Asymmetric Aging Avoidance EDA Tool

Freddy Gabbay[1], Avi Mendelson[2], Basel Salameh[2] and Majd Ganaiem[2]

Ruppin Academic Center[1]
freddyg@ruppin.ac.il

Technion – Israel Institute of Technology[2]
{avi.mendelson, basel238, majd.ga}@technion.ac.il

2021 34th SBC/SBMicro/IEEE/ACM Symposium on Integrated Circuits and Systems Design (SBCCI) | 978-1-6654-2170-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/SBCCI53441.2021.9529984

***Abstract*— The latest process technologies have become highly susceptible to asymmetric aging, whereby the timing of logical elements degrades at unequal rates over the element lifetime, causing severe reliability concerns. Although several tools are available to handle asymmetric aging, such tools mainly rely on circuit or physical design approaches and offer a limited capability to handle large-scale ICs. In this paper, we introduce a flow and a tool to minimize the asymmetric aging effect in data path design structures. The proposed tool can be straightforwardly integrated as part of standard design flows of large-scale ICs. In addition, the tool can automatically analyze various designs at RTL or gate-level and identify logical elements which are suspectable to asymmetric aging. As part of the design flow, the tool automatically embeds a special logical circuitry in the design to eliminate asymmetric aging. Our experimental analysis shows that the proposed design flow can minimize the asymmetric aging effect and eliminate reliability concerns while introducing minor power and silicon area overhead.**

***Index Terms*— Asymmetric Aging, Reliability, Bias Temperature Instability, Asymmetric Aging aware EDA**

I. Introduction

Advanced VLSI process nodes have become highly susceptible to reliability concerns. New applications such as data centers, medical appliances, and automotive systems introduce major challenges for the semiconductors industry by imposing severe requirements on reliability [1, 2]. Today, transistor aging has been identified as one of the major reliability concerns in integrated circuits (ICs). Two physical mechanisms induce transistor aging: hot carrier injection (HCI) and bias-temperature instability (BTI) [2, 3, 6], both of which lead to transistor failure and degraded performance. The common approach to handling transistor aging is to apply extra timing margins to the clock cycle time. When timing degradation due to aging is symmetric, this can be a suitable solution; however, many logical elements may undergo asymmetric aging, whereby logical paths may experience critical timing violations even if the delay shift is relatively small.

BTI, which may arise when a constant voltage is applied to transistor gates, is the main cause of asymmetric aging [2]. Prior studies [3] report that, for BTI to induce transistor timing degradation, such voltages must be applied for durations on the order of seconds up to several weeks. This observation is highly disturbing because constant logical values, when applied to any logical element, may result in severe reliability issues. Prior work has found that dynamic power-saving techniques, which apply constant logical states, accelerate the likelihood of asymmetric aging [4].

The common techniques [5–18] to handle asymmetric aging rely on physical design solutions that are complex and may involve nonscalable analysis and simulation methods. Today, the problem has been further intensified because common Electronic Design Automation (EDA) tools have limited capabilities to offer adequate solutions to asymmetric aging.

In this paper, we extend and generalize the method presented in Ref. [5] and introduce an automated tool and design flow to avoid asymmetric aging. Whereas Ref. [5] introduced mechanisms to be manually inserted into the design to mitigate asymmetric aging, the present study proposes an automated EDA tool for data path logical structures. The proposed tool automatically analyzes general data-path design structures at RTL or gate-level and identifies logical elements that are suspectable to asymmetric aging. The proposed tool uses a special testbench that injects random data patterns into the tested circuitry while automatically monitoring the logical element signal probability and toggle rate. The tool can automatically generate special logical circuitry that is embedded in the design to avoid asymmetric aging while the device is in mission mode. The proposed design flow can be straightforwardly integrated as part of standard design flows of large-scale ICs.

The contributions of this paper is summarized as follows:

1. The proposed tool automatically analyzes data-path design structures at RTL or gate-level and identifies logical elements suspectable to asymmetric aging.
2. The tool automatically generates a special circuitry to avoid asymmetric aging. The embedded circuitry injects pseudorandom data at low rates to avoid static BTI stress.
3. The proposed design flow can be straightforwardly integrated as part of standard design flows of large-scale ICs.
4. An analysis of the experimental results shows that the design flow incurs negligible power and area overhead.
5. Timing simulations combined with aging models indicate that the proposed tool efficiently eliminates asymmetric delay shift and thereby eliminates reliability issues due to BTI.

