

Analysis and Optimization of NBTI Induced Clock Skew in Gated Clock Trees

Ashutosh Chakraborty Gokul Ganesan Anand Rajaram David Z. Pan
ECE Department, University of Texas at Austin, TX 78712, USA
ashutosh@cerc.utexas.edu, goganesa@cisco.com, anandr@mail.utexas.edu, dpan@ece.utexas.edu

ABSTRACT

NBTI (Negative Bias Temperature Instability) has emerged as the dominant PMOS device failure mechanism for sub-100nm VLSI designs. There is little research to quantify its impact on skew of clock trees. This paper demonstrates a mathematical framework to compute the impact of NBTI on gating-enabled clock tree considering their workload dependent temperature variation. Circuit design techniques are proposed to deal with NBTI induced clock skew by achieving balance in NBTI degradation of clock devices. Our technique achieves up-to 70% reduction in clock skew degradation with miniscule (<0.1%) power and area penalty.

1. INTRODUCTION

As VLSI technology advances to sub 100nm regimes, NBTI has emerged as the primary device failure mechanism [1] which dictates the lifetime of the device compared to other reliability issues. NBTI causes slow threshold voltage (V_{th}) increase of PMOS device, thus 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. The shift in V_{th} of the PMOS is due to the generation of interface traps under negative gate-to-source bias referred to as *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 undergoes stress and recovery phases.

There has been no prior work to analyze the impact of NBTI on clock skew. As the clock signal switches every cycle, the PMOS devices in clock buffer experiences alternate/stress and recovery phase. All modern clock trees aggressively employ clock gating [3] technique which shuts down part of the clock tree. However, a PMOS device in a gated clocktree does not experience alternate stress and recovery phases and causing its threshold voltage degradation to go out of sync w.r.t. rest of the clock buffers. This can increase the skew of the clock tree leading to timing violations. Further, the workload dependent variation in the temperature of the clock buffers need to be concurrently analysed due to strong dependence of NBTI effect on temperature. This forms the motivation of our work.

2. PREVIOUS WORK

The power-law dependence of NBTI [4] has been observed by most of the existing works. A predictive NBTI model based on physical understanding and published experimental data was presented in [5]. [6] demonstrated analytical iterative equations for computation of NBTI impact for ar-

bitrary waveforms. Numerical methods to solve the exact reaction/diffusion (R/D) physics based model of NBTI has been proposed in [7]. Though the above models are accurate, they are inadequate for simulating a gigahertz signal for the duration of chip's lifetime due to the sheer computation complexity. One of the recent works, [1], has tackled this issue by proposing a tight upper-bound of NBTI degradation for long term computation. The key results proposed in that work computes long-term NBTI induced V_{th} degradation of a cell at time t as:

$$\Delta V_{th,t} \propto \left((V_{gs} - V_{th})^2 e^{-\frac{2E_a}{kT}} \right)^{\frac{1}{6}} \quad (1)$$

where V_{gs} , V_{th} are gate-to-source voltage and threshold voltage, E_a is the activation energy ($=0.49\text{eV}$ for H_2 diffusion), k is Boltzmann constant and T is the absolute temperature. The patent [8] is the closest attempt known to us that considers NBTI's effect on clock skew together. In this work the clock skew degradation due to NBTI is computed and used to guard-band the clock tree generation tool. As our results will show, the NBTI induced clock skew is much larger than expected skew therefore any guard-banding is difficult. Our main contributions in this work are as follows. We:

1. Introduce and derive the concept of effective temperature of a circuit element to ascertain the NBTI impact.
2. Propose a methodology for temperature dependent NBTI aware clock skew degradation computation.
3. Propose low overhead *design* techniques to overcome the NBTI induced skew degradation of clock tree.

