Identifying NBTI-Critical Paths in Nanoscale Logic

Raimund Ubar[1], Fabian Vargas[2], Maksim Jenihhin[1], Jaan Raik[1], Sergei Kostin[1], Letícia Bolzani Poehls[2]
[1]Tallinn University of Technology, Tallinn, ESTONIA
[2]Catholic University – PUCRS, Porto Alegre, BRAZIL
{raiub | maksim | jaan | skostin}@ati.ttu.ee, {vargas | leticia.poehls}@pucrs.br

***Abstract*— One of the main reliability concerns in the nanoscale logic is the time-dependent variation caused by Negative Bias Temperature Instability (NBTI). It may increase the switching threshold voltage of pMOS transistors and as a result slow down signal propagation along the paths between flip-flops thus causing functional failures in the circuit. In this paper we propose an approach to identify NBTI-critical paths in nanoscale logic that is based on analyzing combination in different degrees of the three parameters: delay-critical paths, gate input signal probability and the gate fan-out degree along the paths. Further the identified NBTI-critical path can be used e.g. for introduction of aging sensors circuitry, rejuvenation stimuli generation, etc. The proposed approach is demonstrated on an industrial ALU circuit design.**

Keywords—NBTI-critical path, path identification, aging, logic circuit.

I. INTRODUCTION

In Very Deep Sub-Micron (VDSM) technology, lifetime reliability has become one of the key design factors to guarantee CMOS integrated circuit robustness. One of the most critical downsides of technology scaling beyond the 65nm node is related to the non-determinism of the devices electrical parameters due to time-dependent deviations in the operating characteristics of device. Essentially, two sources of time-dependent variations are identified: Negative Bias Temperature Instability (NBTI), and Hot Carrier Injection (HCI) [1]. Both are physical/chemical effects that cause a degradation of the oxide and result in a drift of the threshold voltage over time. However, NBTI has become the most prominent effect due to the fact that it creates interface states along with the whole silicon-oxide interface. NBTI has been shown to be the major reliability limiting factor when the gate oxide is thinner than 4 nm [2].

NBTI is defined as the effect that occurs when a pMOS transistor is negatively biased. The effect manifests itself as an increase of the threshold voltage over time. This results in drive current reduction and noise increase, which in turn causes a degradation of the device delay. The threshold variation is estimated to be 5-15% per year [3], [6] depending on the targeted technology and its environment, while with a smaller magnitude though, the path delay degradation follows the same trend. NBTI's effect on the long-term stability of functional logic expresses itself through the incapability of storing a correct value at memory elements such as flip-flops due to the de-synchronization between clock distribution and signal propagation through the logic paths of a circuit. The de-synchronization effect is observed when the summation of all gate delays along a given path is larger than the time slack allocated by the designer for that path. When this condition happens, the propagated signal reaches the memory elements at the end of the path in a time instant later than the clock front and as consequence, an incorrect value is stored at that point of the circuit.

The analysis of the NBTI effect is more complicated than other traditional reliability issues, such as hot-carrier injection [2], as it includes stress and recovery phases. The stress phase occurs when a pMOS transistor is under a negatively biased condition, i.e., $V_{GS} = -V_{DD}$. In this situation, the interaction between the inversion layer holes and the hydrogen-passivated *Si* atoms breaks the *Si–H* bond generated during the oxidation process. Then, the *H* atom converts into H_2 molecules. When H_2 molecules diffuse away, interface traps are left. When interface traps accumulate between silicon and the gate oxide interface, they cause a shift in the threshold voltage V_{th}. However, in the recovery phase, when the biased voltage is removed, the reverse reaction is performed. Some hydrogen diffuses back toward the interface and bonds with *Si*, which reduces the number of interface traps and the NBTI effect. Although the recovery phase can reverse the NBTI effect, it does not eliminate all the interface traps generated during the stress phase, and the pMOS threshold voltage will increase in the long term.

