-Adjusting Clock Tree Architecture

Fig. 5 depicts a pipeline with our proposed Self-Adjusting Clock Tree Architecture. The white triangles represent the ATA skew buffers. The relationship between their delay and temperature is expressed as $s_i - k_i\Delta\theta$. s_i is the delay of the skew buffer at the worst case temperature $\theta_{\max}$. We refer to this delay value as the *base delay* of the ATA skew buffer. k_i is the temperature sensitivity coefficient. Here, $\Delta\theta$ is defined as $\theta_{\max} - \theta$, i.e., the difference between $\theta_{\max}$ and actual operating temperature in the vicinity of the skew buffer. The gray triangles represent the fixed skew buffers (we refer to those as the fixed buffers) with *base delay* f_i. The details of the design of the fixed buffers and the ATA skew buffers will be presented in the next section.

Using this architecture, the clock skew of the ith pipeline stage will be

$$x_i - x_{i+1} = f_i - f_{i+1} - s_i + k_i(\theta_{\max} - \theta). \qquad (10)$$

In order to ensure that this function is linearly dependent on the local temperature $\theta_{i,i+1}$ of the pipeline stage i, all we need to do is to place the ith ATA skew buffer close to the logic of the ith pipeline stage on the circuit layout. Spatial correlation will enable the coupling between the temperature variable θ in (10) and the local temperature $\theta_{i,i+1}$.

It can be easily seen that the purpose of the ATA buffers is to generate the temperature-dependent coupling between pipeline latencies and the skew needed to ensure timing correctness. This can only create negative skew between consecutive registers. However, positive skew might also be needed. The purpose of the fixed buffers is to provide such positive skew.

D. Application of SACTA Onto Different Systems and Limitations

With different temperature profiles the skew distribution of SACTA will clearly be different. As a result the total skew created within the pipeline (i.e., the arithmetic sum of all skews across the pipeline stages) will be variable with temperature. This means, that there is not a single fixed total skew value between any two registers. In fact, this is the special property of SACTA that enables us to achieve adaptability. At the boundary case, this can be interpreted as having a nonzero skew between the input registers and the output registers, which is also variable over time with temperature.

For a 1-D linear pipeline this does not pose any limitations. On the other hand, if the pipeline is expected to communicate synchronously with another entity, then this variable nonzero skew might become an issue. In such a case, one possibility is to isolate the last pipeline stage from SACTA, thereby confining the end-to-end skew. In systems such as the Globally Asynchronous Locally Synchronous (GALS) system, this would not be needed. Then, SACTA can be safely applied to the entire pipeline. Since for GALS intermodule communication is done asynchronously, zero off-module clock skew is not a requirement.

For those pipeline stages lying on the feedback loops, the ATA skew buffers of SACTA cannot completely cover the associated combinational logic along this loop. However, if those uncovered pipeline stages do not lie on the critical path, then this will not be crucial.

In the remainder of our discussions, each synchronous pipeline will be modeled as a 1-D pipeline corresponding to a submodule. This model is representative for a large number of synchronized circuits. Linear pipelines are utilized in various ASICs for Application Specific Processors, particularly popular for signal processing and multimedia applications [15]–[19].

IV. Skew Buffer Design

In this section, the designs of the fixed skew buffer and the ATA skew buffer are presented.

A. Temperature-Insensitive (Fixed) Skew Buffer

Skew buffers can be designed to be insensitive to temperature variations by biasing the gate to the zero-Ttemperature-coefficient (ZTC) point. It was briefly mentioned in Section III-A that the overall temperature dependence of the drain current also depends on the gate-to-source voltage. The ZTC point is a gate bias where the effect of change in the threshold voltage cancels out that of the mobility, thereby making the drain current independent of temperature [20], [21]. Note that we mention before that the ZTC point depends also on the operation conditions, especially on the slope of the input signal. For skew buffers placed on the clock tree, it is reasonable to assume that the slope of their input (the clock signal) is very large and holds the same value during operation. Therefore, there is a single ZTC point for all the skew buffers on the clock tree. As a consequence, fixed skew buffers can be designed by using inverters whose gate-to-source voltage corresponds to that ZTC point.

In order to bias the gate to the ZTC point, a voltage reference circuit that generates the ZTC voltage is needed. Fig. 6(a) shows the voltage reference circuit using a transistor $(M4)$ whose drain is connected to its gate and a current source. By a careful sizing of $M1$ and $M2$, the output of the inverter can be controlled to generate the ZTC voltage.

In order to create a desired amount of delay for each buffer, a mixture of sizing and cascading of inverters is used. Larger L

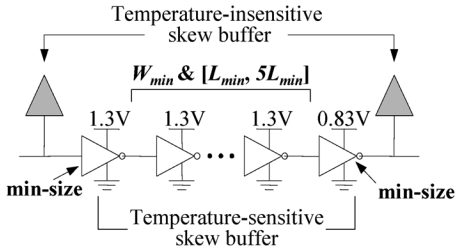


Fig. 7. Design of the automatic temperature-adjustable skew buffer.

increases the delay, but inverters with L that is too large may not be able to generate full voltage swing at the output in a given clock cycle, possibly leading to functional errors especially after a chain of several inverters. Hence, $5L_{\min}$ is used as the upper limit in L to avoid such errors for clock frequency of 2.0 GHz in the 65-nm technology, as determined from HSPICE simulation. On the other hand, increasing L results in the additional benefit of exponential reduction in the leakage power due to $V_{\rm th}$ rolloff [22].

Likewise, one can cascade more inverters to create more delay in the buffer as long as the correct logic level is preserved (i.e., even number of inverters). L of the cascaded inverters is varied between $L_{\min}$ and $5L_{\min}$ except for the first and the last inverter, while W is kept at minimum size for all. Minimum size is used for the last inverter to recover fast transition time and full voltage swing of the signal before it reaches the register. Furthermore, the use of minimum size for the first inverter synchronizes and minimizes the load driven by the last inverter in the ATA skew buffer which will be discussed in the subsequent section (Fig. 7). Besides the first and the last inverters, there is a degree of freedom for the number of inverters and their channel lengths. Fig. 6(b) depicts the design of the fixed skew buffer.

B. Automatic Temperature-Adjustable (ATA) Skew Buffer

The temperature-sensitivity of skew buffers can also be adjusted by a mixture of sizing and cascading of inverters. From (3) through (6), the temperature sensitivity of the gate delay [i.e., the temperature coefficient k in (7)] is related to gate sizing by

$$k = \left.\frac{\partial \tau}{\partial \theta}\right|_{\theta=\theta_0} \propto \left.\frac{CV_{dd}L\left[\eta(V_{gs}-V_{\rm th}(\theta))^{\alpha}-\alpha\kappa\mu(\theta)(V_{gs}-V_{\rm th}(\theta))^{\alpha-1}\right]}{W\mu(\theta)^2(V_{gs}-V_{\rm th}(\theta))^{2\alpha}}\right|_{\theta=\theta_0}. \tag{11}$$

As indicated by (11), increasing L amplifies the temperature sensitivity, which results in a longer delay as well. Furthermore, when several inverters are cascaded together, increasing L of an inverter also increases its gate capacitance, which increases the delay and the temperature sensitivity of the previous inverter [increase in C in expression (11)]. The increase in the absolute value of the delay is not a problem in the clock distribution network as long as the relative arrival times of the clock signals are well controlled.

