Generation of Minimal Leakage Input Vectors with Constrained NBTI Degradation

Pramod Subramanyan*, Ram Rakesh Jangir‡, Jaynarayan Tudu*, Erik Larsson† and Virendra Singh*
*Computer Design and Test Lab, Indian Institute of Science, Bangalore.
†Linköping University, Linköping, Sweden.
‡Government Polytechnic, Hisar, India.

***Abstract*—Technology scaling has caused Negative Bias Temperature Instability (NBTI) to emerge as a major circuit reliability concern. Simultaneously leakage power is becoming a greater fraction of the total power dissipated by logic circuits. As both NBTI and leakage power are highly dependent on vectors applied at the circuit's inputs, they can be minimized by applying carefully chosen input vectors during periods when the circuit is in standby or idle mode. Unfortunately input vectors that minimize leakage power are not the ones that minimize NBTI degradation, so there is a need for a methodology to generate input vectors that minimize both of these variables.**

This paper proposes such a systematic methodology for the generation of input vectors which minimize leakage power under the constraint that NBTI degradation does not exceed a specified limit. These input vectors can be applied at the primary inputs of a circuit when it is in standby/idle mode and are such that the gates dissipate only a small amount of leakage power and also allow a large majority of the transistors on critical paths to be in the "recovery" phase of NBTI degradation. The advantage of this methodology is that allowing circuit designers to constrain NBTI degradation to below a specified limit enables tighter guardbanding, increasing performance. Our methodology guarantees that the generated input vector dissipates the least leakage power among all the input vectors that satisfy the degradation constraint.

We formulate the problem as a zero-one integer linear program and show that this formulation produces input vectors whose leakage power is within 1% of a minimum leakage vector selected by a search algorithm *and* simultaneously reduces NBTI by about 5.75% of maximum circuit delay as compared to the worst case NBTI degradation. Our paper also proposes two new algorithms for the identification of circuit paths that are affected the most by NBTI degradation. The number of such paths identified by our algorithms are an order of magnitude fewer than previously proposed heuristics.

I. Introduction

Relentless technology scaling has caused Negative Bias Temperature Instability (NBTI) to emerge as a major concern for circuit reliability. NTBI occurs when PMOS transistors are negatively biased, (i.e., $V_{gs} = -V_{DD}$) at elevated temperature causing a shift in the threshold voltage (V_{th}). Over a long period of time, such accumulated shifts can cause a significant increase the delay of these transistors, resulting in a degradation of the circuit's operating frequency. A number of techniques have been proposed to combat the effects of NBTI degradation, such as gate sizing [4, 11], adaptive body biasing [5], adjustment of supply voltage, signal probability etc. [8] and bit flipping [2].

The reduction of leakage power is also an important design goal in modern technologies. A popular method for leakage power reduction is Input Vector Control (IVC) [1, 13]. The basic idea behind this method is that when the circuit is idle or sleeping, input vectors which dissipate minimal leakage power are applied. The technique is effective because there is a significant difference in the leakage power dissipated by different input vectors.

The technique of input vector control (IVC) is attractive from the point of view combating NBTI degradation as well because IVC can mitigate the impact of NBTI over and above the effect of techniques like gate sizing. The basic idea behind IVC is to apply an input vector such that a large number of the PMOS transistors of the gates along the critical path of the circuit are not negatively biased (i.e. input value 1). Ideally we would like to have input value 1 at all points in the circuit, but this is not possible for real word circuits because the presence of logical negations implies that some values will necessarily be the complement of others. Thus the IVC problem for NBTI is to select an input vector that ensures that a large number of PMOS transistors are in recovery (i.e. receiving input 1) along the critical and near critical paths in the circuit.

As noted in previous work [10], input vectors that minimize leakage power may not be the ones that minimize degradation. As a result it is necessary to select vectors that *co-optimize* leakage and NBTI degradation. In this paper, our approach is to select input vectors that minimize leakage power while simultaneously constraining the NBTI degradation that occurs due to that vector to be less than a certain value called the *degradation limit*. Thus, circuit designers can use the well known technique of guardbanding to design their circuits with the required amount of delay slack to account for delay degradation due to NBTI while simultaneously minimizing leakage power. Being able to limit NBTI degradation (for known values of gate input probabilities and utilization) gives designers a easy way to trade-off leakage power for performance, by allowing circuit designers to use a smaller degradation limit, (i.e. smaller guardband) in exchange for higher leakage power.