Previous works found in the literature have addressed the NBTI problem. Among them, [3] proposed an approach as a means of alleviating the NBTI-induced aging effects in static random access memories (SRAMs). Another approach [4] uses an experimentally verified NBTI model to study DC noise margins in conventional 6T SRAM cells as a function of NBTI degradation in the presence of process variations. Considering functional logic, authors proposed in [5] a transistor sizing technique that not only mitigates NBTI induced delay of the gate under consideration, but also minimizes its impact on the adjacent gates. This technique seems to be very effective, but it is mandatory to identify the critical gates and paths within the circuit in order to apply such technique. Otherwise, this technique should be applied to all the circuit gates, which would result in an unacceptable area overhead and eventually, excessive power consumption. In [7] authors present an interesting method to characterize the delay of every gate in a standard cell library as a function of the signal probability (SP) of each of its inputs. In the sequence, a technology-mapping technique that incorporates NBTI stress and recovery effects in order to ensure optimal performance of the circuit during its entire lifetime is applied. However, even though the use of this technique results in reasonable area and power savings with respect to the worst-case where the SP for

DOI 10.1109/DSD.2013.23

each gate is not taken into account (i.e., SP is fixed to "1"), it must be applied to all nodes of the circuit in order to find the best-delay match for each gate as a function of the SP. This is the main difference between this technique and the one we propose herein: instead of indiscriminately replacing all the gates by their minimum-NBTI counterpart along with all the logic cones of the circuit, we propose first to identify the delay-critical paths along the circuit. Then, based on SP and fan-out information for the nodes settled along these critical paths, we select the NBTI-critical paths and only to this latter reduced set of gates and paths, apply NBTI-recovery strategies. Such strategies can be, for instance, the replacement of the gates by their minimum-NBTI counterpart as proposed by [7].

As mentioned above several works found in the literature have addressed the NBTI problem. However, to the best of the authors' knowledge, there is no previous publication devoted to identify the critical paths from the NBTI point of view. These NBTI-critical paths are those resulting from the combination in different degrees of three parameters: (a) delay-critical paths, (b) gate input signal probability SP and (c) the gate fan-out degree along with the delay-critical paths. Therefore, the purpose of this paper is to identify these NBTI-induced critical paths in a given functional logic. To do so, first we propose to identify the delay-critical paths of a functional logic. Then, according to SP and fan-out information for all nodes along with these delay-critical paths, the NBTI-critical paths are selected.

The remaining of the paper is organized as follows: Section 2 introduces the basic principles behind the proposed approach, Section 3 describes the strategy carried out to compute the gate-level delay, Section 4 details the proposed approach for NBTI-critical paths identification, Section 5 demonstrates experimental results on a case study design, while Section 6 concludes the paper.

II. PRELIMINARIES

Assuming that NBTI degrades long-term signal propagation due to aging of pMOS devices, one could expect an increase in the occurrence of timing (i.e., delay) faults during system lifetime. Therefore, *delay-critical paths* along with the circuit are the primary candidates to fail.

Two major parameters should be taken into account in order to properly predict NBTI effect at the circuit level. These parameters are applied to the delay-critical paths of the circuit and are, respectively, the *input signal probability* (SP) and the *gate fan-out degree.* The reasons are described below.

When a pMOS transistor is under constant stress, this situation is called *static NBTI*. On the contrary, if both stress and recovery phases exist, this condition is called *dynamic NBTI*. The threshold voltage increases when the circuit is in an active mode and can be estimated using dynamic NBTI models. However, when the circuit enters a standby mode, it is affected by static NBTI. The *input signal probability* is defined as the *probability that input signal is at logic 0*. When the input signal probability is '1', it means the pMOS transistor's input signal is constant '0', and the pMOS transistor is under maximum constant stress. In order to correctly predict the NBTI effect on circuit degradation, it is important to estimate both dynamic and static NBTI effects.

Consider that for two similar gates, the one that drives the highest output capacitance is probably the one that experiences the highest probability to suffer from delay faults when the pMOS current driving capability is degraded by NBTI. Noting also that as high is the output capacitance of a gate, higher is the probability of this gate to present a large fan-out degree, then gates presenting high fan-out degrees along with the critical paths of a circuit are the primary candidates to fail.

