[[1]](#footnote-1)

A Design Flow and Tool for Avoiding Asymmetric Aging

Freddy Gabbay1, Avi Mendelson2, Basel Salameh2, and Majd Ganaiem2

|  |  |
| --- | --- |
| Ruppin Academic Center1freddyg@ruppin.ac.il | Technion – Israel Institute of Technology2{avi.mendelson, basel238, majd.ga}@technion.ac.il |

*Abstract* — Reliability is a crucial requirement in any modern integrated circuit (IC) because it assures correct execution over the IC lifetime. As mission-critical components are becoming common in commodity systems, the demand for reliable processing continues to grow. The latest process technologies have aggravated the situation by further extending IC vulnerability to reliability issues. Lately, asymmetric aging has emerged as a major contributor to reliability concerns. With asymmetric aging, logical elements suffer from unequal timing degradation over their lifetime and, consequently, may cause system failure. Few tools are available to handle asymmetric aging, and such tools as exist mainly rely on circuit- or physical-design approaches and are hard-pressed to handle large-scale ICs. We thus introduce herein a design flow and a tool to minimize asymmetric aging in data-path design structures. The proposed tool is easy to integrate into standard design flows of large-scale ICs. In addition, it can automatically analyze various designs at resistor-transistor-logic or gate level and identify logical elements that are suspectable to asymmetric aging. As part of the design flow, the tool automatically embeds a special logical circuitry into the design to counter asymmetric aging. The experimental analysis and simulations show that the proposed design flow minimizes asymmetric aging and eliminates major reliability concerns while introducing minor power and silicon-area overhead.

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

# INTRODUCTION

Reliability is a crucial requirement for modern integrated circuits (ICs) to assure correct execution over their lifetime. New mission-critical computation-intensive applications (e.g., autonomous vehicles, data centers, cloud computing, life-support systems) impose strict requirements on IC reliability through the lifetime and operating conditions. For example, the automotive industry expects an IC to function reliably for 10–15 years at a given temperature (usually about 125 °C [1,2]) and under various workloads. None of these reliability-sensitive applications can afford IC faults caused by reliability issues. As the requirements for reliability have become substantially more stringent, advanced FinFET VLSI technologies (28 nm and lower) have become highly suspectable to reliability issues, and in particular to transistor aging. Transistor aging is the deterioration of transistors due to charge carriers trapped at the dielectric insulator of a transistor gate. This phenomenon is induced by hot-carrier injection (HCI) and bias-temperature instability (BTI), which are further discussed in Section 2. The common approach to handle such degradation in digital circuits is to provide extra timing margin to the clock cycle time and thereby take the timing degradation into account (however, as a result, ICs incur performance degradation). This solution is adequate provided that aging is symmetric (i.e., all transistors age at the same rate).

Unfortunately, many digital circuits experience asymmetric transistor aging. As a result, different paths in the circuit may incur different degradations that may result in critical timing violations. Asymmetric aging is mainly induced by the BTI effect [3–5], which results from applying constant voltage to transistor gates for long periods. The time required for the transistor to incur such degradation may vary from tens of seconds up to several weeks [4]. As a result, even if the asymmetric delay shift is relatively small, critical timing constraints in the logical circuit may be compromised. This observation renders logical elements that are maintained under constant logical values for long duration susceptible to severe reliability concerns. Past studies indicate that, from the architectural point of view, asymmetric aging is commonly induced by dynamic-power-saving techniques (e.g., clock gating), which enforce static states on logical circuits and, as a result, cause BTI stress [6].

 Many prior studies (described in Section 2.3) have approached asymmetric aging from the physical-design point of view. Such approaches have proven quite complicated because they involve highly complex simulations and analysis methods. Even many of the common electronic design automation (EDA) tools today lack such a capability and are extremely limited for analyzing very-large-scale circuits, which could employ billions of transistors.

 This work generalizes the method we previously presented [7] and introduces an extended design flow and a tool for avoiding asymmetric aging. The proposed tool automatically analyzes data path design structures at the RTL or gate level and identifies logical elements that are suspectable to asymmetric aging induced by static BTI stress. The proposed tool uses a special testbench that injects random data patterns into the circuitry being tested while automatically monitoring the logical elements’ signal probability and toggle rate. The tool automatically generates special logical circuitry, which is embedded in the design, to avoid asymmetric aging while the device is in mission mode. The proposed design flow is straightforward to integrate as part of the standard design flow of large-scale ICs.

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

# Asymmetric Aging

The susceptibility of modern process technologies to reliability-related issues has grown dramatically over the last decade. Design efforts dedicated to reliability have substantially increased for 28 nm process technology and below (16, 7, 5, and 3 nm). The design community has focused on enhancing physical design flows to minimize and eliminate reliability-related issues. Such flows involve substantial design efforts and, in many cases, require multiple iterations to make the IC comply with the design rules (also known as the “sign-off process”). In the remainder of this section, we first describe the asymmetric aging effect, following which we provide an overview of previous studies.

