Skew Management of NBTI Impacted Gated Clock Trees

Ashutosh Chakraborty, *Member, IEEE,* and David Z. Pan, *Senior Member, IEEE*

***Abstract*—Negative bias temperature instability (NBTI) has emerged as the dominant failure mechanism for PMOS devices in nanometer integrated circuit (IC) designs, thus limiting their lifetime. There are several existing research works that mitigate impact of NBTI on gate delay and reliability. However, its impact on one of the most important components of modern IC design—the clock tree—has not been researched enough. Clock gating impacts the extent of NBTI-induced V_{TH} degradation of clock buffers leading to nonuniform NBTI degradation and, thus, increased clock skew. In this paper, we propose a practical design-time technique of modifying the clock gating implementation by selecting NAND or NOR gate as output stage of integrated clock gating cells with the objective of minimizing NBTI-induced clock skew. This selection intelligently modulates the signal probability and delay equations of clock signal paths with no extra hardware penalty. We formulate the skew minimization problem as an integer linear program which determines the optimal NAND or NOR assignment of clock gating buffer. Experimental results demonstrate the effectiveness of our method as the NBTI-induced clock skew is reduced by more than 74% compared to the traditional method. The impact of voltage and temperature variation on the proposed technique was analyzed and we observed reduced but still significant reduction in clock skew under variation as compared to the traditional clock gating technique.**

***Index Terms*—Aging, clocks, integer linear programming.**

I. Introduction

NEGATIVE bias temperature instability (NBTI) has emerged as the primary PMOS failure mechanism [1] for advanced sub-65-nm very large scale integration (VLSI) technology. With the reduction in gate oxide thickness, NBTI dictates the lifetime of the device as compared to other reliability issues, such as hot carrier injection, time dependence dielectric breakdown, and so on. NBTI causes slow threshold voltage (V_{TH}) degradation (i.e., increase) of PMOS device consequently reducing its drive current and performance over time. Over a period of 10 years, the V_{TH} of the PMOS device can increase by up to 50 mV [2], causing timing violation and functional failures. As feature sizes shrink, NBTI effects will worsen exponentially due to higher operating temperatures. The instability of PMOS transistor parameters (e.g., threshold voltage, transconductance, saturation current, and so on) under negative (inversion) bias and relatively high temperature has been known to be a reliability concern since the 1970s, and modeling effort to understand this is also just as old [3]. Though a similar effect also occurs in NMOS devices under positive bias [4], the degradation in PMOS devices is much higher than in NMOS devices.

The shift in V_{TH} of the PMOS undergoing NBTI is due to the generation of interface traps under negative gate-to-source bias. These interface traps are the result of breaking of weak Si–H bonds which are formed due to crystal mismatch at the channel–gate interface [5]. Fig. 1 from [3] illustrates the complete process of interface trap generation. The breaking of Si–H bonds at the $Si–SiO_2$ interface creates Si^+ (interface traps) and H atom. Initial interface-trap generation rate depends on Si–H bond dissociation (reaction), while the steady-state rate depends on H removal (diffusion) rates. The presence of Si^+ at the surface requires larger gate voltage for channel inversion. This is the reason why V_{TH} of a PMOS device increases due to NBTI. The negatively biased duration of the PMOS is said to be *stress* stage. Removal of the negative gate-to-source bias helps in annealing *some* of these interface traps, thereby leading to partial recovery. This phase is known as *recovery stage*.

As the internal nodes in a circuit switch during regular operation, each PMOS device experiences a sequence of stress and recovery phases. It must be noted that the recovery is never complete. Reference [6] noted that stressing a device even for 1% of the time followed by recovery phase for 99% of the time is still sufficient to slowly build up interface charge. However, the recovery phase is very important to be considered for the correct estimation of the NBTI effect. The lifetime estimation of NBTI without considering the recovery phase can be an order of magnitude lower than the actual value [7]. To account for the relative time of stress and recovery phases caused in a PMOS device due to its input, it is common to analyze the signal probability of the input pin. Since the NBTI effect happens due to negative bias (i.e., input gate voltage at logic LOW), we are interested in tracking probability that a given signal is at logic LOW. In the remainder of this paper, for any signal we will denote the probability of a signal to be at logic LOW as SP, for simplicity.

Clock is one of the most critical signals in the VLSI chip and special attention is paid to generate reliable, low-power clock trees while meeting stringent skew constraints. Clock

Manuscript received March 24, 2011; accepted May 16, 2011. Date of current version May 15, 2013. This paper was recommended by Associate Editor Y.-W. Chang.

A. Chakraborty is with Oracle Microelectronics, Austin, TX 78727 USA (e-mail: ashutoshc@ieee.org).

D. Z. Pan is with the Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX 78731 USA (e-mail: dpan@ece.utexas.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2012.2195002

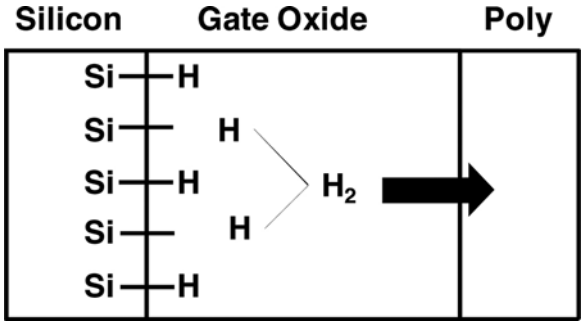

Fig. 1. Illustration of NBTI phenomenon. Breaking of Si–H bonds creates Si^+ interstitial causing V_{TH} increase.

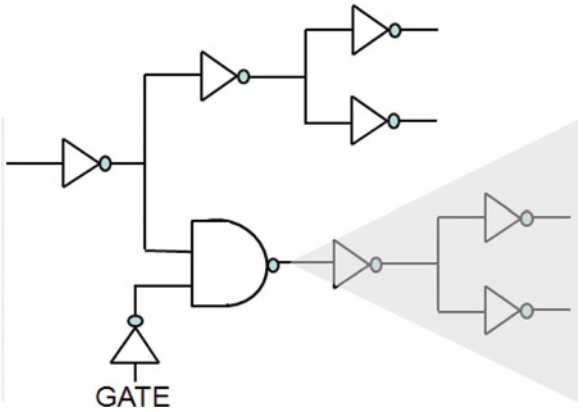

Fig. 2. Example of clock gating. Alternate PMOS in the shaded subtree experience constant stress during clock gating.

skew is defined as the maximum difference in the arrival times of the clock signal at all those sequential elements (flops) which can interact with each other due to the presence of a path between them. A large value of clock skew implies that the clock signal reaches the flops at very different times. Clock skew has one-on-one impact on the maximum frequency of operation of a chip; therefore, decreasing it is a major design concern. The minimum period of the clock signal is given by the following formula:

$$ClockPeriod >= T_{\text{cq}} + T_{\text{pd}} + T_{\text{skew}} + T_{\text{setup}} \quad (1)$$

where T_{cq} is the clock-to-Q delay of the flop, T_{pd} is the propagation delay through the combinational block, T_{skew} is the clock skew, and T_{setup} is the setup time of capturing flop. As can be seen from (1), reducing clock skew directly reduces the minimum time period of operation. A high-quality low-skew clock tree often requires symmetric clock tree topology and advanced clock signal routing with the aim of equalizing the arrival times of the clock signal at all the sequential elements. In recent times, due to several effects such as parametric variation, lack of high-quality interconnect models and environment factors such as spatial temperature variation, maintaining low clock skew has become a challenge. The challenge is to come up with schemes such that all the path delays can be made as close to each other as possible, thus maximizing frequency of operation.