3. COMPUTING EFFECTIVE TEMPERATURE

3.1 Capturing Temperature Distribution

We assigned random switching activities to primary inputs of our benchmark circuit which was then simulated to run for 10 million cycles. At the end of 10 million cycles, another set of random input switching activity was generated and the simulation was repeated. This process was performed 10000 times corresponding to a total of 100 billion cycles. The power consumption of each cell was found with *Design Compiler* at interval of 10 million cycles. These power numbers along with the placement information of the cells was fed to thermal simulator *HotSpot* [9] to obtain a sequence of temperature maps for the chip. On plotting the obtained temperature values for an arbitrary grid, we observed that the spread of temperature is near-gaussian¹. For example, for the benchmark *des*, the spread of temperature of an arbitrary block is shown in Figure 1. In rest of the text, we will assume that the temperature of a cell i is gaussian with mean and standard deviation

¹One may also fit the profile to a triangle. However, we observed that as the simulation points increase, the triangular profile developed into gaussian profile

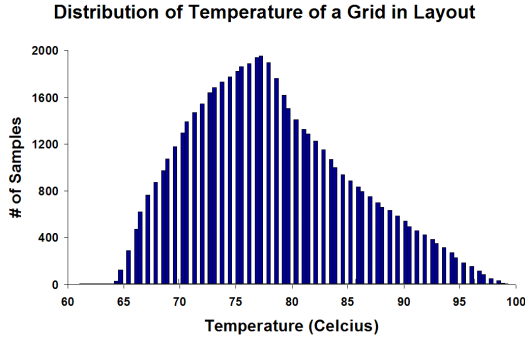


Figure 1: Frequency distribution of temperature of a grid. Sampling frequency = 1 million cycles.

of μ_i and σ_i respectively. Thus, the probability distribution of temperature of a cell i , $P(T_i)$ can be written as:

$$P(T_i) \approx \frac{1}{\sqrt{2\pi}\sigma_i} \times e^{-\frac{(x-\mu_i)^2}{2\sigma_i^2}} \quad (2)$$

3.2 Temperature Dependent NBTI Impact

The temperature dependence of NBTI can be found by substituting $V_{th} = -0.22$ and $V_{gs} = 1V$ (for our 45nm library [10]) in Eqn 1. Increase in temperature affects NBTI in two ways: a) following the Arrhenius dependency, the exponential term increase significantly, and b) V_{th} reduction with temperature causes further NBTI aggravation. As the cell library we used did not specify spice parameter TCV (temperature dependence of threshold voltage), we used its value as $-1mV/^{\circ}C$ based on PTM and HSpice manuals [11].

In the range of temperatures typically associated with VLSI chip operation, the dependence of NBTI on temperature using Eqn 1 can be very closely fitted by a quadratic curve. Figure 2 shows the original (using Eqn 1 and associated parameters), and the fitted quadratic curve of NBTI degradation of a cell in the range of 300K-380K normalized w.r.t. the degradation at 300K. At temperature T_i , the fit can be described as:

$$NBTI(T_i) \approx a \times T_i^2 + b \times T_i + c \quad (3)$$

where $a=1.98 \times 10^{-4}$, $b=-9.83 \times 10^{-2}$, and $c=12.67$. The R-square goodness of the above fit is 0.9999.

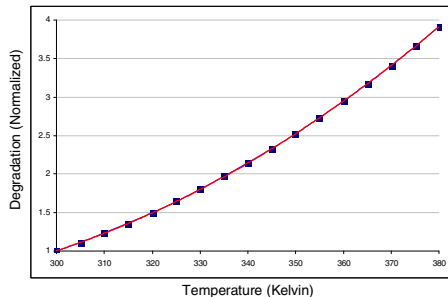


Figure 2: NBTI degradation normalized w.r.t. degradation at 300K (dot) and the quadratic fit (red line)

3.3 T^{eff} Computation

Using Eqn 2 and Eqn 3, the NBTI degradation of cell i is given by the expected value of product of temperature distribution curve and NBTI degradation at that temperature.