This paper makes the following contributions. We present a 0/1 ILP formulation that can be used to obtain the input vector that dissipates the minimum leakage power under the constraint that NBTI degradation does not exceed a certain factor. Although previous work [9, 10] has attempted to obtain input vectors that minimize both leakage power and NBTI degradation, these methods are not guaranteed to provide optimal solutions. Furthermore, these methods also do not provide a rigorous way of trading-off leakage power to limit NBTI degradation or vice versa. On the other hand, our method provides fine grained control over the trade-off between performance, reliability and power.

Our results show that our formulation can produce input vectors that dissipate leakage power that is within 1% of that leakage power dissipated by an input vector generated by a search algorithm *and* simultaneously reduces NBTI degradation by 5.75% from the worst case. Our other main contribution is the introduction of two new algorithms for identifying critical paths (referred to as *potentially critical paths*, see section III-C) that will be affected most by NBTI degradation. We show that our algorithms select an order of magnitude fewer paths as compared to previous work [7, 12].

The rest of the paper is structured as follows. Section II presents the NBTI degradation model that we assume in the rest of the paper. Section III develops the ILP formulation of the problem. Section IV presents our algorithms for identifying critical paths and gates that are

most likely to be affected by NBTI degradation. Section V presents evaluates our technique and Section VI concludes.

II. NBTI Degradation Model

Vattikonda et al. [6] proposed a predictive NBTI degradation model for both static and dynamic NBTI The dynamic NBTI model takes into account the recovery processes that mitigate the effect of NBTI degradation that occur when the reverse bias on the PMOS transistors is removed. They provide equations that model the change in N_{it}, the number of positive interface traps. These equations can be used to compute the change in N_{it} over the circuit's lifetime, from which we can derive the change in the threshold voltage ΔV_{th} and the corresponding delay degradation. The disadvantage of this model is that it is computationally expensive as ΔV_{th} degradation has to be calculated by simulation over the entire circuit lifetime. To mitigate this problem, a number of models [3, 7] have been proposed based on a curve-fitting approach to obtain a closed form approximation for ΔV_{th}.

For the results presented in this paper, we use the model proposed by Luo et al. [3]. According to this model, the degradation in threshold voltage ΔV_{th} is given by the following equation.

$$\Delta V_{th} = \eta_0 \cdot p_s^{0.27p_s+0.28} \cdot t^{1/4} \tag{1}$$

Where the coefficient η_0 is given by:

$$\eta_0 = A \cdot T_{ox}\sqrt{C_{ox}(V_{gs} - V_{th})} \cdot exp\Big(\frac{E_{ox}}{E_0}\Big) exp\Big(\frac{-E_a}{k_b T}\Big) \tag{2}$$

We obtained the parameters in Equations (1), (2) for the 65-nm technology node from [6] and [14].

It is well known that the delay of a gate g is given by the following equation:

$$d_g = \frac{\alpha \cdot V_{dd}}{(V_{dd} - V_{th})^{\beta}} \tag{3}$$

We do not require the value of the constant α as we interested in ratio of the degraded delay to the original delay of the gate.

We note that our formulation is readily extensible to other NBTI degradation models, e.g. those that take into account the stacking effect.

III. Problem Formulation

We formulate this problem as an 0-1 Integer Linear Program (ILP). The variables of our ILP are the inputs and outputs of all the gates in the circuit and some additional variables that are added as described in the latter parts of this section. We have three types of constraints. One set of constraints express the input-output relation between gates. These are called the *I/O constraints*. A second set of constraints ensures that the maximum NBTI degraded delay of of the circuit does not increase beyond the degradation limit. We call these the *path delay constraints*. Finally, we have a third set of constraints called *linearization constraints* that are added for technical reasons. These convert a nonlinear integer program to an ILP.

In the following subsections, we first show how to encode various aspects of the problem as a *nonlinear* integer program. In section III-D we show how to linearize the formulation using a well known transformation.