All modern clock trees employ a clock gating technique which selectively shuts down unused parts of the clock tree. Typically, integrated clock gating (ICG) cells are inserted in the design, which conceptually are composed of a latch followed by AND/OR gate. The presence of latch avoids glitches and premature ending of clock signal. When using the NAND gate as the output stage of ICG, the value coming out of latch should be the controlling value, i.e., logic LOW. Similarly, when using the NOR gate at the output stage of ICG, the controlling value of logic HIGH needs to come out of latch. As the clock signal switches every cycle, the PMOS devices in clock buffer experience alternate stress and recovery phases of equal duration. However, PMOS devices that are part of heavily gated clock buffers do not experience stress and recovery for equal durations. The degradation of V_{TH} in these clock buffers can go out-of-sync compared to the rest of the clock buffers. This nonuniformity in V_{TH} degradation can cause a substantial increase in clock skew of a clock tree causing timing violations. In this paper, we propose a novel scheme to tackle this problem.

For example, consider the clock tree shown in Fig. 2 where the shaded subtree is currently gated by active low GATE signal. As a result of this, though the PMOS device inside a nongated clock buffer experiences alternate NBTI stress and recovery cycles, the PMOS inside a gated buffer undergoes constant stress. This may cause increase in skew of the whole clock tree. Also of importance is the workload-dependent temporal variation in the temperature of the clock buffers. Due to strong dependence of the NBTI effect on duty cycle and temperature, the study of its impact on skew degradation of clock tree is strongly mandated. This forms the motivation of our work.

In Section II, we will highlight the key previous works in this area and introduce the original contributions of this paper. Our proposed design technique along with the optimization formulation is described in Section III. In Section IV, we explain our experimental setup and the results achieved by our technique. The impact of variability on the clock tree generated by our technique is analyzed in Section V. We conclude our paper in Section VI.

II. Previous Works

NBTI and its effects have been considerably researched in recent years [3], [5], [8], [9]. These works include lifetime prediction of a chip considering AC stress, NBTI-aware timing analysis, reliability of memories due to PMOS V_{TH} degradation, and so on. The power-law dependence of NBTI on time [10] has been reported by most of the existing research. A predictive NBTI model based on physical understanding and published experimental data was presented in [9]. Reference [8] demonstrated analytical iterative equations for computation of NBTI impact for arbitrary waveforms. Numerical methods to solve the exact reaction/diffusion physics-based model of NBTI were proposed in [11]. The work in [1] proposed a tight upper bound of NBTI degradation for long-term computation. In [6], the impact of NBTI on circuit timing has been tackled by making logic synthesis aware of NBTI. The authors prepared a NBTI-aware delay look-up table along with propaga-

tion of the probability of signal to be at logic LOW, to synthesize circuits. The impact of input SP on typical gates, such as NAND, NOR, INV, and others, are also discussed in this paper. Though the above works are very important to understand the nature of NBTI effect, none of these works has targeted the interplay of NBTI and the skew of clock trees directly.

The patent [12] is the first work to combine the NBTI effect and clock skew together. This paper calculates the clock skew degradation due to NBTI and uses it to guard band the clock tree generation tool by half of the skew degradation. This paper has two main deficiencies: 1) their technique can overconstrain the clock tree synthesis tool, and 2) this method cannot work for large skew degradation. The first direct technique known to us which tries to combat the impact of NBTI in clock tree is [13]. Reference [13] proposed a scheme that relies on equalizing the SP of all clock tree trunks and derived a compact formula to compute equivalent temperatures for NBTI estimation under Gaussian temporal temperature variation. Their technique chooses at runtime whether the clock signal being gated should be frozen at logic 0 or logic 1. The choice is based on the value of a secondary, very slow clock. Though this is the first work to propose choice of stopping the clock at logic 0 or logic 1 for NBTI relief, there are several serious shortcomings of this approach: 1) the number of transistors in the proposed gating element is more than double compared to a clock buffer, which it replaces; 2) routing the second clock signal wastes precious routing resources; and 3) most importantly, switching of secondary independent clock can lead to spurious clock pulses and thus to logic failures.

More recently, [14] proposed an antiaging zero skew clock tree design by considering the critical PMOS transistors that lie on each clock root to clock tree leaf paths. This scheme requires that the clock tree should be type matching, i.e., all the gates lying at a particular distance down from the clock tree root are of the same type. For example, at a distance d down the clock root, in the case of binary cock tree there are 2^d clock buffers and if even one of these needs to be of clock gating type (say, using NAND gate), then all the 2^d gates at the dth level must be all NAND gates. Reference [14] formulated an integer linear program (ILP) to minimize the clock power while meeting zero skew. One of the key differentiating factors in our work with respect to [14] is that in our work we do not constrain ourselves to only type matching trees. Though the scheme proposed in [15] is novel, due to the way typical clock gating is implemented, its power and area cost can be substantial. This is because of the fact that in practice clock gating cells are not just NAND gates, they are large ICG cells consisting of a latch and NAND gate. Replacing all dth level gates with these large ICGs can add substantial power and area to the circuit. In our proposed scheme, we do not add any new ICGs to the circuit and only replace some of the NAND gates with NOR gates. One advantage of the scheme in [14] is the use of smaller *types* of gates (NAND and INV, i.e., two types) used in the clock tree compared to our scheme (NAND, NOR, and INV, i.e., three types). Using lesser types of gates can potentially improve resilience to process variation.

Our contribution: We propose a new scheme to exploit the ability of gating the clock tree at logic 0 or logic 1, which does not suffer from the above problems. Being a static approach (as opposed to the dynamic approach of [13]), generation and routing of secondary clock is avoided. There are no extra transistors required for implementing our technique and the impact of NBTI on the proposed gating element itself is already calibrated in our model. In addition, there are no spurious glitches in the clock signal.

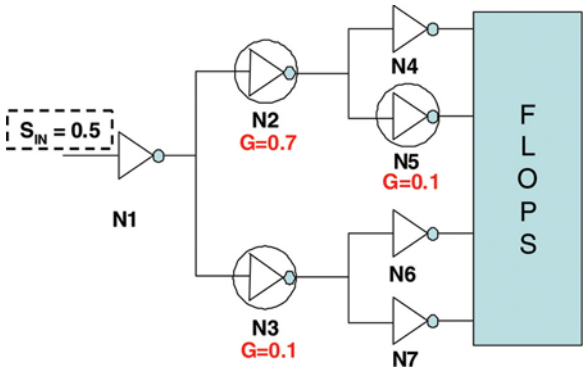

Fig. 3. Example of a small clock tree. Nodes that allow clock gating are circled and their probability of clock gating indicated by symbol G. Input SP was set as 0.5 (i.e., square wave clock signal).

