POLITECNICO DI TORINO
Repository ISTITUZIONALE

On NBTI-induced Aging Analysis in IEEE 1687 Reconfigurable Scan Networks

Original
On NBTI-induced Aging Analysis in IEEE 1687 Reconfigurable Scan Networks / Damljanovic, A., Squillero, G., Cem Gursoy, C., Jenihhin, M.. - ELETTRONICO. - (2019). (27th IFIP/IEEE International Conference on Very Large Scale Integration Cuzco, Peru 6-9 October 2019) [10.1109/VLSI-SoC.2019.8920313].

Availability:
This version is available at: 11583/2742355 since: 2020-04-26T19:04:45Z

Publisher:
IEEE

Published
DOI:10.1109/VLSI-SoC.2019.8920313

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

17 September 2026

On NBTI-induced Aging Analysis in IEEE 1687 Reconfigurable Scan Networks

Aleksa Damljanovic, Giovanni Squillero
Politecnico di Torino,
Torino, Italy

Cemil Cem Gürsoy, Maksim Jenihhin
Tallinn University of Technology,
Tallinn, Estonia

***Abstract*—The Negative Bias Temperature Instability (NBTI) phenomenon is one of the main reliability issues in today's nanoelectronic systems. It causes increase in threshold voltage of pMOS transistors, thus degrading signal propagation delay in logic paths between flip-flops. Recently, IEEE published a new standard IEEE 1687 for Reconfigurable Scan Networks (RSN) to facilitate access to embedded instrumentation within an integrated circuit. In the field, the RSN infrastructure is often exploited for fault-management in failure-sensitive critical parts of the system. Therefore, the severity level of a fault in the RSN itself is very high, thus, amplifying the impact of the reliability issues caused by the aforementioned effect. To the best of the authors' knowledge no approach has been proposed to investigate or address this issue so far. In this paper, we analyze the effect of NBTI-induced aging in RSNs from architectural and operational (functional) perspectives and present a novel technique to mitigate the degradation. The methodology is demonstrated on a a case-study example and the effectiveness of our approach is evaluated on a sub-set of ITC2016 benchmark RSN designs.**

***Index Terms*—IJTAG, NBTI, Aging, Mitigation**

I. Introduction

The expansion of contexts where electronic systems serve people has also led to a significant growth of the number and variety of safety- and mission-critical embedded systems. This comes along with the trend of implementation technology miniaturization that allows boosting the nanoelectronic systems' functionality, but brings the lifetime reliability concern to the front. Autonomous and unmanned vehicles, robotic systems, fly-by-wire aircrafts and complex industry automation machines are often empowered by advanced computing systems and require extreme levels of safety and dependability for years of the operational lifetime.

The phenomenon of nanoelectronics aging was addressed by numerous related works focusing e.g., on the degradation issues caused by the Negative Bias Temperature Instability (NBTI) in memories [1], [2] and in functional logic [3], [4]. To mitigate such effects in functional logic, approaches exist at different levels, e.g., some are based on redesign or transistor sizing techniques [5], while others rely on modifying voltage and frequency of the circuit [6] or resort to NBTI-aware synthesis [7]. Mitigation on a circuit level has been exploited in [8] where authors propose using idle-time of the processor and unused bits in source operands of the instruction.

An efficient approach to mitigate reliability issues in complex critical systems is their instrumentation with additional embedded fault-tolerance and fault-management infrastructures. The recent continuous increase of embedded instrumentation that is used to support monitoring, test, debug, calibration and configuration as well as the systems complexity in general, made it difficult to organize and perform access to such instruments in an efficient manner. Additionally, the need to provide a scalable and reusable interface for managing and incorporating different types of instruments became another important requirement. To alleviate such problems IEEE society has recently published a standard referred to as the IEEE Std 1687 (IJTAG) [9]. To the best of authors' knowledge, no work has addressed the reliability issues caused by NBTI-induced aging in the IJTAG RSNs, so far. In [10]–[12], different techniques for generating patterns to test permanent faults affecting reconfigurable modules in RSNs have been presented.

