

An Efficient Method to Identify Critical Gates under Circuit Aging

Wenping Wang¹, Zile Wei², Shengqi Yang³, Yu Cao¹

¹Department of Electrical Engineering, Arizona State University, Tempe, AZ 85287, USA

²Department of EECS, University of California, Berkeley, CA 94720, USA

³Intel Corporation, Chandler, AZ 85226, USA

Abstract—Negative Bias Temperature Instability (NBTI) is the leading factor of circuit performance degradation. Due to its complex dependence on operating conditions, especially signal probability, it is a tremendous challenge to accurately predict the degradation rate in reality. On the other hand, we demonstrate in this work that it is feasible to reliably predict the relative importance of gates under NBTI. By identifying critical gates that are the most important ones for timing degradation, we will be able to effectively protect the circuit from aging, with the minimum design overhead. The proposed method is based on a new timing analysis framework that integrates a NBTI-aware library. For each potential critical path, we prove that there exists a particular signal probability, which leads to the worst case of timing degradation. The search of such worst case signal probability provides a safe guardband for the degradation, yet avoiding overly pessimistic analysis. By applying this method to ISCAS and ITC benchmark circuits at the 65nm node, we demonstrate that in average only 1% of total gates need to be protected in order to control the timing degradation within 10% in ten years. Since this method only requires one-time analysis of each critical path, it is very efficient in computation. With the information of critical gates available, it further enables other resilient design techniques to mitigate circuit aging under NBTI.

I. INTRODUCTION

The relentless scaling of CMOS technology inevitably leads to new reliability concerns, particularly NBTI [3], [4], [11], [14], [16], [17]. NBTI primarily increases the threshold voltage (V_{th}) of PMOS devices, which in turn causes the degradation of circuit speed [3]–[5], [12], [13], [17], [18]. To cope with this threat and guarantee circuit lifetime, it is critical to include NBTI into circuit analysis and adaptively develop effective techniques to mitigate its negative impact on performance. However, for a VLSI circuit, an accurate prediction of circuit performance degradation under NBTI remains as a tremendous challenge. As shown in [18], NBTI has a strong dependence on dynamic operation conditions, such as supply voltage (V_{dd}), temperature (T) and signal probability (α). Usually these parameters are not spatially or temporally uniform, but vary significantly from gate to gate and from time to time. Even if we can use a high temperature and voltage as guardbands [13], the uncertainty in signal probability may lead to more than 5X difference in the prediction of timing degradation [18]. While a worst case analysis is overly pessimistic, an exhaustive analysis with various signal probability will be extremely expensive in computation and thus, both approaches are not acceptable for a VLSI design [10]. To overcome this fundamental barrier, we propose to shift the focus toward the search of “critical gates”, which are the most

important gates under circuit aging, rather than a detailed analysis of timing degradation. By identifying the critical units, it will enable resilient design techniques, such as [2], to protect the circuit from degradation with the minimum design overhead. Although a general prediction of the degradation rate is impractical, we demonstrate in this work that it is still feasible to reliably predict *the relative importance of gates under NBTI* with sufficient computation efficiency. The key idea that supports this approach is to understand *the worst case signal probability* of each path that results in the maximum amount of degradation without over-margining. Built upon such worst case signal probability, we are thus able to conduct static path-based analysis, which is fast in computation and reliable in the protection from aging under all circumstances. The specific contributions of this work include:

- A NBTI-aware timing analysis framework: This framework embodies transistor-level NBTI models, gate-level timing models with the sensitivity to NBTI-induced V_{th} shift, and path-based timing analysis. This framework serves as the platform for this study to incorporate NBTI into conventional timing analysis and predict the degradation rate under various operating conditions.
- A fast method for path-based degradation analysis: Based on the path topology and static gate timing information. This method searches the worst case signal probability for path timing degradation. By considering the logic correlation among gates on the same path, it reduces the margin in the prediction and still ensures the safety of the guardband.
- An algorithm to identify critical gates under NBTI: By integrating the above methods, we develop a systematic approach that guarantees the selection of critical gates, given the margin of timing degradation. Comprehensive simulation results demonstrate that typically we only need to protect 1% of all gates for 10% degradation in ten years. Even with a stringent criterion of 5% degradation in ten years, the ratio of critical gates on an average only increases by 2%. Therefore, the approach to localizing critical gates is very promising and efficient for further resilient design exploration.

The outline of the rest of the paper is as follows: Section II describes the NBTI-aware timing analysis framework; Section III presents the fast method for path-based degradation analysis; Section IV explains the details of the integral algorithm for the identification of critical gates under NBTI; Section V evaluates the experiment results with ISCAS and ITC benchmark circuits at the 65nm node; Section VI further explores potential design benefit with the new approach.