III. Proposed Design Technique

The aim of the proposed design technique is to reduce the skew of a clock tree arising due to the asymmetry in the V_{TH} degradation of the clock buffers. This asymmetry is due to difference in probability of signal to be at logic LOW (SP) at different parts of the clock tree due to clock gating. We propose to use *both* NAND and NOR gates[1] (instead of just one of them) to implement clock gating. Using NAND (NOR) gate to shut down the clock allows freezing the clock tree at logic HIGH (LOW), thus decreasing (increasing) the SP for all clock buffers in the fanout cone. By intelligently choosing which gate to use for each clock gating element, the SP of the clock tree can be modulated to reduce the clock skew.

The input to our technique is a clock tree constructed using inverters or buffers. Using RTL simulation, clock gating opportunities at some of the clock inverters are identified. In traditional clock gating, these inverters would be replaced by ICGs with NAND gates at their output stage with the second input of NAND gate tied to active LOW clock gating enable signal. However, in our approach we would replace some of these ICGs with those that have NOR gate at their output stage. Owing to the different characteristics (such as intrinsic delay and input gate capacitance) of the NAND and NOR gates, the clock path delays may need to be tweaked by appropriate sizing of the NAND/NOR gates and moving the merging point of clock routing. For minute delay differences, techniques such as interconnect snaking can be applied if it does not introduce congestion. We next present an example to demonstrate the possibility of skew reduction by selecting ICGs with NAND or NOR gate.

[1]Using NAND or NOR gate is equivalent to using AND or OR gate with inverted signals. Since NAND/NOR gates are single-stage-like inverter, we prefer to use them. In clock trees containing buffers instead of inverters, our technique will choose between AND and OR gates.

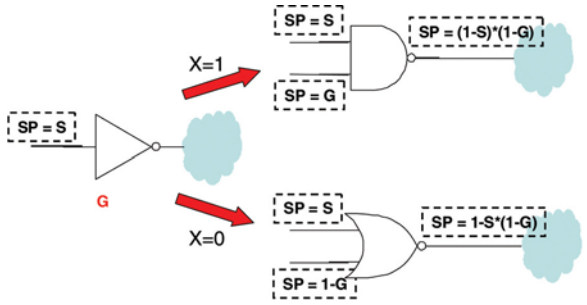

Fig. 4. Clock inverter (left) with input signal probability of S and probability of clock gating G can be replaced by either a NAND gate (right top) or a NOR gate (right bottom) with the indicated SP at their inputs. The output SP is also shown for both the cases.

Motivating example: Consider the clock tree shown in Fig. 3 that drives four latches. The clock tree nodes (represented as inverters indexed by the name under it) that have clock gating ability are circled and referred to as gated nodes. The nominal skew of the clock tree is zero due to symmetry. For the gated nodes, the probability of clock gating (G) of each node is also shown. A value of $G = 0$ implies that this particular node is never gated, whereas $G = 1$ means that clock is always gated. Assuming 50% duty cycle of clock at input (i.e., $S_{\text{in}} = 0.5$), we computed the skew at clock tree leaves using HSPICE after aging the circuit by 10 years. When all the gated nodes are implemented as NAND-gated ICGs, the skew of the clock tree is 1.90 ps, whereas for a configuration of all-NOR-gated ICGs, the clock skew is 1.36 ps. The best configuration is obtained when N2 = NOR, N3 = NOR, N5 = NAND with a value of skew of only 0.16 ps, a reduction of almost 90%. This proves that simply choosing all gates as NAND or all gates as NOR is not the right choice.

A. NAND/NOR-Aware SP Propagation

In this section, we set up the ground rules for propagation of SP and delay when implementing clock gating through NAND or NOR gates. Recall that the probability of a signal to be at logic LOW is denoted by SP. Consider a clock tree inverter shown on the left-hand side in Fig. 4.

The input SP of the inverter is S and the probability of clock gating is G. If this inverter is replaced by a NAND-gated ICG, the SP of the clock gating signal would be G itself because logic LOW is the controlling value for NAND. In such a scenario, the output SP of the NAND gate is $(1-G)*(1-S)$. On the other hand, if the inverter is replaced by a NOR gate, the SP of the clock gating signal would be $(1-G)$. This is because logic HIGH is the controlling value of NOR and gating probability (GP) of G means $(1-G)$ period of nongating when the logic LOW is present. The output SP can then be obtained as $1-S*(1-G)$. Let the binary variable X represent this choice between using NAND or NOR gate. $X = 1$ implies using NAND gate for clock gating and $X = 0$ implies choosing NOR gate. For the regular inverters in the clock tree (i.e., those that do not have clock gating ability), the output SP is trivially equal to $(1-S)$. Let us assume the delay of an INV, NAND, and NOR gate is $\text{D}_{INV}(S, G)$, $\text{D}_{NAND}(S, G)$, and $\text{D}_{NOR}(S, G)$, respectively, which are functions of the switching probability (S) and clock GP (G). If $X = 1$, the delay of the cell is $\text{D}_{NAND}(S, G)$, else it is $\text{D}_{NOR}(S, G)$. Delay of a clock inverter that does not clock gating capability is simply $\text{D}_{INV}(S, G)$. Table I summarizes these observations that are used for propagating the symbolic SP and delay values through the clock tree from the root to the clock leaves.

TABLE I
FORMULA FOR OUTPUT SP AND DELAY OF DIFFERENT GATES

Choice	Variable X	Output SP	Delay
NAND	1	$(1-G)*(1-S)$	$\text{D}_{NAND}(S, G)$
NOR	0	$1-S*(1-G)$	$\text{D}_{NOR}(S, G)$
INV	–	$1-S$	$\text{D}_{INV}(S, G)$

Binary variable $X = 1$ if NAND is chosen, 0 if NOR is chosen.

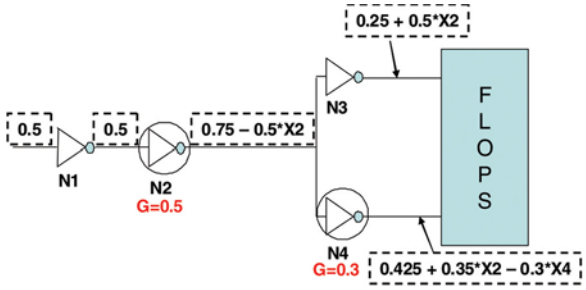

Fig. 5. Example showing the propagated value of SP (in dashed boxes) as a function of $X2$ and $X4$ indicating the choice between NAND and NOR gates at nodes N2 and N4, respectively. Clock GP used for calculation is represented by symbol G.

The information from Table I can be combined to get the following expression for the output SP and delay through a clock gating enabled gate in terms of the binary variable X:

$$SP_{\text{out}} = 1 + S * G - S - X * G \tag{2}$$

$$D = X * D_{NAND}(S, G) + \overline{X} * D_{NOR}(S, G). \tag{3}$$

We prove an important property of the delay expression of any gate, which will help make the ILP formulation tractable later in this paper.

Lemma 1: Signal probability of any gate is at most a linear function of X.

Proof: Consider (2). If input signal probability, S, is linear or constant in X then output SP_{out} is also linear since G is a constant. As SP_{out} becomes the input signal probability for a fanout gate, the linearity property remains recursively true. As the base case, SP of clock tree root is a constant number. ■

Lemma 2: If cell delay is linearly dependent on input signal probability, delay expression of any gate is at most a quadratic function of X.