Another important issue that arises is how to generate efficiently reconfiguration patterns that provide minimum access time. Works such as [13], [14] address optimized access and retargeting techniques. A number of works propose to reuse the IJTAG infrastructure for system's reliability and fault management to address aging effects in the system itself. In [15]–[18] different frameworks are proposed for on-line health monitoring, fault management, and system's degradation.

In this paper, we analyze the effect of NBTI on logic paths in IJTAG RSNs by estimating delay resorting to the model introduced in [4]. It should be noted that this work does not aim at developing new or extending already existing technology-level models for NBTI.

The remainder of this paper is structured as follows. Section II describes the background, introducing RSNs and some basic terminology, together with a hierarchical NBTI-induced gate-delay modeling approach. Section III proposes, first, a methodology for assessment of the aging effect depending on the RSN architecture and configuration and, second, a corresponding mitigation technique based on workload alteration. Section IV presents a case study. Section V reports preliminary experimental results. Finally, Section VI draws conclusions for the paper.

II. Background

This work relies on two fundamental aspects and joins them. The first one is known as a Reconfigurable Scan Network for which we present some basic constructs and the idea behind

its structure together with how they are operated. The second one is the model for computing the NBTI-induced delay at the gate-level, based on intensive SPICE simulations of individual gates.

A. Basics of Reconfigurable Scan Networks (RSNs)

A designer may acquire different strategies when organizing the access to the set of instruments. To avoid overhead that was present in IEEE 1149.1-compliant circuits where all scan registers are connected serially, the designer may introduce hierarchy with insertable scan segments or incorporate mutually excludable scan segments into the circuit, which is supported by the IEEE 1687 standard. The latter is the key feature of RSNs since it enables to partition the set of instruments into segments controlled by programmable components, and then dynamically decide which segments are currently accessible and which are bypassed. One of the programmable component introduced by IEEE 1687 is a Segment Insertion Bit (SIB), used to bypass a segment of a network. As a segment can be simply one or several TDRs or a sub-network consisting of TDRs and other programmable components, it is possible to create a hierarchical network.

Simplified schematic of a possible implementation of a SIB based on a one-bit shift-update register and a two-input multiplexer is given in Fig. 1(a). SIBs can be programmed by shifting a bit into their S flip-flop and latching that bit into the parallel U latch. SIB is said to be *de-asserted* if the latched value is 0, and the scan-path is from the si terminal, to the so terminal via the S flip-flop, bypassing the segment between the tsi and fso terminals. On the other hand, if the value latched in the bit is 1, the SIB is in *asserted* state and the scan-path includes the segment connected between the tsi and fso terminals of that SIB. In this paper, the symbol shown in Fig. 1(b) is used to represent a SIB.

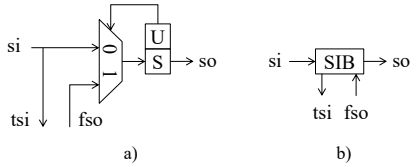

Fig. 1. Simplified schematic of a SIB module (a) and its symbol (b)

Another type of IEEE 1687 RSN architecture is achieved by using shift-update registers and ScanMuxes. As an example, consider the network shown in Fig. 2(a) in which a two-bit shift-update register is used to select among four inputs of a 4-to-1 ScanMux. Similarly to the SIB, the configuration of the ScanMux can be performed by shifting the required values into the S shift flip-flops of the control register and then latching the shifted bits into the U latches. In the rest of this paper, the symbol shown in Fig. 2(b) is used to represent the shift-update register that controls a ScanMux.

In addition to the multiplexers and sequential elements composing registers, these networks also include combinational logic and control signals. In order to keep the drawings simple, the clock, reset, control signals (namely, *shift*, *update*, and *capture*), and the *select* signal that is used to gate the control signals are not shown.

Changing the configuration of the network corresponds to changing the states of SIBs and ScanMuxes, thus choosing a subset of TDRs that are to be accessed. For the network in Fig. 4 with seven TDRs, six SIBs and one ScanMux a number of $2^7 = 128$ configurations is possible, although some of them are equivalent from the functional point of view: diverse configurations result in having same set of accessible registers (instruments) and therefore the same active path (length). The length of the active path is equal to the sum of lengths of individual registers placed along that path, starting from the serial input to the serial output of the circuit.