This project is supported by the Gigascale Systems Research Focus Center, one of five research centers funded under the Focus Center Research Program, a Semiconductor Research Corporation program.

II. OVERVIEW OF THE FRAMEWORK

Figure 1 shows the structure of this framework. It includes a NBTI aware library based on our developed NBTI threshold voltage degradation and delay prediction model and a path-based fast critical gates identification methodology. For a given circuit, we begin with the standard Static Timing Analysis (STA) without considering NBTI effect. This process will generate the timing information for all the paths inside the circuit. Among all the circuit paths, we want to find the Potential Critical Paths (PCPs) and group them into a set. Moreover, timing degradation is analyzed for all the paths inside this PCP set by loading the NBTI aware library. Under certain timing constraint, for example, the maximal delay after degradation is less than 110% of the maximal delay before degradation, some paths in the PCP set may exceed the time margin requirement. And these paths are considered as the Protected Paths (PPs) which should be protected for reducing the NBTI effect on timing.

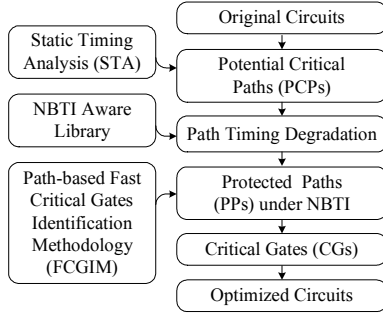


Fig. 1. NBTI-based static timing analysis framework.

For the gates in the PPs, a fast critical gates identification algorithm is carried out to obtain the Critical Gates (CGs). Optimization techniques are then applied to minimize the delay degradation of these critical gates until the timing for all the protected paths satisfies the requirement. Finally, the delays for all the paths in the given circuit meet the timing requirement even under worst case NBTI degradation. Each block in Figure 1 will be described in the following sections. A clear view of all the key terms, such as PCP, PP, and CG will be given once we proceed to the corresponding sections. As an extension, this framework can also be used to analyze other circuit aging effects, such as Hot Carrier Instability (HCI).

III. WORST-CASE PATH DEGRADATION ANALYSIS

A. Model Development

Several short-term analytical NBTI models for the degradation of threshold voltage, ΔV_{th} , have been proposed [3], [4], [12], [17]. However, these cycle-by-cycle models are not efficient enough for long stress time simulation. Faced with this situation, we have developed a long term NBTI prediction model which can accurately estimate ΔV_{th} at a given time t . Equation (1) shows the formula of ΔV_{th} caused by NBTI, where K_v is a function of electrical field, temperature and carriers concentration, n is the time exponential constant and equals to 0.16, α is the signal probability and it reflects the fraction of time spent in stress state for a period of time.

For a given time t , ΔV_{th} can be predicted by the long term prediction model, as shown in Figure 2.

$$\Delta V_{th} = \left(\sqrt{K_v^2 \cdot T_{clk} \cdot \alpha / (1 - \beta_t^{1/2n})} \right)^{2n}, \quad (1)$$

$$\text{where } \beta_t = 1 - \frac{2\xi_1 \cdot t_e + \sqrt{\xi_2 \cdot C \cdot (1 - \alpha) \cdot T_{clk}}}{2t_{ox} + \sqrt{C \cdot t}}. \quad (2)$$

By observing equation (1), we can further simplify this prediction model for fast circuit timing analysis. For a specific technology node and a given set of environmental conditions (such as V_{dd} and T), ΔV_{th} can be expressed as a function of the input signal probability (α), i.e., $\Delta V_{th} = b \cdot \alpha^n \cdot t^n$, where $b = 3.9 \times 10^{-3} V \cdot s^{-1/6}$. We verify this simplified model by comparing it with the long term prediction model as shown in equation (1). And the comparison result is shown in Figure 3. Obviously, this simplified model accurately predicts the V_{th} degradation for different long stress time periods.

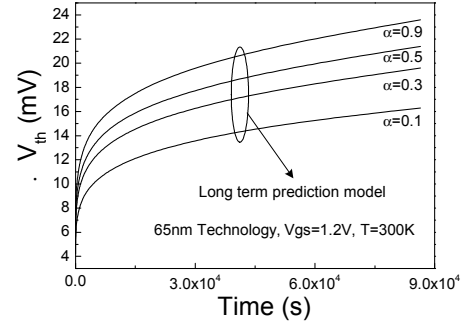


Fig. 2. The long term prediction model for different α .