The remainder of this paper is organized as follows: Section 2 presents asymmetric aging and discusses previous studies. Section 3 presents our proposed tool and design flow. Section 4 describes our experimental results and, finally, Section 5 summarizes the study and suggests future research directions.

II. Asymmetric Aging

Starting at 28 nm process technology nodes and below (16, 7, and 5 nm), the vulnerability of ICs to reliability issues grows significantly. Reliability issues introduce major challenges to EDA tools and ICs mainly because they involve physical phenomena that require complex simulation and analysis for various combinations of temperature, voltage, and process corners. The semiconductor industry has invested major efforts to cope with reliability issues, which mainly consisted of attempting to enhance physical design implementation flows. However, the effectiveness of such flows is quite limited, primarily due to their small scale and the substantial design efforts they involve. This section starts by reviewing asymmetric aging and then

978-1-6654-2170-6/21/$31.00 ©2021 IEEE

discusses prior works.

A. Asymmetric Transistor Aging

Transistor aging is the process whereby silicon transistors develop faults in their circuitry over time [6]. This degradation is induced by the two physical mechanisms HCI and BTI. In both cases, charge carriers from the transistor channel are trapped at the dielectric insulator of the transistor gate. When HCI is involved, excessive energy causes charge carriers from the current between the transistor source and drain to be trapped at the gate oxide. For BTI, the charge carriers are trapped whenever a constant voltage is applied to the transistor gate, but no current flow is involved in this case. Damage to the transistor as a result of BTI is partially recovered a few seconds after the gate voltage is removed. Both HCI and BTI lead to an increase in transistor threshold voltage, which degrades the transistor speed and mandates a higher voltage to switch on the transistor. This imposes extra timing margins in the clock-cycle time to allow ICs to continue operating reliably throughout their lifetime. Various approaches have been proposed to handle transistor aging, including physical-design- and circuit-based solutions [7].

When the aging degradation of transistors is not uniformly distributed, the problem becomes even more complicated. This may happen when p- and n-type devices age unequally and, as a result, the rising and falling transients incur asymmetric shifts in delay. The problem can also occur when the timing degradation between different logical paths becomes asymmetric [8]. This may result in severe setup and hold-timing violations, which cannot be identified by conventional timing verification tools. Hold violations are more severe than setup violations because they cannot be mitigated by reducing the clock frequency or by increasing the timing margins of the clock-cycle time. This phenomenon, referred to as "asymmetric aging," is today a major reliability concern in ICs.

Asymmetric aging is induced by long periods of unequal static stress applied to logic elements in different logical paths, which may vary between tens of seconds up to several weeks [9]. BTI, which is the main contributor to asymmetric aging, can degrade the speed of both p-type (NBTI) and n-type (PBTI) devices. The impact of NBTI is much more significant relative to PBTI, although, in advance process nodes, the impact of PBTI has become considerable.

Asymmetric aging introduces major challenges to IC designers due to the extremely complex modeling and analysis required to eliminate its impact in large-scale circuits. In particular, timing-verification tools that model BTI effects are nontrivial because they depend not only on the operating conditions and technology specifications but also on the functional modes of operation through the IC lifetime [10] (e.g., use of constant values, switching into standby modes, and activation of clock gates for long periods). From an architectural point of view, asymmetric aging in many cases is a result of dynamic power-saving techniques that impose static states on logical circuits, which leads to the BTI effect.

B. Prior studies

Many of the prior works approached asymmetric aging from the physical design point of view. Such an approach is not straightforward, since the process of simulating, analyzing, and fixing asymmetric aging issues in large-scale circuits is highly complex [11], with only a few EDA tools (e.g., BERT, RelXpert) having such a (limited) capability [12, 13]. The suggested strategies attempted to cope with the problem from various directions. One approach was to enhance the process node to reduce the impact of the BTI effect. This, however, became highly challenging due to the down-scaling dimension of the gate oxide. Traditional approaches relied on taking margins in timing closures for both setup and hold that would take into account the asymmetric aging effect. This was found to necessitate a highly complex analysis and, in many cases, ended up in overdesign. Other studies attempted to model and predict the degradation as a result of NBTI [3, 8–10] and suggested various solutions such as transistor sizing, V_{DD} tuning, duty cycle reduction, and decreasing the transistor channel length. Agrawal et al. [14] presented a mechanism for circuit failure prediction by collecting data from special sensors placed in different locations in the silicon die. Their results indicate that by using these sensors, they can reduce the conservative margins used by the traditional design flows and improve chip performance. Further studies [15] also introduced methods for analyzing digital circuits and identifying critical gates that are the most susceptible to NBTI stress. This was done by employing an aging model and an aging-aware timing analysis.