A. Modeling Leakage Power

Consider the example of a two input gate, with inputs x_1 and x_2. Let $\hat{x_i}$ be the value of input x_i during sleep/standby mode. This value can be controlled by appropriately setting the primary inputs. We can model the leakage power dissipated by this gate using the following equation:

$$P(\hat{x_1}, \hat{x_2}) = c_0 + c_1\hat{x_1} + c_2\hat{x_2} + c_3\hat{x_1}\hat{x_2} \tag{4}$$

There are four different values that $(\hat{x_1}, \hat{x_2})$ can take and so we have four different values of P. We can use these four sets of values for $(\hat{x_1}, \hat{x_2})$ and P to generate and solve four linear equations in c_0, c_1, c_2 and c_3.

This approach can be generalized to gates with n inputs in the following manner. Let $\rho_n = \{S_0, S_1, \ldots, S_{k-1}\}$ be the set of all subsets of $\{\hat{x_1}, \hat{x_2}, \ldots, \hat{x_n}\}$. Clearly, ρ_n has $k = 2^n$ elements. Define $\pi(S_i)$ as the product of the elements of S_i. We set $\pi(\Phi) = 1$. For example, $\rho_3 = \{\Phi, \{\hat{x_1}\}, \{\hat{x_2}\}, \{\hat{x_3}\}, \{\hat{x_1}, \hat{x_2}\}, \{\hat{x_2}, \hat{x_3}\}, \{\hat{x_3}, \hat{x_1}\}, \{\hat{x_1}, \hat{x_2}, \hat{x_3}\}\}$, and $\pi(\{\hat{x_1}, \hat{x_2}\}) = \hat{x_1}\hat{x_2}$. Then we can write:

$$P(\hat{x_1}, \hat{x_2}, \ldots, \hat{x_n}) = \sum_{i=0}^{k-1} c_i \pi(S_i) \tag{5}$$

Since $(\hat{x_1}, \hat{x_2}, \ldots, \hat{x_n})$ and $P(\hat{x_1}, \hat{x_2}, \ldots, \hat{x_n})$ can take on 2^n different values, we can substitute these values to obtain 2^n equations in c_i which can then be solved in a straightforward manner to obtain a closed form expression for $P(\hat{x_1}, \hat{x_2}, \ldots, \hat{x_n})$.

For example consider Table I which shows leakage power values obtained from [10] for a 3-input NAND gate. Using the method detailed above we obtain the following equation[1].

$$\begin{aligned} P(\hat{x_1}, \hat{x_2}, \hat{x_3}) = \; & 30.1 + 24.8 \cdot \hat{x_1} + 24.6 \cdot \hat{x_2} + 25 \cdot \hat{x_3} \\ & + 169.6 \cdot \hat{x_1}\hat{x_2} + 179.3 \cdot \hat{x_1}\hat{x_3} + 230.1 \cdot \hat{x_2}\hat{x_3} \\ & + 19.8 \cdot \hat{x_1}\hat{x_2}\hat{x_3} \end{aligned} \tag{6}$$

Input	Leakage (pW)	Input	Leakage (pW)
000	30.1	100	55.1
001	54.9	101	259.2
010	54.7	110	309.8
011	249.1	111	703.3

TABLE I
Typical leakage power values for a 3-input NAND gate.

Since we want to minimize leakage power, we express the objective of the ILP as $minimize \;\; \sum_{j=1}^{N} P_j$, where P_j is the power dissipated by gate j of a circuit consisting of N gates.

B. I/O Constraints

To encode the relation between gate inputs and outputs, we introduce a set of constraints. Let the inputs of a n input gate be $\hat{x_1}, \hat{x_2}, \ldots, \hat{x_n}$, and its output be $\hat{y}$. For NOT, NAND and NOR gates respectively, we have the following relations:

$$\hat{y} = 1 - \hat{x_1} \tag{7}$$

$$\hat{y} = 1 - \prod_i \hat{x_i} \tag{8}$$

$$\hat{y} = 1 - \sum_i \hat{x_i} + \sum_{i,j} \hat{x_i} \cdot \hat{x_j} + \cdots + (-1)^n \prod_i \hat{x_i} \tag{9}$$

Similar constraints can be derived for other types of gates.