Cascading more inverters is another effective way to adjust the temperature sensitivity. Since the total temperature sensitivity of a chain of m inverters is given by

$$k = \left.\frac{\partial \tau}{\partial \theta}\right|_{\theta=\theta_0} = \sum_{i=1}^{m} \left.\frac{\partial \tau_i}{\partial \theta}\right|_{\theta=\theta_0} \tag{12}$$

where $\partial\tau/\partial\theta_i$ is the temperature sensitivity of ith inverter in the chain. Cascading more inverters increases the total temperature sensitivity as well as the total delay.

Fig. 7 shows the design of the ATA skew buffer, connected to the fixed skew buffers described in the preceding section. The ATA skew buffer consists of a chain of cascaded inverters. The supply rail of the last inverter is connected to the ZTC voltage instead of V_{dd} (1.3 V) because its output is connected to the gate of the fixed skew buffer, which need to be biased to the ZTC voltage. The first inverter in the next ATA skew buffer then converts the voltage swing back to 1.3 V. Minimum W is used for all the inverters in the chain since increased width reduces the temperature sensitivity. As for L, a range between $L_{\min}$ and $5L_{\min}$ is used for all the inverters in the chain except for the first and the last inverter ($L_{\min}$ is used for the first and the last inverter for the same reasons explained in the preceding section).

V. Systematic Design Framework for the Clock Tree

So far, we established the fundamental models governing the relationship between temperature and delay. Furthermore, we demonstrated that we can design a delay element to achieve the ATA delay adjustment. Now, we need a systematic approach to determine the physical specifications of the buffer elements in SACTA in order to be able to design the clock tree for a given circuit. In this section, we present a systematic framework to determine the skew buffer configurations. Before presenting our framework, we will first elaborate on fundamental links between the circuit-level issues and their corresponding high-level constraints in the optimization framework.

A. Linking the Circuit-Level Aspects With the Systematic Design Framework

Various physical aspects of the skew buffer structures have a direct impact on the formulation of the optimization framework. The first significant phenomenon is that the base delay of an ATA skew buffer is proportional to its temperature sensitivity coefficient. In fact, according to (3), (4), and (12)

$$\begin{aligned} k &= \left.\frac{\partial \tau}{\partial \theta}\right|_{\theta=\theta_0} = \sum_{i=1}^{m} \left.\frac{\partial \tau_i}{\partial \theta}\right|_{\theta=\theta_0} \\ &= \sum_{i=1}^{m} \frac{C_i V_{dd}}{\frac{W_i}{L_i}\mu(\theta_{\max})\left(V_{gs}-V_{\rm th}(\theta_{\max})\right)^{\alpha}} \frac{\partial}{\partial \theta} \\ &\quad \times \left.\frac{\mu(\theta_{\max})\left(V_{gs}-V_{\rm th}(\theta_{\max})\right)^{\alpha}}{\mu(\theta)\left(V_{gs}-V_{\rm th}(\theta)\right)^{\alpha}}\right|_{\theta=\theta_0} \\ &= \frac{\partial}{\partial \theta} \left.\frac{\mu(\theta_{\max})\left(V_{gs}-V_{\rm th}(\theta_{\max})\right)^{\alpha}}{\mu(\theta)\left(V_{gs}-V_{\rm th}(\theta)\right)^{\alpha}}\right|_{\theta=\theta_0} \tau(\theta_{\max}\cdot) \end{aligned}$$

Therefore, we can relate the temperature sensitivity coefficient and the base delay by

$$k = \lambda \tau(\theta_{\max}) \tag{13}$$

where λ is a constant called *relative temperature sensitivity* whose analytical expression is given by

$$\lambda = \frac{\partial}{\partial\theta}\frac{\mu(\theta_{\max})\,(V_{gs}-V_{\rm th}(\theta_{\max}))^{\alpha}}{\mu(\theta)\,(V_{gs}-V_{\rm th}(\theta))^{\alpha}}\bigg|_{\theta=\theta_0}. \tag{14}$$

Therefore, our optimization scheme cannot assume these parameters as independent. As a result, this relationship must be accounted for in the form of a constraint in our optimization framework [see constraint (21)].

Another observation is that the base delays of both types of skew buffers cannot be made arbitrarily small. As discussed in Section IV-B, both types of skew buffers contain at least a head and a tail minimum sized inverters. Therefore, their minimal achievable base delays will be bounded by the sum of the delays of these two inverters. These constraints can be expressed with the following:

$$s \geq s_{\min} \quad f \geq f_{\min}. \tag{15}$$

Finally, the formulation of our optimization objective utilizes a function in the form of a linear combination of the skew buffer base delays. Various properties of SACTA can be represented with this generalized cost function. Particularly, the overhead of SACTA, i.e., the number of inverters used to implement the additional skew buffers can be represented with this function. According to the discussion in Section IV, for both types of buffers, the only way to obtain large base delays is to size up L. However, there is an upper bound on this sizing $(5L_{\min})$. Therefore, to achieve a given base delay value, there is a minimal number of inverters required. Therefore, a large value of f_i and s_i will correspond to a higher number of inverters required. In other words, f_i and s_i are directly related to the overhead of implementing the skew buffers. Thus, we can use the summation of the base delays of all buffers as a metric for our optimization framework.

This objective function can be used to represent other design metrics as well. For example, the base delay is related to power. A skew buffer with longer base delay consumes larger amount of power, since sizing up L increases the active area, resulting in larger dynamic power consumption. The base delay values f_i and s_i can also be related to the susceptibility of the skew buffers toward process variation. The magnitude of f_i and s_i are positively related with the length L of the transistors in the skew buffers. Larger values of L will indicate less sensitivity toward lithography-induced variations.

B. Clock Tree Optimization Framework

We will formulate the problem of determining the physical delay parameters s_i and f_i as a generalized min-cost flow problem. We describe this formulation in the following.

Problem 2. Base Delay Calculation for Skew Buffers: Given a 1-D pipeline driven by a clock signal with period time $T_{\rm cp}$, determine the base delays of the skew buffers, such that their sum is minimum, while satisfying the setup and hold time constraints.

This problem can be formulated as the following linear programming problem:

$$\text{Minimize} \quad \sum_{i=1}^{n} f_i + \sum_{i=1}^{n-1} s_i \tag{16}$$

$$\text{s.t.} \quad f_i - s_i - f_{i+1} \leq T_{\rm cp} - D_{i,i+1} \tag{17}$$

$$f_i - s_i - f_{i+1} \geq -d_{i,i+1} \tag{18}$$

$$f_i - s_i + k_i\Delta\theta_M - f_{i+1} \leq T_{\rm cp} - D_{i,i+1} + \Gamma_{i,i+1}\Delta\theta_M \tag{19}$$

$$f_i - s_i + k_i\Delta\theta_M - f_{i+1} \geq -d_{i,i+1} + \gamma_{i,i+1}\Delta\theta_M \tag{20}$$

$$k_i - \lambda s_i = 0 \tag{21}$$

$$s_i \geq s_{\min}, f_i, f_{i+1} \geq f_{\min} \tag{22}$$

$$i = 1, 2, \ldots, n-1. \tag{23}$$

Here f_i, s_i, $D_{i,i+1}$, $d_{i,i+1}$ are the delay values at $\theta_{\max}$. $\Delta\theta_M$ denotes $\theta_{\max} - \theta_{\min}$, i.e., the gap between the worst-case temperature and minimum temperature. $\Gamma_{i,i+1}$ and $\gamma_{i,i+1}$ denotes the temperature sensitivity coefficient of the longest and shortest combinational paths of the ith pipeline stage.