While the approach adopted by many studies to cope with asymmetric aging relied on enhancements in physical design, only a limited number of works have suggested architectural solutions. Firouzi et al. suggested a NOP instruction insertion to reduce the impact of NBTI on MIPS processors [16], which was found to provide limited improvement [17]. Abbas et al. suggested running anti-aging programs when the processor is not utilized [17]. This technique was efficient; however, it required a complex analysis of the critical paths and the requisite anti-aging values. Chen et al. examined the performance degradation due to asymmetric aging in multicore systems [18]. A technique to reduce the impact of asymmetric aging on FPGA was introduced by [19], who suggested bundling unused FPGA elements in logical chains and toggling them at low rates to prevent the constant NBTI stress. Other studies proposed solutions for asymmetric aging in memory systems [4].

III. Design-Flow and Tool for Asymmetric Aging Avoidance

In this section, we present a new design flow and a tool to cope with the asymmetric aging problem. The proposed flow and tool are based on a prior study [5] where it has been observed that data path circuits may be highly susceptible to asymmetric aging under different workloads. Past experimental analysis, which has examined various benchmarks and applications, indicates that microprocessors execution units, such as integer ALU, FP adder, multiplier, etc., may incur very long static BTI stress. For example, when integer workloads are used, the utilization of FP execution units is extremely low, resulting in excessive BTI stress. Reference [5] introduced a novel scheme to mitigate BTI stress over FP adder. The proposed scheme uses a pseudorandom sequence bit (PRBS) generator [20] that generates pseudorandom patterns into the data path of the FP adder unit to prevent extended periods of constant stress.

In this study, we generalize the prior work and extend it to automatically handle any data path logical structure. We introduce a full design flow and an automated EDA tool to minimize the asymmetric aging effect in data path logical structures. We first start by presenting our proposed design flow and then describe the implementation and configuration details of our automated tool.

A. Asymmetric Aging Avoidance Design Flow

The proposed design flow is depicted in Figure 1 consists of two iterative phases. In the first phase, the analysis is done at the RTL level, and the second phase is performed on the synthesized design at the gate-level. In the RTL phase, our tool runs on the original design, and the run setting is specified through a special configuration file, which will be described later. The tool automatically generates two modules: a testbench module and a synthesizable module, which are presented in Figure 2. In this figure, all the logical components that are automatically added by the tool are shown by the light and dark gray colors. For the synthesizable design module, the tool generates a top-level wrapper that instantiates the original data path, a pseudorandom bit sequence (PRBS) generator, and a multiplexor. The PRBS generator, which is activated by a slow frequency clock, generates pseudorandom patterns that are fed into the data path module through a multiplexor to prevent extended periods of constant stress. The added multiplexor (illustrated in Figure 2) arbitrates between the functional inputs and the PRBS outputs. The slow clock frequency can be in the order of MHz or even lower to minimize dynamic power overhead. Varieties of PRBSs generators are used for communication and security applications. We examine a simple PRBS circuit [20] as part of our study, which introduces very small logic and power overhead while being able to generate random patterns that are sufficient to effectively toggle the design at a low rate.

The testbench module is used to quantitively measure through simulations the effectiveness of the asymmetric aging avoidance circuitry, which is integrated as part of the synthesizable module. The testbench module instantiates the synthesizable module with the clock generators and a set of signal probability counters. The tool automatically maps all the nets in all hierarchies of the original data path design and associates each net with an individual counter. Through the RTL simulation, the counters measure the number of clock cycles in which the corresponding net is in the logical state of 1. The signal probability 1, denoted as SP(1), of every net in the synthesizable design is calculated by dividing the corresponding counter value by the number of simulated clock cycles.

In the next step of our design flow, the tool invokes an RTL simulation of the testbench. When the simulation is completed, it dumps the signal-probability counter values and the corresponding net names into a temporary file. The file is post-processed by the tool, and a report is generated. The report summarizes the SP(1) of every net in the design and will be further described as part of the tool implementation detail. The report, which is presented for the designer review, highlights all the nets with excessive BTI stress that could not be mitigated by the asymmetric aging avoidance circuitry. If some of these cases can be fixed by the designer at the RTL level, the process will repeat itself till there are not manual fixes left. An example of such a manual fix is illustrated in Figure 3.