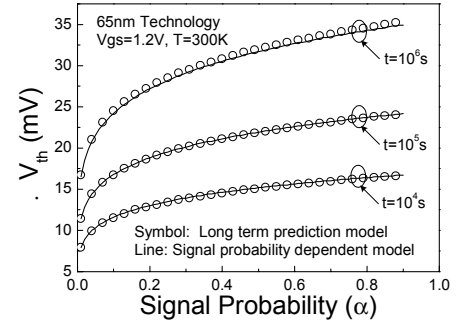


Fig. 3. The signal probability dependent model verification.

In addition to this simplified ΔV_{th} model which only depends on input signal probability and time, we can easily develop a very simple gate delay model with consideration of NBTI effect. Previous work [15] has demonstrated that the propagation delay (t_{pi}) of a gate, as a first order approximation, is linearly proportional to ΔV_{th} . Now, we can safely write the following equation:

$$t_{pi} = a_{0i} + a_{1i} \cdot \alpha^n \cdot t^n, \quad (3)$$

where a_{0i} is the intrinsic delay of the gate without NBTI degradation and a_{1i} is a constant parameter. For NBTI aware circuit timing analysis, library characterization is built for different logic units. In this work, we use a 65nm PTM [19] model in SPICE simulation to extract the delay information for an INVERTER, NAND and NOR gates at discrete time units. By fitting the simulation results with equation (3), we can get the coefficient values for a_{0i} and a_{1i} .

B. Worst Case Path Degradation

Traditional worst case timing analysis does not consider the correlation among gates on the same path. The worst case path delay is just the sum of the worst case gates delay on the path, which is overly pessimistic. In this subsection, we propose a fast method for path-based degradation analysis, which reduces the margin in the prediction and still ensures the safety of the guardband.

Figure 4 (a) shows an example to illustrate how to find the worst case path degradation. There are four gates in this path, t_{pi} is the gate delay under NBTI. For each gate, we only let one input signal switch and disable all other input signals. For instance, if the input signal probability of the first inverter is α , the signal probability of the second gate (NAND gate) is $(1-\alpha)$. Similarly, the signal probability for the second inverter is α and it is $(1-\alpha)$ for the NOR gate.

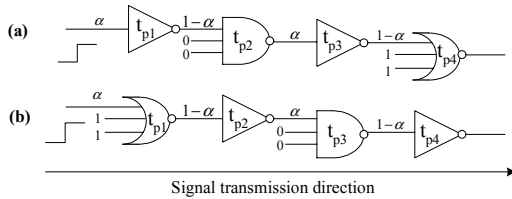


Fig. 4. Path-based degradation illustration examples: (a) Case 2; (b) Case 3.

For any gate under the setting mentioned above, only the signal probability of one input is of interest to us. As a result, we do not need consider the probabilities of all other input signals. Once we have the input signal probability of the first gate in the path, the total path delay under NBTI can be given by:

$$\begin{aligned} T &= \sum (a_{0i} + a_{1i} \cdot \alpha^n \cdot t^n + a_{0j} + a_{1j} \cdot (1-\alpha)^n \cdot t^n) \\ &= A_0 + (A_1 \cdot \alpha^n + A_2 \cdot (1-\alpha)^n) \cdot t^n, \end{aligned} \quad (4)$$

where A_0 is the sum of the intrinsic delay of all gates in the path, A_1 is the sum of a_{1i} for all gates with signal probability to be α and A_2 is the sum of a_{1j} for all gates with signal probability to be $(1-\alpha)$. The value range for α is $[0,1]$. When α is equal to a certain value, T reaches its maximum (T_{max}). This α is called as α_{worst} . Figure 5 illustrates the relation of the path delay versus α . We investigate three different paths: the first one is a four-stage inverter chain; the second one is the path shown in Figure 4 (a) and the third one is the path shown in Figure 4 (b).

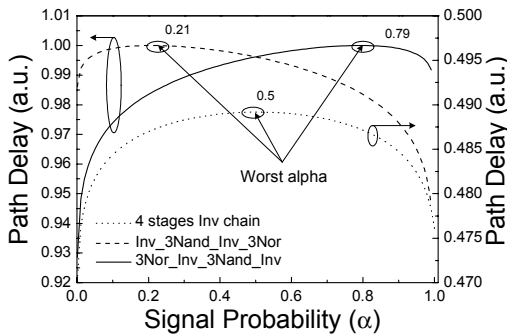


Fig. 5. Worst case path degradation for different investigated cases.