B. Hierarchical Modelling of NBTI-Induced Delays

Bias Temperature Instability (BTI) phenomenon causes threshold voltage V_{TH} shift on MOS transistors. Two types of BTI are defined depending on the type of stressed transistor. Negative BTI (NBTI) caused by the negative gate stress occurs on pMOS transistors, while the Positive BTI (PBTI) is related to nMOS transistors due to the positive gate stress. This paper focuses on NBTI [19] as it is considered to be a dominant aging mechanism for the current implementation technologies.

In [4], fast yet accurate modeling of NBTI-induced delays at the gate level is proposed. First, technology and environment dependent curve of the threshold voltage shift as a function of the transistor's gate input signal probability $\Delta V_{THp}(P_z)$ has to be obtained at the transistor level. Then, technology and environment dependent curves of the gate delay degradation as a function of the threshold voltage shift $\Delta t(\Delta V_{THp})$ for each gate type in the netlist (e.g. INV, 2NAND, 2NOR) assumed for gate-level implementation.

In the NBTI effect analysis, we rely on a reaction-diffusion (R-D) based predictive model for dynamic NBTI presented in [19], [20]. This model predicts the long term threshold voltage V_{THp} degradation due to NBTI at a time $t > 1,000s$ at high frequencies [19]. It captures the dependence of NBTI on a gate input signal probability P_z (probability that the related pMOS transistor is under stress) in addition to its dependence on other key process and design parameters as presented in [19]. The values of the involved technology and

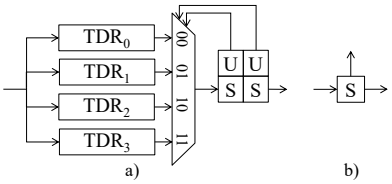

Fig. 2. Simplified schematic of a ScanMux module (a) and its control register's symbol (b)

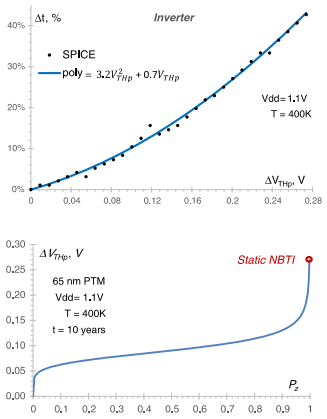

Fig. 3. Threshold voltage shift V_{THp} as a function of signal probability P_z (up). Gate delay increase Δt dependency on voltage threshold shift ΔV_{THp} in an inverter gate (down).

environmental parameters can be summarized by a parameter γ in the following form:

$$|\Delta V_{THp}| = \gamma(\frac{P_z}{1-P_z})^n \quad (1)$$

Note that Equation 1 is valid only for dynamic stress, as ΔV_{THp} becomes infinite when P_z reaches the value 1. Therefore, the upper limit of ΔV_{THp} is defined by static NBTI models [19]. Equation 1 represents a convenient mathematical function of the threshold voltage V_{THp} degradation dependence on the signal probability for the gate input signal $P_z(x_i)$ of a pMOS transistor. In the equation, $n = 1/6$ represents the variety of the dominant diffusion species (H or H_2) expressed by the time exponent parameter and $\gamma = 0.0904$ represents a parameter that incorporates the selected technology and environmental variables. In Fig. 3 (up), the corresponding dependence is illustrated for PTM 65 nm technology [20] after 10 years of NBTI-induced degradation at constant temperature $T = 400K$ with supply voltage $V_{DD} = 1.1V$. The calculated value of ΔV_{THp} for static NBTI is $0.27V$. The model allows fast estimation of NBTI-induced V_{THp} shifts.