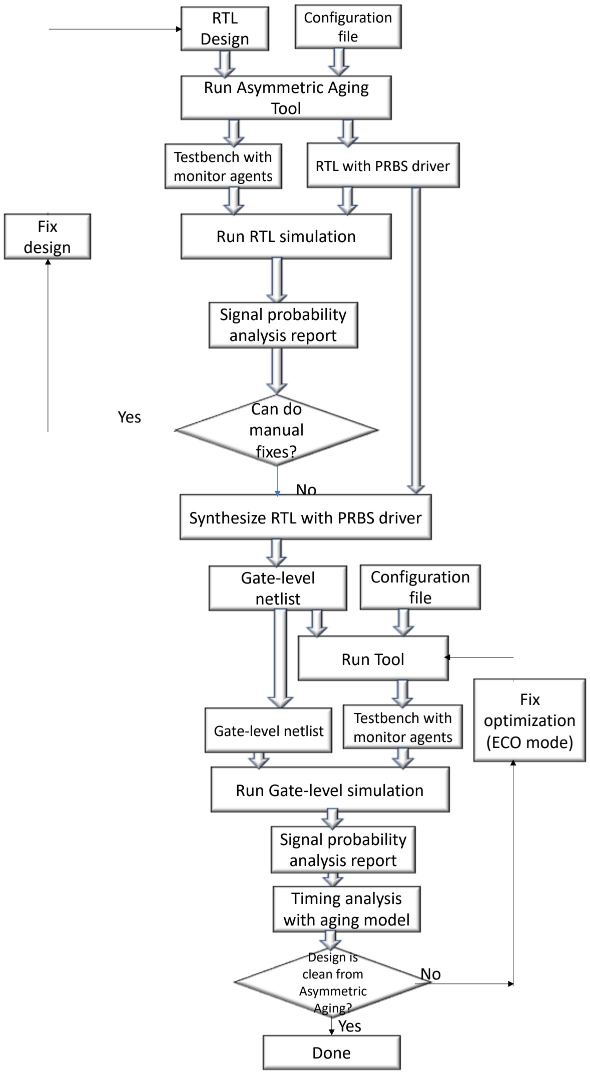
RTL Design
Configuration file
Run Asymmetric Aging Tool
Testbench with monitor agents
RTL with PRBS driver
Fix design
Run RTL simulation
Signal probability analysis report
Yes
Can do manual fixes?
No
Synthesize RTL with PRBS driver
Gate-level netlist
Configuration file
Run Tool
Gate-level netlist
Testbench with monitor agents
Fix optimization (ECO mode)
Run Gate-level simulation
Signal probability analysis report
Timing analysis with aging model
Design is clean from Asymmetric Aging?
No
Yes
Done

Figure 1 – Asymmetric aging avoidance design flow

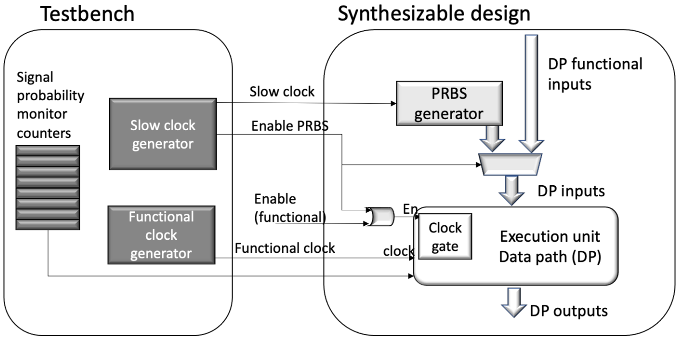

Figure 2 – Automatically generated design with asymmetric aging avoidance circuitry

In this example, one of the multiplexor inputs is connected to logical 0, inducing excessive BTI stress no matter what logical values are applied to the other multiplexor inputs. As illustrated by Figure 3, this stress can be simply eliminated by replacing the constant 0 with a NOT gate which is connected to the multiplexor selector and input. The manually fixed circuit is capable of eliminating the static stress assuming that the multiplexer select signal has SP(1)=0.5.