From Figure 5, we have two observations:

(1) The α_{worst} strongly depends on what kinds of gates exist in the path. For example, the four-stage inverter chain only has one kind of gate and the dependence of path delay on signal probability is quite symmetric. And the α_{worst} is around 0.5. However, for the path shown in Figure 4 (a), it has three different kinds of gates and the dependence is asymmetric and α_{worst} is approximately 0.21.

(2) Also, the α_{worst} also depends on how these gates are arranged in the path. Even if two paths have the same kind of gates (for example, case 2 and case 3), their α_{worst} can be quite different due to the different arrangement of these gates. In this example, α_{worst} is equal to 0.21 for case 2 and it is 0.79 for case 3.

Table I compares two path delay degradation estimation methods. They are the Pessimistic Worst Case Analysis (PWCA) and the Fast Worst Case Analysis (FWCA). The PWCA method estimates a path delay by setting all gates independently to their own worst cases. In another word, PWCA makes the input signal probability for all the four gates shown in Figure 4 (a) to be 1 instead of α and $(1-\alpha)$. The FWCA method is our proposed method described above. In Table I, T_o is the intrinsic path delay without NBTI degradation, T_{max} is the path delay estimated by FWCA, T_{worst} is the pessimistic worst case delay achieved by PWCA and $\Delta\%$ is the overestimation percentage by PWCA compared with FWCA. For the four-stage inverter chain, PWCA overestimates 2.12%, which will increase the number of required critical gates for protection. In summary, equation (4) can fully capture the path properties. It can efficiently predict the circuit delay degradation under worst case NBTI.

TABLE I
WORST CASE DELAY COMPARISON

	$T_o(ps)$	$T_{max}(ps)$	$T_{worst}(ps)$	$\Delta\%$
4-stage Inverter Chain	34.05	40.14	40.86	2.12
Inv-3Nand-Inv-3Nor	69.33	82.06	83.20	1.62
3Nor-Inv-3Nand-Inv	69.33	82.06	83.20	1.62

IV. CRITICAL GATES IDENTIFICATION ALGORITHM

The purpose of this section is proposing a fast searching algorithm which can efficiently find some circuit gates. Once these gates are protected, the timing of the circuit satisfies the requirement under NBTI impact.

A. Problem Definition

A given circuit may have thousands or millions of paths and nodes. The questions we want to answer are: how do we decide which circuit nodes should be protected to reduce the timing degradation of each path to a desired level? What is the fastest approach to find these nodes? Obviously, we cannot pick up all circuit nodes for protection because of the chip resource constraint. The first step to differentiate these nodes is to find which nodes are in some particular circuit paths that interests us the most. The following will define the potential critical paths.

B. Potential Critical Paths Definition

Without considering NBTI effect, we can very safely say that the critical path for a given circuit will not change with an

increase in time. By incorporating NBTI effect into timing analysis, the above argument is in risk and may not hold true anymore. For example, there are two paths P_i and P_j in a given circuit, and P_i is the critical path without considering NBTI. After some time stress, if the delay degradation caused by NBTI for the noncritical path P_j is much larger than that for the critical path P_i , originally critical path (P_i) can be non-critical one and vice-versa. It is quite obvious that the path delay degradation caused by NBTI effects play an important role in the static timing analysis. At this point, we can give a clear definition of Potential Critical Paths (PCPs). A potential critical path is a path that can potentially be the critical path after some period with delay degradation caused by NBTI effect.

Figure 6 demonstrates the criteria on how to select the potential critical paths. Suppose there are N paths in a circuit, the timing information of each path will be available once the traditional static timing analysis is finished without considering NBTI effect. Sorting procedure is carried out for the N paths according to their delay value. We name these paths as $P_1, P_2, P_3, \dots, P_{N-1}, P_N$, where P_1 means the first path with the longest delay and P_N means the N^{th} path with the smallest delay. The delay values for these paths are $T_1, T_2, T_3, \dots, T_{N-1}, T_N$, where T_1 is the delay for P_1 and T_N is the delay for P_N , with $T_1 \geq T_2 \geq T_3 \dots \geq T_N$.

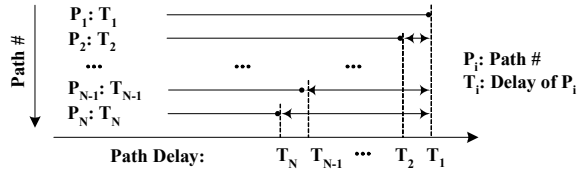


Fig. 6. Path delay information without consideration of NBTI.