$$Degr(i) = E(NBTI(T_i) \times P(T_i)) \quad (4)$$

The product of $P(T_i)$ and $NBTI(T_i)$ vanishes at both very low and very high temperatures. Considering this, Eqn 4 can

be simplified by approximating the $erfc$ functions to obtain:

$$Degr(i) = (a\mu_i^2 + a\sigma_i^2 + b\mu_i + c) \quad (5)$$

The effective temperature of cell i , T_i^{eff} , can now be defined as the temperature at which if the cell was operating throughout, produces NBTI degradation equal to $Degr(i)$.

$$T_i^{eff} = \frac{-b + \sqrt{(b + 2a\mu_i)^2 + 4a^2\sigma_i^2}}{2a} \quad (6)$$

EXAMPLE 1. Consider a block with a mean temperature of 340K and std. deviation of 40K. The T^{eff} of this block by Eqn 6 as 348K. Simply considering the mean value (i.e. 340K) instead of T^{eff} causes 14% mismatch in computation.

To ascertain the accuracy of Eqn 6, we computed the NBTI degradation of a cell using Eqn 6 as well as through sequential calculation using first principles for gaussian random temperature values. A difference of only 0.2% was observed thus proving that for the purpose of NBTI calculation, a single value of T^{eff} can be used with insignificant loss of accuracy.

4. SKEW COMPUTATION METHODOLOGY

The skew computation methodology proposed by us is as follows. From RTL simulation of the design, the fraction of time each clock buffer remain gated is extracted. Next, the T^{eff} of each clock buffer is computed (see Section 3) based on placement of the cells in layout. The next step is to generate the spice netlist of the clock tree of the design. The T^{eff} and gating fraction (GF) of each clock buffer are now used to compute its ΔV_{th} due to NBTI. We used the cycle-by-cycle method for ΔV_{th} calculation detailed in [5]. Each clock buffer in the spice netlist of clock tree is then annotated with its T^{eff} and ΔV_{th} value and a spice simulation is run to compute the clock skew.

4.1 Clock Skew Results

We created the layout of several benchmark circuits from *opencores.org* [12] including clock tree generation with SOC Encounter using 45nm open-source *nangate* technology library [10] and computed its NBTI aware clock skew. The details of the benchmark circuits are in Table 1.

Table 1: Benchmarks (from [12]) used in this work.

Name	Num Cells	Num Nets	Num Flops	Num Clk Buf	Area μm^2
DES*	441K	453K	53K	5K	1168K
DES	103K	105K	8.8K	1.2K	198K
ethernet*	238K	242K	53K	4.3K	884K
ethernet	48K	49K	11K	998	180K
DLX	14K	15K	1.6K	174	38K
vga	3.2K	4K	702	65	14K

* means several (6) copies were simultaneously placed.

The above benchmarks vary in size from 3K to nearly half a million gates. Except for the couple of largest benchmark, rest are much smaller than typical industrial design, therefore the temperature increase we obtained using thermal simulation was very small. To capture the effect of our technique, we linearly scaled the temperature values obtained through T^{eff} computation to a spectrum of temperature between $50^{\circ}C$ to $110^{\circ}C$ to mimic a large industrial design [13] with significant thermal gradient.

Table 2 shows the temperature aware NBTI induced clock skew degradation for our benchmark circuits. For each benchmark, four skew values are reported. Colm A presents the

skew reported by the clock tree generator which is un-aware of the temperature gradient. Colm B shows the skew value after considering impact of temperature on delay of the buffers. Colm C and D show the clock skew after including NBTI as well as temperature considerations after 5 and 10 years of circuit aging respectively. Comparison of Colm A and Colm B shows that due to *just* temperature gradients, the skew of clock tree can be very different from what is reported by the clock tree synthesis tool. However, much more troublesome problem is evident by comparison of Colm A and Colm C or Colm D. The effect of NBTI drastically increases the clock skew due to clock gating and temperature dependent NBTI effect. In particular, the clock skew can become as much as 7X compared to the skew reported by clock tree synthesis tool (Colm A vs Colm D, *DES** benchmark after 10 year aging). This brings home the point that previously proposed technique in [8] of over-constraining the clock tree synthesis tool by artificially tightening the clock skew constraint cannot be relied on since there might not be enough margin available in original clock skew budget itself.

