

# Skew Management of NBTI Impacted Gated Clock Trees

Ashutosh Chakraborty  
ECE Department  
The University of Texas at Austin  
Austin, TX 78703, USA  
ashutosh@cerc.utexas.edu

David Z. Pan  
ECE Department  
The University of Texas at Austin  
Austin, TX 78703, USA  
dpan@ece.utexas.edu

## ABSTRACT

NBTI (Negative Bias Temperature Instability) has emerged as the dominant failure mechanism for PMOS in nanometer IC designs. 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 clock skew violation. Our work proposes a practical design-time technique of selecting NAND or NOR gate as output stage of integrated clock gating (ICG) cells with the objective of minimizing NBTI induced clock skew. This selection intelligently modulates the signal probability and delay equation of clock signal paths with no extra hardware penalty. We formulate the skew minimization problem as an integer linear program (ILP). Experimental results demonstrate the effectiveness of our method as the NBTI induced clock skew is reduced by more than 74% compared to traditional method.

## Categories and Subject Descriptors

B.7.1 [Integrated Circuits]: Design Styles—VLSI

## General Terms

Design, Performance

## Keywords

NBTI, Clock skew, Clock gating

## 1. INTRODUCTION

NBTI (Negative Bias Temperature Instability) has emerged as the primary PMOS failure mechanism [1] for advanced sub-65nm 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 etc. 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 50mV [2] causing timing violation and functional failures. As feature sizes shrink, NBTI effects will worsen exponentially due to higher operating temperatures.

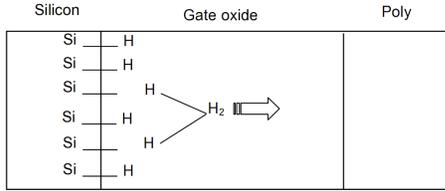
The shift in  $V_{TH}$  of the PMOS undergoing NBTI is due to the generation of interface traps under negative gate-to-source bias. These interface traps are the result of breaking of weak Si-H bonds which are formed due to crystal mismatch at the channel-gate interface [3]. Figure 1 from [4] illustrates the complete process of interface trap generation. The breaking of Si-H bonds at the Si-SiO<sub>2</sub> interface creates Si<sup>+</sup> (interface traps) and H atom. The presence of Si<sup>+</sup> at the surface requires larger gate voltage for channel inversion. This is the reason  $V_{TH}$  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. Existing literature [5] has 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 correct estimation of NBTI effect. The lifetime estimation of NBTI without considering the recovery phase can be an order of magnitude lower than the actual value [6]. To account for the relative time of stress and recovery phases, a common method is to deal in terms of signal probability (SP). Typically, SP is defined as the ratio of time a signal is at logic HIGH. However, since NBTI effect happens due to negative bias (i.e. input gate voltage at logic LOW), *we will denote the ratio of time a signal remains at logic LOW* to the total clock period as SP.

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

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ISPD'10, March 14–17, 2010, San Francisco, California, USA.

Copyright 2010 ACM 978-1-60558-920-6/10/03 ...\$10.00.



**Figure 1: Illustration of NBTI phenomenon. Breaking of SiH bonds creates  $\text{Si}^+$  interstitial causing  $V_{th}$  increase. Picture taken from [4].**

frequency of operation of a chip, therefore decreasing it is a major design concern. All modern clock trees employ 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 AND or 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 OR or NOR gate at 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 experiences alternate stress and recovery phases of equal duration. However, PMOS devices which are part of heavily gated clock buffer 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 non-uniformity in  $V_{TH}$  degradation can cause substantial increase in clock skew of a clock tree causing timing violations. In this paper we propose a novel scheme to tackle this problem.

In the next section, 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 3. In Section 4, we explain our experimental setup and the results achieved by our technique. We conclude our paper in Section 5.

## 2. PREVIOUS WORKS

NBTI and its effects have been considerably researched in recent years [3, 4, 7, 8]. These works include lifetime prediction of a chip considering AC stress, NBTI aware timing analysis, reliability of memories due to PMOS  $V_{th}$  degradation etc. The power-law dependence of NBTI on time [9] has been reported by most of the existing researches. A predictive NBTI model based on physical understanding and published experimental data was presented in [8]. [7] demonstrated analytical iterative equations for computation of NBTI impact for arbitrary waveforms. Numerical methods to solve the exact reaction/diffusion (R/D) physics based model of NBTI has been proposed in [10]. The work in [1], proposed a tight upper-bound of NBTI degradation for long term computation. In [5], the impact of NBTI on circuit timing has been tackled by making logic synthesis aware of NBTI. None of these works has targeted the interplay of NBTI and the skew of clock trees directly.

