A Design Flow and Tool for Avoiding Asymmetric Aging

Freddy Gabbay,1, 2 Avi Mendelson,2 Basel Salameh,2 and Majd Ganaiem2

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

*Abstract* — Advance process nodes may incur severe reliability concerns due to asymmetric aging, which induces unequal timing degradation of logic elements over time. Most existing tools can handle asymmetric aging in relatively small circuitry and rely on a physical design approach. This paper extends asymmetric aging avoidance design flows using automated engineering change order (ECO) flow. The proposed tool and design flow can be simply integrated as part of standard flows of large-scale integrated circuits. Our experimental analysis of various data-path logic structures reveals that the tool can eliminate reliability issues while introducing minor overhead.

*Keywords* — Asymmetric Aging, Reliability, Bias Temperature Instability, Electronic Design Automation, EDA

# INTRODUCTION

 Advanced very-large-scale integration (VLSI) process nodes can be highly susceptible to reliability concerns. At the same time, new application areas such as data centers, medical appliances, and automotive systems are imposing severe reliability requirements on integrated circuits (ICs) [1, 2]. Transistor aging, which is governed by hot carrier injection (HCI) and bias temperature instability (BTI) [2–4], is now considered one of the most important reliability concerns in IC design. Both HCI and BTI can lead to transistor degraded performance and circuit failure. Applying extra margins to the clock cycle time can mitigate the reliability concern if the degradation is symmetric. However, such an approach may not work when aging degradation is asymmetric. This may lead to a critical timing violation where both hold and setup timing slacks may not be met.

 When a constant voltage is applied to a transistor gate for durations ranging from seconds to several weeks [2], it produces BTI stress, which is the main contributor to asymmetric transistor aging. Various electronic design automation (EDA) tools using physical design-based techniques [5-17] have attempted to cope with asymmetric aging, however, they involved complex simulations and thereby have been effective only for small circuits.

 In this paper, we extend our previous work [6] where we introduced an EDA tool that reduces the impact of asymmetric aging in data-path designs. Our prior study analyzes data-path logic structures and identifies elements that are suspectable to asymmetric aging. The tool generates a special circuitry embedded in the design that attempts to avoid asymmetric aging while the device is in mission mode. The design flow has been demonstrated on two data-path modules: integer arithmetic logic unit (ALU) and floating-point (FP) divider.

This paper extends our previous work and introduces the following contributions:

1. We extend our experimental analysis and examine the asymmetric aging effect in additional data-path structures such as FP multiplier, adder, divider, subtractor, and round logic. We observe that our prior design flow may incur limitations in mitigating asymmetric aging impact. These are further discussed in this study.
2. We extend our previous work and introduce a new automated engineering change order (ECO) flow that can overcome any asymmetric aging reliability issues that cannot be handled by the prior tool.
3. Our experimental analysis shows that the extended tool can fully eliminate asymmetric aging reliability concerns.

After a brief review of asymmetric aging and prior studies, we illustrate how the proposed tool automatically analyzes data-path structures, identifies elements suspectable to asymmetric aging, and generates a special circuitry to avoid asymmetric aging. Our timing simulations combined with aging models indicate that the tool efficiently eliminates asymmetric delay shift and thereby avoids BTI reliability issues. Experimental results show that the design flow incurs negligible power and area overhead and that it can be simply integrated as part of standard design flows of large-scale ICs.

# Asymmetric Aging

ICs designed in advance process nodes are highly susceptible to reliability issues. The analysis of reliability issues is a major challenge to EDA tools since it involves highly complex simulations of numerous combinations of process, temperature, and voltage corners. Existing design-for-reliability flows have been mainly focused on physical design approaches. Unfortunately, such approaches are typically limited to small circuits and may involve heavy simulations.

## Asymmetric Transistor Aging

Transistor aging is the process whereby silicon transistors develop faults in their circuitry over time [2]. This degradation is induced by two physical mechanisms: HCI and BTI. When HCI is involved, excessive energy causes charge carriers from the source-drain current 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. 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, in which case the rising and falling transients incur asymmetric delay shifts. The problem can also occur when the timing degradation between different timing paths becomes asymmetric [5]. This may lead to severe setup and hold timing violations, which cannot be identified by conventional timing analysis tools. This phenomenon, referred to as “asymmetric aging,” is a major reliability concern in ICs.