A set of SPICE simulations for each logic cell is used to create a polynomial curve to model the gate delay degradation (see Fig. 3) (down):

$$\Delta t_{gate} = \lambda * \Delta V_{THp}(X_i) + \mu * (\Delta V_{THp}(X_i))^2 \quad (2)$$

Here, Δt_{gate} is the gate output delay increase (in percentage) compared to the nominal gate delay, $\Delta V_{THp}(x_i)$ is the change of V_{THp} for the stressed pMOS transistor at the gate input x_i, while λ and μ are technology dependent constants. For example, in our experiments λ and μ are set to 0.7 and 3.2 for the INV gate. In case a logic gate consists of multiple cascaded pMOS transistors, both their physical location relative to the output node and their $0 \rightarrow 1$ output transition impact the gate delay degradation. Each combination of gate input values is modelled by different values of the constants λ and μ in Equation 2. For an alternative technology and different parameters such as temperature and supply voltage, additional SPICE simulations are required to obtain the curves for modelling the gate-delay degradation.

III. Methodology

This section addresses methodology behind NBTI-induced aging evaluation in RSNs according to the aging model introduced in Section II. Additionally, structural analysis and mitigation approach is presented here.

A. Aging Evaluation

The design has to be synthesized to gate-level to allow inverting gates (i.e. NAND, NOR, INV) and FFs only. Here we present the steps that are required in order to evaluate aging on RSNs.

- The first step is modeling the threshold voltage shift as a function of the transistor's gate input signal probability $\Delta V_{THp}(P_z)$ and the gate delay degradation as a function of the threshold voltage shift $\Delta t(\Delta V_{THp})$ for NAND, NOR and INV gates as described in [4] and summarized in Section II.
- Then, the design is simulated to obtain signal probabilities of the each gate input.
- Next, these signal probabilities are mapped to the curve parameters that were obtained in the first step to calculate NBTI-induced gate delays.
- In the final step, paths between FFs and primary inputs/outputs are extracted to find the critical one after aging. Since the design consists of inverting gates, when a path is activated, all gate inputs on the path will transition to another value. To find aged delay of a path, all gate delays on the path must be summed up. If the output transition of a gate is $0 \rightarrow 1$, NBTI-induced delay must be added to the nominal delay of the gate. Otherwise, only nominal delay of the gate is used when calculating path delays.

B. Architectural analysis

RSNs are interesting from the architectural point of view. Since an RSN is a dynamically reconfigurable network that provides a means for creating a hierarchy, organizing set of TDRs can be performed in numerous ways. It is always a question of trade-off, since this decision may depend on the frequency of access to a certain instrument and overhead in terms of time (clock cycles) required to perform any access in general.

In this paper, we decided to examine simple restructuring of the network by organizing complete set of TDRs in such a way that they can be accessed individually through serially connected SIBs as shown in the case study (Fig. 6).

A SIB may provide access to a scan segment containing additional SIBs, thus deepening the hierarchy. Having a regular structure such as a series of SIBs alleviates identifying the critical logic path(s). It is always the one(s) leading to the FFs related to the longest register. The former, however, for

accessing certain TDRs may require multiple reconfiguration operations, while in the latter one, only one configuration cycle is required to assert/de-assert a SIB and include/exclude corresponding TDR to/from the active path. On the other hand, serially connected SIBs produce overhead since Segment Insertion Bits (configuration bits) always belong to the active path. Removing hierarchy shortens logic paths in general and therefore reduces nominal delay. However, as it has been confirmed experimentally, it does not impact significantly the aging-induced delay.

C. Mitigation by workload alteration

As the structure of the RSNs can be quite regular, some paths share, i.e., they traverse the same gates and the same signal lines. Furthermore, it is common that for each of those paths, the remaining parts have the exactly same structure where the gates of the same type are encountered in the same order. As a consequence, an abundant number of paths has the same nominal delay and ages in the same mode.

To perform mitigation, an algorithm creates a list of most critical registers, i.e. sorts the registers in a descending order based on the value equal to the sum of the hierarchical level of the segment they are positioned in and their length since it impacts the fan-out of some gates that belong to the path between primary input or flip-flop output and the input of the flip-flop in the shift stage of the corresponding register. Further on, a function described in Algorithm 1 is called to generate the set of configurations for reaching desired register(s). From the internal network model, it is possible to find controllable module providing access to the particular scan segment and configuration bit(s) corresponding to the aforementioned module. *configureMux* is called recursively saving necessary states and updating the list of registers to be accessed removing the ones already encountered while accessing the current one.