The patent [11] is the first work to combine NBTI effect and clock skew together. This work 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 work has two main deficiencies: a) their technique can over-constrain the clock tree synthesis tool, and b) this method cannot work for large skew degradation. The only direct technique known to us that tries to combat the impact of NBTI in clock tree is [12]. [12] proposed a scheme which relies on equalizing the signal probability of all clock tree trunks. Their technique chooses at run-time 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. The main shortcomings of this approach are: a) the number of transistors in the proposed gating element is more than double compared to a clock buffer, which it replaces; b) routing the second clock signal wastes precious routing resources; and c) most importantly, switching of secondary independent clock can lead to spurious clock pulses and thus to logic failures.

**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 dynamic approach of [12]), 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.

## 3. PROPOSED DESIGN TECHNIQUE

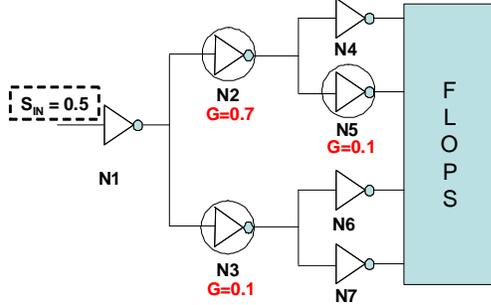
The aim of the proposed design technique is to reduce the skew of a clock tree arising due to the asymmetry in the  $V_{TH}$  degradation of the clock buffers. This asymmetry is due to difference in the signal probability (SP) at different parts of the clock tree due to clock gating. We propose to use *both* NAND and NOR gates<sup>1</sup> (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 signal probability for all clock buffers in the fanout cone. By intelligently choosing which gate to use for each clock gating element, the signal probability 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. Normally, these inverters would be replaced by ICGs (integrated clock gates) 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 which have NOR gate at their output stage. We next present an example to demonstrate the possibility of skew reduction by selecting ICGs with NAND or NOR gates.

**Motivating Example:** Consider the clock tree shown in Figure 2 that drives four latches. The clock tree nodes (represented as inverters indexed by 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

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

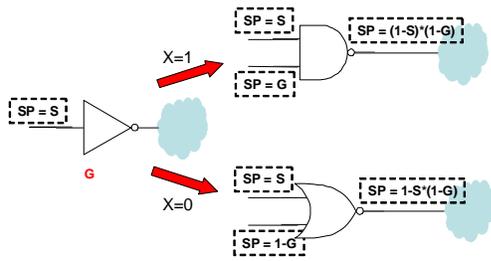
symmetry. For the gated nodes, the probability of clock gating ( $G$ ) of each node is also shown. A value of  $G=0$  implies that this particular node is never gated, whereas  $G=1$  means that this clock is always gated. Assuming 50% duty cycle of clock at input (i.e.  $S_{in}=0.5$ ), we computed the 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 picoseconds (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.



**Figure 2:** Example of a small clock tree. Nodes that allow clock gating are circled and their probability of clock gating indicated by symbol  $G$ . Input signal probability was set as 0.5 (i.e. nominal clock signal).

### 3.1 NAND/NOR Aware SP Propagation

In this section we set up the ground rules for propagation of signal probability (SP) and delay when implementing clock gating through NAND or NOR gates. We would like to reiterate that we are defining SP of a signal as the ratio of time it is logic LOW, different from the conventional opposite meaning. Consider a clock tree inverter shown on left side in Figure 3. The input SP of the inverter is  $S$  and the



**Figure 3:** A 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 signal probability (SP) at their inputs. The output SP is also shown for both the cases.

probability of clock gating is  $G$ . If this inverter is replaced by an 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 of  $G$  means  $(1 - G)$  period of non-gating 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  $D_{INV}(S,G)$ ,  $D_{NAND}(S,G)$ , and  $D_{NOR}(S,G)$  respectively, which are functions of the switching probability ( $S$ ) and Clock Gating probability ( $G$ ). If  $X = 1$ , the delay of the cell is  $D_{NAND}(S, G)$ , else it is  $D_{NOR}(S, G)$ . Delay of a clock inverter which does not clock gating capability is simply  $D_{INV}(S, G)$ . Table 1 summarizes these observations which are used for propagating the symbolic signal probability and delay values through the clock tree from the root to the clock leaves.