Table 2: Clock Skew (in ps) obtained with: A) Temperature unaware; B-Temp aware; C-Temp + NBTI (5 yrs); D-Temp + NBTI (10 yrs) delay computation.

Name	Skew A	Skew B	Skew C	Skew D
DES*	84.46	246.62	458.55	482.54
DES	117.59	279.23	374.59	392.95
ethernet*	64.86	160.42	231.57	257.48
ethernet	16.83	40.13	86.56	92.19
DLX	29.53	89.39	107.96	118.02
vga	8.05	17.42	58.54	59.21

With the above results in mind, in the next section we will propose our solution to efficiently deal with the NBTI menace to manage the skew of a clock tree over the lifetime of the design.

5. PROPOSED SCHEME: GBLV

Though NBTI degrades each clock buffer, the clock *skew* is primarily impacted due to *difference* in degradation of individual clock buffers. There are two sources for this difference: a) non-uniform temperatures, and b) non-uniform gating ratio (GR) among different clock buffers. In the post-placement stage, little can be done to eliminate thermal gradients, however equalizing GRs of different clock buffers is still possible. To exploit this optimization knob, we propose our scheme called **Gating with Both Logic Value (GBLV)** scheme.

GBLV Scheme: When gating a clock sub-tree, choose between logic 0 and logic 1 to apply to the frozen clock tree with the aim of equalizing the times these two choices are made.

Each clock buffer is internally made up of two inverters in series. Freezing the clock tree with logic 1 provides recovery (stress) time to the first (second) inverter in each of the gated clock buffer². On the other hand, freezing the clock tree with logic 0 has the opposite effect. We will refer to these two scenarios as 1-Gating and 0-Gating respectively. The ability to choose between 1-Gating and 0-Gating allows balancing of NBTI degradation thus reducing clock skew.

²The reasoning remains the same if the clock tree is implemented with inverters instead of buffers.

5.1 Hardware Implementation

GBLV technique requires that each clock gating cell should have the ability to freeze its output at logic value of 1 or 0 instead of traditional method of one of them. Figure 3 shows a simple optimized design of such a clock gating cell.

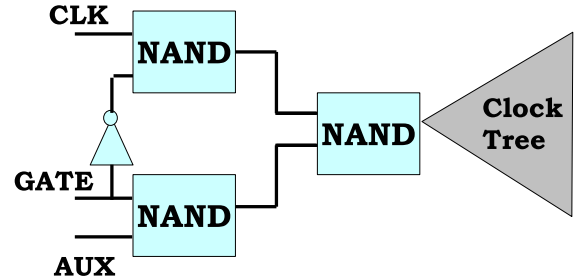


Figure 3: Gate-level implementation of optimized 0/1 choosing clock gating element.

When the GATE signal is low, the CLK signal passes through and the clock tree functions as usual. When GATE signal is high (which signals a clock gating opportunity), the clock tree is frozen with the logic value corresponding to signal AUX. Thus, signal AUX provides a handle to implement GBLV scheme.

Clock Gating Cell Overhead: We compared the overhead of our gating cell w.r.t. traditional AND gate based implementation [3] using our 45nm library [10]. For a clock frequency of 1GHz, 10% slew, and fanout load of 10 flops, the penalty in leakage power, dynamic power and area is 29.4% ($2.2\mu\text{W}$ vs $1.7\mu\text{W}$), 10.5% ($25.2\mu\text{W}$ vs $22.8\mu\text{W}$), and 33% ($2.12\mu\text{m}^2$ vs $1.60\mu\text{m}^2$) respectively. However, one must realize that even with fine-grain clock gating, only a fraction of clock buffers are converted as clock gating elements. A typical 1 million cell high performance design with logic depth of 6 may have 160K flops and 10K clock buffers. If 10% of these are clock gating elements, the overhead in