Now we want to determine a cutting point and find the i^{th} path, whose delay meets $T_i \times (1 + p\%) < T_1$, where $p\%$ is the worst case degradation parameter. It means the highest degradation percentage a circuit may suffer for a given technology. This criterion means that for all the paths from $P_{(i+1)}$ to P_N , even if they are under the worst input vectors and their delays are degraded by $p\%$, they cannot become the longest critical path. However, for the paths from P_1 to P_{i-1} they have the potential to become the longest critical path under worst case NBTI stress or worst case signal probability. We don't need optimize or protect the gates which are in those paths. We define the potential critical path set and non-potential critical path set as:

$$PCP_{set} = \{P_1, P_2, \dots, P_{i-2}, P_{i-1}\}$$

$$nonPCP_{set} = \{P_i, P_{i+1}, \dots, P_{N-1}, P_N\}$$

Reference [18] shows that after ten-year stress stress, the maximum timing degradation for a path due to NBTI is below 20%. In this work, we select $p\% = 20\%$. As an example, if any path in the PCP_{set} as shown above is degraded by 20%, it can be the most critical path which is the longest path considering NBTI degradation. However, if any path in the $nonPCP_{set}$ is degraded even by the worst 20%, it still cannot become the critical path. In the following steps, we just need to analyze these PCPs.

C. Protected Paths Identification

For a given circuit, we can get the PCP_{set} according to the above analysis. However, we cannot protect all these paths since this set may still be large and includes hundreds or thousands of paths. This set need to be shrunk further. In this section, we will talk about how to identify the protected paths (PPs) according to theoretical study of the worst case path degradation described in section III.

When designing a circuit, designer normally leave some timing margin to ensure the circuit to work well after a long period of time. We define this margin as $q\%$ of the critical path delay (T_1) for a circuit without NBTI consideration. Under NBTI, the delay of these PCPs degrades with different rate. In order to satisfy the timing requirement for all the PCPs, we first predict the worst case delay of the PCPs. For a single PCP, we use the fast worst case path degradation methodology which was described in section III to get the worst propagation delay of each PCP. For paths $P_1, P_2, P_3, \dots, P_{i-2}, P_{i-1}$, under worst NBTI assumption, the delay values for these paths become $T_1', T_2', T_3', \dots, T_{i-2}', T_{i-1}'$, where T_1' is the delay for P_1 and T_{i-1}' is the delay for P_{i-1} with considering of worst NBTI delay degradation, as shown in Figure 7.

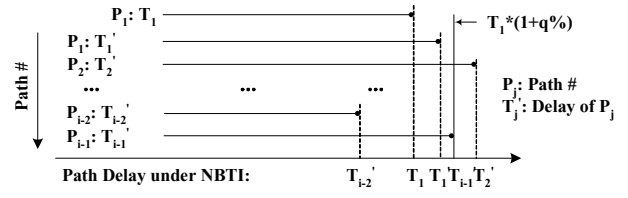


Fig. 7. Path delay information with consideration of NBTI.

These PCPs in Figure 7, can be divided into two types:

- (1) Paths whose worst delay are less than $(T_1 \times (1 + q\%))$, for example P_1 . This kind of paths do not need protection since even under worst case degradation, their delay will not exceed the timing margin and can not become the critical path.
- (2) Paths whose worst delay are equal to or larger than $T_1 \times (1 + q\%)$, for example P_2 and P_{i-1} . This kind of paths are the idea candidate for protection since their delays may exceed the margin which may result in a functional failure of the whole circuit. Thus, we obtain the protected paths set, which can be defined as:

$$PP_{set} = \{P_{p1}, P_{p2}, \dots, P_{p(k-1)}, P_{pk}\}$$

The delay values for these paths are $T_{p1}, T_{p2}, \dots, T_{p(k-1)}, T_{pk}$, where T_{p1} is the delay for P_{p1} and T_{pk} is the delay for P_{pk} , with $T_{p1} \geq T_{p2}, \dots, T_{p(k-1)} \geq T_{pk}$.

D. Critical Gates Identification

Once we get the PP_{set} , what we need to do next is to optimize some particular gates in the PPs for reducing the delay degradation of the PPs to the desired level, or within the time margin. These gates are called Critical Gates (CGs) which are a group of gates with largest delay degradations. The criterion to find the critical gates can be calculated as follows.