Algorithm 1 Listing programmable modules (SIBs, SMs) and corresponding states to be applied in order to reach particular TDR

```
function GENERATECONFIGURATIONS(gen, len, i, d)
    tdr ← regList(0)
    mux ← GETMUX(tdr)
    while mux ≠ null do
        currentMuxEnc ← mux current configuration
        regSegEncoding ← tdr's segment encoding
        if currentMuxEnc ≠ regSegEncoding then
            CONFIGUREMUX(gen, mux, regSegEncoding)
        PUTONPATH(mux)
        index ← index + 1
        tdr ← regList(i)
        mux ← GETMUX(tdr)
```

One of the advantages of the RSNs is that the mitigation can be performed in parallel while executing required operations without affecting the final result. Of course, the overhead of such operations exists in terms of additional clock cycles required to access targeted registers and perform read/write operations.

IV. CASE STUDY

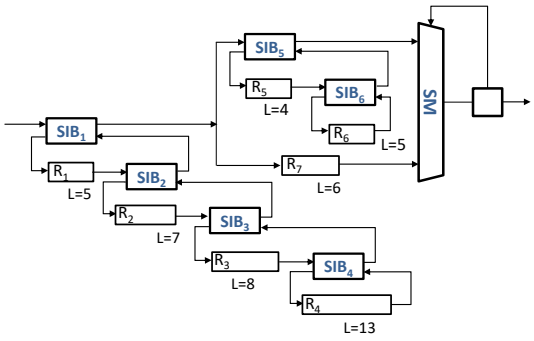

Fig. 4. Case-study RSN with hierarchy levels - one SM, six SIBs and seven TDRs

The following case study illustrates how we analyzed and identified the most critical logic path at the gate level of the RSN and the effect of NBTI-induced aging in RSNs. The mitigation technique is also applied to this example and obtained results are reported.

The considered RSN, shown in Fig. 4, is motivated from an automotive context and it consists of two sub-networks. The first one has four TDRs - R_1 to R_4 which are placed hierarchically behind 4 SIBs - SIB_1 to SIB_4 and are accessed less frequently. The second section contains registers that are to be accessed more often. It includes three TDRs - R_5 to R_7, two of which are placed behind SIB_5 and SIB_6 located on one input segment of the ScanMux (SM). The remaining register R_7 has been placed directly on the second input segment of the multiplexer. There are no remotely controlled modules, i.e, all of them are controlled in-line. As it has been outlined previously in Section II-A, the function of these instruments may vary since they can be used for monitoring (sensors), debug and calibration/configuration, or BIST control.

For the circuit shown in Fig. 4, we created a workload containing series of operations to change the network configurations and read, i.e., write to/from the registers R_5, R_6 and R_7. In the gate-level design we have identified the critical (longest) NBTI-degraded path that corresponds to the path having the longest total delay consisting of the nominal gate delay and NBTI-induced delays Δt for the gates along that path.

Gate-level description of the circuit shown in this case study contains 695 gates, while the total number of logic paths sums up to 2238. Fig. 5 shows the path with the largest nominal delay and is at the same time the longest NBTI-degraded path for the workload provided. When the structure of the circuit is taken into account we can identify some regularities. The longest path starts with the top level signal *SEL* and it passes several gates (U20, U19, U18 and U17) up to the *tosel_SIB4* for the SIB_4 module. This signal is used to gate signals *CE* and *SE* responsible for allowing capture, i.e., shift operations. Gates *R4xU138* and *R4xU137* have a fan-out since the signals at their output are used to control single FFs belonging to the same register R_4. In this circuit the register R_4 has the length