Proof: From the above lemma, signal probability is a linear function of X. From (3), the delay expression is a linear combination of two expressions where X is multiplied by the delay of the NAND or NOR cell. As long as the delay of each cell is a linear function of the input signal probability, the delay expression is at most a quadratic function of X. ■

Using the above expressions as well as that for an inverter from Table I, we can start at the root of the clock tree and recursively compute the symbolic SP and the delay from the

root of the clock to each leaf level sink. An example of this is as follows.

Example 1: Consider the toy clock tree shown in Fig. 5, where each clock tree node that implements clock gating is circled. Let the binary variables $X2$ and $X4$ represent the choice of NAND/NOR gate at the nodes N2 and N4. The probability of clock gating (G) is written next to the corresponding nodes. The propagated SP of the path from the clock root to the clock tree leaves are computed for each net based on G and the binary variables $X2$ and $X4$ and noted along the net. Ignoring dependence of NOR gate's delay on G,[2] the delay at upper leaf can be written as $[D_{INV}(0.5)] + [X2 * D_{NAND}(0.5) + \overline{X2} * D_{NOR}(0.5)] + [D_{INV}(0.75 - X2*0.5)]$. Similarly, the delay at the lower leaf can be written as $[D_{INV}(0.5)] + [X2 * D_{NAND}(0.5) + \overline{X2} * D_{NOR}(0.5)] + [X4 * D_{NAND}(0.75 - X2*0.5) + \overline{X4} * D_{NOR}(0.75 - X2*0.5)]$.

From Example 1, we note that the choice of using NAND and NOR gates (i.e., variables $X2$ and $X4$) not only modifies the delay function of the clock path, but also modulates the SP at the fanout cone affecting the delay of output gates. Therefore, properly making this choice can help in reducing clock skew.

B. SP and GP-Aware Delay Model

We sized the NAND and NOR gates to match their rise and fall delay to those of an inverter (INV). In this way, replacement of the INV by any other gate will not change the nominal clock skew. The ratios of PMOS to NMOS width for INV, NAND, and NOR gates in our library that achieved this iso-delay setting are 2.2, 1.36, and 4.46, respectively. The delay computed through HSPICE has a nominal value of 22.69 ps for fanout-4 load at 50 °C. The next step is to characterize the delay of these cells as a function of clock SP and GP. Since NBTI does not impact the output load capacitance of the gate in any way, the load-dependent delay is ignored for degradation analysis. Our aim for delay characterization is to extract simple high fidelity approximations to guide the optimization engine in the right direction. Therefore, we extensively use linearization of near-linear curves.

To consider the impact of SP on delay, we first need to relate SP to V_{TH}. We computed the V_{TH} degradation as a function of SP using the s_k model of [8], extensively employed in other works such as [6], [13], and so on. Using the obtained V_{TH} values, we performed SPICE simulation to obtain the rise and fall delay of the NAND, NOR, and inverter gates. Since NBTI impacts only PMOS devices, the fall delay of the gates was observed to be nearly constant for all SP. However, the rise delay of these gates varies by as much as 10% when the SP increases from 0 to 100%. The dashed curves in Fig. 6 show the rise delay of INV, NAND, and NOR gates as a function of SP.

There is a large increase in delay degradation near very low value of SP of approximately 5%. However, the curve flattens out for larger values of SP. This observation is consistent with those obtained by other authors such as in [6]. Similar to [6], to model this behavior, we performed piecewise linear

[2]This dependence is characterized later in Section III-B.

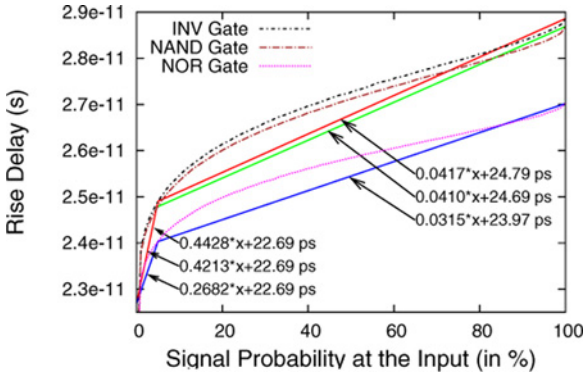

Fig. 6. Rise delay of INV, NAND, and NOR gates as a function of input SP. Initial sharp increase is observed. Change in slope near 5% SP motivated the piecewise linear delay model.

fit for the case of SP $\leq$ 5% and for SP $>$ 5% obtained through the Gnuplot tool with R^2 coefficient of fit as 0.88. The actual curves and the fit were forced to coincide at SP = 0%, SP = 5%, and SP = 99%. These linear fits are as follows:

$$D_{INV}(SP) = \begin{cases} (0.4428*\text{SP} + 22.69)\,\text{ps}: & SP \leq 0.05 \\ (0.0417*\text{SP} + 24.79)\,\text{ps}: & SP > 0.05 \end{cases}$$

$$D_{NAND}(SP) = \begin{cases} (0.4213*\text{SP} + 22.69)\,\text{ps}: & SP \leq 0.05 \\ (0.0410*\text{SP} + 24.69)\,\text{ps}: & SP > 0.05 \end{cases}$$

$$D_{NOR}(SP) = \begin{cases} (0.2682*\text{SP} + 22.69)\,\text{ps}: & SP \leq 0.05 \\ (0.0315*\text{SP} + 23.97)\,\text{ps}: & SP > 0.05. \end{cases}$$

From the previous discussion, it is clear that clock SP has direct impact on the delay of the fanout gate. Next, we consider the impact of GP of a NAND/NOR gate on its own delay. Both NAND and NOR gates have two PMOS transistors driven by two separate pins. One of the input pins is driven by the clock signal from the previous stage of clock tree with probability of logic LOW as SP and the other pin is driven by gating enable signal latched in the ICG with probability of logic LOW equal to GP. In the case of NAND gate, due to parallel paths to V_{DD} through the two PMOS devices, even if the PMOS connected to gating enable signal degrades, the net impact on the rise time is negligible. On the other hand, in the case of NOR gate, different values of gating enable probability lead to different V_{TH} degradation of the PMOS driven by a gating enable signal. This directly affects the pull-up capability of the NOR gate due to the inherent PMOS stack in it. In short, for a NOR gate we must consider the impact of degradation of both PMOS transistors. This was first pointed out by [6].

To capture this effect, we simulated the rise delay of the NOR gate as a function of the V_{TH} degradation of PMOS driven by an input clock for different gating enable probabilities driving the second PMOS input. Fig. 7 shows the rise delay of the NOR gate as a function of the GP at it on the x-axis (i.e., how frequently it is gated), for different SP of the clock pin. From this figure, it is immediately visible that GP can play a significant role in determining the rise delay of the NOR gate due to the stacking effect. Fortunately, in the most relevant range of signal probabilities the delay dependence can be very closely approximated by a linear dependence.

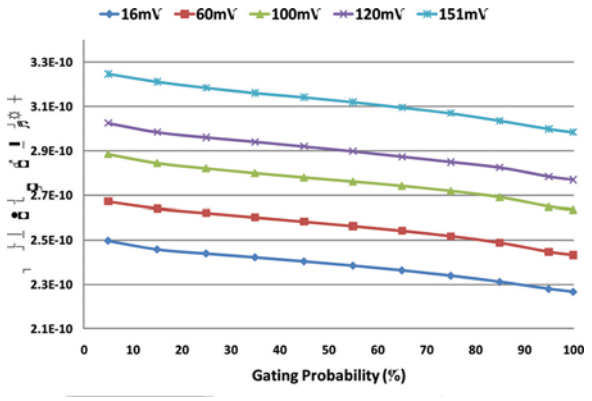

Fig. 7. Rise delay of NOR gate as a function of clock GP for various NBTI degraded V_{TH} values at the clock input pin.