## Asymmetric Transistor Aging

Transistor aging is the deterioration of transistors in logical elements [8, 9] and is caused by charge carriers from the transistor inversion channel being trapped at the dielectric insulator of a transistor gate. There are two physical mechanisms that trap charge carriers: (1) HCI, which involves charge carriers that flow from transistor source to drain. With this mechanism, charge carriers occupying abnormally high energy levels can get trapped in the gate oxide. (2) BTI, whereby charge carriers also become trapped in the dielectric gate insulator, but this time no current flow is required between the source and drain of the transistor; instead, charges can become trapped whenever a voltage is applied to the transistor gate. If the gate voltage is applied for a short period of time (<10s), the damage is partially reversible (i.e., some of the charge carriers are detached).

Both BTI and HCI increase the transistor threshold voltage, reduce charge-carrier mobility in the channel, and mandate a higher voltage to switch on the transistor. In addition, the reduced transistor current reduces the transistor speed. As a result, ICs may experience major frequency degradation over their lifetime. Proposed methods for dealing with aging effects include physical-design or circuit-based solutions [9–11], with the most common approach being to provide an extra margin in the clock-cycle time to compensate for the expected performance degradation over the lifetime of the device.

 Recent studies have discovered that degradation due to aging may not be uniformly distributed. This may happen inside a logical cell when p-devices and n-devices age unequally and, as a result, rising and falling transient edges may experience different shifts in timing delays. In addition, the same phenomenon can also occur between different paths in a logical circuit [7, 12] that age unevenly, possibly leading to violations of critical timing constraints. When such violations involve setup-timing constraints, they can be mitigated by reducing the clock frequency; however, when hold constraints are violated, the circuit will become plagued by severe reliability issues that cannot be mitigated. This phenomenon is referred to as “asymmetric aging” and has become a major reliability concern in mission-critical systems.

Asymmetric aging is induced by unequal static stress applied to logic elements for long periods, which may vary from tens of seconds up to several weeks [7]. BTI is the main contributor to this phenomenon, which may affect both p-type (NBTI) and n-type (PBTI) transistors. The impact of NBTI exceeds that of PBTI by several orders of magnitudes, although in advanced process technologies, PBTI also has considerable impact. BTI causes logical paths that are under different static stress to age asymmetrically and introduces new timing violations that cannot be identified by conventional timing-verification methods.

Asymmetric aging is difficult to model, analyze, predict, and avoid in very large-scale ICs, making it a major reliability issue. In addition, timing analyses that consider asymmetric aging are non-trivial because they depend not only on the mode of operation (static versus dynamic stress) but also on the operating conditions and technology specifications. Conventional timing verification tools [13] lack information related to the lifetime activation modes of the digital circuit (e.g., standby modes, constant values, and activation of clock gates) that are applied for long periods.

 From an architectural point of view, asymmetric aging in many cases results from dynamic power-saving techniques that enforce a static state on logical circuits, leading to BTI. This is demonstrated in Figure 1 on two typical scenarios that are common to various logic circuits. Figure 1(a) depicts a logical shifter implemented by a multiplexor where the input data are left-shifted in accordance with the multiplexor control. The static logical 0 padding in the multiplexer input introduces static BTI stress, which can lead to asymmetric aging in the multiplexor. Figure 1(b) shows another example in which an idle data path of an execution unit (e.g., a floating-point adder or multiplier) is under static stress. The inputs to the block are stored by clock-gated registers and, when the unit is not in use, the clock is disabled. This induces a static logical state on all gates in the execution unit.



Figure - Asymmetric aging in (a) logical left-shifter and (b) execution unit with gated clock.

## Prior studies

Many prior works approached asymmetric aging from the physical design point of view, which is not a straightforward approach because the process of simulating, analyzing, and fixing asymmetric aging issues in large-scale circuits is highly complex [15], with only a few EDA tools (e.g., BERT, RelXpert) currently offering a limited capability in this area [16, 17]. The strategies proposed to cope with the problem have attacked it from various directions. One approach is to enhance the process node to reduce the impact of the BTI. However, this becomes extremely challenging because the dimensions of the gate oxide must be downscaled. Traditional approaches rely on allowing margins in timing closures for both setup and hold that are designed to take into account asymmetric aging. This necessitates a highly complex analysis and, in many cases, terminates in overdesign. Other studies model and predict the degradation as a result of NBTI [3–5, 7, 12, 13, 18–20] and suggest various solutions such as transistor sizing, *V*DD tuning, duty cycle reduction, and decreasing the length of the transistor channel. Agrawal et al. [21] presented a mechanism to predict circuit failure that involves collecting data from special sensors placed in different locations in the silicon die. Their results indicate that the use of these sensors leads to the reduction of the conservative margins used by traditional design flows and thereby improves chip performance. Further studies [22–25] 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 (consisting of BTI-aware libraries) and an aging-aware timing analysis.

While many studies attempted to cope with asymmetric aging from the physical design point of view, only a limited number of works have examined this phenomenon from an architectural point of view. Firouzi et al. suggested inserting a NOP instruction to reduce the impact of NBTI on the execution stage of MIPS processors [26]. Unfortunately, this method provides limited improvement [27].