In this paper we propose to identify *NBTI-critical paths* in nanoscale logic by analyzing combination of all the three parameters: delay-critical paths, gate input signal probability and the gate fan-out degree along the paths.

III. GATE-LEVEL PATH DELAY CALCULATION

The basis for the path delay calculation is the information about individual delays in the logic gates along the path. These delays can be pre-calculated using the following relative approach [8].

Let $d(g)$ be a *delay of a logic gate* $g \in G$ where G is a set of gates. I.e. the delay between each input s_i of and output s_j of g is $D(s_i, s_j)=d(g)$. Then

$d(g)= f(g) + p_g$

where $f(g)$ is the *effort delay* of g and p_g is the *parasitic delay* of g.

$f(g)= e_g h(g)$

where e_g is the *logical effort* which is the relative ability of the gate g topology to deliver current and defined to be 1 for an inverter gate. $h(g)$ is the *electrical effort* influenced by fan-out and equal to the ratio of the output to input capacitance, i.e., $h(g)=C_{out}/C_{in}$. Therefore:

$$d(g) = e_g \left(\frac{C_{out}}{C_{in}}\right) + p_g \qquad (1)$$

Here the logical effort e_g and the parasitic delay p_g can be estimated for some CMOS gates as follows:

$e_{\text{inverter}} = 1$
$e_{n\text{-input NAND}} = (n+2)/3$
$e_{n\text{-input NOR}} = (2n+1)/3$
$e_{n\text{-way multiplexer}} = 2$
$e_{2\text{-input XOR/XNOR}} = 4$

$p_{\text{inverter}} \approx 1$
$p_{n\text{-input NAND}} = np_{\text{inverter}}$
$p_{n\text{-input NOR}} = np_{\text{inverter}}$
$p_{n\text{-way multiplexer}} = 2np_{\text{inverter}}$
$p_{2\text{-input XOR/XNOR}} = 4np_{\text{inverter}}$

While each particular technology defines specific values for the logic gate cells, the described above approach is relative and can be considered as technology independent. At the same time it is sufficient to select the delay-critical paths in the given CMOS circuit.

As an example the delay in the XOR gate closest to output $\bar{F}_0$ (bottom-right corner) of the *ALU_4bit* circuit in Fig. 2 can be calculated as follows:

$$d(g_{EO1}) = 4\frac{2}{1} + 4 = 12$$

The total delay of a path in a combinational circuit is estimated as sum of delays in individual gates along the path. The paths with the longest delay, e.g. close to the time slack, are delay-critical paths.

IV. Algorithm for Identifying NBTI-Critical Paths

In this Section, a method for identifying paths which are most critical to NBTI effect is proposed. The method selects a restricted number of delay-critical paths and evaluates these with respect to NBTI caused aging.

Consider a combinational gate-level logic circuit represented as a network $N = (G, S)$ where G is a set of gates and S is a set of signal lines. The set S is partitioned into three subsets: a set of lines IN to represent the primary inputs of the network, a set of internal nodes FN to represent the signal lines connecting the gates, and a set of lines OUT to represent the primary outputs of the network.

In order to evaluate aging effects at different primary outputs OUT of the network N we have to perform an analysis on a sequence of input patterns T with a length n. To characterize the aging effects we introduce the following parameters:

$D(s_i, s_j)$ – delay between the lines s_i and s_j,

$P_z(s_i)$ – gate input signal probability, i.e. probability of '0' at line s_i, and

$B(s_i)$ – gate fan-out degree, i.e. number of fanout branches from the gate with output line s_i.

The delay parameters $D(s_i, s_j)$ for each gate $g \in G$ from its input s_i to output s_j are calculated by Formula 1 in Section III. The values $P_z(s_j)$ for all $s_j \in S$ will be calculated by simulation of T.