**Table 1: Formula for output SP and delay of different gates. Binary variable  $X = 1$  if NAND is chosen, 0 if NOR is chosen.**

Choice	Variable X	Output SP	Delay
NAND	1	$(1-G)*(1-S)$	$D_{NAND}(S, G)$
NOR	0	$1-S*(1-G)$	$D_{NOR}(S, G)$
INV	-	$1-S$	$D_{INV}(S, G)$

The information from Table 1 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  $X$ .

$$SP_{out} = 1 + S * G - S - X * G \quad (1)$$

$$D = X * D_{NAND}(S, G) + \bar{X} * D_{NOR}(S, G) \quad (2)$$

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

**LEMMA 1.** *Signal probability of any gate is at most a linear function of  $X$ .*

**PROOF.** Consider Eqn 1. If input signal probability,  $S$ , is linear or constant in  $X$  then output signal probability  $SP_{out}$  is also linear since  $G$  is a constant. As  $SP_{out}$  becomes the input signal probability for fanout gate, the linearity property remains recursively true. As the base case, signal probability of clock tree root is a constant number.  $\square$

**LEMMA 2.** *Delay expression of any gate is at most a quadratic function of  $X$ .*

**PROOF.** From above lemma, signal probability is linear function of  $X$ . From Equation 2, delay expression is linear combination of two expressions where  $X$  is multiplied by an expression at most a linear function of  $X$ . Hence, the delay expression is at most a quadratic function of  $X$ .  $\square$

Using the above expressions as well as that for inverter from Table 1, we can start at the root of the clock tree and recursively compute the symbolic signal probability 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 Figure 4 where each clock tree node that implements clock gating is circled. Let the binary variable  $X_2$  and  $X_4$  represent the choice of NAND/NOR gate at the nodes  $N_2$  and  $N_4$ . The probability of clock gating ( $G$ ) is written next to the corresponding nodes. The propagated signal probability of the path from clock root to the clock tree leaves are computed for each net based on  $G$  and the binary variables  $X_2$  and  $X_4$  and noted along the net. Ignoring dependence of NOR gate's delay on  $G^2$ , the delay at upper leaf can be written as  $[D_{INV}(0.5)] + [X_2 * D_{NAND}(0.5) + \bar{X}_2 * D_{NOR}(0.5)] + [D_{INV}(0.75 - X_2 * 0.5)]$ . Similarly, the delay at the lower leaf can be written as  $[D_{INV}(0.5)] + [X_2 * D_{NAND}(0.5) + \bar{X}_2 * D_{NOR}(0.5)] + [X_4 * D_{NAND}(0.75 - X_2 * 0.5) + \bar{X}_4 * D_{NOR}(0.75 - X_2 * 0.5)]$ .

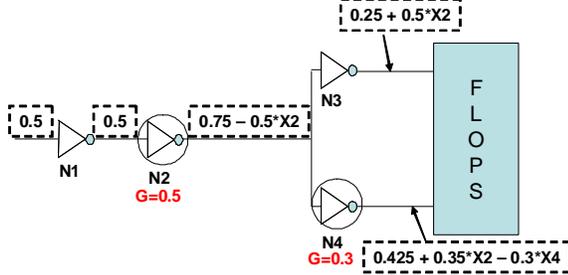


Figure 4: Example showing the propagated value of SP (in dashed boxes) as a function of  $X_2$  &  $X_4$  indicating the choice between NAND and NOR gate at nodes  $N_2$  and  $N_4$  respectively. Clock gating probability used for calculation is represented as symbol  $G$ .

From the above example, we note that the choice of using NAND and NOR gate (i.e. variable  $X_2$  and  $X_4$ ) not only modifies the delay function of the clock path but also modulates the signal probability (SP) at the fanout cone affecting the delay of output gates. Therefore, properly making this choice can help in reducing clock skew.

### 3.2 SP & GP Aware Delay Model

We sized the NAND and NOR gates to match their rise and fall delay to those of an INV (inverter). 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 gate 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. Next step is to characterize the delay of these cells as function of clock signal probability (SP) and gating probability (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 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 [7] extensively employed in other works such as [12] [5] etc. Using the obtained  $V_{TH}$  values, we performed spice simulation to obtain the rise and

fall delay of the NAND, NOR and inverter gate. 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 Figure 5 shows the rise delay of INV, NAND and NOR gate as a function of SP.

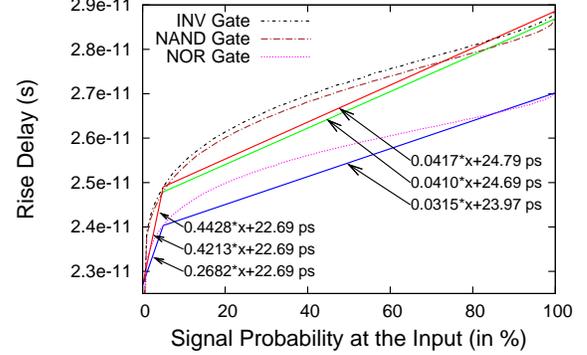


Figure 5: Rise delay of INV, NAND and NOR gate as a function of input signal probability (SP). Initial sharp increase is observed. Change in slope near 5% SP motivated the piecewise linear delay model.