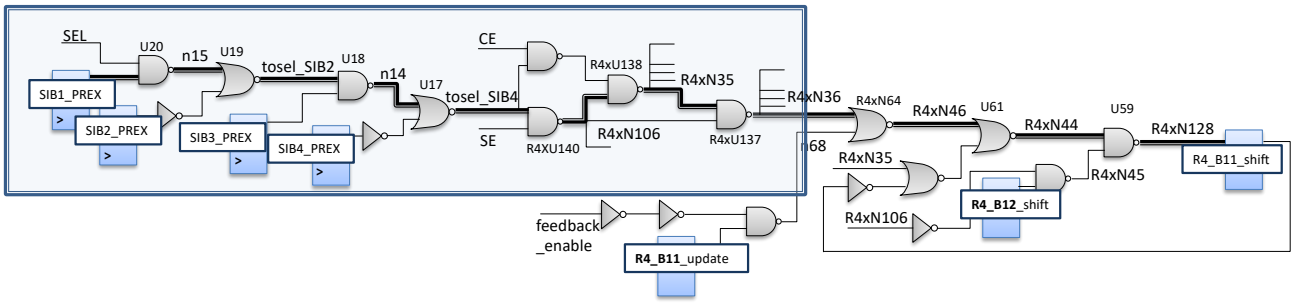

Fig. 5. Gate-level schematic of the longest logic path in the case-study RSN

of 13, and therefore consists of 13 FFs (the shift stage), gate R4xU138 has a fan-out $13 + 1 = 14$, while gate R4xU137 has a fan-out of 13. The section within the borders is a source of the signals and is shared between logic paths leading to the shift flip-flops in a register. The signals belonging to the section out of borders in Fig. 5 lead to the second FF in R_4. They are in control whether the same value is kept in the flip-flop if all operations are disabled, or the value from the preceding FF should be stored, or the value from the update stage should be captured since the feedback functionality is used. The identification of the longest path takes into account the hierarchy levels and the length of the registers.

For circuits containing registers with a large length, i.e., with the great number of FFs, some gates may have a high fan-out. In such case, the synthesis tool may introduce additional gates such as buffers (double inverters) to reduce the overall load on the driving gate and decrease the transition time of the net.

For the longest path, the nominal delay is 86.67 time units, while the total delay after estimating NBTI-induced delay equals to 103.63 time units. Therefore, the delay increase given in percentages is 19.58%. After applying the mitigation technique as already discussed in Section III, the overall delay after NBTI-degradation is 92.5 time units, thus resulting in only 6.73% of delay degradation, which is significantly less with respect to the value obtained when applying the original workload.

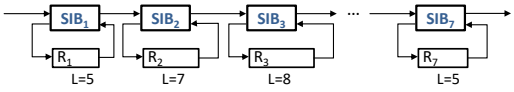

Fig. 6. Simplified schematic of a SIB module (left) and its symbol (right)

V. Experimental results

In this work, we report the results on two networks from the ITC2016 set of benchmark networks [21] - Mingle and N17D3, for which some basic properties are given in Table I. For both benchmarks, a flattened network with a separate SIB for each of the registers is generated. The workload is adjusted preserving the access order, set of activated instruments and written values from the workload created for original networks. In Table II for each of the considered benchmarks and its flattened version the following data are reported: nominal delay (t_{nom}) and total delay ($t_{nom} + \Delta t$) with increase in percentages before and after mitigation. To show its effectiveness, we report the ratio of the delay increase in percentages, before and after mitigation. For example, in Mingle network, we have been able to reduce the NBTI-aging effect by 2.1 times, since the delay increase has changed from 19.5% to 9.4%.

TABLE I
Benchmark networks

Network	SIB	SM	Conf. bits	Max depth	Max path	Scan cells
Mingle	10	3	13	4	171	270
N17D3	7	8	15	4	372	462

Synopsys Design Compiler tool was used for synthesising the circuit at the gate-level from the description in RTL (VHDL) with the flattening of the hierarchy and without any optimization. For this purpose, 65 nm technology library has been chosen with imposing certain constraints primarily related to the choice of primitive gates and flip-flops (2-input NAND gate, 2-input NOR gate, inverter gate, flip-flop with the reset functionality). The aforementioned comes from the limited availability of data regarding the model and aging characterization (65 nm PTM). For reporting results on different technology nodes the same procedure from Section II-B has to be repeated. Open-source tool zamiaCAD [22] has been used for simulation of the design. Simulation results are used to record the signal probabilities of the gate inputs. All calculations (nominal/NBTI-induced gate delays, path extraction and their delays) are automated in zamiaCAD using Python scripts.