For a given path P_{pj} with M gates, we sort the gates according to their delay degradations, i.e., the delay increases caused by NBTI effect under the worst case assumption shown in

TABLE II
EXPERIMENTAL RESULTS FOR ISCAS85, 89 AND ITC99 CIRCUIT BENCHMARK (DYNAMIC CIRCUITS)

Circuit	PCP Gates		$q\% = 5\%$					$q\% = 10\%$					$q\% = 15\%$				
			PP	PR%	CG	G%	RT (s)	PP	PR%	CG	G%	RT (s)	PP	PR%	CG	G%	RT (s)
B14	315066	4485	14820	4.70	91	2.03	27.790	5271	1.67	46	1.03	29.038	368	0.12	16	0.36	33.87
B15	746232	8849	128704	17.25	172	1.94	214.550	46208	6.19	110	1.24	206.073	6656	0.89	90	1.02	202.48
B20	1109228	11729	3075	0.28	98	0.84	62.920	615	0.06	49	0.42	63.800	74	0.01	19	0.16	73.22
B21	1109641	11602	4256	0.38	116	1.00	334.240	1174	0.11	44	0.38	274.169	86	0.01	16	0.14	268.15
B22	2192997	16991	8504	0.39	151	0.89	361.750	1862	0.08	78	0.46	516.100	118	0.01	17	0.10	361.82
C5315	100746	1677	8797	8.73	32	1.91	1.900	1890	1.88	18	1.07	2.100	56	0.06	7	0.42	2.30
C7552	17127	1946	1101	6.43	44	2.26	0.250	146	0.85	18	0.92	0.284	10	0.06	5	0.26	0.33
DES	80800	4367	19936	24.67	623	14.27	0.690	4708	5.83	271	6.21	0.776	340	0.42	30	0.69	0.85
K2	2449	1541	272	11.11	41	2.66	0.024	53	2.16	15	0.97	0.028	20	0.82	3	0.19	0.03
S9234	2118	1770	251	11.85	51	2.88	0.032	88	4.15	26	1.47	0.004	20	0.94	5	0.28	0.04
S13207	64528	2691	6970	10.80	70	2.60	0.884	715	1.11	18	0.67	0.948	2	0.00	4	0.15	0.93
S15850	3972408	3530	292791	7.37	59	1.67	71.740	30562	0.77	24	0.68	79.520	328	0.01	8	0.23	84.97
S35932	5632	10920	224	3.98	176	1.61	0.044	128	2.27	96	0.88	0.036	32	0.57	32	0.29	0.05
S38417	192977	10556	40418	20.94	161	1.53	3.016	58	0.03	13	0.12	3.488	14	0.01	5	0.05	3.40
S38584	5475	12651	90	1.64	38	0.30	0.168	22	0.40	18	0.14	0.168	4	0.07	8	0.06	0.16

PCP: Number of potential critical path

Gates: Total number of the gates in the circuit

PP: Number of Protected path

PR%: The ratio of PP to PCP

CG: Number of critical gates

G%: The ratio of CG to Gates

RT: Running time

q%: Design timing margin

section III. For gates $G_1, G_2, G_3, \dots, G_{M-1}, G_M$ in path P_{pj} , the delay degradations of these gates can be expressed as: $\Delta t_{G_1}, \Delta t_{G_2}, \Delta t_{G_3}, \dots, \Delta t_{G_{(M-1)}}, \Delta t_{G_M}$ with $\Delta t_{G_1} \geq \Delta t_{G_2} \geq \Delta t_{G_3}, \dots, \Delta t_{G_{(M-1)}} \geq \Delta t_{G_M}$.

Since this path does not meet the timing requirement, we need reduce the delay of this path be to less than $T_1 \times (1 + q\%)$. There are several design techniques, which we will talk about in Section VI, to reduce the gate delay degradation. Here we use fresh identical gate to replace the degraded one, i.e., Δt_{G_i} is zero once it is replaced. Now we need find a minimum l to meet the following criterion: $T_{pj} - (\Delta t_{G_1} + \Delta t_{G_2} + \dots + \Delta t_{G_l}) < T_1 \times (1 + q\%)$. As a result, G_1, G_2, \dots, G_l are the critical gates which need to be optimized. We define the critical gates set for path P_{pj} as:

$$CG_{set} = \{G_1, G_2, \dots, G_l\}$$

So far, we already solved the problem proposed at the beginning of Section IV. By optimizing these critical gates, all the paths delay are with the margin even under worst case stress.

V. EXPERIMENTAL RESULTS AND DISCUSSION

The proposed fast critical gates identification methodology has been implemented in C. Experiments are run on 64-bit Linux servers with Xeon 2.33Ghz, 4MB level2 cache and 16G memory. This section describes the results and key insights obtained by performing timing analysis on ISCAS85, 89 and ITC99 [7]–[9] circuit benchmark. A PTM [19] 65nm technology is used throughout this section [19]. Since NBTI-induced degradation is relatively insensitive to switching frequency (f) [4], in the experiment we fix f at a value above 100Hz for all the analysis.