There is a large increase in delay degradation near very low value of signal probability 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 [5]. To model this behavior, we performed piecewise linear fit for the case of  $SP \leq 5\%$  and for  $SP > 5\%$  obtained through Gnuplot tool. These linear fits are as follows:

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

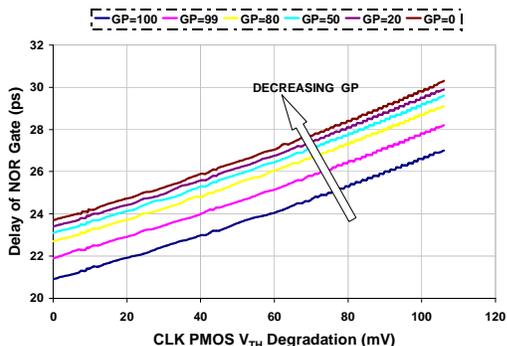
$$D_{NAND}(SP) = \begin{cases} (0.4213 * SP + 22.69) \text{ ps} & : x \leq 0.05 \\ (0.0410 * SP + 24.69) \text{ ps} & : x > 0.05 \end{cases}$$

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

From the previous discussion it is clear that clock signal probability has direct impact on the delay of the fanout gate. Next we consider the impact of gating probability (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 pin is driven by the clock signal from previous stage of clock tree with signal probability of SP and the other pin is driven by gating enable signal latched in the ICG (integrated clock gate) with signal probability 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 gating enable signal. This directly affects the pull-up capability of the NOR gate due to the inherent PMOS stack in it. In

<sup>2</sup>This dependence is characterized later in Section 3.2

short, for NOR gate we must consider the impact of degradation of both PMOS transistors. This was first pointed out by [5]. To capture this effect, we simulated the rise delay of NOR gate as a function of the  $V_{TH}$  degradation of PMOS driven by input clock for different gating enable probabilities driving the second PMOS input. The delay variation obtained for the gating probability of 0% (meaning never clock gated), 20%, 60%, 90%, 99%, and 100% (meaning always gated) are shown in Figure 6.



**Figure 6: Rise delay of NOR gate as a function of input signal probability (SP) for various clock gating probability (GP).**

Clearly, the higher is the signal probability of the gating enable signal (i.e. of clock being gated), the higher is 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 gating probability varies from 0% to 100%. Hence, we incorporated this gating probability dependence by linearly scaling the NOR delay as follows <sup>3</sup>:

$$D_{NOR}(SP, GP) = D_{NOR}(SP) * (1 - 0.08 * GP) \quad (3)$$

Using the expression for dependence on signal probability (SP) and gating probability (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.

### 3.3 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 presence of NBTI degradation after 10 years of aging. Let the set of sinks in the clock tree be given as  $S$ . For the  $i$ -th sink  $s_i$ , using the piecewise linear delay model developed in Section 3.2, we can obtain the formula for arrival time of the clock signal. Obviously, the arrival time is a function of the signal probability of the clock interconnects and gating probability 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.

<sup>3</sup>As explained earlier, NAND gate and INV gate do not have any dependence on GP.

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 (SAT) 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:

$$\begin{aligned} & \text{Minimize : } (MAX - MIN) \\ & \text{Subject To :} \\ & AT_i(\{X_i\}, \{SP_i\}) \leq MAX \quad \forall i \in S \\ & AT_i(\{X_i\}, \{SP_i\}) \geq MIN \quad \forall i \in S \\ & X_i \in \{0, 1\} \quad \forall i \in S \\ & MAX \geq 0 \\ & MIN \geq 0 \end{aligned} \quad (4)$$

In this, MAX and MIN are dummy variables which 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 are clearly  $O(n)$ . Assuming 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 3.1 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

$$\begin{aligned} X_A + X_B & \leq 1 + X_{AB} \\ (1 - X_A) + (1 - X_B) & \leq 2 - 2 * X_{AB} \end{aligned}$$

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

## 4. EXPERIMENTAL SETUP & RESULTS

For all our transistor level simulations, we used the post extraction spice models from the open-source 45nm Nangate library [13]. Simulations were run using Synopsys HSPICE Version A-2008.03-SP1. A C++ program was written to symbolically propagate the signal probability, to compute the symbolic delay equation of each clock sink and to write the optimization problem. We used tool CPLEX to solve our optimization 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 dual core 2.67 GHz workstation running Linux operating system.