Clearly, the higher the GP at the gating enable signal (i.e., of clock being gated), the higher the proportion of the time logic HIGH (controlling value for NOR gate) is fed to the NOR gate, which translates into *lower* NBTI degradation. The rise delay of the NOR was observed to decrease approximately 8% in a near-linear manner when the clock GP varies from 0% to 100%. Hence, we incorporated this GP dependence by linearly scaling the NOR delay as follows:[3]

$$D_{NOR}(SP, GP) = D_{NOR}(SP) * (1 - 0.08 * GP). \tag{4}$$

Using the expression for dependence on SP and GP, now we can analytically write the delay of each of the three cells for any combination of these variables. These expressions can be used for optimizing clock skew of large scale circuits using integer programming formulation described next.

C. Skew Reduction Formulation

Using the models developed in the previous section, we will present our optimization program formulation for skew reduction of a clock tree in the presence of NBTI degradation after 10 years of aging. Let the set of sinks in the clock tree be given as S. For the ith sink s_i, using the piecewise linear delay model developed in Section III-B, we can obtain the formula for arrival time of the clock signal. Obviously, the arrival time is a function of the SP of the clock interconnects and GP of clock buffers connecting sink s_i to the clock signal root. This can be represented as $AT_i(X_i, SP_i)$, where X_i and SP_i capture the binary variables for the choice of NAND/NOR clock gating and signal probabilities along the path.

There are two interesting problems that can be formulated. The first is an *optimization* problem to identify the optimal choice of NAND/NOR gate configuration to minimize the skew of the clock tree. This case is of special importance for high performance designs or when the clock tree structure is already fixed but timing closure is difficult to achieve. The second problem is a satisfiability problem which, given a clock tree structure, decides whether there is any configuration of NAND/NOR assignment that meets a particular skew constraint after circuit aging. We will focus on the *optimization* problem due to its more practical use. Consider the following formulation:

$$\begin{aligned} &\text{Minimize } (MAX - MIN) \\ &\text{subject to} \\ &AT_i(\{X_i\}, \{SP_i\}) \le MAX \quad \forall i \in S \\ &AT_i(\{X_i\}, \{SP_i\}) \ge MIN \quad \forall i \in S \\ &X_i \in \{0, 1\} \quad \forall i \in S \\ &MAX \ge 0 \\ &MIN \ge 0. \end{aligned} \tag{5}$$

In this formulation, *MAX* and *MIN* are dummy variables that represent the largest and the smallest arrival time of the clock signal among all sinks, as indicated by the first two constraints over all sinks. All the X_i variables are constrained to be binary. By minimizing the objective $(MAX-MIN)$, we are effectively minimizing the clock skew of the whole clock tree. For a clock tree with n sinks, the number of constraints is clearly O(n). Assuming a balanced tree structure, there are log(n) levels thus each of the $AT_i(X_i, SP_i)$ has at most O(log(n)) binary variables.

The expression of arrival time contains multiplication of binary variables which can cause solvers to fail. In Section III-A, we proved that the delay expression can have multiplication of at most two binary variables. To decompose such expressions, we use the following transformation. Let X_A and X_B be the two binary variables whose multiplication appears in arrival time expression. We introduce a new binary variable X_{AB} such that

$$\begin{aligned} X_A + X_B &\le 1 + X_{AB} \\ (1 - X_A) + (1 - X_B) &\le 2 - 2 \times X_{AB}. \end{aligned}$$

By replacing $X_A \times X_B$ by X_{AB}, and adding the above constraints to the ILP, the new problem is equivalent but without any multiplication of binary variables.

Consider a design with 100k cells out of which 10%, i.e., 10k of these cells may be flops. Assuming each leaf level clock buffer can drive ten flops, the clock tree will have 1k leaf level clock buffers or approximately 2k total clock buffers in the clock tree. Even if 10% of these clock buffers are gating enabled, we have at most 200 binary variables. Therefore, the above formulation can be solved readily by solvers, branch and bound techniques, or simulated annealing methods to get the optimal solution for skew minimization.

IV. Experimental Setup and Results

For all our transistor level simulations, we used the postextraction SPICE models from the open-source 45 nm nangate library [15]. Simulations were run using Synopsys HSPICE Version A-2008.03-SP1. A C++ program was written to symbolically propagate the SP, to compute the symbolic delay equation of each clock sink and to write the optimization problem. We used the tool CPLEX to solve our optimization

[3]As explained earlier, NAND and INV gates do not have any dependence on GP.

TABLE II
CLOCK TREE BENCHMARKS USED IN THIS PAPER

Name	Depth	Fanout	# Buf^{+}	# Sinks*	# Gated
A	7	4	21 845	87 380	331
B	8	3	9841	8748	144
C	9	3	29 524	26 244	426
D	8	4	87 381	348 520	1251
E	9	3	29 524	26 244	430
F	8	3	9841	8748	138
G	8	4	87 381	348 520	1267
H	7	4	21 845	87 380	326

Depth of the tree, the fanout of each clock inverter, the number of buffers, sinks (flops) and clock gating enabled inverters shown in consecutive columns.
$^{+}$Number of buffers include all levels, not just leaves.
*Number of sink assuming fanout of 4 at leaves.

problem. For managing the long symbolic expressions, we used the symbolic expression simplifier built in the tool Mathematica. All the above steps were performed on a dual-core 2.67 GHz workstation running the Linux operating system. All skew numbers are obtained as the value of the objective function after solving (5) either to optimality (for our proposed method) or under predetermined NAND/NOR assignment. In other words, we did not run the clock tree synthesis tool after our optimization.

For benchmarks, we generated several instances of clock trees with varying levels (depth of clock tree) and the fanout number of each clock buffer. Approximately, 2% of the clock buffers were randomly picked to be gating enabled. The gating ratio of these buffers was chosen randomly between 20% and 70%. The reason why we do not allow clock gating ratio of less than 20% is that clock gating does not come for free. There can be appreciable power dissipation in the logic gates that generate a signal whether a certain logic block must be gated in the next clock cycle or not. The power dissipation of such auxiliary gates must be weighed in against the possible savings in power by clock gating of that module. In other words, we assume that if a module is going to be turned off less than 20% of the times, the cost of clock gating signal generation may be more than power savings expected. The input SP to the clock tree root was assumed to be 50%. Because of perfect symmetry of output load and matching of delay of NAND/NOR gate to that of the inverter, the initial clock skew for all benchmarks is 0 ps. Table II shows the characteristics of the benchmarks used in this paper. In the table, the second column onward contains the depth of the clock tree, the number of fanout nodes of each clock buffer, the total number of buffers and flops at the sinks of the clock tree. The last column shows the number of buffers that have clock gating capability. Each one of such buffers is associated with one binary decision variable for choosing NAND or NOR implementation.