C. Path Delay Constraints

This set of constraints ensures that the the maximum NBTI degradation along each path is limited by L, the *degradation limit*.

Let us call any circuit path which could, depending on the input vector selected for use during sleep/standby mode, degrade so that its delay is greater than the degradation limit as a *potentially critical path (PCP)* [7].

In this section we do not address the problem of how to select PCPs, but assume that they are already known. In the next section we will present two novel algorithms for selecting these paths, and compare the performance of these with previous work.

The delay of a gate after NBTI degradation depends on two factors: (1) the probability that the inputs of the gate are stressed during normal circuit operation and (2) the values at the inputs of the gate when the circuit is in standby/sleep mode. If the gate's input is a primary input,

[1]Here $\hat{x_1}$ is the lsb, and $\hat{x_3}$ is the msb.

than we can directly control the value applied at the input of the gate. If it is an internal node, then its value depends on the value of the primary inputs in an indirect way.

Define the nominal (undegraded) delay of a gate g as d_g. Define the degraded gate delay of gate g on input values $(\hat{x_1}, \hat{x_2}, \ldots, \hat{x_n})$ as $\mathcal{D}_g(\hat{x_1}, \hat{x_2}, \ldots, \hat{x_n})$. In the rest of this subsection we show how to obtain closed form expression for $\mathcal{D}_g$.

Define $p_0(x_i)$ as the probability that input x_i is stressed during normal circuit operation, i.e. $p_0(x_i) = Pr\{v(x_i) = 0\}$, where $v(x_i)$ is the value of the input x_i during normal circuit operation. Note that $v(x_i)$ is a random variable whose distribution is application dependent. Although $v(x_i)$ is itself not known, we can measure the value of $p_0(x_i)$ during circuit operation by conduction appropriate simulations like in [11]. We also define $\mathcal{U}$ as the circuit utilization, i.e. the fraction of time the circuit is *not* in standby/sleep mode. Then we can write the probability that the input x_i will be stressed $p_s(x_i)$ as:

$$p_s(x_i) = p_0(x_i) \cdot \mathcal{U} + (1 - \mathcal{U})\hat{x_i} \tag{10}$$

We can evaluate the $p_s(x_i)$ for each input value $\hat{x_i}$ to obtain the degradation due to NBTI on each input of the gate. Selecting the maximum degradation among these gives us the degraded delay of the gate.

$$\mathcal{D}_g(\hat{x_1}, \hat{x_2}, \ldots, \hat{x_n}) = \max_i \Big\{ \Delta_d\big(p_s(x_i), t\big) \Big\} \cdot d_g \tag{11}$$

$\mathcal{D}_g$ gives the degraded delay of each gate as a function of its inputs. We can use the method outlined in Section III-A to obtain a closed form expression for $\mathcal{D}_g$ in terms of $(\hat{x_1}, \hat{x_2}, \ldots, \hat{x_n})$.

Using this closed form expression for $\mathcal{D}_g$ for each gate g we can enumerate all the potentially critical paths and bound the sum of delays of the gates along the path to be less than the *degradation limit*. To be more precise for each potentially critical path $P = (g_1, g_2, \ldots, g_k)$, we can write the constraint as $\sum_{i=1}^{k} \mathcal{D}_{g_i} \leq L$

D. Linearization Constraints

All the constraints we have derived so far use nonlinear functions, where some variables are the product of other variables. We can easily convert these to linear functions by introducing a new variable and three additional constraints for each product variable. For example, consider the following constraint:

$$z = 1 - xy \tag{12}$$

This can be linearized by introducing a new variable p representing the product of x and y and replacing the constaint (12) with the constaints in (13).

$$\begin{aligned} z &= 1 - p \\ p &\leq x \\ p &\leq y \\ x + y - p &\leq 1 \end{aligned} \tag{13}$$

It is easily verified that p is always the product xy when $x, y \in \{0, 1\}$. We use this method to convert all the nonlinear terms to linear terms in the constraints presented in the previous sections.

IV. Identifying Potentially Critical Paths

In this section, we address the question of selecting potentially critical paths (PCPs).

A simple approach proposed in [12] is to regard the top 10% of the longest paths in the circuit as PCPs.