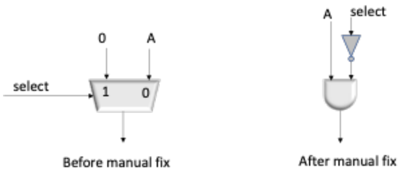

Figure 3 – Manual fix at RTL level

The second phase of our design flow runs at the gate-level. The synthesizable module only, which includes the original data path, and the asymmetric aging avoidance circuitry is synthesized using standard synthesis tools. The gate-level netlist is presented to the tool as an input. The tool automatically generates a testbench module, as illustrated in Figure 2, similar to the RTL run. In this phase, the tool automatically analyzes the gate-level netlist and uniquely associates every net to an individual signal probability counter. In the next step, the tool invokes a gate-level simulation of the testbench and the gate-level netlist. Similar to the RTL phase, once the simulation completes, all counter values are dumped into a temporary file and a signal probability report is automatically created and presented to the designer. Since the pattern generated by the PRBS circuit may not be able to propagate to all the nets in the design, a step of incremental fixes might be needed. The incremental fixes and optimizations can be manually made by the designer similar to the RTL phase.

B. Asymmetric Aging Avoidance Tool Architecture

The block diagram of our asymmetric aging avoidance tool is illustrated in Figure 4. The tool has been designed in Tcl language and includes the following software modules: configuration file parser, HDL generation engine, signal probability analyzer, and report generator. The configuration file parser reads the configuration file supplied by the designer. TABLE *1* summarizes the main configuration parameters of the tool.

TABLE 1 – CONFIGURATION PARAMETERS

Parameter	Details
`prbs_type`	Specifies PRBS type: PRBS9, PRBS31 or none.
`Design_type`	Species data path design type: RTL or gate-level
`inv_synt`	If TRUE, the tool will launch the synthesis tool
`report`	`histogram` – generates SP(1) histogram.
	`Netlist` – generate SP(1) textual report.

The parameters parsed from the configuration are provided to all other internal modules of the tool. The HDL generation engine creates a new top-level wrapper for the synthesizable design and a testbench module which are depicted in Figure 2. The HDL generation engine also invokes the simulation and, once the simulation is completed, a dump file of the SP(1) information is created. The dump file is analyzed, and the information is provided to the report generator. The report generator can generate any combination of the reports provided by the report command specified in the configuration file (as presented in TABLE *1*).

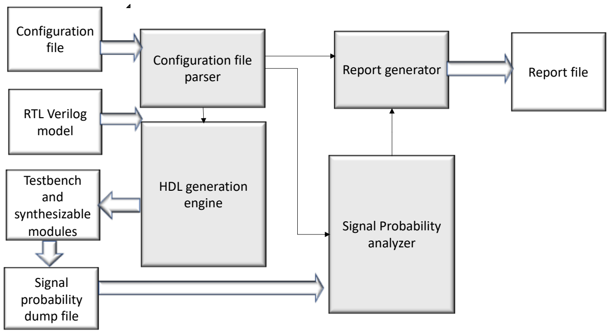
Configuration file
Configuration file parser
Report generator
Report file
RTL Verilog model
HDL generation engine
Testbench and synthesizable modules
Signal Probability analyzer
Signal probability dump file

Figure 4 – Asymmetric Aging Avoidance Tool Block Diagram

IV. Experimental Results

In this section, we present the experimental results of our design flow and tool to mitigate the asymmetric aging effect. Our experimental analysis consists of three parts: first, we present the signal probability improvement that is gained by our tool. Second, we summarize the impact on power and area as a result of our design flow, and last, we present timing analysis with aging models, which demonstrates the reliability improvement gained by our design flow.

TABLE 2 – PROCESS TECHNOLOGY PARAMETERS

Core Model	
Process	28nm
PDK	Synopsys SAED 28nm
Core V_{DD}	1.05 V
Clock frequency	420 MHz
Synthesis tool	Synopsys Design Compiler Q-2019.12-SP1
Process corner	SS_1p05_125C (Slow-Slow corner)

As part of our experimental analysis, we examine the effectiveness of the full flow presented in Figure 1 on two data path units: integer ALU and FP divider, which have been obtained from OpenCores[1] repository. Our analysis is performed on a 28nm process technology using the Synopsys® Digital Cell Libraries (SAED_EDK28_CORE) [21]. The process technology and synthesis parameters are summarized in TABLE *2*. For every examined block, we run our tool in RTL-level and gate-level. The PRBS generator that we use is PRBS9. We examined different PRBS generators, and we have found that this type of PRBS has optimal performance with minimal overhead. The runtime of the simulation is 1 million clock cycles. The next subsections summarize our experimental results.

A. Signal probability experimental analysis

In the first part of the experimental analysis, we examine the SP(1) distribution on our data path modules. Initially, we run the tool at the RTL level and generate histogram reports of the signal probability for every module. In the next step, we

[1] www.opencores.org

synthesize every module with the asymmetric aging avoidance circuitry. Finally, we run the tool on the gate-level netlist and generate histogram reports for the SP(1) distribution. The histograms generated by the tool are presented in Figure 5.