Constraints (17)–(20) are derived from (8)–(10). They guarantee that the line $(x_i - x_{i1})(\theta_{i,i+1})$ will conform strictly to the timing constraints and the temperature bounds (as depicted in Fig. 4).

This linear programming problem can be solved with a LP solver. However, LP-based techniques are prone to numerical instability and exhibit slow running times, and hence, combinatorial algorithms are preferred. We will show that the constraints of this problem have a special structure, which enables us to use a generalized min-cost flow-based algorithm to solve it optimally in polynomial time.

Simple transformations on the constraints will help reveal this special structure. First, substituting (21) into (19) and (20) yields

$$f_i - s_i(1-\lambda\Delta\theta_M) - f_{i+1} \leq T_{\rm cp} - D_{i,i+1} + \Gamma_{i,i+1}\Delta\theta_M \tag{24}$$

$$f_i - s_i(1-\lambda\Delta\theta_M) - f_{i+1} \geq -D_{i,i+1} + \gamma_{i,i+1}\Delta\theta_M. \tag{25}$$

Defining $f_i^{\Delta} = f_i - f_{\min}$, $s_i^{\Delta} = s_i - s_{\min}$, $u_i = f_i - s_i - f_{i+1} + d_{i,i+1}$, $v_i = f_i - s_i(1-\lambda\Delta\theta_M) - f_{i+1} + d_{i,i+1} - \gamma_{i,i+1}\Delta\theta_M$, constraints (17)–(18) and (24)–(25) can be rewritten as

$$-f_i^{\Delta} + s_i^{\Delta} + f_{i+1}^{\Delta} + u_i = d_{i,i+1} + s_{\min} \tag{26}$$

$$-f_i^{\Delta} + (1-\lambda\Delta\theta_M)s_i^{\Delta} + f_{i+1}^{\Delta} + v_i = d_{i,i+1} - \gamma_{i,i+1}\Delta\theta_M + (1-\lambda\Delta\theta_M)s_{\min} \tag{27}$$

$$0 \leq u_i \leq T_{\rm cp} - D_{i,i+1} + d_{i,i+1} \tag{28}$$

$$0 \leq v_i \leq T_{\rm cp} - D_{i,i+1} + d_{i,i+1} + (\Gamma_{i,i+1} - \gamma_{i,i+1})\Delta\theta_M. \tag{29}$$

Constraints (26) and (27) give us

$$-(\lambda\Delta\theta_M)s_i^{\Delta} - u_i + v_i = -\gamma_{i,i+1}\Delta\theta_M - (\lambda\Delta\theta_M)s_{\min}. \tag{30}$$

Equivalently, constraint (27) can be replaced by constraint (30). The above transformation yields the following new formulation:

$$\text{Minimize} \quad \sum_{i=1}^{n} f_i^{\Delta} + \sum_{i=1}^{n-1} s_i^{\Delta} \tag{31}$$

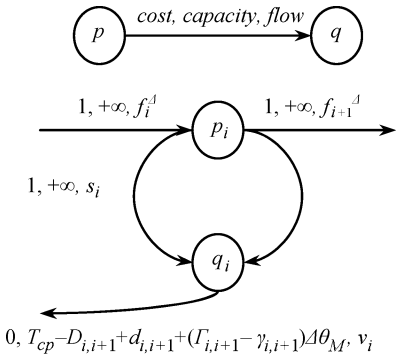


Fig. 8. Graph-based depiction of the constraints (32)–(35).

$$\text{s.t.} \quad -f_i^{\Delta}+s_i^{\Delta}+f_{i+1}^{\Delta}+u_i = d_{i,i+1}+s_{\min} \tag{32}$$

$$-(\lambda\Delta\theta_M)s_i^{\Delta}-u_i+v_i = -\gamma_{i,i+1}\Delta\theta_M-(\lambda\Delta\theta_M)s_{\min} \tag{33}$$

$$0 \le u_i \le T_{\text{cp}}-D_{i,i+1}+d_{i,i+1} \tag{34}$$

$$0 \le v_i \le T_{\text{cp}}-D_{i,i+1}+d_{i,i+1}+(\Gamma_{i,i+1}-\gamma_{i,i+1})\Delta\theta_M \tag{35}$$

$$s_i^{\Delta}, f_i^{\Delta}, f_{i+1}^{\Delta} \ge 0 \tag{36}$$

$$i = 1,2,\ldots,n-1 \tag{37}$$

Note that we have replaced the cost function with

$$\sum_{i=1}^{n} f_i^{\Delta}+\sum_{i=1}^{n-1} s_i^{\Delta}.$$

In fact, this new cost function and the original one are the same (they only differ by a constant). Therefore the new optimization problem (31)–(37) is equivalent to the original one. This formulation is actually a generalized min-cost flow formulation. Variables s_i^{Δ} and f_i^{Δ} can be any real number between 0 and $+\infty$. They can be viewed as flows on directed edges, each having a capacity of $+\infty$. Also, according to the objective function given in expression (31), each of these edges should be associated with cost 1. Likewise, each u_i (or v_i) can be modeled as a flow on a directed edge with cost 0 and having capacity of $T_{\text{cp}}-D_{i,i+1}+d_{i,i+1}$ (or $T_{\text{cp}}-D_{i,i+1}+d_{i,i+1}+(\Gamma_{i,i+1}-\gamma_{i,i+1})\Delta\theta_M$). Equation (32) [or (33)], which has the form of flow conservation condition, can be modeled as four (or three) directed edges intersecting at a node. The balance of these intersection nodes can be expressed as

$$p_i = d_{i,i+1}+s_{\min}\,(or\ q_i = -\gamma_{i,i+1}\Delta\theta_M-(\lambda\Delta\theta_M)s_{\min}). \tag{38}$$

This is also depicted in Fig. 8. Note that in constraint (33), the coefficient of s_i^{Δ} is not 1. However, this kind of constraints can still be handled using generalized min-cost flow algorithms [23].

Based on the graph representation of the constraints (32)–(33), we can model constraints (32)–(37) as shown in Fig. 9. In Fig. 9, $p_1, p_2, \ldots, p_{n-1}$ and $q_1, q_2, \ldots, q_{n-1}$ (the white vertices) are the nodes representing the constraints

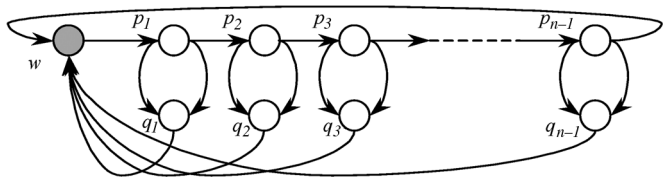


Fig. 9. Graph-based depiction of the constraints (32)–(37).

(32)–(33) for all $i = 1,2,\ldots,n-1$. Note that we add a gray node w whose balance is expressed as

$$-\sum_{i=1}^{n-1}\left[(d_{i,i+1}-\gamma_{i,i+1}\Delta\theta_M)+(1-\lambda\Delta\theta_M)s_{\min}\right]$$

ensuring the sum of balances of all the vertices is zero. It is easy to prove that the constraints (32)–(37) have a feasible solution if and only if there is a feasible flow on this graph.

Problem 2 is then reduced to the problem of finding a feasible flow that minimizes the cost function given by expression (31). There are several efficient generalized min-cost flow algorithms in the literature that guarantee polynomial running time [24].