A more rigorous selection procedure was introduced by Wang et al. in [7], where all paths P_i which have delay T_i such that $T_i \times (1+p) \geq T_{max}$ are selected as PCPs. Here T_{max} is the delay of the longest path in the circuit and p is the worst case degradation parameter. The authors assume that worst case NBTI degradation cannot exceeed 20%, and so select $p = 0.2$. In effect, this method selects all paths P_i such that $T_i \geq T_{max}/(1 + p)$.

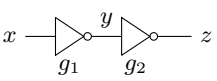

Fig. 1. Simple circuit demonstrating over-estimation of worst case degradation of delay calculated as part of Algorithm 1

Algorithm 1: Selecting PCPs Using the Worst Case Degradation
Consider a gate g with inputs $x_1, x_2, \ldots, x_n$. Note that the inputs x_i may be outputs of gates themselves. The maximum possible degradation of this gate will occur when all the input values, $\hat{x_i} = 0$. In this case the probability of the input x_i being stressed, $p_{s_{max}}(x_i)$ for circuit utilization $\mathcal{U}$ is given by:

$$p_{s_{max}}(x_i) = p_0(x_i) \cdot \mathcal{U} + 1 - \mathcal{U} \tag{14}$$

The *worst case degraded ready time* of the gate g, represented by $\mathcal{R}_{max}(g)$, is given by:

$$\mathcal{R}_{max}(g) = \max_i \Big\{ \Delta_d\big(p_{s_{max}}(x_i), t\big) \cdot d_g + \mathcal{R}_{max}(x_i) \Big\} \tag{15}$$

Recall that $\Delta_d(p_s, t)$ is the degradation factor for stress probability p_s over time t, and d_g is the undegraded delay of the gate g.

Using equation (15) we can calculate $\mathcal{R}_{max}(g)$ for each gate g in the circuit. Once $\mathcal{R}_{max}(g)$ is known, we can perform a depth first search from the primary outputs to select all paths from a primary input to a primary output which have worst case degradation greater than the degradation limit as potentially critical paths.

Algorithm 2: Improved Estimatation of the Worst Case Degradation
Consider the circuit show in Figure 1. In this circuit the worst case degradation of gate g_1 occurs when $\hat{x} = 0$, while the worst case degradation of gate g_2 occurs when $\hat{x} = 0$. The algorithm presented in the previous section will calculate the worst case degraded ready time $\mathcal{R}_{max}(g_2)$ as the sum of the worst case degraded delays of g_1 and g_2. Note that $\hat{x} = 0$ and $\hat{y} = 0$ cannot simultaneously occur. So $\mathcal{R}_{max}(g_2)$ is a *pessimistic* estimate of the degradation that can occur in the circuit.

We can improve the accuracy of the estimates by maintaining separate values of $\mathcal{R}_{max}(g)$ for when the output value of the gate is 1 and when the output value of the gate is 0. Define $\mathcal{R}^1_{max}(g)$ as the worst case degraded ready time of gate g among all the input vectors that lead to output 1, and $\mathcal{R}^0_{max}(g)$ as the worst case degraded ready time of gate g among all the input vectors that lead to output 0.

$$\begin{aligned} \mathcal{R}^0_{max}(g) &= \max_i^{\hat{g}=0} \Big\{ \Delta_d\big(p_s(x_i), t\big) \cdot d_g + \mathcal{R}^{\hat{x_i}}_{max}(x_i) \Big\} \\ \mathcal{R}^1_{max}(g) &= \max_i^{\hat{g}=1} \Big\{ \Delta_d\big(p_s(x_i), t\big) \cdot d_g + \mathcal{R}^{\hat{x_i}}_{max}(x_i) \Big\} \end{aligned} \tag{16}$$

The meaning of the equations (16) is as follows. We iterate over all the input vectors of a gate g[2]. For each input vector we calculate the output value. We calculate the degradation of the ith transistor using the ith component of the input vector by the $\Delta_d\big(p_s(x_i), t\big) \cdot d_g$ term. We calculate the worst case degraded ready time of that input gate using the $\mathcal{R}^{\hat{x_i}}_{max}(x_i)$ term. The sum of these gives the degraded delay along the path through that the input x_i. The maximum of these for output values 0 and 1 are chosen as $\mathcal{R}^0_{max}(g)$ and $\mathcal{R}^1_{max}(g)$ respectively.