Abbas et al. suggested running anti-aging programs instead of idle tasks when the processor is not in use [27]. Anti-aging programs generate specific values to counteract BTI asymmetric aging in the execution unit combinatorial circuits. Although this technique has proven efficient, it requires complex analysis of the critical paths and the requisite anti-aging values. Moreover, it is limited to execution-stage combinatorial circuits and assumes a scalar processor. With out-of-order processors, such techniques may have difficulty mapping the anti-aging patterns to the multiple execution units.

Chen et al. examined performance degradation due to asymmetric aging in multicore systems where processors asymmetrically age because of different workloads and uses [34]. They suggested reserving certain cores at the early stages of the system lifetime to be used for executing critical missions at the later stages.

Field programable gate array (FPGA) devices may also be highly susceptible to NBTI. Unused FPGA logic can suffer from constant, long-term logical stress and, when the same logic is used again, it may cause asymmetric aging. A technique to reduce the impact of asymmetric aging on FPGAs was introduced in Ref. [29], which suggests bundling unused FPGA elements in logical chains and toggling them at low rates to prevent constant NBTI stress. Other studies proposed remedies for asymmetric aging in processor memory systems [7, 30–33]. Various techniques aim to mitigate degradation in SRAMs due to asymmetric aging by balancing the signal probability of 0 and 1 states.

# A Design Flow and Tool for Avoiding Asymmetric Aging

This section presents a new design flow and tool to cope with the problem of asymmetric aging. The proposed flow and tool are based on our prior study [7], which showed that data-path circuits might be highly susceptible to asymmetric aging when exposed to different workloads. The experimental analysis, which examines various benchmarks and applications, indicates that microprocessor execution units, such as integer ALU, FP adder, and multiplier, may incur long-term static BTI stress. For example, when integer benchmarks are used, FP execution units see extremely little use, resulting in excessive BTI stress. In previous work [7], we proposed a scheme to mitigate BTI stress over FP adders that uses a pseudorandom sequence bit (PRBS) generator [35] that inputs pseudorandom patterns into the data path of the FP adder unit to prevent extended periods of constant stress.

The present study generalizes our previous work and extends it to automatically handle any data path logical structure. We introduce a full design flow and a tool to minimize asymmetric aging in data-path logical structures. We start by presenting the proposed design flow, following which we describe the implementation and configuration of the automated tool.

## Design Flow for Avoiding Asymmetric Aging

 Figure 2 shows the proposed design flow, which consists of two iterative phases. In the first phase, the analysis is done at the RTL level and, once completed, the second phase involves the synthesized design at the gate level. At the RTL level, the proposed tool runs on the original design, and the run setting is specified through a special configuration file, described below. The tool automatically generates two modules: a testbench module and a synthesizable module. As presented in Figure 3, all logical components automatically added by the tool appear in light and dark gray. For the synthesizable design module, the tool generates a top-level wrapper that instantiates the original data path, a 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 multiplexor, connected to the data path inputs, is controlled by the slow clock and arbitrates between the functional inputs and the PRBS outputs. To minimize dynamic power overhead, the slow clock frequency can be on the order of megahertz or even less. Different varieties of PRBS generators are used for communication and security applications; we examine herein a simple PRBS circuit [35] that introduces very small logic and power overhead while generating random patterns that are sufficient to toggle the execution unit at a low rate.

The testbench module serves to quantitively measure through RTL 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 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 of a logical 1 of every net in the synthesizable design is denoted SP(1) and 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 a RTL simulation of the testbench. When the simulation is complete, it dumps the signal-probability counter values and the corresponding net names into a temporary file. The file is post-processed by the tool, which generates a report. The report summarizes SP(1) for every net in the design and is further described below as part of the tool implementation. The report is reviewed by the designer and 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 repeats itself until no manual fixes remain. Figure 4 illustrates an example of such a manual fix. In this example, one of the multiplexor inputs is connected to logical 0. This, of course, induces excessive BTI stress independently of the logical value of the multiplexor select signal. As seen in Figure 4, this stress can be eliminated simply by replacing the constant 0 with a NOT gate which is connected to the multiplexor selector. The manually fixed circuit is logically equivalent to the original circuit and, assuming that SP(1) = 0.5 for the multiplexer select signal, eliminates the static stress. Note that the proposed design flow does not necessarily rely on such manual fixes by the designer but rather offers them to the designer as an optional step.



Figure – Design flow to avoid asymmetric aging.



Figure – Automatically generated design with asymmetric aging avoidance agent.



Figure – Manual fix at RTL level.

The second phase of the proposed design flow runs at the gate level. The synthesizable module, which includes the original data path, and the asymmetric aging avoidance circuitry are synthesized by using standard synthesis tools. The gate-level netlist is input into the tool. The tool automatically generates a testbench module, as illustrated in Figure 3, like the run in the RTL phase. In this phase, the tool automatically analyzes the gate-level netlist and associates every net to a unique individual signal-probability counter. In the next step, the tool invokes a gate-level simulation of the testbench and the gate-level netlist. Like the RTL phase, once the simulation completes, all counter values are dumped into a temporary file, and a signal-probability report is automatically generated and presented to the designer. Given that the pattern generated by the PRBS circuit may not propagate to all nets in the design, a step of incremental fixes might be needed. These incremental fixes and optimizations may be done manually by the designer, as done in the RTL phase, although in this case, the tools can be configured to automatically perform incremental fixes on nets with asymmetric BTI stress.