The generalized min-cost flow-based algorithm determines the base delay value for each skew buffer while minimizing the total number of inverters used to construct the skew buffers. An inverter chain for each skew buffer is then constructed according to the base delay value for that buffer. In each chain, we set the length of all inverters in the chain except the last one to be $5L_{\min}$, and finally, we scale the length of the last inverter to the proper value to provide the required base delay.

VI. Handling Multi-V_{TH}/V_{DD} Designs

In Section II and Section III, we have developed the delay model for the pipelined circuitry. Particularly, we proposed to characterize both the maximum and minimum latency of a pipeline stage as linear functions of the local temperature. One implicit assumption we have made is that the longest and shortest paths (in terms of delay) of a given pipeline stage remain the same ones across different temperatures. In the following, we will extend our techniques to handle more general cases.

A. Pipeline Timing Constraints Revisited

The concept of relative temperature sensitivity, defined for the ATA skew buffers by (15) (which is rewritten as (40) in the following), can be similarly defined for CMOS gates:

$$\lambda = \frac{k}{\tau(\theta_{\max})} = \frac{\partial}{\partial\theta}\frac{\mu(\theta_{\max})\left(V_{gs}-V_{\text{th}}(\theta_{\max})\right)^{\alpha}}{\mu(\theta)\left(V_{gs}-V_{\text{th}}(\theta)\right)^{\alpha}}\bigg|_{\theta=\theta_0}. \tag{39}$$

According the above equation, if the all the gates used in the circuit share the same V_{gs} and V_{th}, they should have the same λ value. Hence, all the paths in a pipeline stage exhibit the same relative temperature sensitivity. As a consequence, the relative order of different paths in terms of delay remains the same regardless of the local temperature. This enable us to use a quadrangle (as depicted in Fig. 4) to represent the timing and temperature range constraints.

In the cases that multi-$V_{\rm th}$ cell library and/or voltage island technique are/is used, the above argument may no longer hold true. However, we will show later that the timing and temperature range constraints can still be represented by a convex polygon.

Lemma 1: Given a pipeline stage driven by a clock with period time $T_{\rm cp}$, denoting its maximum latency (as a function of the local temperature θ) by $D(\theta)$, the quantity $T_{\rm cp} - D(\theta)$ is a concave function of θ.

Proof: We denote the delay of a path path$_i$ by $D^{(i)}(\theta)$. As $D^{(i)}(\theta)$ equals to the sum of the delay of the gates along path$_i$, $D^{(i)}(\theta)$ should be a linear function of θ. Let λ_i represent the relative temperature sensitivity of this path, we have

$$D^{(i)}(\theta) = D^{(i)}(\theta_{\max}) \cdot (1 - \lambda_i(\theta_{\max} - \theta)).$$

Therefore

$$T_{\rm cp} - D(\theta) = T_{\rm cp} - \max_i \left\{ D^{(i)}(\theta_{\max}) \cdot (1 - \lambda_i(\theta_{\max} - \theta)) \right\}.$$

Given a real number ρ between 0 and 1, temperature values θ_1 and θ_2 between $\theta_{\min}$ and $\theta_{\max}$, we have

$$\begin{aligned}
&T_{\rm cp} - D(\rho\theta_1 + (1-\rho)\theta_2) \\
&\quad = T_{\rm cp} - \max_i \Big\{ D^{(i)}(\theta_{\max}) \\
&\qquad\qquad \cdot (1 - \lambda_i(\theta_{\max} - \rho\theta_1 - (1-\rho)\theta_2)) \Big\} \\
&\quad = \max_i \left\{ T_{\rm cp} - D^{(i)}(\theta_{\max}) \cdot (1 - \lambda_i\theta_{\max}) \right\} \\
&\qquad - \max_i \left\{ \rho D^{(i)}(\theta_{\max})\lambda_i\theta_1 + (1-\rho) D^{(i)}(\theta_{\max})\lambda_i\theta_2 \right\} \\
&\quad \geq \rho \cdot \max_i \left\{ T_{\rm cp} - D^{(i)}(\theta_{\max}) \cdot (1 - \lambda_i\theta_{\max}) \right\} \\
&\qquad - (1-\rho) \cdot \max_i \left\{ (1-\rho) D^{(i)}(\theta_{\max}) \cdot (1 - \lambda_i\theta_{\max}) \right\} \\
&\qquad - \rho \cdot \max_i \left\{ D^{(i)}(\theta_{\max})\lambda_i\theta_1 \right\} \\
&\qquad - (1-\rho) \cdot \max_i \left\{ D^{(i)}(\theta_{\max})\lambda_i\theta_2 \right\} \\
&\quad = \rho \cdot \left(T_{\rm cp} - \max_i \left\{ D^{(i)}(\theta_{\max}) \cdot (1 - \lambda_i(\theta_{\max} - \theta_1)) \right\} \right) \\
&\qquad + (1-\rho) \cdot \Big(T_{\rm cp} - \max_i \Big\{ D^{(i)}(\theta_{\max}) \\
&\qquad\qquad \cdot (1 - \lambda_i(\theta_{\max} - \theta_2)) \Big\} \Big) \\
&\quad = \rho \cdot (T_{\rm cp} - D(\theta_1)) + (1-\rho) \cdot (T_{\rm cp} - D(\theta_2)).
\end{aligned}$$

The above inequality directly leads to the concavity of the function. □

Lemma 2: Given a pipeline stage driven by a clock with period time $T_{\rm cp}$, denoting its minimum latency (as a function of the local temperature θ) by $d(\theta)$ the quantity $-d(\theta)$ is a convex function of θ.

The proof of Lemma 2 is similar to that of Lemma 1 and is omitted here. Combining Lemma 1 and Lemma 2, and noticing that the temperature range constraint can be represented by two vertical lines, we have the following theorem.

Theorem 1: Given a pipeline stage driven by a clock with period time $T_{\rm cp}$, denote its maximum and minimum latency (as a functions of the local temperature θ) by $D(\theta)$ and $d(\theta)$, and let x and x' represent the delay of the clock signal delays from the clock source to the two registers associated with the pipeline stage. The timing and temperature range constraints of this pipeline stage

$$-d(\theta) \leq x - x' \leq T_{\rm cp} - D(\theta), \quad \forall\theta \in [\theta_{\min}, \theta_{\max}]$$

can be represented by a convex polygon.

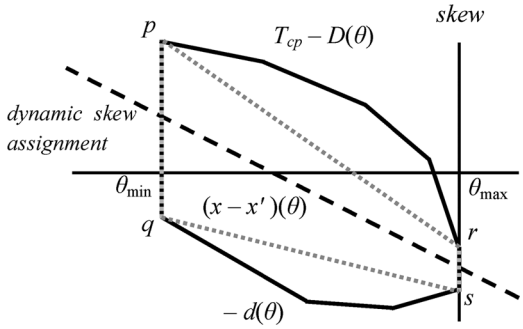


Fig. 10. Convex polygon representation of the timing and temperature range constraints.

Fig. 10 gives an example of the convex polygon representation of the timing and temperature range constraints.

B. Extended Clock Tree Design Framework

In the following, we will extend our clock tree design framework to handle the multi-$V_{\rm th}/V_{\rm dd}$ designs.

As depicted by Fig. 10, functions $T_{\rm cp} - D(\theta)$ and $-d(\theta)$ intersect with vertical lines $\theta = \theta_{\min}$ and $\theta = \theta_{\max}$ at four points p, q, r, and s. These four points induce a quadrangle depicted in gray dotted lines. It is obvious that the timing and temperature range constraints are satisfied if and only if the dynamic clock skew $(x - x')(\theta)$ lies strictly within the quadrangle. Thus, we have reduced the dynamic clock skew scheduling problem for multi-$V_{\rm th}/V_{\rm dd}$ designs to its simplified version discussed in Section III.