Table III contains our results. Column 2 shows the CPU time to run the ILP solver on our optimization program. The next two columns show the fraction (in %) of the clock gating buffers converted to NAND and NOR gates, respectively, in our solution. The skew reported by our optimization flow is shown in column 6. Next, we compare our solution to the three strategies: choosing all NAND gates (symbol $\forall$ NAND), choosing NOR gates (symbol $\forall$ NOR), and running ten random assignments of NAND and NOR gates and picking the best among these. For each one of these strategies has two columns in the table. The first one reports the skew value obtained by that strategy and the second column contains the % penalty in skew that this strategy has over our optimal solution.

From the skew numbers in Table III, we observe that our proposed method gives very good results. As compared to our optimal clock gating solution, the clock skew of designs implemented using all-NAND, all-NOR, and random choices is, on average, 56%, 219%, and 133% higher. This proves that the use of our method can significantly tighten the skew budget helping high performance designs. In some of the benchmarks (see benchmark D in Table III), the optimal solution was only 37% better than the trivial solution of using all-NAND gates. However, on other occasions, our solution was up to 74% better. In general, the skew when using only NAND gates was lesser than using only NOR gates. We believe this is due to the different delay dependence curves of NOR gate in Fig. 6, compared to the seemingly similar curves for INV and NAND gates. However, it does not mean that the optimal solution has most of the clock gated buffers chosen to be NAND gates. From the runtime perspective, we note that the maximum CPU runtime for solving these testcases is less than 2 s. Benchmark G, which has the largest number of gated clock buffers, is solved in less than half a second.

Validation: In Section III-B, we extensively used linearization and approximations for developing easy model for optimization program generation. In this process, we would have lost some accuracy. However, as long as the optimization program is guided in the right direction, a good solution will be achieved. To check how accurate our approximations were, we did the following experiment. For all benchmarks, we computed the clock skew directly using SPICE simulation of the fastest and slowest clock paths for each of the three configurations: all NAND, all NOR, and the optimal configuration obtained by us. Though the exact skew numbers were different from the numbers reported in Table III, the penalty of using the configuration of all-NAND and all-NOR gates matched with our results. For example, in the case of benchmark D, the skew penalty reported by our model is 56% and 219% for all-NAND and all-NOR, respectively, while the HSPICE returned numbers are 58% and 216% respectively. This proves that our linearized model has good fidelity and can be used for skew optimization.

V. IMPACT OF VARIABILITY

High quality variation tolerant clock tree design relies on making the clock tree as symmetric as possible. This is achieved by having the same *signature* of each clock source to sink (i.e., latch) path. This signature is comprised of the interconnect length, number of clock buffers inserted, and the sizing of the clock buffers used in each source to sink path. In a clock tree without clock gating, it is relatively easier to obtain such symmetry though at the cost of power or chip area. In the case of traditional clock gating, the clock trees are exclusively made up of clock buffers and ICG elements with NAND gate at their output. Even for this clock gating scheme,

TABLE III
CLOCK SKEW ACHIEVED BY OUR SOLUTION IS COMPARED WITH ALL-NAND, ALL-NOR, AND RANDOM BEST-AMONG-10-TRIALS STRATEGIES

Name	Solver CPU (s)	% of NAND	% of NOR	Our Skew (ps)	∀ NAND Skew (ps)	Penalty (%)	∀ NOR Skew (ps)	Penalty (%)	Rand* Skew (ps)	Penalty (%)
A	0.14	77%	23%	2.80	4.41	57.50%	9.02	299.03%	7.24	158.57%
B	0.06	97%	3%	2.18	3.23	48.26%	5.84	167.28%	4.96	125.68%
C	1.41	71%	29%	4.13	6.64	56.14%	9.28	124.69%	7.05	70.70%
D	0.81	81%	19%	3.03	5.04	37.81%	9.74	221.45%	6.21	104.95%
E	0.12	73%	27%	2.76	5.46	66.33%	10.21	269.92%	7.04	225.92%
F	0.09	60%	40%	3.94	6.21	57.61%	12.23	210.40%	11.82	200.00%
G	0.47	77%	23%	3.88	6.75	73.94%	13.07	237.11%	10.58	172.84%
H	0.09	83%	17%	2.59	3.91	50.95%	8.44	225.86%	5.38	107.72%
Avg.		77%	23%			56.07%		219.45%		133.75%

Data preparation time and solver time are reported in seconds. All "penalty" columns show the extent of penalty (in %) of using the corresponding technique instead of the proposed optimal solution.
*Best result chosen among ten random tries.

there is inherently a source of asymmetry because not all paths from the clock source to sink have clock gating enabled on them. The clock tree with a traditional clock gating scheme comprises INV and NAND gates and the delay response of these two types of gates to various variability sources could be different. This asymmetry can lead to increase in clock skew in the presence of variations.

One concern with our proposed approach is that instead of *two* types of gates that are used in traditional clock gating methodology (i.e., INV and NAND), the proposed technique uses *three* types of gates by adding NOR gate in the clock tree. This can further increase the asymmetry between the different paths in the clock tree leading to higher skew. The results from Section IV show that our proposed technique can significantly reduce clock skew due to NBTI degradation of the various gates; however, that analysis is done without considering any device or operation condition variations.

To fully understand how the proposed technique performs in the presence of variation, we performed the following experiments. We identified few sources of variation and subjected the clock trees to these variation conditions one at a time. The skew of the two clock trees: 1) traditional clock gating (with INV and NAND gates), and 2) proposed approach (with INV, NAND, and NOR gates) was computed under these operating conditions. In all cases, it was assumed that the clock skew at the beginning of the clock tree's lifetime ($T = 0$) *under nominal condition* is zero, which is practical because typically the clock buffer library is rich and the correct gate sizes for the gates can be picked to satisfy this condition. Certainly, under variation even the starting skew is nonzero. The value of clock skew under variation computed in this section is at the end of life (i.e., $T = 10$ years) of the VLSI product.

A. Temperature Variation

In this experiment, we changed the spatially uniform operating temperature of the clock tree and computed the clock skew. Though it is possible to also vary the temperature spatially, however, as the different clock gates (INV, NAND, NOR) can be spread randomly across the chip, the difference in their delay variation due to different sensitivity to temperature variation can possibly average out. In that sense, spatially uniform temperature variation gives a deterministic way to capture the impact of variation. Fig. 8 shows the variation in the delay of the three types of gates as a function of operating temperature.

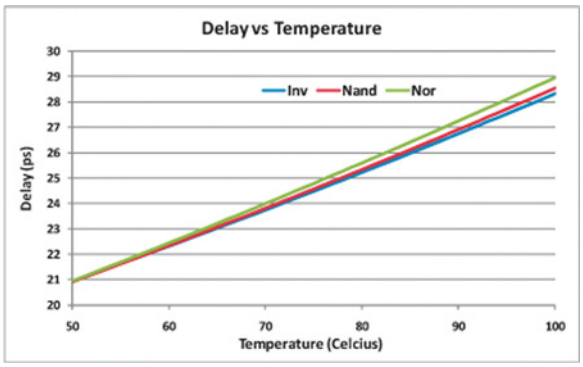

Fig. 8. Delay of INV, NAND, and NOR gates as a function of operating temperature.