The automatic fix procedure is done by the tool as an engineering change order on the gate-level netlist. The tool provides the designer with two options (which can also be combined) to specify the required nets to be fixed. In the first option, the designer provides a list of nets to be fixed by the tool and, in the second option, the designer provides two SP(1) thresholds: threshold-high (thH) and threshold-low (thL). Given the two thresholds, the tool fixes all nets with SP(1) in the range 0 to thL and thH to 1. For example, if thL = 0.2 and thH = 0.7, then all nets with SP(1) ≤ 0.2 or SP(1) ≥ 0.7 are fixed.

Identifying the nets that require incremental fixing involves running a standard timing verification combined with aging models. To accurately model the aging degradation of each cell, the aging models use the signal probability from the flow. Once a timing report is generated, the candidate nets for incremental fixes are obtained from all the violating paths in the report.

The fixes that can be done by the tool are illustrated in Figure 5. The tool automatically detects all the nets to be fixed and inserts a multiplexor to drive the input of the original net. One of the inputs of the multiplexer is connected to the original driver cell, which drives the net before the fix, whereas the second input is connected to one of the outputs of the PRBS generators. The inserted multiplexor selector is controlled by the slow clock, which injects a pseudorandom logical value into the fixed nets. With this method, the tool ensures that the pseudorandom pattern propagates through a selective number of nets in the design. Note that the fix process is fully controlled by the designer. The designer can adjust for the overall design considerations and initiates the needed tradeoffs between the number of fixes required to meet reliability requirements and the power and area overhead associated with the incremental fixes (which is also presented by the tool as part of its report). A broader discussion on overhead is presented in Section 4 as part of the experimental results.



Figure - Automatic incremental fix of a net with asymmetric BTI stress.

##  Architecture of Asymmetric Aging Avoidance Tool

Figure 6 shows a block diagram of our asymmetric aging avoidance tool. The tool was coded in the Tcl language and includes the following software modules: configuration-file parser, hardware description language (HDL) generation engine, signal-probability analyzer, incremental fix engine, 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.

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, both of which are depicted in Figure 3. The HDL generation engine also launches the simulation that, once completed, dumps the SP(1) information into a file. The dump file is analyzed, and the information is input into the report generator and incremental-fix engine. The report generator can generate any combination of reports demanded by the report command specified in the configuration file (see Table 1). Finally, the incremental-fix engine identifies all the candidate nets for fixes, as identified by the signal-probability analyzer, and inserts the incremental fixes shown in Figure 5 into the synthesizable design.

Table 1

Configuration Parameters

|  |  |
| --- | --- |
| Parameter | Details |
| prbs\_type | Specifies PRBS type: PRBS9, PRBS31 or none. |
| design\_type | Species the data path design type: RTL or gate-level |
| inv\_synt | If set to TRUE, the tool will launch the synthesis tool |
| inc\_fix | If set to TRUE, the tool will perform incremental fix at gate-level. |
| inc\_fix\_nets | Specifies a list of nets to be fixed provided by an input file |
| thH and thL | Specify the threshold high and low values when incremented fix is executed. |
| report | histogram – generates SP(1) histogram.netlist – generate SP(1) textual report.Violations – generates textual report of all violating nets with SP(1)<thL and SP(1)>thHarea – reports estimated area overhead as a result of incremental fixes.power – reports estimated power overhead as a result of incremental fixes. |
| fix\_cell\_type | Specifies the standard library cell to be used in the incremental-fix procedure. |

# Experimental Results

This section presents the results of experimentation with our design flow and tool to mitigate asymmetric aging. The experimental analysis has three parts: first, we present the improvement in signal probability gained by applying the proposed tool. Second, we summarize the impact on power and area produced by applying the proposed design flow. Finally, we analyze the timing by applying aging models, which demonstrates the improved reliability gained by the proposed design flow. In the last part, we also combine the incremental-fix mode that is offered by our tool.

As part of our experimental analysis, we examine the effectiveness of the full flow on various data-path units (see Figure 2). The examined data-path execution units consist of integer ALU, FP adder, FP multiplier, FP divider, and FP round logic. All modules were obtained from the OpenCores[[2]](#footnote-2) repository.

The analysis was done on 28 nm process technology using the Synopsys® Digital Cell Libraries (SAED\_EDK28\_CORE) [38]. The process technology and synthesis parameters are summarized in Table 2. For every block examined, we run our tool at the RTL level and gate level, and we use the PRBS generator “PRBS9.” After examining different PRBS generators, we found that this type of PRBS generator performs optimally with minimal overhead. The simulation time of our testbench is 1 million clock cycles. The next subsections summarize the experimental observations and results.