Regarding the results, NBTI-critical path for the N17D3 network changes after mitigation. Before altering the workload, critical path did not correspond to the path with the largest nominal delay, since the latter lead to the register which was accessed in the workload. After performing/adding additional TDR accesses to reduce the aging effect, the NBTI-critical path becomes the one with the largest nominal delay.

Although the mitigation approach is scalable and applicable for larger and more complex designs (runtime less than

TABLE II
EXPERIMENTAL RESULTS

Network	Original circuit					Flattened hierarchy				
	t_{nom}	$t_{nom} + \Delta t$	t_{nom} (mit.)	$t_{nom} + \Delta t$ (mit.)	Decrease	t_{nom}	$t_{nom} + \Delta t$	t_{nom} (mit.)	$t_{nom} + \Delta t$ (mit.)	Decrease
Mingle	125.67	150.21 (19.5%)	125.67	137.54 (9.4%)	2.1	104.33	127.35 (22.1%)	104.33	115.13 (10.4%)	2.1
N17D3	126.67	150.76 (19.0%)	134.33	144.22 (7.4%)	2.6	115.33	135.86 (17.8%)	115.33	128.13 (11.1%)	1.6

one second for the reported benchmarks), simulating design with the test-bench stimuli generated from the workload for obtaining signal probabilities and calculating delay is computationally expensive and therefore time-demanding. The whole framework was run on a modest laptop with a dual-core CPU.

VI. CONCLUSIONS

The paper has proposed a methodology for assessment and mitigation of NBTI aging induced delays in logic paths within IEEE 1687 IJTAG Reconfigurable Scan Networks. While RSNs are commonly used to provide fault management and embedded instrumentation access, such as safety mechanisms, in advanced safety- and mission-critical electronic systems, a failure in such infrastructure itself has a high severity. The methodology is based on a scalable hierarchical (transistor-to-architecture) modelling of the NBTI impact on timing-critical logic paths in RSN implementations. The evaluation implies analysis of gate input signal probabilities based on the configurations and test data selected for the RSN infrastructure. The details of the methodology are demonstrated by a case study on an example RSN and the feasibility and efficiency are validated by experiments on a subset of ITC2016 RSN benchmarks. The experimental results demonstrate that RSNs can be impacted by significant NBTI-induced logic path delays and a simple proposed mitigation technique can reduce such delays up to 2.6 times. The future work is aimed at a comparative analysis of aging in the RSN gates and the functional part of the circuit.

ACKNOWLEDGEMENTS

The work has been supported by the European Commission through the Horizon 2020 RESCUE-ITN project under the agreement No.722325, by the Estonian research grant no. IUT19-1 and by the European Structural and Regional Development Funds.

REFERENCES

[1] F. Ahmed and L. Milor, "Reliable cache design with on-chip monitoring of nbti degradation in sram cells using bist," in *2010 28th VLSI Test Symposium (VTS)*, April 2010, pp. 63–68.

[2] C. Ferri, D. Papagiannopoulou, R. I. Bahar, and A. Calimera, "Nbti-aware data allocation strategies for scratchpad memory based embedded systems," in *2011 12th Latin American Test Workshop (LATW)*, March 2011, pp. 1–6.

[3] H. Kkner *et al.*, "Comparison of reaction-diffusion and atomistic trap-based bti models for logic gates," *IEEE Transactions on Device and Materials Reliability*, vol. 14, no. 1, pp. 182–193, March 2014.

[4] M. Jenihhin *et al.*, "Identification and rejuvenation of nbti-critical logic paths in nanoscale circuits," *Journal of Electronic Testin*, vol. 32, no. 3, pp. 273–289, Jun 2016.