The linear program (16)–(23) can be easily generalized to determine the base delay of the skew buffers. We have rewritten the generalized LP in the following:

$$\text{Minimize} \quad \sum_{i=1}^{n} f_i + \sum_{i=1}^{n-1} s_i \tag{40}$$

$$\text{s.t.} \quad f_i - s_i - f_{i+1} \leq T_{\rm cp} - D_{i,i+1} \tag{41}$$

$$f_i - s_i - f_{i+1} \geq -d_{i,i+1} \tag{42}$$

$$f_i - s_i + k_i\Delta\theta_M - f_{i+1} \leq T_{\rm cp} - D_{i,i+1} + \Gamma_{i,i+1}\Delta\theta_M \tag{43}$$

$$f_i - s_i + k_i\Delta\theta_M - f_{i+1} \geq -d_{i,i+1} + \gamma_{i,i+1}\Delta\theta_M \tag{44}$$

$$\Gamma_{i,i+1} = \left(D'_{i,i+1} - D_{i,i+1}\right)/\Delta\theta_M \tag{45}$$

$$\gamma_{i,i+1} = \left(d'_{i,i+1} - d_{i,i+1}\right)/\Delta\theta_M \tag{46}$$

$$k_i - \lambda s_i = 0 \tag{47}$$

$$s_i \geq s_{\min}, f_i, f_{i+1} \geq f\min \tag{48}$$

$$i = 1, 2, \ldots, n-1. \tag{49}$$

Notice that compared to LP (16)–(23), the interpretation of parameters $\Gamma_{i,i+1}$ and $\gamma_{i,i+1}$ ischanged. In the generalized formulation, $\Gamma_{i,i+1}$ is equal to $(D'_{i,i+1} - D_{i,i+1})/\Delta\theta_M$, i.e., the slope of segment pr in Fig. 10. Here $D'_{i,i+1}$ and $D_{i,i+1}$ are the maximum latency of the pipeline stage at $\theta_{\min}$ and $\theta_{\max}$, respectively. Similarly, $\gamma_{i,i+1}$ is set to the slope of segment qs.

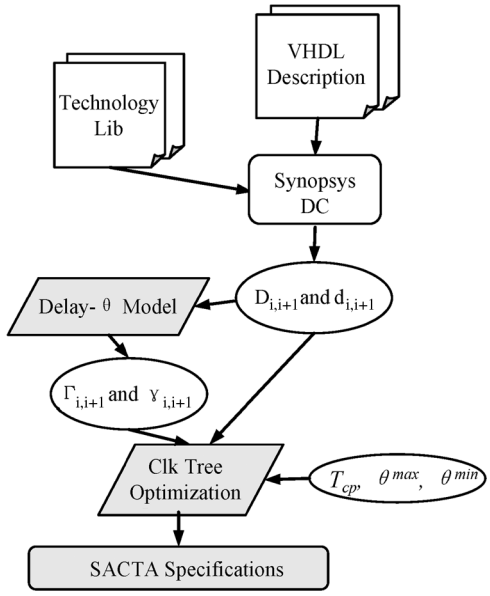


Fig. 11. Experimental flow.

TABLE I
PIPELINE PARTITION OF THE BENCHMARKS

Circuit \ Partition	Balanced					Unbalanced				
	s1	s2	s3	s4	s5	s1	s2	s3	s4	s5
PolyEval	4	4	4	4	4	2	6	2	5	3
RSDecoder	4	4	4	4	4	2	6	2	5	3
FiniteFieldMult	4	4	4	4	4	2	6	2	5	3

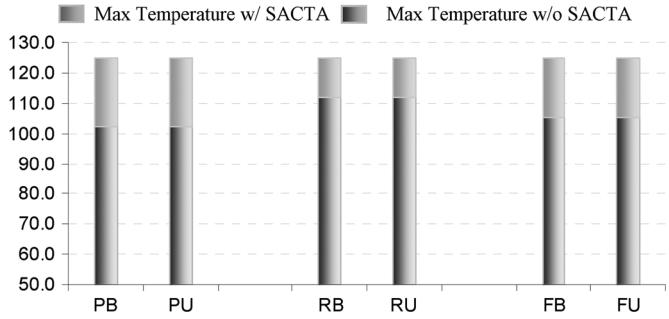


Fig. 12. Maximum permissible temperature for single-$V_{\rm th}$ designs.

The network-flow-based algorithm can be similarly extended to handle the multi-$V_{\rm th}/V_{\rm dd}$ designs. The only change to the generalized min-cost flow formulation (31)–(37) is to replace $\Gamma_{i,i+1}$ and $\gamma_{i,i+1}$ with $(D'_{i,i+1} - D_{i,i+1})/\Delta\theta_M$ and $(d'_{i,i+1} - d_{i,i+1})/\Delta\theta_M$, respectively. The underlying graph structure remains unchanged.

Note that in deep submicron technologies, CMOS cells may exhibit inversed temperature dependence (ITD), where the delay of the cells might decrease with rising temperature. The proposed technique needs to be enhanced to handle ITD to achieve timing closure. To be more specific, ATA skew buffers with inversed temperature dependence need to be developed and the clock tree optimization framework needs to be changed accordingly.

VII. EXPERIMENTAL RESULTS

In this section we first describe our experimental flow. Next, we present our results demonstrating the effectiveness of our proposed self-adjusting clock tree architecture.

A. Experimental Setup

Fig. 11 illustrates our experimental flow. We use the Synopsys Design Compiler to synthesize the benchmarks onto the TSMC 65-nm technology library. We experimented with both single-$V_{\rm th}$ and multi-$V_{\rm th}$ libraries. Design Compiler reports the longest and shortest paths for each pipeline stage in the benchmarks. According to the analysis in Section VI., when the multi-$V_{\rm th}$ library is used, the critical path at low temperature might not be the most critical one at a higher temperature. Therefore, we let Design Compiler report multiple paths to cover all the potential longest/shortest paths. Also, based on our validated Delay-θ model at 65 nm, the temperature sensitivity of the longest/shortest paths, i.e., the Γ and γ values for each pipeline stage, can be determined. Then, we feed the $D_{i,i+1}$, $d_{i,i+1}$, $\Gamma_{i,i+1}$, and $\gamma_{i,i+1}$ values, as well as the given $T_{\rm cp}$ into our clock tree optimization framework. The other two parameters for the optimization algorithm, i.e., the boundaries of the operating temperature range, are set to 25 °C and 125°C. The optimization subroutine determines the appropriate buffer parameters, which are further used as the guidelines for finalizing the skew buffer design. In our implementation, the LP formulations for skew buffer configuration are solved using a commercial linear programming solver CPLEX ver10.1.

We have used a benchmark set consisting of systolic array circuits. This set includes a polynomial expression evaluator [18], the Reed-Solomon decoder [19], and a fast digital-serial multiplier for finite field [15]. These circuits share a common structure. A set of processing elements (PEs) is connected in series. In order to improve system throughput, the circuits are pipelined by inserting registers between the PEs. In our experiments, we divide each benchmark into five pipeline stages. To better evaluate our technique, we allow both balanced and unbalanced pipeline partitions. The pipeline partitions are summarized in Table I. The columns s1-s5 denote the number of PEs in each pipeline stage. For example, the unbalanced-partitioned benchmark PolyEval, contains 2, 6, 2, 5, and 3 PEs, in pipeline stages s1 through s5, respectively.