Figure - Tool Block Diagram to Avoid Asymmetric Aging.

Table 2

Process Technology Parameters.

|  |
| --- |
| Core model |
| Process | 28 nm |
| PDK | Synopsys SAED 28 nm |
| Core VDD | 1.05 V |
| Clock frequency | 420 MHz |
| Synthesis tool | Synopsys Design Compiler Q-2019.12-SP1 |
| Process corner | SS\_1p05\_125C (Slow-Slow corner) |

## Signal-Probability Experimental Analysis

In the first part of the experimental analysis, we examine the SP(1) distribution over our data-path modules. Initially, we run the tool at RTL level and generate histogram reports of the signal probability for every module. In the next step, we synthesize every module with the asymmetric aging avoidance circuitry. Finally, we run the tool on the gate-level netlist and generate histogram reports of the SP(1) distribution. Figure 7 presents the histograms generated by the tool (for both RTL and gate levels). The proposed tool and asymmetric aging avoidance circuitry clearly eliminate continuous BTI stress from most nets. The FPU adder, FPU multiplier, and ALU all have SP(1) in the range of 30%–70% for the majority of nets. The FPU divider and FPU round modules have an even tighter distribution, and the majority of their nets have SP(1) in the range of 40%–60%. The histograms show that a smaller fraction of nets cannot be toggled effectively and remain static through most of the simulations. We have determined that this behavior is due to:

1. constant values in the design which are logical 0 or 1;
2. large logical shifters in the FPU adder, FPU multiplier, and FPU divider that involve many nets with constant zero-padding.

Such constant nets are easily fixed by running the proposed tool in incremental-fix mode. However, this may not be needed in this case because such nets are not typically on the critical path in an asymmetrically aged design. In subsection C we discuss how the SP(1) distribution affects reliability and violates timing paths as a result of asymmetric aging.

 

 

 

Figure - SP(1) histograms for integer ALU, FP adder, FP multiplier, FP divider, and FP round execution units.

## Analysis of Power and Area Overhead

As part of our experimental analysis, we integrated all data-path modules and wrapped them together in a top-level module representing an execution unit that combines multiple processing elements and then hierarchically synthesized the execution unit (see Table 3). *Table 2* shows the overall area and power of each module using the process technology parameters. We ran the proposed asymmetric aging flow on the combined gate-level netlist with no incremental-fix part (which will be examined as part of the timing analysis). Table 3 also presents the area of the asymmetric aging avoidance circuitry and the power overhead, both of which are very small (1.5% and 1.6%, respectively).

Table - Area and power overhead of asymmetric aging aware scheme for execution units.

|  |  |  |
| --- | --- | --- |
| Module | Area [um2] | Power [mW] |
| FP Adder | 10 299 | 2.27 |
| FP Multiplier | 60 172 | 7.74 |
| FP Divider | 22 415 | 3.40 |
| FP Round | 3095 | 0.96 |
| Integer ALU | 49 563 | 4.36 |
| Total area w/o asymmetric aging avoidance circuitry | 145 544 | 18.73 |
| Asymmetric aging avoidance Circuitry area overhead  | 2226 (1.5%) | 0.3 (1.6%) |

## Timing Analysis Based on Aging Models

In the last part of our experimental analysis, we analyze the timing based on aging models to examine whether the reliability improves as a result of our proposed design flow. We use aging diagnosis methods like those introduced in previous studies [7, 44], which are more practical for large-scale circuits. The aging model is based on a simplified BTI aging-aware digital library that serves to model the gate-delay degradation (as a function of SP) combined with conventional timing analysis for aging diagnosis. In the aging-aware library, the rising cell delays are derated by their corresponding NBTI degradation factors, whereas the falling delays remain unchanged. The derate factors for the aging libraries were generated by using SPICE simulations to replace the nominal *V*th values with 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 industry and research [36, 47–50]. The reaction-diffusion model produces the following equation for *V*th degradation *ΔV*th resulting from static NBTI stress:

$$∆V\_{th}∝K\_{s}e^{-\frac{E\_{a}}{kT}}(t-t\_{0})^{\frac{1}{6}}$$

Equation – Diffusion-reaction model for Vth degradation.

where *Ks* is a technology-dependent constant, *Ea* is the activation energy of Si, *T* is the operating temperature, *k* is the Boltzmann constant, *t*0 is the time of the onset of the NBTI stress, and *t* is the overall time.

The aging model used herein produces a maximum delay shift of approximately 6% over a ten-year lifetime, which is similar to results of previous studies and to industry observations, which report that BTI degradations may even reach 10% delay shift when stressed [7, 39–44]. Figure 8 summarizes the frequency degradation predicted by the aging models as a result of the BTI stress over a ten-year lifetime.