Asymmetric aging is induced by long periods of unequal static stress applied to logic elements in different timing paths, which may range from tens of seconds to several weeks [2]. As the main contributor to asymmetric aging, BTI can degrade the speed of both p-type (NBTI) and n-type (PBTI) devices. The NBTI’s impact is more significant, although in advanced process nodes PBTI’s impact is also considerable.

Asymmetric aging introduces major challenges due to the 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 on the functional modes of operation throughout the IC’s lifetime. From an architectural point of view, asymmetric aging in many cases is a result of dynamic power-saving techniques that impose static states on logic circuits.

## Prior Studies

Many prior studies attempted to cope with asymmetric aging using physical design approaches [7]. However, the complexity level incurred by these methods was very high and introduced entry barriers in large-scale circuits is highly complex. Current chip design flows run highly complex analysis tools and use extra timing margins to mitigate the asymmetric aging effect. Other techniques attempted to predict the transistor speed degradation resulting from NBTI [2] and offered various mitigations such as *V*DD tuning, transistor sizing, duty cycle reduction, and decreasing channel length.

A limited number of studies attempted to offer architectural solutions for asymmetric aging. Firouzi et al. used an embedded no operation (NOP) instruction to mitigate NBTI effect on microprocessors [8]. Abbas et al. used anti-aging programs when the processor was not utilized [9]. This approach seemed to be quite efficient, however, it required complex analysis of the critical paths and the requisite anti-aging values. Other studies have proposed solutions for asymmetric aging in memory systems [4].

Gabbay et al. examined asymmetric aging in microprocessors and identified three main causes of BTI stress: microprocessor’s backward-compatibility features, new forward-compatible features, and microprocessors being by their nature general-purpose machines. In their study, they introduced an asymmetric aging-aware microarchitecture that aimed to reduce the BTI effect in microprocessors’ execution units, register files, and cache memories. Their further study [6] introduced an EDA tool that was aimed at avoiding BTI impact on data-path structures. Their tool embedded a special circuitry that periodically injected a pseudorandom bit sequence into the design to avoid long periods of BTI stress and balance signal-probability 0 and 1. A further examination of the prior EDA tool has found that it may incur several limitations that are addressed by this study.

# Design Flow and Tool for Asymmetric Aging Avoidance

This study extends the design flow and tool presented in our previous work and introduces an enhanced automated ECO flow that can overcome any asymmetric aging reliability issues that cannot be handled by the prior tool. The configuration of the tools and design flow is specified through a designated configuration file which will be discussed later. Figure 1 (a) illustrates the design flow, which consists of two stages: a register transfer level (RTL) stage and a gate-level stage. In the first stage, the tool runs on the original design at the RTL level and produces a testbench module and a synthesizable module. In the two modules, which are illustrated in Figure 1(b), the light and dark gray colors illustrate the elements that are automatically inserted by the tool. The synthesizable module hierarchy consists of a top-level wrapper, the original data path, a pseudorandom binary sequence (PRBS) generator, and a multiplexor. The tool supports two types of PRBS, PRBS9 and PRBS31 ([6]), that can be specified in the configuration file. The PRBS module runs at a slow clock rate and injects a pseudorandom pattern into the data path to avoid extended periods of BTI stress when the module is unused. The slow clock rate can be on the order of megahertz or lower to minimize dynamic power overhead. The multiplexor, shown in Figure 1(b), can switch between the PRBS pattern and the functional inputs of the data path. The multiplexor is controlled by the PRBS enable signal, which is activated whenever the data path is idle.

The testbench module is run in RTL simulation to measure the signal-probability 1, denoted as SP(1), of every net in the synthesizable design. We use SP(1) to measure the efficiency of the asymmetric aging avoidance circuitry where the ideal goal is to achieve SP(1) of 50%. The testbench instantiates the synthesizable design, the clock generators. and a set of signal-probability counters. All nets in all design hierarchies are mapped by the tool and each net is associated with an individual counter. Every counter measures the number of clock cycles in which the corresponding net is in the logic state of 1. The corresponding SP(1) is calculated by dividing the counter value by the number of simulated clock cycles.