Once $\mathcal{R}^1_{max}(g)$ and $\mathcal{R}^0_{max}(g)$ have been calculated for all the gates in the circuit g, we can calculate $\mathcal{R'}_{max}(g)$ as the maximum of these two.

$$\mathcal{R'}_{max}(g) = max\Big(\mathcal{R}^0_{max}(g), \mathcal{R}^1_{max}(g)\Big) \tag{17}$$

$\mathcal{R'}_{max}(g)$ is a tighter bound on the worst case degraded delay of the gate g than the bound suggested in the previous section. Once this bound is calculated for each gate, we can use it enumerate all paths from primary input to a primary output that have a worst case degraded delay greater than the degradation limit.

Table II shows the number of paths selected by our proposed algorithms compared against the method of Wang et al. [7]. For both of our

[2]Note that we only iterating over all the input vectors of a single gate.

Circuit	Algorithm 2	Algorithm 1	Wang et al. $p = 12\%$	Wang et al. $p = 15\%$
c1355	196608	786432	2523136	2779136
c1908	128	1077	58979	113563
c2670	168	1864	43832	69292
c3540	414	3468	368292	913898
c432	29430	74844	404946	432702
c499	196608	557056	2523136	2686976
c5315	164	1384	74616	131878
c7552	50	964	41214	63174
c880	12	24	196	329

TABLE II
NUMBER OF POTENTIALLY CRITICAL PATHS IDENTIFIED BY OUR ALGORITHM AS COMPARED TO THE METHOD OF WANG ET AL. [7]

.

Circuit	RND leakage max (μW)	RND leakage min (μW)	ILP leakage (μW)	ILP degradation	Worst case
c880	0.32	0.29	0.29	10%	16%
c1908	0.71	0.68	0.68	10%	16%
c3540	1.55	1.50	1.51	11%	16%
c432	0.14	0.12	0.12	11%	17%

TABLE III
RESULTS OF EXPERIMENTS ON ISCAS '85 BENCHMARKS

algorithms, we set the degradation limit $L = 1.12 \cdot T_{max}$. T_{max} is the maximum delay of the undegraded circuit. In other words, we are enumerating the number of paths that can potentially degrade to more than 12% of the "original" critical path. Recall that p is worst case degradation due to NBTI. Therefore the method of Wang et al. assumes that no path will degrade by more than $p\%$ of initial value and enumerates all paths which could become the critical path under this assumption. Selecting $p = 12\%$ is a quite optimistic assumption about the worst case degradation and we note that our algorithms do not make any such assumption.

It is clear from the table for many circuits, our algorithms produce reductions of more than an order of magnitude in the number of PCPs as compared to the heuristics proposed by Wang et al.

V. EVALUATION

A. Methodology

We developed a C++ simulator that generates the ILP model which takes as input a synthesized netlist, estimates the probability that a given input is stressed during normal circuit operation and generates the constraints and the objectives of the ILP.

Since NBTI degradation is circuit as well as input dependent, it is not possible to use a single degradation limit for all circuits. Therefore, we developed another C++ tool that generates a large number (e.g. $20,000$) random vectors and searches for the vector with minimum NBTI degradation among these. We then select the degradation limit as $L_{min}+\delta$ where L_{min} is the minimum of the degraded delays and δ is a small constant less than 0.1% of the maximum circuit delay that leaves some "slack" to find an input vector with lesser leakage power. Like in [9, 10] we assume that circuit utilization is $\mathcal{U} = 10\%$.

B. Results

Table III shows the results of our experiments on some of the ISCAS '85 benchmarks. To provide a reference for comparison, we show the minimum and maximum leakage power values obtained when searching over a large number (e.g. $20,000$) random input vectors. While this method can uncover some random vectors with low leakage power, it is difficult to find a vector that simultaneously minimizes leakage power and NBTI degradation. The results of the random search algorithm are shown in the first two columns of table III. The results of our ILP solution are shown in the third and fourth columns. We can see that the ILP solution is able to find a single input vector that minimizes both leakage power and NBTI degradation. The final column shows that the input vector derived using the ILP solution is significantly better than the worst case degradation.