B. Experimental Results

1) Single-$V_{\rm th}$ Design: The first set of experiments is carried out on the systolic pipeline designs synthesized using TSMC low-$V_{\rm th}$ libraries only.

Our first set of results, depicted in Fig. 12, presents the maximum operating temperature with guaranteed timing correctness for all the pipelines under spatially uniform temperature distribution. The required clock periods for the pipelines are given in Table II. PB (PU) denotes the balanced (unbalanced) pipeline of benchmark PolyEval; RB (RU) represents the balanced (unbalanced) pipeline of benchmark RSDecoder; and FB (FU) stands for the balanced (unbalanced) pipeline of benchmark FiniteFieldMult.

TABLE II
CLOCK FREQUENCY FOR THE PIPELINES (SINGLE-$V_{\rm th}$ DESIGNS)

	PB	PU	RB	RU	FB	FU
Freq*/GHz*	0.4900	0.3267	0.1908	0.1272	0.4097	0.2731

TABLE III
TIMING CORRECTNESS OF THE PIPELINES UNDER DIFFERENT THERMAL PROFILES (SINGLE-$V_{\rm th}$ DESIGNS)

Thermal Profiles/°C					**Pipelines w/o SACTA**						**Pipelines w/ SACTA**					
s1	s2	s3	s4	s5	PB	PU	RB	RU	FB	FU	PB	PU	RB	RU	FB	FU
125	120	115	112	110	X	X	X	X	X	X	√	√	√	√	√	√
110	112	115	120	125	X	X	X	X	X	X	√	√	√	√	√	√
100	105	110	105	100	X	X	√	√	X	X	√	√	√	√	√	√
110	105	100	105	110	X	X	√	√	X	X	√	√	√	√	√	√
105	100	102	105	102	X	X	√	√	√	√	√	√	√	√	√	√

In our first set of experiments we evaluate the resilience of SACTA against temperature variations. We compare two cases: utilizing SACTA versus utilizing static clock skew scheduling in the pipeline designs. For each pipeline, we keep increasing its operating temperature until the timing constraint is violated. This temperature is recorded as the maximum tolerable temperature for that pipeline under the given clock period constraint.

We observe that for the pipelines with SACTA, the maximum tolerable temperature can be dramatically increased—all the pipelines employing SACTA can function correctly until chip temperature attains the designed maximal value: 125 °C. As a comparison, unbalanced PolyEval pipelines without SACTA fail at about 102 °C, indicating a 23 °C increase in maximal tolerable temperature. Approximately the same improvement is observed for the benchmark PolyEval. On average, the tolerable temperature range increases by 18.5 °C.

Next, we experiment with different thermal profiles. Table III presents our results, where symbol "√" signifies that circuit functions correctly, and symbol "×" indicates timing constraints are violated. Obviously examining all possible thermal profiles is impractical. Here, we only consider some representative thermal profiles. The first one represents a monotonically decreasing profile across the pipeline stages. The second profile is monotonically increasing. The third (fourth) profile exhibits a profile that is first increasing (decreasing) and then decreasing (increasing). The fifth row represents a profile which is decreasing, increasing, and decreasing again. The clock period times for the pipelines are set according to Table II. We observe that for all the thermal profiles, no timing violation occurs on pipelines with SACTA. In comparison, the pipelines without SACTA fail in most cases.

Our experimental results also indicate that SACTA can enhance system performance. In this experiment, the chip is assumed to uniformly execute at the worst case operating temperature 125 °C. For the pipelines with SACTA, we set f_i and s_i to the values determined by the clock tree optimization algorithm. For the pipelines without SACTA, as mentioned in Section III-B, the best we can do to prevent a thermal-induced timing violation is to set the skew value of the ith pipeline stage to $-d_{i,i+1}(\theta_{\min})$. We then keep increasing the system clock frequency until a timing violation occurs. Fig. 13 plots the relative performance gain, for each pipeline, the maximal achievable frequency without SACTA is normalized to 1, and the maximal achievable frequency with SACTA is normalized to

$$\frac{\text{Max achievable Freq w/ SACTA}}{\text{Max achievable Freq w/o SACTA}}.$$

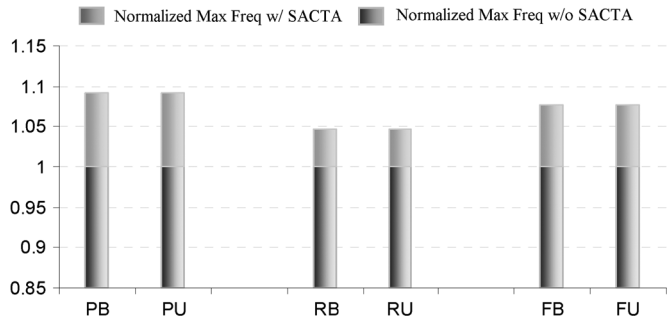


Fig. 13. Relative performance improvement for single-$V_{\rm th}$ designs.

TABLE IV
ACHIEVABLE CLOCK FREQUENCIES (SINGLE-$V_{\rm th}$ DESIGNS)

	PB	PU	RB	RU	FB	FU
Orig. */GHz*	0.4487	0.2992	0.1823	0.1216	0.3804	0.2536
SACTA */GHz*	0.4900	0.3267	0.1908	0.1272	0.4097	0.2731
Impr%	9.20%	9.19%	4.66%	4.61%	7.70%	7.69%

For these benchmarks, by employing SACTA, the maximum achievable clock frequency can be increased. For PolyEval, the relative improvement is almost 10%. We also provide the absolute values of the maximum achievable frequencies as well as the exact number of relative improvement, which are listed in Table IV.

Notice that the improvements gained by adopting SACTA are different for different benchmarks. In fact, as most clock-skew-scheduling-based techniques, our technique works better on benchmarks whose pipeline stages are more balanced. This is because, in general, the more balanced a pipeline stage is, the more timing slacks it can offer to the adjacent stages via clock skew scheduling. A pipeline stage is said to be *fully balanced* if the gap between the longest and shortest paths in the pipeline stages is zero. In general, the more balanced the pipeline stages are, the more benefit one can obtain by adopting clock skew scheduling. In our experiment, the benchmarks have different levels of balance. Hence, the relative performance improvements for different benchmarks are different.