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 input signal probability 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 inverter, the initial clock skew for all benchmarks is 0ps. Table 2 shows the characteristics of the benchmarks used in this work. In the table, the second column onwards contain 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 are associated with one binary decision variable for choosing NAND or NOR implementation.

**Table 2: Clock tree benchmarks used in this work. 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.**

Name	Depth	Fanout	# Buf <sup>+</sup>	# Sinks <sup>*</sup>	# Gated
A	7	4	21845	87380	331
B	8	3	9841	8748	144
C	9	3	29524	26244	426
D	8	4	87381	348520	1251
E	9	3	29524	26244	430
F	8	3	9841	8748	138
G	8	4	87381	348520	1267
H	7	4	21845	87380	326

<sup>+</sup>Number of buffers include all levels, not just leaves.

<sup>\*</sup>Number of sink assuming fanout of 4 at leaves.

Table 3 contains our results. Column 2 shows the CPU time to run 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 10 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 3, 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 are on an 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 3), 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 Figure 5 compared to the seemingly similar curves for INV and NAND gates. From the runtime perspective, we note that the maximum CPU runtime for

solving these testcases is less than 2 seconds. Benchmark G which has the largest number of gated clock buffers is solved in less than half a second.

**Validation:** In Section 3.2, 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 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 3, 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.

## 5. DISCUSSIONS & CONCLUSIONS

In this paper, for the first time we have proposed a static method for controlling NBTI degraded clock skew due to clock gating. Our method relies on 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 two degrees of freedom: firstly, the choice of the gate changes the delay function of that clock branch and secondly, it modulates the signal probability 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 signal probability at the input and the degradation of the clock gating element itself. The skew minimization problem was formulated as an integer linear program (ILP) and solved using commercial solvers. By exploiting our technique, we are able to reduce the NBTI induced clock skew by up to 74% compared to all-NAND implementation and 300% compared to all-NOR implementation. Our future work will be targeted towards 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.

## 6. REFERENCES

- [1] S. Bhardwaj *et al.*, "Predictive modeling of the nbtI effect for reliable design," *IEEE Custom Integrated Circuits*, pp. 189–192, Sept. 2006.
- [2] W. Wang *et al.*, "The impact of nbtI on the performance of combinational and sequential circuits," *DAC '07. 44th ACM/IEEE*, pp. 364–369, June 2007.
- [3] S. Kumar *et al.*, "Impact of nbtI on sram read stability and design for reliability," *ISQED '06.*, pp. 6 pp.–, March 2006.
- [4] M. A. Alam *et al.*, "A comprehensive model of pmos nbtI degradation," *Microelectronic Reliability*, pp. 71–81, Sept. 2005.
- [5] S. Kumar *et al.*, "NbtI-aware synthesis of digital circuits," in *DAC. 44th ACM/IEEE*, pp. 370–375,

**Table 3: Clock skew achieved by our solution is compared with all-NAND, all-NOR, and random best-among-10-trials strategies. 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.**

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%

\*Best result chosen among 10 random tries.

June 2007.

- [6] G. Chen *et al.*, “Dynamic nbti of pmos transistors and its impact on device lifetime,” in *Reliability Physics Symposium Proceedings, 2003. 41st Annual. 2003 IEEE International*, pp. 196–202, March-4 April 2003.
- [7] S. V. Kumar *et al.*, “An analytical model for negative bias temperature instability,” in *ICCAD '06*, (New York, NY, USA), pp. 493–496, ACM, 2006.
- [8] R. Vattikonda *et al.*, “Modeling and minimization of pmos nbti effect for robust nanometer design,” in *DAC*, (New York, NY, USA), pp. 1047–1052, ACM, 2006.
- [9] S. Mahapatra *et al.*, “On the generation and recovery

of interface traps in mosfets subjected to nbti, fn, and hci stress,” *Electron Devices, IEEE Transactions on*, vol. 53, pp. 1583–1592, July 2006.

- [10] R. Vattikonda *et al.*, “A new simulation method for nbti analysis in spice environment,” *ISQED*, pp. 41–46, March 2007.
- [11] J. M. Cohn, “Method for reducing design effect of wearout mechanisms on signal skew in integrated circuit design,” November 2003.
- [12] A. Chakraborty *et al.*, “Analysis and optimization of nbti induced clock skew in gated clock trees,” *DATE*, pp. 849–855, March 2009.
- [13] “Corporate website.” <http://www.nangate.com>.