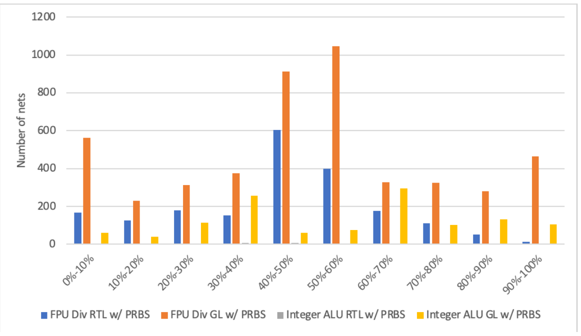

Figure 5 – SP(1) histograms for integer ALU and FP divider

As previously indicated, designs that allow static BTI stress for long periods may incur serious reliability issues due to asymmetric aging. For example, microprocessors that run integer applications do not use FP execution units. In some cases, even the available integer ALUs cannot be fully utilized. Prior studies [5] show that, in such cases, approximately 50% of the nets are static logical 1 while the rest are static logical 0. Clearly, Figure 5 shows that the proposed tool and circuitry can remove the continuous BTI stress from the majority of nets. The ALU and FP divider have SP(1) in the range of 30%–70% and 40%–60%, respectively, for the majority of their nets. The histograms show that a smaller portion of nets cannot be toggled effectively and remain static through most of the simulations. We have identified this behavior as due to (1) constant values in the design (either logical 0 or 1) and (2) big logical shifters in the integer ALU and in the FPU divider that involve many nets with constant zero-padding.

This can be fixed easily by forcing the PRBS patterns to be injected into this group of signals so that, as a result, the constant state is avoided. Despite the fact that such a fix may be quite simple, it may not necessarily be needed since such nets typically are not on the critical path of the asymmetrically aged design. In Subsection 4.C, we present the impact of the SP(1) distribution on reliability and violated timing paths as a result of asymmetric aging.

B. *Power and area overhead analysis*

As part of our experimental analysis, we have integrated all data path modules and wrapped them together in a top-level module, which represents an execution unit that combines multiple processing elements. We hierarchically synthesized the combined execution unit and summarize in TABLE *3* the overall area and power of every module using the process technology parameters, which are presented in TABLE *2*. We run our asymmetric aging flow on the combined gate-level netlist. The asymmetric aging avoidance circuitry area and power overhead are also presented in TABLE *3*. As it can be observed by this table, power and area overhead are very small (1.8% and 2.7%, respectively).

TABLE 3 – AREA AND POWER OVERHEAD

Module	Area [um^2]	Power [mW]
FP Divider	22,415	3.40
Integer ALU	49,563	4.36
Total area w/o asymmetric aging avoidance circuitry	71,798	7.76
Asymmetric aging avoidance circuitry area overhead	1328 (1.8%)	0.21 (2.7%)

C. *Timing analysis based on aging models*

In the last part of our experimental analysis, we perform a timing analysis based on aging models to examine the improvement in reliability gained by the proposed design flow. The timing analysis method that we use is more practical for large-circuit analysis. This approach is similar to that introduced in Ref. [9] and relies on aging-aware libraries. In this method, the conventional timing models are extended with BTI-aging models where the rising cell delays are derated by their corresponding NBTI degradation factors (as a function of the SP), whereas the falling delays remain unchanged. The derate factors for the aging libraries were generated by using SPICE simulations with the nominal V_{th} values replaced by aged V_{th} values that correspond to the lifetime and SP. The V_{th} degradation model that we rely on is based on the reaction-diffusion model, which is the most widely accepted model for BTI aging in both the industry and research communities [22]. The reaction-diffusion model generates the following relation for V_{th} degradation, ΔV_{th}, as a result of static NBTI stress:

$$\Delta V_{th} \propto K_s e^{-\frac{E_a}{kT}}(t-t_0)^{\frac{1}{6}} \qquad (1)$$

where K_s is a technology-dependent constant, E_a is the activation energy of Si, T is the operating temperature, k is the Boltzmann constant, t_0 is the time when the NBTI stress starts, and t is the overall time.