Table V reports the overhead of SACTA. The second, the third, and the fourth rows report the number of standard cells, the area, and the power consumption of the benchmarks without SACTA. The fifth, the sixth, and the seventh rows provide the increase in the number of standard cells (i.e., the inverters on SACTA), the area, and the power consumption. These numbers are generated by Design Compiler and our clock tree optimization algorithm. The numbers in the parentheses are the relative increases compared to the original benchmark. In most cases,

TABLE V
OVERHEAD OF SACTA (SINGLE-$V_{\rm th}$ DESIGNS)

		PB	PU	RB	RU	FB	FU
Orig. Bench	***Num**$_{cell}$*	1492	1362	2472	2244	718	654
	Area */μm^2*	14964.7	13346.9	40851.7	36435.3	2923.7	2607.66
	Power */mW*	27.92	24.90	64.30	53.05	5.55	4.58
SACTA	***Δ Num**$_{cell}$*	97	57	157	60	137	53
	Δ Area */μm^2* **(Δ%)**	104.8(0.70%)	61.56(0.46%)	169.6(0.42%)	64.8(0.18%)	148.0(5.06%)	57.24(2.20%)
	Δ Power */mW* **(Δ%)**	0.118(0.42%)	0.070(0.28%)	0.192(0.30%)	0.073(0.14%)	0.167(3.01%)	0.065(1.41%)

TABLE VI
CLOCK FREQUENCY FOR THE PIPELINES (MULTI-$V_{\rm th}$ DESIGNS)

	PB	PU	RB	RU	FB	FU
Freq*/GHz*	0.4902	0.3268	0.1799	0.1199	0.4165	0.2777

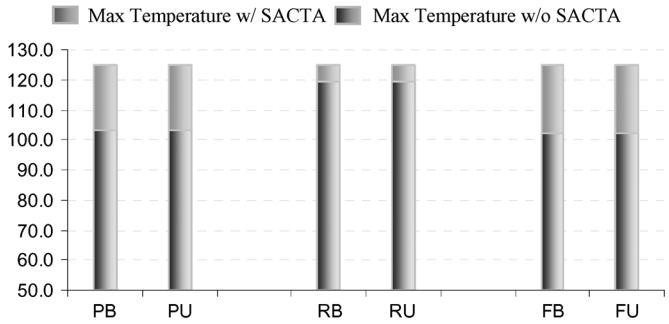


Fig. 14. Maximum permissible temperature for multi-$V_{\rm th}$ designs.

the area and power overheads are negligible. Even for the worst case (FB), the area and power overheads are only 5.06% and 3.01%, respectively.

2) Multi-$V_{\rm th}$ Design: We have also carried out our experiments on the systolic pipeline designs synthesized using TSMC multi-$V_{\rm th}$ libraries (containing high-$V_{\rm th}$, standard-$V_{\rm th}$, and low-$V_{\rm th}$ cells). The temperature sensitivities of different gates with different threshold voltages are obtained by HSPICE simulation using PTM 65-nm transistor models [14].

Similar to the single-$V_{\rm th}$ case, we first determine the maximum operating temperature with guaranteed timing correctness for all the pipelines under spatially uniform temperature distribution. The required clock periods for the pipelines are given in Table VI. The results are depicted in Fig. 14. We observe that for the pipelines with SACTA, all the pipelines employing SACTA can operate correctly until chip temperature attains the designed maximal value – 125 °C. As a comparison, both balanced and unbalanced PolyEval pipelines without SACTA fail at 103 °C, indicating an over 22 °C increase in maximal tolerable temperature. On average, the tolerable temperature range increases by 16.9 °C.

Next, we present the time correctness of the pipelines under different thermal profiles in Table VII. The clock frequencies of different pipelines are set according to Table VI. Similar to the single-$V_{\rm th}$ case, for all the thermal profiles, no timing violation occurs on pipelines with SACTA. However, the pipelines without SACTA fail in most cases.

TABLE VII
TIMING CORRECTNESS OF THE PIPELINES UNDER DIFFERENT THERMAL PROFILES (MULTI-$V_{\rm th}$ DESIGNS)

Thermal Profiles/°C					**Pipelines w/o SACTA**						**Pipelines w/ SACTA**					
s1	s2	s3	s4	s5	PB	PU	RB	RU	FB	FU	PB	PU	RB	RU	FB	FU
125	120	115	112	110	X	X	X	X	X	X	√	√	√	√	√	√
110	112	115	120	125	X	X	X	X	X	X	√	√	√	√	√	√
100	105	110	105	100	X	X	√	√	X	X	√	√	√	√	√	√
110	105	100	105	110	X	X	√	√	X	X	√	√	√	√	√	√
105	100	102	105	102	X	X	X	X	X	X	√	√	√	√	√	√

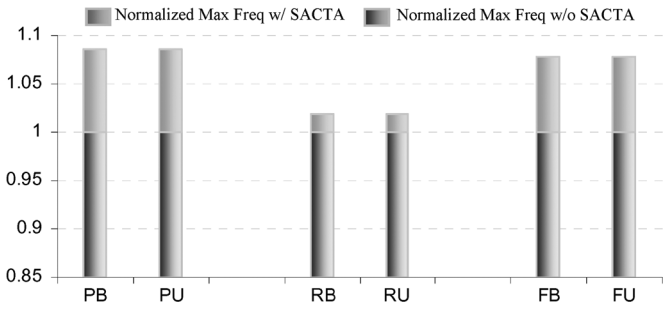


Fig. 15. Relative performance improvement for multi-$V_{\rm th}$ designs.

TABLE VIII
ACHIEVABLE CLOCK FREQUENCIES (SINGLE-$V_{\rm th}$ DESIGNS)

	PB	PU	RB	RU	FB	FU
Orig. */GHz*	0.4514	0.3009	0.1765	0.1177	0.3863	0.2575
SACTA */GHz*	0.4902	0.3268	0.1799	0.1199	0.4165	0.2777
Impr%	8.60%	8.61%	1.93%	1.87%	7.82%	7.84%

Fig. 15 presents the improvement in system performance gained by adopting SACTA. It is clear that SACTA enhances the performance of the mult-$V_{\rm th}$ designs also. For PolyEval, the relative improvement is almost 10%. The absolute values of the maximum achievable clock frequencies and the exact values of the relative improvement are provided in Table VIII.

Finally, we report the hardware overhead of SACTA in the multi-$V_{\rm th}$ case in Table IX. The average area and power overhead are 1.53% and 0.92%, respectively. Even in the worst case, area and power overhead of SACTA are only 5.09% and 3.30%.

TABLE IX
OVERHEAD OF SACTA (MULTI-$V_{\rm th}$ DESIGNS)

		PB	PU	RB	RU	FB	FU
Orig. Bench	Num_{cell}	1492	1362	2552	2316	738	672
	$Area\ /\mu m^2$	15278.0	13626.4	30656.0	27341.8	2906.7	2592.5
	$Power\ /mW$	27.66	26.91	52.23	47.01	5.51	5.00
SACTA	ΔNum_{cell}	127	54	102	43	137	54
	$\Delta Area\ /\mu m^2\ (\Delta\%)$	137.2(0.90%)	58.32(0.43%)	110.2(0.36%)	46.44(0.17%)	148.0(5.09%)	58.32(2.25%)
	$\Delta Power\ /mW\ (\Delta\%)$	0.155(0.56%)	0.066(0.24%)	0.124(0.24%)	0.052(0.11%)	0.167(3.03%)	0.066(1.33%)

VIII. CONCLUSION

We have proposed SACTA, a self-adjusting clock tree architecture, and a dynamic clock scheduling scheme to improve performance and reliability of pipelined circuits. We designed automatic temperature-adjustable skew buffers to create useful temperature-dependent clock skews. We developed a two-step technique for design of a power-optimized clock tree. Experimental results show that our scheme can dramatically improve the temperature tolerance of both single-$V_{\rm th}$ and multi-$V_{\rm th}$ designs. On average, the maximal tolerable temperature is enhanced by more than 15 °C.

REFERENCES

[1] S.A. Bota *et al.*, "Within die thermal gradient impact on clock skew: A new type of delay-fault mechanism," presented at the Int. Test Conf. 2004, .

[2] A. Chakraborty and K. Duraisami, "Dynamic thermal clock skew compensation using tunable delay buffers," presented at the Int. Symp. Low Power Electronics and Design, 2006.

[3] M. Cho, S. Ahmed, and D. Z. Pan, "TACO: Temperature Aware Clock-tree Optimizaton," presented at the Int. Conf. Computer Aided Design, 2005.