[5] S. Khan and S. Hamdioui, "Modeling and mitigating nbti in nanoscale circuits," in *2011 IEEE 17th International On-Line Testing Symposium*, July 2011, pp. 1–6.

[6] A. Tiwari and J. Torrellas, "Facelift: Hiding and slowing down aging in multicores," in *2008 41st IEEE/ACM International Symposium on Microarchitecture*, Nov 2008, pp. 129–140.

[7] S. V. Kumar, C. H. Kim, and S. S. Sapatnekar, "Nbti-aware synthesis of digital circuits," in *2007 44th ACM/IEEE Design Automation Conference*, June 2007, pp. 370–375.

[8] J. Abella, X. Vera, and A. Gonzalez, "Penelope: The nbti-aware processor," in *40th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO 2007)*, Dec 2007, pp. 85–96.

[9] "IEEE standard for access and control of instrumentation embedded within a semiconductor device," *IEEE Std 1687-2014*, pp. 1–283, Dec 2014.

[10] R. Cantoro, M. Montazeri, M. Sonza Reorda, F. G. Zadegan, and E. Larsson, "On the testability of IEEE 1687 networks," in *Test Symposium (ATS), 2015 IEEE 24th Asian*. IEEE, 2015, pp. 211–216.

[11] R. Cantoro, F. G. Zadegan, M. Palena, P. Pasini, E. Larsson, and M. S. Reorda, "Test of reconfigurable modules in scan networks," *IEEE Transactions on Computers*, 2018.

[12] R. Cantoro, A. Damljanovic, M. Sonza Reorda, and G. Squillero, "A new technique to generate effective test sequences for reconfigurable scan networks," in *IEEE International Test Conference (ITC), 2018*.

[13] R. Krenz-Baath, F. G. Zadegan, and E. Larsson, "Access time minimization in ieee 1687 networks," in *2015 IEEE International Test Conference (ITC)*, Oct 2015, pp. 1–10.

[14] R. Baranowski, M. A. Kochte, and H.-J. Wunderlich, "Reconfigurable scan networks: Modeling, verification, and optimal pattern generation," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 20, no. 2, p. 30, 2015.

[15] A. Jutman *et al.*, "Effective scalable ieee 1687 instrumentation network for fault management," *IEEE Design Test*, vol. 30, no. 5, pp. 26–35, Oct 2013.

[16] A. Tsertov, A. Jutman, K. Shibin, and S. Devadze, "Ieee 1687 compliant ecosystem for embedded instrumentation access and in-field health monitoring," in *AUTOTESTCON 2018*, 09 2018, pp. 1–9.

[17] F. G. Zadegan, D. Nikolov, and E. Larsson, "On-chip fault monitoring using self-reconfiguring ieee 1687 networks," *IEEE Transactions on Computers*, vol. 67, no. 2, pp. 237–251, Feb 2018.

[18] A. Ibrahim and H. G. Kerkhoff, "Efficient utilization of hierarchical ijtag networks for interrupts management," in *2016 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, Sep. 2016, pp. 97–102.

[19] S. Bhardwaj, W. Wang, R. Vattikonda, Y. Cao, and S. Vrudhula, "Predictive modeling of the nbti effect for reliable design," in *IEEE Custom Integrated Circuits Conference 2006*, Sep. 2006, pp. 189–192.

[20] W. Wang *et al.*, "Compact modeling and simulation of circuit reliability for 65-nm cmos technology," *IEEE Transactions on Device and Materials Reliability*, vol. 7, no. 4, pp. 509–517, Dec 2007.

[21] A. Tšertov *et al.*, "A suite of IEEE 1687 benchmark networks," in *Test Conference (ITC), 2016 IEEE International*. IEEE, 2016, pp. 1–10.

[22] A. Tepurov, G. Bartsch, R. Dorsch, M. Jenihhin, J. Raik, and V. Tihhomirov, "A scalable model based rtl framework zamiacad for static analysis," in *2012 IEEE/IFIP 20th International Conference on VLSI and System-on-Chip (VLSI-SoC)*, 2012, pp. 171–176.