The RTL simulation is run next by the tool and an SP(1) report is generated for all the nets in the design. The report also summarizes the nets with excessive BTI stress that could not be eliminated by the asymmetric aging avoidance circuitry. In certain cases, such nets can be handled manually as illustrated by the example in Figure 1 (c). In such a case the process will repeat itself until there are no manual fixes left.



Figure 1 – (a) Asymmetric aging avoidance design flow, (b) an automatically generated design with asymmetric aging avoidance circuitry, and (c) manual fix at the RTL.

The RTL stage is followed by the gate-level stage. Both the synthesizable module and the asymmetric aging avoidance circuitry are synthesized, and the gate-level netlist is generated. Similar to the RTL stage, the tool analyzes the SP(1) of every net, this time at gate-level. Because the PRBS generator may not be able to achieve the ideal SP(1) for all nets, additional steps are required to handle the remaining elements that are susceptible to asymmetric aging. This can be done manually as previously illustrated in Figure 1 (c), or by performing automatically by the tool. The automatic ECO fix procedure, illustrated by the gray color in Figure 1 (a), is the enhancement of our previous work introduced in this paper [6]. The new ECO fix flow is done by the tool on the gate-level netlist and can overcome any asymmetric aging reliability issues that cannot be handled by all previous stages of the design flow.

At the automatic ECO stage, the tool provides the designer with two options (which can be combined) to specify the required fixes. In the first option, the designer provides a list of nets for the tool to fix. 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. Identifying the nets that require ECO fixing involves running a timing verification with aging models in conjunction with the SP(1) values from the flow. All the nets from the violating paths are candidates for automatic ECO fixes.

Figure 2 illustrates the ECO fixes that can be provided by the tool to avoid asymmetric aging. The tool detects all the nets to be fixed and inserts a multiplexor to drive the original net’s input. One of the inputs of the multiplexer is connected to the original driver cell, and a second input is connected to one of the outputs of the PRBS generators. The multiplexor selector signal is controlled by the ‘enable PRBS’ signal. 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 overall design considerations and initiates the needed tradeoffs between the required number of fixes and the power and area overhead associated with the incremental fixes (also included in the tool’s report).



Figure 2 – Automatic ECO fix.

Figure 3 illustrates a block diagram of our tool. The tool, which we designed in the Tcl language, 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 (see Figure 3). The HDL generation engine creates a new top-level wrapper for the synthesizable design and a testbench module. The HDL generation engine invokes the simulation, and once 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 create the reports described in Figure 3. Finally, the incremental-fix engine identifies all nets that require fixes and inserts the needed incremental fixes.



Figure 3 – Asymmetric aging avoidance tool block diagram.

# Experimental Results

As part of our experimental analysis, we examine the effectiveness of the full design flow presented in Figure 1 (a) on integer ALU, FP adder, FP multiplier, FP divider, and FP round logic. All modules were obtained from the OpenCores[[1]](#footnote-1) repository. Our analysis is performed on 28 nm process technology using the Synopsys® Digital Cell Libraries [10]. The synthesis tool (Synopsys Design Compiler Q-2019.12-SP1) was executed using nominal signoff conditions with a core voltage of 1.05 V, a clock frequency of 420 MHz, and a process corner SS\_1p05\_125C (Slow-Slow corner). We used the PRBS9 generator [6] and a simulation runtime of 1 million clock cycles.

## Signal-Probability Experimental Analysis

 At the first stage, our experimental analysis examined the SP(1) distribution. We ran the full flow presented in Section 3 and generated SP(1) histogram reports for each module at RTL level and gate-level. Figure 4 presents the histograms for both RTL and gate-level. Figure 4 shows that the proposed tool and circuitry can remove the continuous BTI stress from most nets. It should be noted that continuous static BTI stress results in approximately 50% of nets being static logical 1, while the rest are static logical 0 [5,6]. After running the asymmetric aging avoidance tool, we observe that at gate-level the FP adder, FP multiplier, and ALU all have an SP(1) in the range of 30% to 70% for the majority of nets. The FP divider and FP round modules have an even tighter distribution at gate-level with the majority of their nets having an SP(1) in the range of 40% to 60%. The histograms show that a smaller portion of nets cannot be toggled effectively and remain static through most of the simulations. We have identified that this behavior is due to (1) constant values in the design and (2) logical shifters in the integer ALU and in the FP divider that involve many nets with constant zero padding. Such nets are easily fixed by running the proposed tool in automatic ECO fix mode. This may only be necessary for a small number of nets because such nets are not typically on the critical path in an asymmetrically aged design.