It is evident from the figure that the behavior of the NAND gate remains similar to that of INV gate, whereas the NOR gate's delay diverges slightly more. Since skew is computed by difference in the delay of different clock paths, even small difference in the absolute delay values of the gates can translate into significantly higher skew. We computed the clock skew for our benchmark circuits for operating temperatures of 60 °C, 80 °C, and 100 °C, respectively. Fig. 9 shows the clock skew by using the traditional clock gating that uses INV and NAND gates versus our proposed technique that uses INV, NAND, and NOR gates under temperature variation. To preserve image quality, only the data for benchmarks A, C, and E are shown though the data from other benchmarks follow similar trends.

From Fig. 9, we observe that with variation in operating temperature, in general, the saving in clock skew achieved by our approach reduces as compared to the savings under nominal operating condition. Due to different temperature sensitivity of the constituent gates (as seen in Fig. 8), the fastest and slowest path from root to the clock tree sinks can change at different temperatures. This can lead to some outliers such as benchmark "C" in Fig. 9 for which the skew reduction actually improves at lower temperatures. This is because while the traditional gating suffers from different delay variations of NAND and INV gates, clock tree built with our approach also

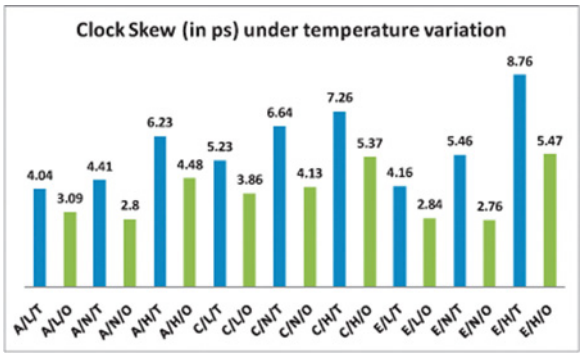

Fig. 9. Clock skew for different benchmarks under temperature variation for benchmarks A, C, and E. x-axis identifies three properties separated by "/." Benchmark name (A, C, E) followed by temperature type (L: low, N: nominal, H: high), followed by clock tree type (T: traditional, O: ours). All blue bars are for traditional clock gating, green bars are for our proposed scheme.

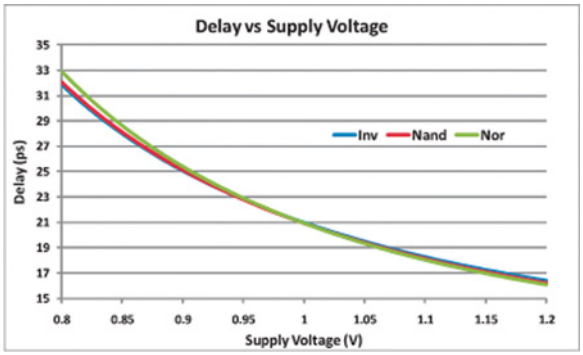

Fig. 10. Delay of INV, NAND, and NOR gates as a function of operating supply voltage.

suffers from delay variation of the NOR gates. Over these three benchmarks, the clock skew saving reduces from 41% under nominal condition to 27% under low-temperature, and 30% under high-temperature operating conditions.

B. Supply Voltage Variation

The timing behavior and functionality of a logic circuit is susceptible to both power (VDD) and ground (GND) voltage variation. However, NBTI-degraded circuits are more susceptible to VDD variations (than GND variation of the same amount), because NBTI weakens the PMOS devices which are responsible for injecting charge from VDD rails to the logic gates. In this section, we measured the variation of clock skew of the clock tree by modifying the power supply voltage (VDD) of the circuit by a spatially uniform value. Fig. 10 shows the variation in the delay of the three types of gates as a function of supply voltage.

From Fig. 10, it is evident that the NOR gate's delay is most divergent with variation in supply voltage. Note that all the three gates are made to be iso-delay at a nominal operating voltage of 1 V. At operating voltages lesser than the nominal voltages, the delay of the NOR gate is the largest among the three gates. However, at operating voltages higher than the nominal voltages, the delay of the NOR gate is least among the three gates. We also notice that the behavior of the INV and NAND gates is relatively similar. Fig. 11 shows the clock skew

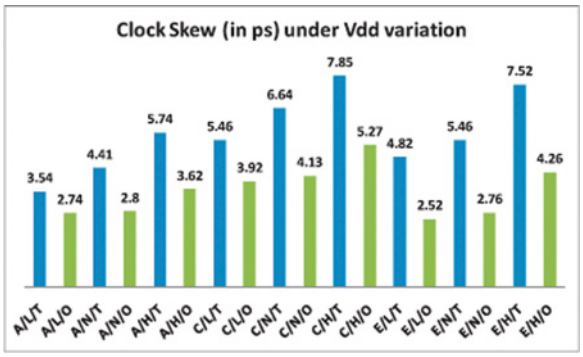

Fig. 11. Clock skew for different benchmarks under voltage variation for benchmarks A, C, and E. x-axis identifies three properties separated by "/." Benchmark name (A, C, E) followed by supply voltage type (L: low, N: nominal, H: high), followed by clock tree type (T: traditional, O: ours). All blue bars are for traditional clock gating, green bars are for our proposed scheme.

by using the traditional clock gating that uses INV and NAND gates versus our proposed technique that uses INV, NAND, and NOR gates under operating supply voltages of 0.9 V, 1.0 V, and 1.1 V.

From Fig. 11, we observe similar behavior as observed for temperature variation. The saving in clock skew achieved by our approach reduces as compared to the savings in nominal operating condition under supply voltage variation. Over these three benchmarks, the clock skew saving reduces from 41% under nominal condition to 32% under low supply voltage, and 37% under high supply voltage operating conditions.

VI. Discussions and Conclusion

In this paper, for the first time we proposed a static method for controlling NBTI-degraded clock skew due to clock gating. Our method relies on a design time intelligent choice of determining which clock buffer will freeze the clock at logic 0 (using NOR gate) or logic 1 (using NAND gate) during clock gating. This choice provides us with two degrees of freedom: first, the choice of the gate changes the delay function of that clock branch and second, it modulates the SP of the fanout cone, affecting the delay of gates downstream. We derived high fidelity piecewise linear models and corrective terms for computing the impact of SP at the input and the degradation of the clock gating element itself. The skew minimization problem was formulated as an ILP and solved using commercial solvers. By exploiting our technique, we were able to reduce the NBTI-induced clock skew by up to 74% compared to all-NAND implementation and 300% compared to all-NOR implementation. The impact of supply voltage and temperature variation on the proposed clock tree methodology was explored and we observed increased sensitivity to variation using the new approach due to the presence of a different type (NOR) gate compared to the traditional clock gating structure.

Our future work will be targeted toward finding ways of performing clock tree synthesis aware of NBTI instead of fixing the clock tree later. In addition, we will explore the possibility of applying our proposed technique for clock trees with prescribed skew.

REFERENCES

[1] S. Bhardwaj, W. Wang, R. Vattikonda, Y. Cao, and S. Vrudhula, "Predictive modeling of the NBTI effect for reliable design," in *Proc. IEEE Custom Integr. Circuits Conf.*, Sep. 2006, pp. 189–192.

[2] W. Wang, S. Yang, S. Bhardwaj, R. Vattikonda, S. Vrudhula, F. Liu, and Y. Cao, "The impact of NBTI on the performance of combinational and sequential circuits," in *Proc. ACM/IEEE Des. Autom. Conf.*, Jun. 2007, pp. 364–369.