The worst delay shift predicted by our aging model is approximately 6% over a ten-year lifetime. This result is similar to those of previous studies and the industry observations, which found that BTI degradations may even reach a 10% delay shift under stress conditions [9, 23]. Figure 6 illustrates the frequency degradation predicted by the BTI aging models over a ten-year lifetime. It can be observed that, for small values of SP(1) (gates are more likely to be under constant logical 0), the frequency degradation is nearly 6%, whereas, for a higher SP(1), the frequency degradation drops to 2%–3%. The absolute delay shift of gates under variable BTI stress relative to gates that are symmetrically aged [with an SP(1) of 0.5] is also illustrated in this figure. Gates with constant stress (when SP(1) is 0 or 1) exhibit a 2%–2.5% asymmetric delay shift relative to gates that avoid the constant stress (SP(1) of 0.5–0.6). In addition, gates with SP(1) in the range of 30%–70% experience significantly smaller delay shift of approximately 1% or lower. It should be noted that the observed asymmetric delay shift, even one as small as 2%–3%, may critically impact the circuit reliability by introducing unbalanced clock tree, setup, and hold timing violations. This implication is further supported by the timing analysis comparison of fresh, aged, and asymmetric-aging-aware designs (using our proposed design flow and tool) that is summarized in TABLE *4*. The timing analysis, which was done using aged and fresh library models, shows that the examined modules incur setup violations when asymmetric aging is considered. The FP divider also experiences a significant number of hold violations. It should be noted that even if hold constraints are still met in the integer ALU, its positive slack becomes smaller, and thereby the design becomes marginal. In addition, it can be observed that the asymmetric aging-aware design using our flow is clean of any violated path due to the symmetrical aging degradation. Therefore, the minimization of the asymmetric delay shift, which our design flow accomplishes, is essential for circuit reliability.

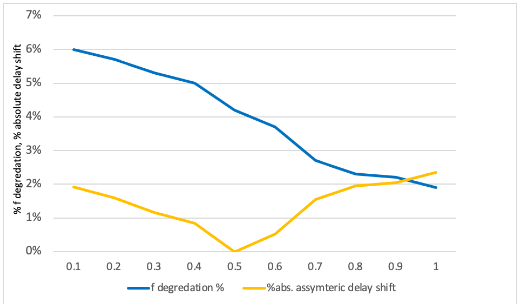

Figure 6 – Frequency degradation and absolute asymmetric delay shift over a ten-year lifetime.

TABLE 4 – NUMBER OF SETUP AND HOLD VIOLATED ENDPOINT PATHS FOR FRESH, AGED AND ASYMMETRIC-AWARE DESGINS

	Max Delay (Setup)			Min Delay (Hold)		
	Fresh	Aged	Asym. Aging-Aware Tool	Fresh	Aged	Asym. Aging-Aware Tool
FP Div.	0 / 992	4 / 992	0 / 992	0 / 992	116 / 992	0 / 992
Int. ALU	0 / 134	2 / 134	0 / 134	0 / 134	0 / 134	0 / 134

V. CONCLUSIONS AND DISCUSSION

IC reliability is a crucial requirement that has been highly challenged by advanced process technologies and new computation-intensive applications. Recent advanced process nodes have become highly susceptible to asymmetric aging that can cause critical timing violations in ICs and overall system failure. In this paper, we introduce a novel tool and design flow to mitigate the asymmetric aging effect. Further studies on EDA flows to cope with asymmetric aging are highly encouraged since the effect of BTI stress is further intensified in the most advanced process node. This is a major challenge for both industry and research communities to find practical solutions to allow the development of future reliable ICs. A possible future direction of research is to further extend the proposed tool and design flow to automatically eliminate any need for manual fixes. This can be done by extending the design flow of the tool to detect such violations and mitigating them by adopting a similar approach of pseudorandom data injection.

REFERENCES

[1] Failure Mechanism Based Stress Test Qualification for Integrated Circuit. Automotive Electronics Council, Component Technical Committee – AEC – Q100 – REV-G standard.

[2] M. A. Alam, H. Kufluoglu, D. Varghese, and S. Mahapatra, “A comprehensive model for PMOS NBTI degradation,” *Microelectron. Rel.*, vol. 47, no. 6, pp. 853–862, Jun. 2007.

[3] S. Bharadwaj, W. Wang, R. Vattikonda, Y. Cao, and S. Vrudhula, “Predictive modeling of the NBTI effect for reliable design,” in *Proc. Custom Integrated Circuits Conf.,* Sep. 2006, pp. 189–192.