A. Experiment Setup

First, the selected fifteen circuit benchmarks are simplified by a fast logic synthesis tool: ABC [1], which was published by UC Berkeley. We process these circuits using the command *resync2rs* which includes a sequence of optimization operations, for example balancing, rewriting, resubstitution, etc. The final results were saved in BLIF format. We observe significant reduction in both the node number and the circuit depth, compared to the original circuits.

We derive a standard cell library from the NBTI delay model. This library consists of fifteen different cells: one inverter, 2 to 8-input NANDs and NORs. The library is stored in *genlib* format—a generic standard cell library format used in SIS [6]. Then we called SIS to map these fifteen benchmark circuits using the library. After technology mapping, we obtained a circuit implemented by standard cells, which are marked with time and required time using a load-independent delay model. In order to identify the potential critical paths, we traversed circuits from primary outputs to primary inputs by a depth first search algorithm. For the primary outputs, we firstly computed the maximal arrival time. Then, we subtracted the gate delay from the maximal arrival time and compared it to the arrival time of each fan-in. If a fan-in is in the preset margin, then we recursively processed it. If not, we stop looking for a path at this fan-in. When a fan-in is one of primary inputs, we recorded this path as one PCP and moved to check next fan-in. Note that, all PCPs are chosen from the static timing model. The actual delay must include the NBTI effect. Then the worst case α_{worst} is obtained analytically and finally the critical nodes are found according to the algorithm described above.

B. Results and Discussion

Reference [18] shows that for ten-year stress stress, the maximum timing degradation due to NBTI is below 20%. For this experiment, we select $p\% = 20\%$ and $q\% = 5\%, 10\%, 15\%$ separately. Table II shows the experimental results for different circuit benchmarks after ten-year degradation. For a given circuit, PCP represents the number of total potential critical paths; *Gates* is the number of total gates inside the circuits; *PP* means the total number of protected paths; *PR%* is the ratio of *PP* to *PCP*; *CG* is the number of total critical gates picked up for protection, *G%* is the ratio of *CG* to *Gates* and *RT* is the algorithm running time.

From this table, we can conclude that the proposed critical gates identification method is very promising. On an average, only 1% of total gates are identified and protected for all circuit benchmarks in order to meet timing requirement, i.e., 10% degradation. As an example, S38417 only needs 13 out of 10556 gates for protection in order to meet the 10% timing

margin. Further, this critical gate identification methodology is very fast in terms of algorithm execution time. It only takes tens seconds for larger benchmarks analysis.

VI. DESIGN OVERHEAD AND OPTIMIZATION TECHNIQUES

From the above experimental results and analysis, we know that the circuits performance degradation can be minimized if we do some prediction and optimization during the early design stage. The circuit lifetime can be extended by increasing a small amount of timing margin.

Figure 8 shows the dependence of the percentage of protected gates on the timing margin $q\%$. Six percent of the total gates in circuit DES need protection under the 10% degradation assumption. While it is 1.47% for S9234 circuit. With the technology scaling, NBTI degradation rate will increase, therefore, the number of protected gates also increases, as shown in Figure 9. 5.5% more gates need to be protected in circuit 9symml, when the degradation rate increases from 20% to 30%. However, if we want to offset the degradation by using transistor sizing technique, this ratio will be around 10% (dash line in Figure 9) [13]. The proposed critical gates identification methodology efficiently minimize the design overhead.

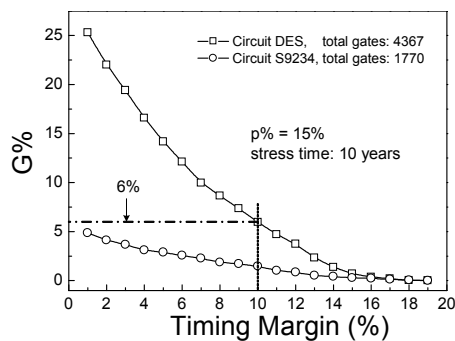


Fig. 8. Protected gates versus timing margin.

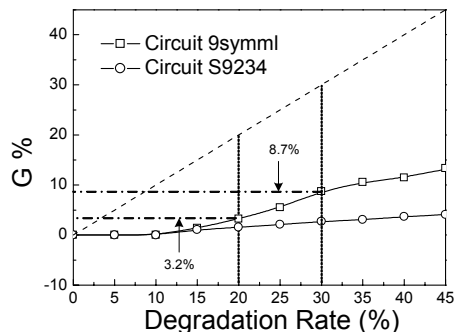


Fig. 9. Protected gates versus degradation rate.

Once the critical gates are identified, design techniques can be applied to reduce the degradation.