- Leakage Power : 0.03%
- Dynamic Power : 0.01%
- Area : 0.01%

5.2 Generating AUX Signal

Generating the *perfect* AUX signal that equalizes duration of 1-Gating and 0-Gating in hardware is impractical because it would require complete knowledge of scheduling and duration of all clock gating activities. Except for a few applications (such as GPU, DSP) this information is seldom possible to obtain. It is possible to easily implement the balancing scheme in software with some performance overhead, however, we will focus our attention to hardware schemes to minimize performance overhead. The AUX signal can be generated using the following two schemes:

1. **Alternate 0 and 1:** If we choose to use 1(0)-Gating to gate a module, the next time this module has to be gated, we use 0(1)-Gating. This scheme can be realized using a 1-bit memory near the gating element or using a Toggle flop. There is some area penalty in this method, however there is no routing overhead.
2. **AUX as slow clock:** The AUX signal can itself be a very slow clock signal (frequency = 0.01Hz) thus flipping periodically. The low frequency is sufficient to control long-term NBTI degradation and desirable for near-zero power penalty. There is routing overhead using this scheme, but with no power or area overhead.

The above two schemes have mutually exclusive strengths. In case the duration for which clock gating is enabled each time is similar but the schedule of clock gating is not, the first scheme works perfectly. This is because alternating stress recovery cycles of equal duration will be applied to each clock buffer. On the other hand, if clock gating is enabled regularly but for unequal duration each time, the second scheme works perfectly. This is because the regular clock gating will align with the very slow changing high/low value of AUX providing the necessary balance of stress and recovery phases.

EXAMPLE 2. A property like “Whenever clock gating is used, its for 1500 cycles” is perfect for the first scheme.

EXAMPLE 3. A property like “In every 10 minutes of operation, the clock is gated for a total of 90 seconds” is perfect for second scheme.

A designer can choose between the above two schemes depending upon the type of clock gating expected in the design.

5.3 Results

By completely eliminating the imbalance between 0-gating and 1-gating, reduced clock skew can be achieved. However, due to the statistical nature of our technique, there might be cases (such as when none of the statistical properties in above examples can be asserted) when the 0-gating and 1-gating do not properly balance. Let the imbalance between 0-gating and 1-gating be described by *mismatch factor* β , which is computed as:

$$\beta = \frac{|(T_0 - T_1)|}{(T_0 + T_1)} \quad (7)$$

where $T_0(T_1)$ is the duration of time when 0(1)-Gating is used. $\beta=0$ implies perfect balance of 0 and 1-Gating and higher magnitude of β implies more imbalance between these two quantities.

Table 3: Clock Skew (in ps) for 10 year aging. (orig) = temperature aware skew; (degr) = temperature+NBTI aware skew without GBLV; last three columns = temperature+NBTI aware skew with GBLV for increasing β .

Name	Skew (orig)	Skew (degr)	Skew ($\beta=0$)	Skew ($\beta=0.1$)	Skew ($\beta=0.2$)
DES*	246.62	482.54	385.21	392.54	387.75
DES	279.23	392.95	342.48	342.43	343.02
ethernet*	160.42	257.48	208.83	202.54	214.99
ethernet	40.13	92.19	65.86	70.33	69.42
DLX	89.39	118.02	98.23	99.64	101.43
vga	17.42	59.21	46.14	49.05	49.65

Table 3 shows the reduction in clock skew obtained by our method. Colm (orig) and (degr) contain the temperature and NBTI aware clock skew value at the beginning and after 10 year lifetime of the circuit respectively. The next three columns denote the clock skew values for three cases: $\beta=0$, $\beta=0.1$ and $\beta=0.2$ for the benchmarks. Because of space constraints, we are showing the numbers for 10 years aging but the trend for skew reduction is similar 5 year aging of devices. From Table 3, we note that our method of balancing 0 and 1-Gating achieves very good results. The improvements achieved in clock skew degradation (%) when using *GBLV* is graphically represented in Figure 4. For the case of perfect balancing, the skew degradation has been reduced by up-to 70% (47% average) after 10 years of aging. For the case of $\delta=0.1$ and $\delta=0.2$ which signify that due to statistical nature of our technique some (10% and 20%) amount of imbalance of 0-Gating and 1-Gating remains, the average reduction in