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

The results show that, when SP(1) is small (i.e., the gates are more likely to be in a constant state of logical 0), the frequency degradation is nearly 6%, whereas, for a greater SP(1), the frequency degradation drops to 2%–3%. This figure also illustrates the absolute delay shift of gates under variable BTI stress relative to gates that age symmetrically [with SP(1) = 0.5]. Such a comparison is of significant value because it demonstrates the asymmetrical delay shift of a circuit under constant BTI stress relative to other circuits within ICs that age symmetrically. Note that gates with constant stress [when SP(1) = 0 or 1] experience a 2.0%–2.5% asymmetric delay shift relative to gates that avoid constant stress [i.e., SP(1) = 0.5–0.6]. In addition, gates with SP(1) in the range 30%–70% experience significantly smaller delay shifts of approximately 1% or lower. Note also that the observed asymmetric delay shift, even one as small as 2%–3%, may crucially affect circuit reliability by introducing unbalanced clock trees, setup, and hold-timing violations. This implication is further supported by the comparison of the timing analysis of fresh, aged, and asymmetric-aging-aware designs (using our proposed design flow and tool), which is summarized in Tables 4 and 5. The timing analysis, which was done using aged and fresh library models, shows that all modules incur setup violations when asymmetric aging is considered. In addition, the FP adder, FP round, and FP divider also experience hold violations. Note that, even if the integer ALU still meets the hold constraints, the positive slack becomes smaller and thereby the design becomes marginal. Tables 4 and 5 also show that the proposed asymmetric aging avoidance tool dramatically reduces the timing violations of the asymmetrically aged design. The FP divider, FP round logic, and integer ALU are now clean from any timing violations. Both FP multiplier and FP adder exhibit a small number of setup and hold violations that are cleaned by applying our incremental-fix flows. In both, the number of paths involved is very small, so the power and area overhead is negligible as a result of the incremental-fix flow.

 Based on our comprehensive experimental results, we conclude that our design flow and tool can cope with different types of data-path design modules by minimizing the asymmetric delay shift and thereby eliminating reliability issues. This has been accomplished with a very small area and power overhead.

Table 4

Number of setup-Violated Endpoint Paths for fresh, aged, and asymmetric-aware desIgns.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Maximum delay (setup) |  |  |
|  | Fresh | Aged  | Asym. aging-avoidance tool | Incremental fix flow |
| FP Add. | 0/687  | 4/687  | 1/687  | 0/687  |
| FP Mul. | 0/1803 | 12/1803 | 3/1803 | 0/1803 |
| FP Div. | 0/992 | 4/992 | 0/992 | 0/992 |
| FP Rou. | 0/322 | 8/322 | 0/322 | 0/322 |
| Int. ALU | 0/134 | 2/134 | 0/134 | 0/134 |

Table 5

Number of HOLD-Violated Endpoint Paths for fresh, aged, and asymmetric-aware desIgns.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Minimum delay (hold) |  |  |
|  | Fresh | Aged  | Asym. aging-avoidance tool | Incremental fix flow |
| FP Add. | 0/687  | 59/687  | 3/687  | 0/687  |
| FP Mul. | 0/803 | 68/803 | 4/1803 | 0/1803 |
| FP Div. | 0/992 | 116/992 | 0/992 | 0/992 |
| FP Rou. | 0/322 | 115/322 | 0/322 | 0/322 |
| Int. ALU | 0/134 | 0/134 | 0/134 | 0/134 |

# Conclusions and discussion

IC reliability is a crucial requirement that has become increasingly difficult to satisfy for advanced process technologies and new computation-intensive applications such as autonomous vehicles, data centers, cloud computing, and life-support systems. Recent advanced process nodes are highly susceptible to asymmetric aging, which can cause critical timing violations in ICs and overall system failure. Asymmetric aging is induced primarily by the BTI when constant voltage is applied for long periods to transistor gates. We summarize the contributions of this paper as follows:

1. We introduce a design flow and tool for asymmetric aging analysis. Our proposed tool can automatically analyze data-path design structures at the RTL or gate level and identify logical elements that are suspectable to asymmetric aging induced by static BTI stress.
2. The proposed tool automatically generates a special circuitry embedded in the design to avoid asymmetric aging while the device is in mission mode. The embedded circuitry injects periodic pseudorandom data at low rates to avoid static BTI stress.
3. The tool introduces an automatic incremental-fix procedure where the asymmetric aging avoidance circuitry is further enhanced to eliminate BTI for gates that are not reachable by the PRBS circuitry.
4. The proposed design flow is easily integrated as part of standard design flows of large-scale ICs.
5. The experimental analysis shows that very little power and area overhead are associated with the asymmetric aging avoidance circuitry.
6. The experimental results indicate that the proposed techniques efficiently eliminated constant BTI stress. The aging model shows that the proposed methods can eliminate an asymmetric delay shift of 2%–3%, which is the threshold for precipitating major reliability issues. Our conclusion is supported by comparing the timing analysis of fresh, aged, and asymmetric-aging-aware designs. The analysis shows that the proposed techniques mitigate the asymmetric timing violations.