Our task is to characterize each output of the circuit by a restricted number K of paths represented by 4-tuples $M_k(s_i)=(D_k(s_i), P_z{}^*_k(s_i), B^*_k(s_i), H_k(s_i) = \{s_0, s_1, ..., s_i\})$, where $k = 1, ..., K$ and

$D_k(s_i)$ – is the delay of the k-th selected path from a primary input to the line s_i;

$P_z{}^*_k(s_i)$ – is the average signal probability along the path from the related primary input to the line s_i;

$B^*(s_i)$ – is the accumulated number of fanout branches from the gate with output s_i along the path;

$H_k(s_i)$ – is the set of signals on the path from a primary input s_0 to the signal s_i.

The calculation of the 4-tuples starts from primary inputs of the network N. In order to avoid an explosion of the number of paths to be analyzed, we introduce a threshold K for the number of longest paths traced up to the current signal line under analysis for continuing the calculations of 4-tuples.

The Algorithm for calculating the 4-tuples $M_k(s_i)$ is as follows:

1. Start with primary inputs, and assign $D_k(s_i) = 0$, $P_z{}^*_k(s_i) = P_z(s_i)$ and $H_k(s_i) = \{s_i\}$ to all $s_i \in IN$. If the input is a fanout, assign $B^*(s_i) = B(s_i)$, otherwise $B^*(s_i) = 0$.
2. For all the internal and primary output lines $s_j \in FN \cup OUT$ of the circuit, proceed along the numeration of the lines, taking into account the order of gates within the circuit. For current line s_j, which is at the output of a gate $g \in G$ calculate the following:
   - For all the 4-tuples $M_k(s_i)$ of gate inputs s_i, $i = 1, 2, ... m$, where m is the number of inputs to the gate g: $D_{i,k}(s_j) = D_k(s_i) + D(s_i, s_j)$;
   - Fix the current ordered set of 4-tuples $\{M_k(s_j)\}$ where k = 1,2, ..., K; the ordering is carried out in decreasing order of the values of $D_k(s_j)$;
   - $H_k(s_j) = H_k(s_i) \cup \{s_j\} = \{s_0, s_1, ..., s_i, s_j\}$
   - If $s_j \notin OUT$ then $P_z{}^*_k(s_j) = (P_z{}^*_k(s_0) + P_z{}^*_k(s_1) + ... +$