[4] V. Nawale and T. W. Chen, "Optimal useful clock skew scheduling in the presence of variations using robust ILP formulations," presented at the IEEE/ACM Int. Conf. Computer-Aided Design, San Jose, CA, 2006.

[5] S. Lee *et al.*, "Reducing pipeline energy demands with local DVS and dynamic retiming," presented at the Int. Symp. Low Power Electronics and Design, 2004.

[6] M. Horowitz *et al.*, "Scaling, power and the future of CMOS," presented at the IEEE Int. Electron Devices Meeting, 2005.

[7] A. Calimera *et al.*, "Temperature-insensitive synthesis using multi-Vt libraries," presented at the Great Lake Symp. on VLSI, 2008.

[8] J. P. Fishburn, "Clock skew optimization," *IEEE Trans. Comput.*, vol. 39, no. 7, pp. 945–951, Jul. 1990.

[9] R. B. Deoka and S. S. Sapatnekar, "A graph-theoretic approach to clock skew optimization," presented at the IEEE Int. Symp. Circuits and Systems, 1994.

[10] E. G. Friedman, "Clock distribution networks in synchronous digital integrated circuits," *Proc. IEEE*, vol. 89, no. 5, pp. 665–692, May 2001.

[11] J. Long *et al.*, "A self-adjusting clock tree architecture to cope with temperature variations," presented at the Int. Conf. Computer-Aided Design, 2007.

[12] T. Sakurai and A. Newton, "Alpha-power law MOSFET model and its application to CMOS inverter delay and other formulas," *IEEE J. Solid-State Circuits*, vol. 25, no. 2, pp. 584–593, Apr. 1990.

[13] J. Ku and Y. Ismail, "Thermal-aware methodology for repeater insertion in low-power VLSI circuits," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 15, no. 8, pp. 963–970, Aug. 2007.

[14] Y. Cao *et al.*, "New paradigm of predictive MOSFET and interconnect modeling for early circuit design," presented at the IEEE Custom Integrated Circuit Conf., 2000.

[15] C. Kim, S. Kwon, and C. Hong, "A fast digit-serial systolic multiplier for finite field GF(2m)," presented at the Asia and South Pacific Design Automation Conf., Shanghai, China, 2005.

[16] S. M. Currie *et al.*, "Implementation of a single chip, pipelined, complex, one-dimensional fast fourier transform in 0.25 μm BulkCMOS," presented at the IEEE Int. Conf. Application-Specific Systems, Architectures, and Processors, 2002.

[17] M. Kovac and P. Ranganathan, "JAGUAR: A high speed VLSI chip for JPEG image compression standard," presented at the Int. Conf. VLSI Design, 1995.

[18] P. C. Mathias and L. M. Patnaik, "Systolic evaluation of polynomial expressions," *IEEE Trans. Comput.*, vol. 39, no. 5, pp. 653–665, May 1990.

[19] K. Iwamura, Y. Dohi, and H. Imai, "A design of reed-solomon decoder with systolic-array structure," *IEEE Trans. Comput.*, vol. 44, no. 1, pp. 118–122, Jan. 1995.

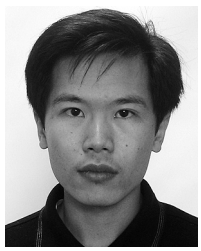
[20] F. S. Shoucair, "Analytical and experimental methods for zero-temperature-coefficient biasing of MOS transistors," *Electron. Lett.*, vol. 25, no. 17, pp. 1196–1198, 1989.

[21] I. M. Filanovsky and A. Allam, "Compensation of mobility and threshold voltae temperature effects with applications in CMOS circuits," *IEEE Trans. Circuits Syst. I*, vol. 48, no. 7, pp. 876–884, Jul. 2001.

[22] K. Roy, S. Muhopadhyay, and H. Mahmoodi-Meimand, "Leakage current mechanism and leakage reduction techniques in deep-submicrometer CMOS circuits," *Proc. IEEE*, vol. 91, no. 2, pp. 305–327, Feb. 2003.

[23] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Upper Saddle River, NJ: Prentice-Hall, 1993.

[24] K. D. Wayne, "A polynomial combinatorial algorithm for generalized minimum cost flow," presented at the ACM Symp. Theory of Computing, 1999.



Jieyi Long (S'07) received the B.S. degree in microelectronics from Peking University, Beijing, China, in 2006. He is currently working toward the Ph.D. degree at the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL.

His research interests include physical design, thermal monitoring, and management for high-performance VLSI systems, and design automation for self-adjusting architectures.

Ja Chun Ku received the B.S. degree in computer engineering and the Ph.D. degree in electrical and computer engineering from Northwestern University, Evanston, IL, in 2003 and 2007, respectively.

In August 2007, he joined the System LSI Division of Samsung Electronics Company, Ltd., Kyungki-Do, Korea, where he is currently involved in the development of Media SoC products as a Senior Engineer. His research interests include low-power and thermal issues in VLSI circuits and microarchitecture. He has published more than ten technical papers in refereed journals and conferences.

Seda Ogrenci Memik (M'98–SM'05) received the B.S. degree in electrical and electronic engineering from Bogaziçi University, Istanbul, Turkey and the Ph.D. degree in computer science from the University of California, Los Angeles.

She is currently an Assistant Professor at the Electrical Engineering and Computer Science Department, Northwestern University, Evanston. Her research interests include embedded and reconfigurable computing, thermal-aware design automation, and thermal management for high-performance microprocessor systems.

Dr. Ogrenci Memik received the National Science Foundation Early Career Development (CAREER) Award in 2006. She served as Technical Program Committee member, Organizing Committee member, Subcommittee and Track Co-Chair of several conferences, including ICCAD, DATE, FPL, and GLSVLSI and is currently serving on the editorial board of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS.

Yehea Ismail was born in Giza, Egypt on November 16, 1971. He attended the School of Engineering, Department of Electronics and Communications, Cairo University, Cairo, Egypt, from 1988-1993, where he received the B.Sc. degree in electronics and communications engineering with distinction and honors. He received his first Masters degree in electronics from Cairo University (with distinction) in 1996. He received his second Masters degree in electrical engineering in 1998 and the Ph.D. degree in 2000, both from the University of Rochester, Rochester, NY.

As one of the top of his class, he was appointed as a Teacher Assistant at the Department of Electrical and Computer Engineering, Cairo University in August 1993. He is currently an Associate Professor at Northwestern University (NU), Evanston, IL, and at Nile University and the Director of the Nanoelectronics Integrated Systems Center at NU. He has coauthored more than 130 technical papers and a book, and several book chapters. He was with IBM Microelectronics from 1997-1999. His primary research interests include digital integrated circuit design, interconnect, noise, innovative circuit simulation, and related circuit-level issues in high-performance VLSI circuits.

Dr. Ismail is the Chair Elect of the CAS VLSI Technical Committee, Editor-in-Chief of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS , was on the editorial board of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I. FUNDAMENTAL THEORY AND APPLICATIONS, and a Guest Editor for a special issue of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS on"On-Chip Inductance in High Speed Integrated Circuits." He has also chaired many conferences such as GLSVLSI, IWSOC, and ISCAS. He was selected as the 2002 IEEE Circuits and Systems Society Outstanding Young Author Award winner, and received the National Science Foundation Career Award in 2002. HE has been given the Best Teacher Award in the ECE Department, NU in 2002-2003.