Asymmetric transistor aging is becoming an increasingly important phenomenon in many fields such as embedded systems, autonomous cars, and memory databases. Many of these environments require system architects to guarantee the product lifetime, which may be contingent on their reliability. Meeting such demands requires further extensive studies by different disciplines: process technology, physical design, EDA tooling, and system microarchitecture. In the most advanced process technologies of 5 and 3 nm, future reliability issues are expected to become even more complex, mainly because the HCI effect becomes more dominant in these technologies. Physical design flows should thus be developed to better analyze and fix asymmetric aging violations in large-scale circuits. This is a major challenge that requires both industry and research to find practical solutions to allow the future development of reliable large-scale ICs.

References

1. Operating Temperature, Wikipedia <https://en.wikipedia.org/wiki/Operating_temperature>.
2. Failure Mechanism Based Stress Test Qualification for Integrated Circuit. Automotive Electronics Council, Component Technical Committee – AEC – Q100 – REV-G standard.
3. 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. [https://doi.org/10.1016/j.microrel.2006.10.012](https://doi.org/10.1016/j.microrel.2006.10.012%22%20%5Ct%20%22_blank)
4. 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.
5. W. Wang, V. Reddy, A. T. Krishnan, R. Vattikonda, S. Krishnan, and Y. Cao, “Compact modeling and simulation of circuit reliability for 65 nm CMOS technology,” IEEE Trans. Device Mater. Rel., vol. 7, no. 4, pp. 509–517, Dec. 2007.
6. 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.
7. F. Gabbay, A. Mendelson. Asymmetric Aging Effect on Modern Microprocessors. Microelectronics Reliability. Vol. 119, April 2021, 114090. ISSN 0026-2714, https://doi.org/10.1016/j.microrel.2021.114090.
8. 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.
9. 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, December 2016, Pages 64-73
10. Hussam Amrouch, Behnam Khaleghi, Andreas Gerstlauer and Jörg Henkel. Reliability-Aware Design to Suppress Aging. DAC '16: Proceedings of the 53rd Annual Design Automation Conference. June 2016. Article No.: 12 Pages 1–6 https://doi.org/10.1145/2897937.2898082
11. Omer Khan and Sandip Kundu. A Self-Adaptive System Architecture to Address Transistor Aging. 2009 Design, Automation & Test in Europe Conference & Exhibition.
12. 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.
13. 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.
14. ~~F. Gabbay, A. Mendelson. Asymmetric Aging Effect on Modern Microprocessors. Microelectronics Reliability. Vol. 119, April 2021, 114090. ISSN 0026-2714, https://doi.org/10.1016/j.microrel.2021.114090.~~
15. S. Ogawa and N. Shiono, “Generalized diffusion-reaction model for the low-field charge build up instability at the Si-SiO2 interface”, Physical Review, 51(7):4218–4230, Feb. 1995.
16. 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.
17. Reliability Simulation in Integrated Circuit Design. [Online]. Available: <http://www.cadence.com/>
18. ~~Jyothi Bhaskarr Velamala. Compact Modeling and Simulation for Digital Circuit Aging. PhD dissertation.~~ [~~https://repository.asu.edu/attachments/97628/content//tmp/package-sAOTlT/Velamala\_asu\_0010E\_12271.pdf~~](https://repository.asu.edu/attachments/97628/content//tmp/package-sAOTlT/Velamala_asu_0010E_12271.pdf)
19. R. Vattikonda, W. Wang, and Y. Cao, “Modeling and minimization of pmos nbti effect for robust nanometer design”, IEEE/ACM Design Automation Conference, pages 1047–1052, Jul. 2006.
20. J. KO and S. CM, “Negative bias stress of mos devices at high electric fields and degradation of mos devices”, Journal of Applied Physics, 48(5):2004– 2014, May 1977.
21. 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.
22. W. Wang, Z. Wei, S. Yang, and Y. Cao, “An efficient method to identify critical gates under circuit aging,” in Proc. Int. Conf. Comput. Aided Des., Nov. 2007, pp. 735–740.
23. B. C. Paul, K. Kang, H. Kufluoglu, M. A. Alam, and K. Roy, “Impact of NBTI on the temporal performance degradation of digital circuits,” IEEE Electron Device Lett., vol. 26, no. 8, pp. 560–562, Aug. 2005.
24. W. Wang, S. Yang, S. Bhardwaj, R. Vattikonda, S. Vrudhula, F. Liu, and Y. Cao, “The impact of NBTI effect on combinational circuit: Modeling, simulation, and analysis,” IEEE Trans. VLSI Syst., vol. 18, no. 2, pp. 173– 183, Feb. 2010.
25. H. Sangwoo and K. Juho, “NBTI-aware statistical timing analysis framework,” in Proc. IEEE Int. SOC Conf., 2010, pp. 158–163.
26. 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.