VI. CONCLUSION

In this paper we introduced an ILP formulation for determining an input vector that minimizes leakage power while simultaneously constraining the maximum NBTI degradation due to that vector. Our results show that these leakage vectors are within 1% of the minimum leakage obtained by searching the input vector space and reduce the NTBI degradation by 5.75% of the maximum circuit delay as compared to the worst case. We also introduced two new algorithms for selecting potentially critical paths that produce order of magnitude reductions in the number of critical paths selected as compared to previous work.

REFERENCES

[1] Feng Gao and J. P. Hayes. Exact and heuristic approaches to input vector control for leakage power reduction. In *Proc. of the 2004 IEEE/ACM International Conference on Computer-aided Design*, 2004.

[2] Sanjay V. Kumar, Chris H. Kim, and Sachin S. Sapatnekar. Impact of NBTI on SRAM Read Stability and Design for Reliability. In *Proc. of the 7th International Symposium on Quality Electronic Design*, pages 210–218, 2006.

[3] Hong Luo, Yu Wang, Ku He, Rong Luo, Huazhong Yang, and Yuan Xie. A Novel Gate-Level NBTI Delay Degradation Model with Stacking Effect . *Integrated Circuits and System Design: Power and Timing Modeling, Optimization and Simulation*, pages 160–170, 2007.

[4] Paul, B.C. and Kunhyuk Kang and Kufluoglu, H. and Alam, M.A. and Roy, K. Negative Bias Temperature Instability: Estimation and Design for Improved Reliability of Nanoscale Circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pages 743–751, April 2007.

[5] Zhenyu Qi and Mircea R. Stan. NBTI resilient circuits using adaptive body biasing. In *Proc. of the 18th ACM Great Lakes Symposium on VLSI*, pages 285–290, 2008.

[6] Rakesh Vattikonda, Wenping Wang, and Yu Cao. Modeling and minimization of pmos nbti effect for robust nanometer design. In *DAC '06: Proceedings of the 43rd annual Design Automation Conference*, pages 1047–1052, 2006.

[7] Wenping Wang, Zile Wei, Shengqi Yang, and Yu Cao. An efficient method to identify critical gates under circuit aging. In *Proc. of the 2007 IEEE/ACM International Conference on Computer-aided Design*, pages 735–740, Piscataway, NJ, USA, 2007.

[8] Wenping Wang, Shengqi Yang, Sarvesh Bhardwaj, Rakesh Vattikonda, Sarma Vrudhula, Frank Liu, and Yu Cao. The impact of NBTI on the performance of combinational and sequential circuits. In *DAC '07: Proc. of the 44th annual Design Automation Conference*, pages 364–369, 2007.

[9] Yu Wang, Hong Luo, Ku He, Rong Luo, Huazhong Yang, and Yuan Xie. Temperature-aware NBTI modeling and the impact of input vector control on performance degradation. In *Proceedings of the Conference on Design, Automation and Test in Europe*, pages 546–551, 2007.

[10] Yu Wang, Xiaoming Chen, Wenping Wang, Varsha Balakrishnan, Yu Cao, Yuan Xie, and Huazhong Yang. On The Efficacy of Input Vector Control to Mitigate NBTI Effects and Leakage Power. In *Proc. of the 2009 10th International Symposium on Quality of Electronic Design*, pages 19–26, 2009.

[11] Xiangning Yang and Kewal Saluja. Combating NBTI Degradation via Gate Sizing. In *Proc. of the 8th International Symposium on Quality Electronic Design*, pages 47–52, 2007.

[12] Xiangning Yang, Eric Weglarz, and Kewal Saluja. On nbti degradation process in digital logic circuits. In *Proc. of the 20th International Conference on VLSI Design*, pages 723–730, 2007.

[13] Lin Yuan and Gang Qu. A combined gate replacement and input vector control approach for leakage current reduction. *IEEE Trans. Very Large Scale Integr. Syst.*, 14(2):173–182, 2006.

[14] Wei Zhao and Yu Cao. Predictive technology model for nano-cmos design exploration. *J. Emerg. Technol. Comput. Syst.*, 3(1):1, 2007. ISSN 1550-4832. doi: http://doi.acm.org/10.1145/1229175.1229176.