[3] M. Alam, H. Kufluoglu, D. Varghese, and S. Mahapatra, "A comprehensive model for PMOS NBTI degradation: Recent progress," *Microelectron. Reliab.*, vol. 47, no. 6, pp. 853–862, 2007.

[4] F. Crupi, C. Pace, G. Cocorullo, G. Groeseneken, M. Aoulaiche, and M. Houssa, "Positive bias temperature instability in nMOSFETs with ultrathin Hf-silicate gate dielectrics," *Microelectron. Eng.*, vol. 80, pp. 130–133, Jun. 2005.

[5] S. Kumar, K. Kim, and S. Sapatnekar, "Impact of NBTI on SRAM read stability and design for reliability," in *Proc. Int. Symp. Quality Electron. Des.*, Mar. 2006, pp. 213–218.

[6] S. Kumar, C. Kim, and S. Sapatnekar, "NBTI-aware synthesis of digital circuits," in *Proc. ACM/IEEE Des. Autom. Conf.*, Jun. 2007, pp. 370–375.

[7] G. Chen, K. Chuah, M. Li, D. Chan, C. Ang, J. Zheng, Y. Jin, and D. Kwong, "Dynamic NBTI of PMOS transistors and its impact on device lifetime," in *Proc. IEEE Int. Reliab. Phys. Symp.*, Mar. 2003, pp. 196–202.

[8] S. Kumar, C. Kim, and S. Sapatnekar, "An analytical model for negative bias temperature instability," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, Nov. 2006, pp. 493–496.

[9] R. Vattikonda, W. Wang, and Y. Cao, "Modeling and minimization of PMOS NBTI effect for robust nanometer design," in *Proc. ACM/IEEE Des. Autom. Conf.*, Jun. 2006, pp. 1047–1052.

[10] S. Mahapatra, D. Saha, D. Varghese, and P. Kumar, "On the generation and recovery of interface traps in MOSFETs subjected to NBTI, FN, and HCI stress," *IEEE Trans. Electron Devices*, vol. 53, no. 7, pp. 1583–1592, Jul. 2006.

[11] R. Vattikonda, Y. Luo, A. Gyure, X. Qi, S. Lo, M. Shahram, Y. Cao, K. Singhal, and D. Toffolon, "A new simulation method for NBTI analysis in SPICE environment," in *Proc. Int. Symp. Quality Electron. Des.*, Mar. 2007, pp. 41–46.

[12] J. M. Cohn, "Method for reducing design effect of wearout mechanisms on signal skew in integrated circuit designs," U.S. Patent 6651230, Nov. 2003.

[13] A. Chakraborty, G. Ganesan, A. Rajaram, and D. Pan, "Analysis and optimization of NBTI induced clock skew in gated clock trees," in *Proc. Des. Autom. Test Eur.*, Apr. 2009, pp. 296–299.

[14] S.-H. Huang, C.-M. Chang, W.-P. Tu, and S.-B. Pan, "Critical-PMOS-aware clock tree design methodology for anti-aging zero skew clock gating," in *Proc. ASP-DAC*, Jan. 2010, pp. 480–485.

[15] *NanGate 45nm PDK Release v1.3* [Online]. Available: http://www.nangate.com.

Ashutosh Chakraborty (S'04–M'11) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology Delhi, New Delhi, India, in 2002, and the M.S. and Ph.D. degrees in computer engineering from the University of Texas, Austin, in 2008 and 2010, respectively.

Since 2010, he has been with Oracle Microelectronics, Austin, as a Senior Hardware Engineer. From 2002 to 2004, he was a Senior Member of Technical Staff with Mentor Graphics, Noida, India, working on emulation products. From 2004 to 2006, he was a Research Assistant with the Politecnico di Torino, Turin, Italy. He was with AMD, Austin, in 2007, with Synopsys, Inc., Hillsboro, OR, in 2009, and with the IBM T. J. Watson Research Center, Yorktown Heights, NY, in 2010, as an Intern. He has published over 20 refereed papers in international conferences and journals. His current research interests include physical design considering the impact of mechanical stress and device reliability on nanometer integrated circuits.

Dr. Chakraborty received the Best Interactive Presentation Award at DATE 2009, the Best Paper nomination at ISPD in 2010, the eASIC Placement Contest Grand Prize in 2009, the University of Texas Continuing Fellowship in 2009, the Government of Italy Fellowship in 2004, and the Rajiv Bhambhavle Best Undergraduate Thesis Award in 2002. He has served as a reviewer for several international conferences and journals, including DAC, SLIP, ICECS, TCAD, TVLSI, TCAS-I, TCAS-II.

David Z. Pan (S'97–M'00–SM'06) received the B.S. degree from Peking University, Beijing, China, and the M.S. and Ph.D. degrees from the University of California, Los Angeles (UCLA).

From 2000 to 2003, he was a Research Staff Member with the IBM T. J. Watson Research Center, Yorktown Heights, NY. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, University of Texas, Austin. He has published over 160 papers in highly refereed conferences and journals, and holds eight U.S. patents. His current research interests include nanometer physical design, design for manufacturability and reliability, vertical integration design and technology, and design or computer-aided design (CAD) for emerging technologies.

Dr. Pan has served as an Associate Editor for the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION SYSTEMS, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I and II, the *Journal of Computer Science and Technology*, and the IEEE CIRCUITS AND SYSTEMS SOCIETY NEWSLETTER. He has served as the chair of the IEEE CANDE Committee and the ACM/SIGDA Physical Design Technical Committee. He is in the Design Technology Working Group of the International Technology Roadmap for Semiconductors. He has served in the technical program committees of major very large scale integration (VLSI)/CAD conferences, including ASPDAC (Subcommittee Chair), DAC (Subcommittee Chair), DATE, ICCAD (Subcommittee Chair), ISPD (Program Chair), ISQED (Topic Chair), ISCAS (CAD Track Chair), SLIP (Publication Chair), GLSVLSI, ACISC (Program Co-Chair), ICICDT (Award Chair), and VLSI-DAT (EDA Track Chair). He was the General Chair of ISPD 2008, the General Co-Chair of ACISC 2009, and the Steering Committee Chair of ISPD 2009. He received a number of awards for his research contributions and professional services, including the ACM/SIGDA Outstanding New Faculty Award in 2005, the NSF CAREER Award in 2007, the SRC Inventor Recognition Award three times, the IBM Faculty Award four times (2004 to 2006, 2010), the UCLA Engineering Distinguished Young Alumnus Award in 2009, eight Best Paper Awards (ASPDAC 2012, ISPD 2011, the IBM Research 2010 Pat Goldberg Memorial Best Paper Award in CS/EE/Math, ASPDAC 2010, DATE 2009, ICICDT 2009, and SRC Techcon in 1998 and 2007), nine other Best Paper Award nominations from DAC/ICCAD/ASPDAC/ISPD, the ISPD Routing Contest Award in 2007, the eASIC Placement Contest Grand Prize in 2009, the IBM Research Bravo Award in 2003, the Dimitris Chorafas Foundation Research Award in 2000, and the ACM Recognition of Service Award in 2007 and 2008. He was an IEEE CAS Society Distinguished Lecturer from 2008 to 2009.