27. 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
28. Y. Chen, I. Lin and J. Ke, "ROAD: Improving Reliability of Multi-core System via Asymmetric Aging," 2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Westminster, CO, USA, 2019, pp. 1-8.
29. Techniques for Reducing Uneven Aging in Integrated Circuits. Intel European Patent Application. Application number 18190097.8. 2018.
30. M. S. Mispan, M. Zwolinski, and B. Halak. Ageing Mitigation Techniques for SRAM Memories. Chapter 4, Ageing of Integrated Circuits - Causes, Effects and Mitigation Techniques, Springer, Cham. ISBN 978-3-030-23781-3
31. Kumar, S. V., Kim, C. H., & Sapatnekar, S. S. (2006). Impact of NBTI on SRAM read stability and design for reliability. In: International symposium on quality electronic design (pp. 210–218).
32. Gebregiorgis, A., Ebrahimi, M., Kiamehr, S., Oboril, F., Hamdioui, S., & Tahoori, M. B. (2015). Aging mitigation in memory arrays using self-controlled bit-flipping technique. In: Asia and South Pacific design automation conference (pp. 231–236).
33. Duan, S., Halak, B., & Zwolinski, M. (2018). Cell flipping with distributed refresh for cache ageing minimization. In: IEEE Asian test symposium (pp. 1–6).
34. 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, doi: 10.1109/TCAD.2013.2287187
35. 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.
36. The SPARC Architecture Manual, Version 8.
37. J. Wang, X. Dong, Y. Xie and N. P. Jouppi, "i2WAP: Improving non-volatile cache lifetime by reducing inter- and intra-set write variations," 2013 IEEE 19th International Symposium on High Performance Computer Architecture (HPCA), Shenzhen, 2013, pp. 234-245, doi: 10.1109/HPCA.2013.6522322.
38. Synopsys Digital Standard Cell Library SAED\_EDK32/28\_Core Databook.
39. Bogdan Tudor, Joddy Wang, Zhaoping Chen, Robin Tan, Weidong Liu, Frank Lee, An accurate MOSFET aging model for 28nm integrated circuit simulation, Microelectronics Reliability, Volume 52, Issue 8, 2012, Pages 1565-1570, ISSN 0026-2714, <https://doi.org/10.1016/j.microrel.2011.12.008>.
40. Bogdan Tudor, Joddy Wang, Zhaoping Chen, Robin Tan, Weidong Liu, Frank Lee, MOS Device Aging Analysis with HSPICE and CustomSim, Synopsys® white paper 2011.
41. Campos-Cruz, A.; Espinosa-Flores-Verdad, G.; Torres-Jacome, A.; Tlelo-Cuautle, E. On the Prediction of the Threshold Voltage Degradation in CMOS Technology Due to Bias-Temperature Instability. Electronics 2018, 7, 427.
42. B. Tudor, J. Wang, C. Sun, Z. Chen, Z. Liao, R. Tan, W. Liu, and F. Lee, "MOSRA: An efficient and versatile MOS aging modeling and reliability analysis solution for 45nm and below," 2010 10th IEEE International Conference on Solid-State and Integrated Circuit Technology, Shanghai, 2010, pp. 1645-1647, doi: 10.1109/ICSICT.2010.5667399.
43. Tsai YS, N. K. Jha, Y.-H. Lee, R. Ranjan, Wayne Wang, J.R. Shih, M. J. Chen, J.H. Lee and K. Wu, Prediction of NBTI degradation for circuit under AC operation. IRPS; 2010. p. 665–9.
44. S. V. Kumar, C. H. Kim and S. S. Sapatnekar, "NBTI-Aware Synthesis of Digital Circuits," 2007 44th ACM/IEEE Design Automation Conference, San Diego, CA, 2007, pp. 370-375.
45. Dabhoiwala, Mehernosh H., "Online Nbti Wear-out Estimation" (2013). Masters Theses 1911 - February 2014. 1117. <https://scholarworks.umass.edu/theses/1117>.
46. K. Kang, M. A. Alam, and K. Roy, “Characterization of NBTI induced temporal performance degradation in nano-scale SRAM array using IDDQ,” in *Proc. ITC*, Nov. 2007, pp. 1–10.
47. S. Bhardwaj, et al. Predictive modeling of the nbti effect for reliable design. In CICC, pages 189–192. IEEE, 2006.
48. S. Chakravarthi, A. Krishnan, V. Reddy, C. F. Machala, and S. Krishnan, “A comprehensive framework for predictive modeling of negative bias temperature instability,” in Proc. IEEE Int. Reliab. Phys. Symp., Apr. 2004, pp. 273–282.
49. M. A. Alam, “Reliability- and process-variation aware design of inte- grated circuits,” Microelectron. Reliab., vol. 48, no. 8, pp. 1114–1122, Aug. 2008.
50. S. Nassif, K. Bernstein, D. Frank, A. Gattiker, W. Haensch, B. Ji, et al., “High performance CMOS variability in the 65-nm regime and beyond,” in Proc. IEDM, Dec. 2007, pp. 569–571.
1. This paper was submitted for a review on August 4th, 2020.

F. Gabbay. Author is with the Electrical Engineering Department, Ruppin Academic Center, Israel (e-mail: freddyg@ruppin.ac.il).

A. Mendelson. Author is with the Computer Science and Electrical Engineering Departments, Technion – Israel Institute of Technology of Technology, Haifa, Israel 3200000 (e-mail: avi.mendelson@technion.ac.il). [↑](#footnote-ref-1)
2. [www.opencores.org](http://www.opencores.org) [↑](#footnote-ref-2)