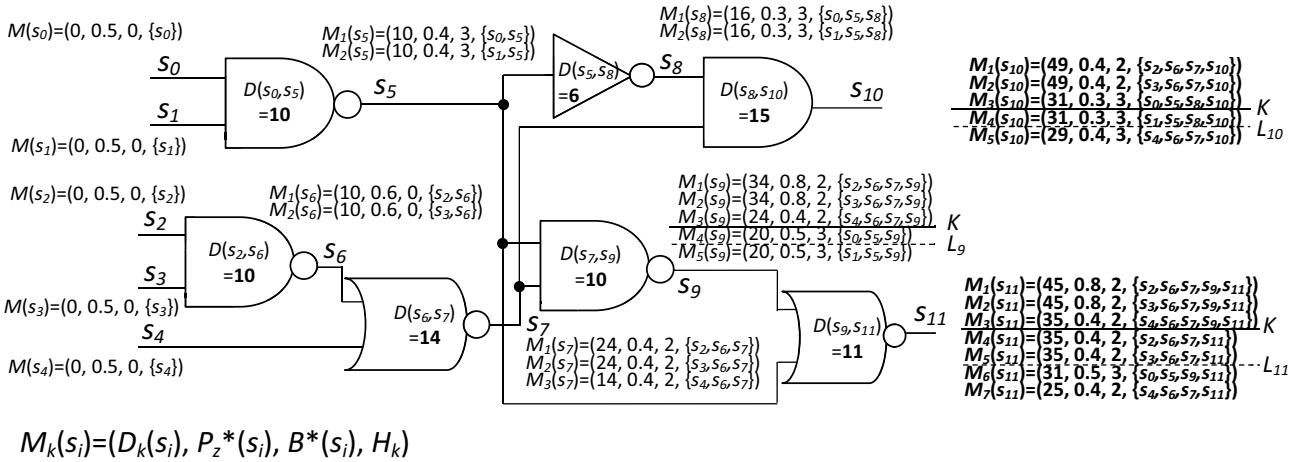

Fig. 1. Calculation flow of the proposed approach on an example

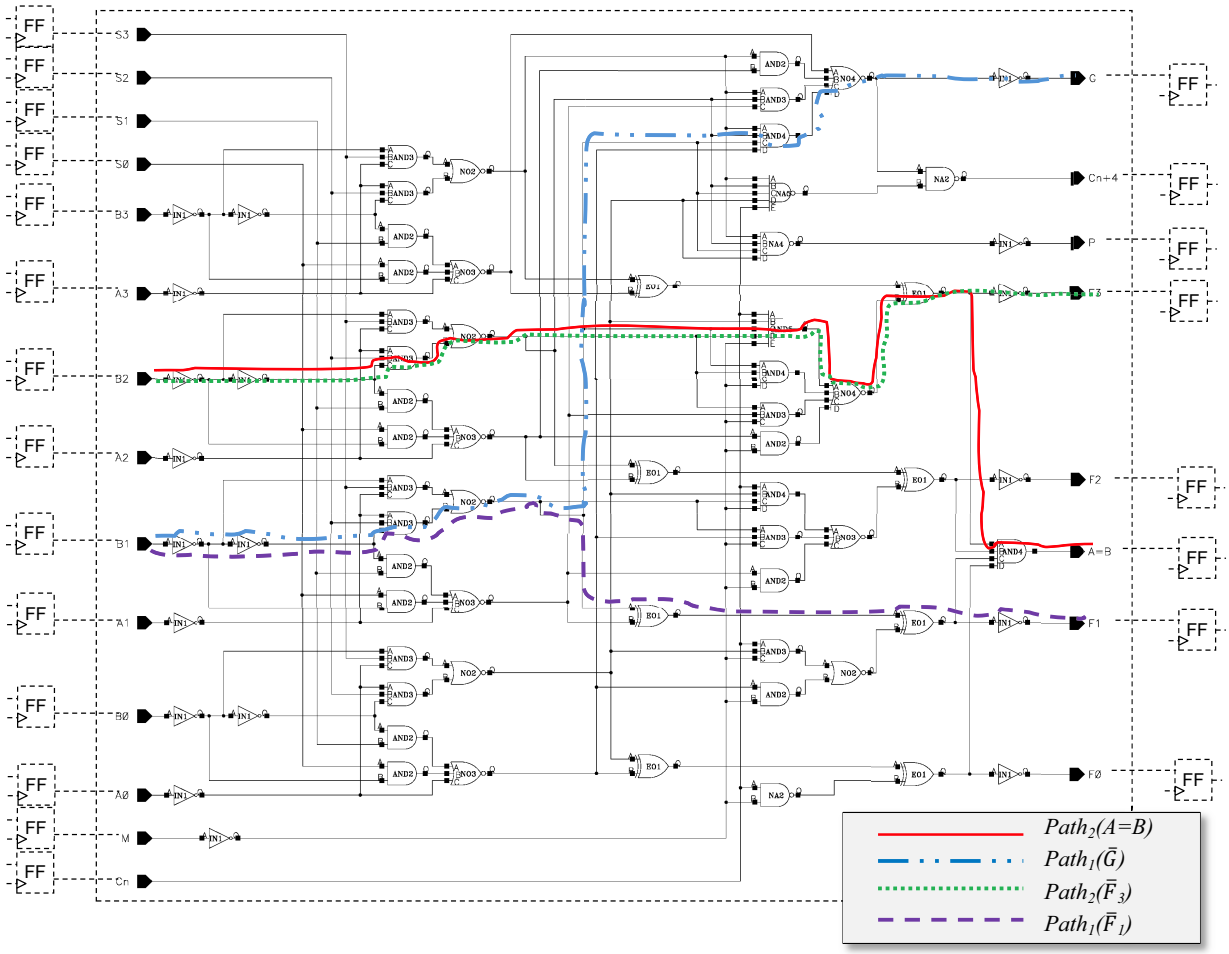

Fig. 2. NBTI-critical path identification case study on the *ALU_4bit* circuit.