Figure 4 – SP(1) histograms for execution units.

## Power and Area Overhead Analysis

Through the experimental analysis, we combined all modules and instantiated them under a top-level module to represent an execution unit that combines multiple processing elements. We synthesized the combined execution unit using the process technology parameters we described earlier. We ran the asymmetric aging avoidance flow on the combined gate-level netlist. The total area and power of all modules and the asymmetric aging avoidance circuitry area and power overhead are summarized in Table 1.

Table 1

Power and Area Overhead

|  |  |  |
| --- | --- | --- |
|  | Power[mW] | Area[mm2] |
| Original design | 0.3 | 0.1455 |
| Asymmetric aging avoidance circuitry overhead | 0.0081(2.7%) | 0.0022 (1.5%) |

## Timing Analysis Based on Aging Models

Our last part in the experimental analysis consists of a timing analysis based on aging models to examine the reliability improvement achieved by the tool. Our timing analysis extended the standard timing models with aging models in which cell delays are derated by their corresponding aging degradation factors (as a function of the SP). The derate factors were obtained from SPICE simulations that were run with aged *V*th values that correspond to the lifetime and SP. The *V*th degradation model used is based on the reaction-diffusion model, which is the most widely accepted model for BTI aging [6, 11, 12].

 Table 2 summarizes our timing analysis comparison of fresh (timing slack is met), aged, and asymmetric-aging-aware designs. The results show 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 decreases, and thereby the design reliability becomes marginal. Table 1 also shows that the tool significantly reduces the timing violations of the asymmetrically aged designs. The FP divider, FP round logic, and integer ALU are clean from any timing violations. Both FP multiplier and FP adder exhibited a small number of setup and hold violations that were fully handled by applying the automatic ECO fix flow. In both modules, the number of paths involved was small, so the power and area overhead were negligible.

Table 2

setup and HOLD VIOLATED Endpoint Paths for fresh, aged, and asymmetric-aware desIgns.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Total timing path |  | Setup |  | Hold |
|  |  | Aged  | Asym. Aging avoidance tool | Aged  | Asym. Aging avoidance tool |
| FP Add. | 687 | 4  | 1  | 59 | 3 |
| FP Mul. | 1803 | 12 | 3 | 68 | 4 |
| FP Div. | 992 | 4 | 0 | 116 | 0 |
| FP Rou. | 322 | 8 | 0 | 115 | 0 |
| ALU | 134 | 2 | 0 | 0 | 0 |

# Conclusions

Asymmetric transistor aging can cause critical timing violations and overall system failure for modern ICs. It introduces serious reliability concerns for ICs that are highly emphasized in advanced VLSI process nodes and new computation-intensive applications. In this paper, we enhanced our prior studies and introduced a new automated ECO flow that can overcome any asymmetric aging reliability issue that could not be handled by the prior tool. In addition, we extended the experimental analysis and examined new data-path structures such as FP multiplier, adder, divider, subtractor, and round logic. Our experimental analysis indicates that the enhanced tool can fully eliminate asymmetric aging reliability concerns

Further exploration of new EDA tools for asymmetric aging avoidance is essential since the impact of this phenomenon is emphasized in advanced process nodes. A possible future direction is to further extend the proposed tool and design flow to control path modules, memory systems, and domain-specific accelerators.

References

1. Failure Mechanism Based Stress Test Qualification for Integrated Circuit. Automotive Electronics Council, Component Technical Committee – AEC – Q100 – REV-G standard. 2007.
2. 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.
3. 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.
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. F. Gabbay, A. Mendelson, B. Salameh and M. Ganaiem Asymmetric Aging Avoidance EDA Tool. 34th SBC/SBMicro/IEEE/ACM Symposium on Integrated Circuits and Systems Design (SBCCI), 2021, pp. 1-6, doi: 10.1109/SBCCI53441.2021.9529984.
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. 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.
9. 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, 2020.
10. Synopsys Digital Standard Cell Lib. SAED\_EDK32/28\_Core Databook.
11. 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
12. 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.
1. [www.opencores.org](http://www.opencores.org) [↑](#footnote-ref-1)