[4] A. Calimera, M. Loghi, E. Macii, and M. Poncino, “Aging effects of leakage optimizations for caches,” in Proc. IEEE Great Lakes Symp. VLSI, May 2010, pp. 95–98.

[5] F. Gabbay, A. Mendelson. Asymmetric Aging Effect on Modern Microprocessors. Microelectronics Reliability. Vol. 119, April 2021, 114090.

[6] A. Campos-Cruz, G. Espinosa-Flores-Verdad, A. Torres-Jacome, and E. Tlelo-Cuautle, On the prediction of the threshold voltage degradation in CMOS technology due to bias-temperature instability. Electronics, vol. 7, no. 12, pp. 427 2018, 7, 427.

[7] M. Taddiken, N. Hellwege, Nils Heidmann, Dagmar Peters-Drolshagen and Steffen Paul. Analysis of aging effects - From transistor to system level. Microelectronics Reliability Volume 67, Dec. 2016, pp. 64-73

[8] A. T. Krishnan et al., “Product drift from NBTI: Guardbanding, circuit and statistical effects,” 2010 International Electron Devices Meeting, San Francisco, CA, 2010, pp. 4.3.1-4.3.4.

[9] J. B. Velamala, K. B. Sutaria, V. S. Ravi, and Y. Cao. Failure analysis of asymmetric aging under NBTI,” IEEE Trans. Device Mater. Rel., vol. 13, no. 2, pp. 340–349, June 2013.

[10] R. Zheng, J. Velamala, V. Reddy, V. Balakrishnan, E. Mintarno, S. Mitra, S. Krishnan, and Y. Cao, “Circuit aging prediction for low-power operation,” in Proc. Custom Integr. Circuits Conf., Sep. 2009, pp. 427–430.

[11] S. Ogawa and N. Shiono, “Generalized diffusion-reaction model for the low-field charge build up instability at the Si-SiO_2 interface”, Physical Review, 51(7):4218–4230, Feb. 1995.

[12] K. L. Chen, S. A. Saller, I. A. Groves, C. C. Li, E. Minami, and D. B. Scott, “Reliability effects on MOS transistors due to hot-carrier injection,” IEEE Trans. Electron Devices, vol. 32, no. 2, pp. 386–393, Feb. 1985.

[13] Reliability Simulation in Integrated Circuit Design. [Online]. Available: http://www.cadence.com/

[14] M. Agarwal, B. C. Paul, Ming Zhang, and S. Mitra, “Circuit failure prediction and its application to transistor aging”, VLSI Test Symposium, pages 277–286, May 2007.

[15] W. Wang, Z. Wei, S. Yang, and Y. Cao, “An efficient method to identify critical gates under circuit aging,” in Proc. Int. Conf. Comp. Aided Des., Nov. 2007, pp. 735–740.

[16] F. Firouzi, S. Kiamehr and M. B. Tahoori, “NBTI mitigation by optimized NOP assignment and insertion,” 2012 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, 2012, pp. 218-223.

[17] Haider Muhi Abbas, Mark Zwolinski, and Basel Halak. Aging Mitigation Techniques for Microprocessors Using Anti-aging Software. Chapter 3, Ageing of Integrated Circuits - Causes, Effects and Mitigation Techniques, Springer, Cham. ISBN 978-3-030-23781-3

[18] Y. Chen, I. Lin and J. Ke, “ROAD: Improving Reliability of Multi-core System via Asymmetric Aging,” 2019 IEEE/ACM International Conf. on Computer-Aided Design (ICCAD), CO, USA, 2019, pp. 1-8.

[19] Techniques for Reducing Uneven Aging in Integrated Circuits. Intel European Patent Application. Application number 18190097.8. 2018.

[20] L. Pavlovic, M. Vidmar and S. Tomazic, “2.5 Gbit/s PRBS Generator and Checker,” Proceedings ELMAR 2006, Zadar, 2006, pp. 363-366, doi: 10.1109/ELMAR.2006.329585.

[21] Synopsys Digital Standard Cell Lib. SAED_EDK32/28_Core Databook.

[22] A. Calimera, M. Loghi, E. Macii and M. Poncino, “Dynamic Indexing: Leakage-Aging Co-Optimization for Caches,” in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 33, no. 2, pp. 251-264, Feb. 2014

[23] Bogdan Tudor, Joddy Wang, Zhaoping Chen, Robin Tan, Weidong Liu, Frank Lee, An accurate MOSFET aging model for 28nm integrated circuit simulation, Microelectronics Reliability, Vol. 52, Issue 8, 2012, pp. 1565-1570