$P_z{}^*_k(s_i)+ P_z{}^*_k(s_j))/|H_k(s_j)|$,

otherwise $P_z{}^*_k(s_j) = P_z{}^*_k(s_i)$;

- $B^*(s_j) = B^*(s_i) + B(s_j)$.

3. In some cases, after ordering the K-th 4-tuple under calculation has an accumulated delay $D_K(s_j)$ that is equal to $D_K(s_j) = D_{K+1}(s_j) = ... = D_{K+Lj}(s_j)$. In this situations $K+L_j$ 4-tuples with the longest accumulated delays will be included to further calculation. The Algorithm will complete when for all the primary outputs $s_j \in OUT$ $K+L_j$ 4-tuples are calculated.
4. After completion the algorithm has calculated up to $K+L_j$ paths with the longest delay for each outputs $s_j \in OUT$. The NBTI-critical paths are selected by analyzing the 4-tuplpes $M_{K+Lj}(s_j)$. These are the paths with delay that can be potentially incremented by NBTI to be larger than the initial time slack allocated by the designer for the path. Uncertainties caused by process variabilities may also be included to the paths selection by the Algorithm.

Consider the following example of a logic circuit fragment in Fig 1. We show the steps of iteratively calculating the 4-tuples M_k for the circuit lines s_i. As a result, five 4-tuples are calculated for the primary output s_{10} and seven 4-tuples for the primary output s_{10}, respectively. Although, the path $M_1(s_{10})$ has a slightly higher delay than $M_1(s_{11})$ (49 versus 45 ns), it is less critical with respect to NBTI because the signal probability of $M_1(s_{11})$ is considerably higher than of $M_1(s_{10})$ with $P_z{}^*(s_{11})$=0.8 and $P_z{}^*(s_{10})$=0.4, respectively. Therefore, path represented by $M_1(s_{11})$ should be considered as a more NBTI-critical. Concerning the fanout points on paths $M_1(s_{10})$ and $M_1(s_{11})$, both paths are equal with $B^*(s_{10})= B^*(s_{11})$=2. In this example the value of K is set to 3. However, four and five 4-tuples are selected for the s_{10} and s_{11} outputs correspondingly because of the equal accumulated delay of several paths.

V. Case Study

The proposed approach is demonstrated on a *ALU_4bit* circuit shown in Fig 2., which is a design of the 4-bit ALU circuit 74HC/HCT181 from Philips [9] with minor modifications. It is a gate-level combinational logic design (as