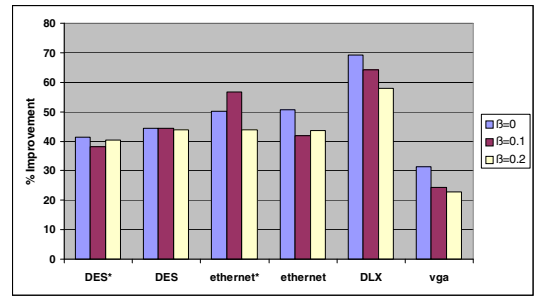


Figure 4: Reduction in skew degradation achieved by *GBLV* on 10 year aged benchmarks for different β

skew degradation is 45% and 42%. This significant reduction in the clock skew can be harnessed to design high performance systems (by shrinking the clock period) or to increase the device lifetime if it is constrained by NBTI affected clock skew violation. As shown earlier, the power and area overhead of using clock gating cell which can freeze clock trees to a choice of logic 0/1 levels is miniscule.

6. CONCLUSIONS

In this work, for the first time we introduced the concept of effective temperature which significantly simplifies NBTI degradation computation with minute error ($< 0.2\%$). We presented the first systematic methodology to compute coupled NBTI and temperature induced clock skew of an aging design. The results have highlighted the previously discounted menace of NBTI effects in gating enabled clock trees - the clock skew can increase by up-to 7X if not corrected explicitly. The technique of **G**ating with **B**oth **L**ogic **V**alue (*GBLV*) is proposed which can freeze a clock sub-tree at high or low logic value ensuring a balance between stress/recovery cycles of clock buffers. Use of *GBLV* reduces the NBTI induced clock skew by up-to 70% (47% average) with very insignificant ($< 0.03\%$) increase in design’s power and area.

Acknowledgment

This work is supported in part by NSF, Sun, and equipment donations from Intel.

7. REFERENCES

- [1] S. Bhardwaj *et al.*, “Predictive modeling of the nbtI effect for reliable design,” *CICC 2006*, pp. 189–192, Sept. 2006.
- [2] W. Wang *et al.*, “The impact of nbtI on the performance of combinational and sequential circuits,” *DAC 2007*, pp. 364–369, June 2007.
- [3] P. Babighian *et al.*, “A scalable algorithm for rtl insertion of gated clocks based on odcS computation,” *TCAD*, vol. 24, pp. 29–42, Jan. 2005.
- [4] 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.
- [5] R. Vattikonda *et al.*, “Modeling and minimization of pmos nbtI effect for robust nanometer design,” in *DAC 2006*, (New York, NY, USA), pp. 1047–1052, ACM, 2006.
- [6] S. V. Kumar *et al.*, “An analytical model for negative bias temperature instability,” in *ICCAD 2006*, (New York, NY, USA), pp. 493–496, ACM, 2006.
- [7] R. Vattikonda *et al.*, “A new simulation method for nbtI analysis in spice environment,” *ISQED 2007*, pp. 41–46, March 2007.
- [8] “Method for reducing design effect of wearout mechanisms on signal skew in integrated circuit design.”
- [9] W. Huang *et al.*, “Compact thermal modeling for temperature-aware design,” in *DAC 2004*, pp. 878–883, 2004.
- [10] “Corporate website.” <http://www.nangate.com>.
- [11] Synopsys, *HSpice User Guide*. 2007.
- [12] <http://opencores.org>.
- [13] E. Karl *et al.*, “Reliability modeling and management in dynamic microprocessor-based systems,” *DAC 002006*, pp. 1057–1060, July 2006.