(1) *Fresh device replacing.* When we design a circuit, for these critical gates whose delay degradation caused by NBTI is significant, one solution is replacing the degraded gate with a fresh one. Reference [2] provides a sensor design technique which can be used to monitor NBTI degradation. Once the degradation reach some value, we can replace it.

(2) *Gate resizing.* Increasing the sizes of the critical gates is another way to reduce gate delay. Reference [13] has

demonstrates that delay degradation due to NBTI effect can be offset using gate sizing (8.7% average circuit size increase).

(3) *Reducing temperature.* Reference [18] shows that lowering the chip temperature can achieve up to 60% reduction on the delay degradation.

VII. CONCLUSIONS

A fast critical gates identification methodology is proposed in this work. By optimizing these critical gates, we will be able to efficiently protect the circuit from aging, with the minimum design overhead. Experimental results show that only 1% of total gates need to be protected in order to control the timing degradation within 10% in ten years. Since this method only requires one-time analysis of each critical path, it is very efficient in computation. For the large benchmark circuit analysis, its running time is just about ten seconds. With the information of critical gates available, it further enables other resilient design techniques to mitigate circuit aging under NBTI.

REFERENCES

- [1] Berkeley logic synthesis and verification group, abc: A system for sequential synthesis and verification, release 61225. available at: <http://www.eecs.berkeley.edu/~alanmi/abc/>, Dec. 2006.
- [2] M. Agarwal, B. C. Paul, M. Zhang, and S. Mitra. Circuit failure prediction and its application to transistor aging. *IEEE VLSI Test Symp.*, pages 277–286, 2007.
- [3] M. A. Alam and S. Mahapatra. A comprehensive model of PMOS NBTI degradation. *Microelectronics Reliability*, 45:71–81, Aug. 2005.
- [4] S. Bhardwaj, W. Wang, R. Vattikonda, Y. Cao, and S. Vrudhula. Predictive modeling of the nbtI effect for reliable design. *CICC*, pages 189–192, Sep. 2006.
- [5] S. Borkar. Electronics beyond nano-scale CMOS. *ACM/IEEE DAC*, pages 807–808, 2006.
- [6] E. M. Sentovich, K. J. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. R. Stephan, R. K. Brayton and A. Sangiovanni-Vincentelli. SIS: A system for sequential circuit synthesis. Technical report, 1992.
- [7] <http://courses.ece.uiuc.edu/ece543/iscas85.html/>.
- [8] <http://www.cbl.ncsu.edu/>.
- [9] <http://www.cerc.utexas.edu/itc99-benchmarks/bench.html>.
- [10] B. Krishnamurthy and I. G. Tollis. Improved techniques for estimating signal probabilities. *IEEE Tran. on Computers*, 38(7):1041–1045, Jul. 1989.
- [11] A. T. Krishnan, C. Chancellor, S. Chakravarthi, P. E. Nicollian, V. Reddy, and A. Varghese. Material dependence of hydrogen diffusion: Implication for nbtI degradation. *IEDM*, Dec. 2005.
- [12] S. V. Kumar, C. H. Kim, and S. S. Sapatnekar. An analytical model for negative bias temperature instability. *ICCAD*, pages 493–496, 2006.
- [13] B. C. Paul, K. Kang, H. Kuflluoglu, M. A. Alam, and K. Roy. Temporal performance degradation under NBTI: Estimation and design for improved reliability of nanoscale circuits. *DATE*, pages 780–785, 2006.
- [14] V. Reddy, A. T. Krishnan, A. Marshall, J. Rodriguez, S. Natarajan, T. Rost, and S. Krishnan. Impact of negative bias temperature instability on digital circuit reliability. *IRPS*, pages 248–254, Apr. 2002.
- [15] T. Sakurai and A. R. Newton. Alpha-power law mosfet model and its application to cmos inverter delay and other formulas. *JSSC*, 25(2):584–594, Apr. 1990.
- [16] D. K. Schroder and J. A. Babcock. Negative bias temperature instability: Road to cross in deep submicron silicon semiconductor manufacturing. *Journal of Applied Physics*, 94(1):1–18, Jul. 2003.
- [17] R. Vattikonda, W. Wang, and Y. Cao. Modeling and minimization of pmos nbtI effect for robust nanometer design. *DAC*, pages 1047–1052, Jul. 2006.
- [18] 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. *DAC*, pages 364–369, Jun. 2007.
- [19] W. Zhao and Y. Cao. New generation of predictive technology model for sub-45nm early design explorations. available at <http://www.eas.asu.edu/~ptm>. *TED*, 53(11):2816–2823, Nov. 2006.