TABLE 1 NBTI-CRITICAL PATHS IN ALU_4BIT CIRCUIT

Output	$\overline{P}$				$\overline{G}$				$\overline{F}_3$				$\overline{F}_2$				$\overline{F}_1$				$\overline{F}_0$				C_{n+4}				A=B			
Param.	*D*	P_z^R	P_z^F	*B*	*D*	P_z^R	P_z^F	*B*	*D*	P_z^R	P_z^F	*B*	*D*	P_z^R	P_z^F	*B*	*D*	P_z^R	P_z^F	*B*	*D*	P_z^R	P_z^F	*B*	*D*	P_z^R	P_z^F	*B*	*D*	P_z^R	P_z^F	*B*
Paths *1*	37.0	0.51	0.67	13	**51.0**	0.57	**0.86**	15	59.3	0.49	0.56	15	56.3	0.49	0.56	15	**51.0**	0.4	**0.56**	15	49.0	0.45	0.56	12	52.3	0.57	0.86	15	65.3	0.49	0.56	15
2	37.0	0.51	0.33	13	51.0	0.57	0.57	15	59.3	0.48	0.78	15	55.0	0.49	0.56	15	50.0	0.52	0.44	12	47.7	0.41	0.44	13	52.3	0.57	0.57	15	65.3	0.48	0.78	15
3	34.0	0.50	0.40	11	49.7	0.56	0.57	15	58.0	0.48	0.78	15	53.3	0.48	0.38	13	50.0	0.47	0.33	13	46.0	0.44	0.63	10	51.0	0.56	0.57	15	64.0	0.48	0.78	15
4	34.0	0.50	0.60	11	49.0	0.63	0.71	12	58.0	0.49	0.56	15	53.3	0.48	0.38	13	48.0	0.38	0.38	13	44.7	0.40	0.63	11	50.3	0.63	0.71	12	64.0	0.49	0.56	15
5	34.0	0.51	0.40	11	48.0	0.58	0.67	13	56.7	0.48	0.56	15	53.3	0.48	0.50	13	48.0	0.38	0.38	13	44.7	0.39	0.50	11	49.3	0.58	0.67	13	62.7	0.48	0.56	15
6	34.0	0.50	0.40	11	48.0	0.58	0.83	13	56.3	0.48	0.75	13	53.0	0.54	0.44	12	48.0	0.38	0.50	13	44.7	0.40	0.63	11	49.3	0.58	0.83	13	62.3	0.48	0.75	13
7	34.0	0.51	0.40	11	48.0	0.59	0.67	13	56.3	0.48	0.63	13	53.0	0.49	0.33	13	47.0	0.46	0.38	11	42.0	0.44	0.75	11	49.3	0.59	0.67	13	62.3	0.48	0.63	13
8	34.0	0.51	0.40	11	48.0	0.59	0.67	13	56.3	0.48	0.63	13	52.0	0.49	0.50	13	47.0	0.46	0.50	11	42.0	0.44	0.75	11	49.3	0.59	0.67	13	62.3	0.48	0.63	13
9	33.7	0.51	0.50	11	48.0	0.58	0.67	13	56.3	0.48	0.63	13	52.0	0.49	0.38	13	47.0	0.46	0.50	11	40.7	0.40	0.38	12	49.3	0.58	0.67	13	62.3	0.48	0.63	13
10	33.7	0.51	0.33	11	48.0	0.58	0.67	13	56.3	0.48	0.63	13	52.0	0.49	0.38	13	47.0	0.51	0.50	10	40.7	0.40	0.57	10	49.3	0.58	0.67	13	62.3	0.48	0.63	13
11									56.3	0.48	0.63	13									40.7	0.40	0.50	12					62.3	0.48	0.63	13
12																													62.3	0.49	0.56	15

outlined by the dashed border) that may have flip-flops at the inputs and outputs.

To identify the NBTI-critical paths in the *ALU_4bit* circuit, first, the individual delays in the logic gates were calculated by Formula 1. Further the Algorithm with *K* set to 10 has selected for each of the eight outputs in the circuit $K+L_j$ delay-critical paths with delay values *D*. Table 1 shows the values of *D* for each of the selected paths in the columns titled *"D"*. The number of selected path was from 10 to 12. Assume, the NBTI caused circuit performance degradation is up to 5%. Then only the first several paths (separated by the bold line in Table 1) from the sets selected for each output shall be left for further analysis. E.g. the third path for the output '$\overline{P}$' with *D*=34 cannot become slower than the first path with *D*=37, i.e. (34+5%) < 37. Therefore in this example *K* set to 10 was sufficient to select all the delay-critical paths necessary NBTI-critical paths analysis (under assumption that NBTI caused performance degradation is up to 5%). Otherwise *K* had to be increased.

Further two sets of stimuli *T* were applied to the circuit. T_R is a set of pseudo-random patterns and T_F is a set of functional patterns generated in accordance with [9]. Columns *"*P_z^R*"* and *"*P_z^F*"* in Table 1 show the values of average gate input signal probability $P_z*(s_i)$ calculated for the selected paths by applying these two stimuli sets correspondingly. Columns *"B"* show the accumulated number of fan-out branches $B*(s_j)$ along the selected paths.

The analysis results show that the outputs '$\overline{F}_3$' and 'A=B' have the most NBTI-critical paths. They are highlighted by gray in the Table 1 and shown in Fig. 2 as $Path_2(A=B)$ and $Path_2(\overline{F}_3)$.

Assume an NBTI mitigation approach requires choosing which path of the other two paths shown in Fig. 2 is more NBTI-critical. One of these two paths is the first path of the output '$\overline{G}$' and the other one is the first path of the output '$\overline{F}_1$' in Table 1 (highlighted by bold and shown as $Path_1(\overline{G})$ and $Path_1(\overline{F}_1)$ in Fig. 2). Both of the paths have the same delay value *D*=51.0. The proposed approach will select the path of the output '$\overline{G}$' as more NBTI-critical because its signal probability value is higher and it is estimated degradation is faster. Also the underlined values of in columns *B* demonstrate that decision about which path is NBTI-critical can be made by considering the gate fan-out degree parameter.

Solutions to mitigate NBTI on the identified NBTI-critical paths will be studied in the future work.

VI. CONCLUSIONS

In this paper we proposed an approach to identify NBTI-critical paths in nanoscale logic that is based on the combined analysis of the following parameters: *gate input signal probability (SP)* and the *gate fan-out degree*. These parameters are applied to all nodes along with the delay-critical paths of the circuit.

The proposed approach was demonstrated on an industrial ALU circuit design. The results have shown the proposed technique is very effective on identifying NBTI-critical paths.

On the continuation of this work, solutions to mitigate NBTI will be studied and applied to the paths identified as NBTI-critical. Among these solutions, three are currently under investigation: the addition of aging sensors at the end of critical paths; the replacement of gates by NBTI-robust gates along these paths; and the application of "rejuvenating" stimuli at the inputs of these paths.

ACKNOWLEDGMENTS

The work has been supported in part by CNPq (Science and Technology Foundation, Brazil) under contracts n. 303701/2011-0 (PQ) and n. 483037/2011-7 (Universal), by Estonian ICT program project FUSETEST, by Research Centre CEBE funded by European Union through the European Structural Funds and by Estonian SF grants 8478 and 9429.

REFERENCES

[1] S. Mahapatra, D. Saha, D. Varghese, P. B. Kumar, On the Generation and Recovery of Interface Traps in MOSFETs Subjected to NBTI, FN, and HCI Stress, IEEE Trans. Electron. Dev. 53, 7, 1583-1592, 2006.

[2] Ing-Chao Lin, Chin-Hong Lin, Kuan-Hui Li, Leakage and Aging Optimization Using Transmission Gate-Based Technique, IEEE Trans. on Computer-Aided Design of Integratedd Circuits and Systems, Vol. 32, No. 1, Jan. 2013, pp. 87-99.

[3] C. Ferri, D. Papagiannopoulou, R. Iris Bahar, A. Calimera, NBTI-Aware Data Allocation Strategies for Scratchpad Memory Based Embedded Systems, IEEE 12th Latin American Test Workshop (LATW'11), March 27-30, 2011, Porto de Galinhas, Brazil.

[4] F. Ahmed, L. Milor, Reliable Cache Design with On-Chip Monitoring of NBTI Degradation in SRAM Cells using BIST, 2010 28th IEEE VLSI Test Symposium, pp. 63-68.

[5] S. Khan, S. Hamdioui, Modeling and Mitigating NBTI in Nanoscal Circuits, 2011 IEEE 17th International On-Line Testing Symposium, pp. 1-6.

[6] Ceratti, A.; Copetti, T.; Bolzani, L.; Vargas, F.; , "Investigating the use of an on-chip sensor to monitor NBTI effect in SRAM," *Test Workshop (LATW), 2012 13th Latin American* , vol., no., pp.1-6, 10-13 April 2012

[7] S. V. Kumar, C. H. Kim, and S. S. Sapatnekar, "NBTI-aware synthesis of digital circuits," in *Proc. Design Autom. Conf.*, 2007, pp. 370–375.

[8] I. Sutherland, R. Sproull, D. Harris "Logical Effort:Designing Fast CMOS Circuits", Morgan Kaufmann, 1999.

[9] Data sheet "74HC/HCT181 4-bit arithmetic logic unit", Philips, 1998