Clock Tree Design Considerations in The Presence of Asymmetric Transistor Aging

1st Freddy Gabbay
Faculty of Engineering
Ruppin Academic Center
Emek Hefer, Israel
freddyg@ruppin.ac.il

2nd Firas Ramadan
Faculty of Electrical and Computer Engineering
Technion - Israel Institute of Technology
Haifa, Israel
firasramadan@campus.technion.ac.il

3rd Majd Ganaiem
Faculty of Electrical and Computer Engineering
Technion - Israel Institute of Technology
Haifa, Israel
majd.ga@campus.technion.ac.il

***Abstract*—Reliability is critical for integrated circuits (ICs) to ensure accurate lifetime operation. With the rise of mission-critical systems, the need for reliability continues to grow. However, recent advancements in semiconductors have uncovered the vulnerability of ICs to reliability issues, particularly those stemming from transistor aging. Transistor aging refers to the gradual deterioration of a transistor's performance over time, mainly influenced by bias temperature instability (BTI). BTI can severely impact IC reliability, leading to performance degradation and critical circuit failures due to timing violations. Additionally, asymmetric aging occurs when the degradation is unevenly distributed, intensifying timing violations and reliability concerns. This paper examines the impact of asymmetric transistor aging on clock tree designs, highlighting the role of useful skew, clock gates, and asymmetry between clock buffer delays and net delays in amplifying reliability concerns. Furthermore, new design flow guidelines are provided to address asymmetric aging-related violations in clock trees.**

***Index Terms*—Transistor aging, BTI, Reliability, Clock-tree**

I. Introduction

In recent decades, VLSI technologies have witnessed remarkable advancements, characterized by several significant trends. First, the continuous development of new process nodes has ensured the consistent miniaturization of transistors to nanometric dimensions, in line with the principles of Moore's law. Second, revolutionary devices and materials have played a pivotal role in driving advancements, resulting in improved performance and reduced energy consumption. However, these advancements have also brought to light the vulnerability of integrated circuits (ICs) to reliability issues, particularly those caused by transistor aging. Transistor aging refers to the declining process in a transistor's performance over time, primarily attributed to bias temperature instability (BTI), which will be further described in Section 2. The impact of BTI on IC reliability is significant, causing performance degradation and critical circuit failures due to timing violations. Moreover, asymmetric aging exacerbates timing violations and amplifies concerns regarding reliability, as it results from unevenly distributed degradation.

Continuing with the ongoing trends of advances in VLSI technologies, semiconductors are increasingly being integrated into mission-critical systems, including autonomous vehicles, medical appliances, finance, and security systems [14], [15]. These emerging applications have raised the standards for ICs in terms of resiliency, reliability, and safety, as enforced by regulatory agencies and industry standards [13]. As a result, the necessity for reliability-aware IC design has become crucial.

This paper investigates the influence of asymmetric aging on clock tree design considerations. Clock trees are crucial circuit resources responsible for distributing a balanced clock signal across the chip die. The reliability of the clock tree is paramount to IC reliability, as even a single point of failure within the clock tree can lead to a complete failure of the entire clock distribution network. In this paper, we show that clock trees are highly vulnerable to reliability issues caused by asymmetric aging. While prior studies [9], [10] mainly focused on the impact of asymmetric aging on gated clocks, our study extends the scope of previous works and indicates that factors such as useful skews and the asymmetry between net and cell delays can contribute to significant timing violations in the clock tree, ultimately resulting in overall IC failure. As a case study, our experimental analysis examines the clock tree susceptibility to asymmetric aging in General Purpose Graphic Units (GPGPUs) and demonstrates the timing violations that arise when considering asymmetric transistor aging. Additionally, we introduce new timing constraints combined with new design flow guidelines to mitigate the impact of asymmetric aging on clock trees. The proposed mitigation guidelines are examined through aging-aware timing analysis.

The remainder of this paper is structured as follows: Section 2 provides background and discusses prior works. Section 3 examines the vulnerability of clock trees in presence of asymmetric transistor aging and introduces new extended timing constraints. In Section 4, we present our experimental results. Finally, Section 5 concludes our work.

II. Background and Prior Works

This section offers background information on transistor aging and BTI. It also provides an overview of previous studies conducted in the field of aging.

A. Transistor Aging

Transistor aging refers to the deterioration process of transistors in digital circuits ([8, 9]), which is caused by the trapping of charge carriers from the transistor inversion channel at

the dielectric insulator of the transistor gate. BTI is recognized as the primary mechanism governing transistor aging. BTI is activated when a constant voltage is applied to the transistor gate, resulting in an elevation of the transistor's threshold voltage. Consequently, this increase in threshold voltage leads to a rise in transistor switching delay and a reduction in transistor speed. Asymmetric aging, which denotes the uneven distribution of performance degradation among transistors within an IC, can lead to severe timing issues, including setup and hold timing violations.

The signal probability (SP) is a prevalent technique [9] for assessing the BTI stress profile on logical elements. SP quantifies the probability of a signal having a logical value of 1. It is defined as the ratio of the time a signal spends in the logical 1 state to the overall time. Decreased SP intensifies the impact of BTI, resulting in performance degradation or potentially causing failures in integrated circuits over time.

B. Prior works

Common approaches propose incorporating additional timing margins to mitigate the effects of asymmetric aging. However, this approach often necessitates complex simulation analysis and can lead to overdesign [1]. Other studies ([2]–[4]) have proposed models for predicting aging degradation and have explored various solutions, including clock cycle time reduction, transistor resizing, VDD tuning and power gating. Agrawal et al. ([6]) proposed a method to predict circuit failure by using sensors placed at various locations within the silicon die. Additional research ([7]) has explored techniques to analyze digital circuits and detect the most vulnerable gates affected by NBTI stress. This involves utilizing an aging model with BTI-aware libraries and conducting aging-aware timing analysis. Abbas et al. ([8]) proposed executing anti-aging programs during periods of low processor utilization instead of idle tasks. In [9], an aging-aware microarchitecture was proposed to minimize the effects of asymmetric aging on execution units, register files, and memory hierarchy in microprocessors, while keeping overhead to a minimum. An analysis of asymmetric aging in the clock tree segments of power-efficient designs has been presented in [10] using a 45nm process node. In their study, the authors examined the BTI impact on the clock tree as a result of clock gates and built-in clock tree skews. However, their study did not consider the impact of useful skew and the asymmetry between net and cell delays on clock tree design. These important factors are extensively addressed by our study using a 28nm process node.

III. The Impact of Asymmetric Transistor Aging on Clock Trees

Clock trees are responsible for delivering the clock signal across digital circuits with minimal insertion delay and minimal clock skew between clock tree endpoints. The clock signal is crucial for the correct logical operation of digital circuits. Any single failure in the clock tree can potentially manifest as an overall circuit failure. Therefore, to ensure reliable operation of the clock signal, setup and hold timing constraints are enforced.

In this paper, we introduce three main factors that encourage asymmetric aging in clock trees, which will be discussed subsequently.

A. Clock Gating

Clock gating is a widely accepted approach for achieving dynamic power savings. It entails selectively blocking the clock signal in inactive sections of the circuit, effectively reducing dynamic power consumption. By deactivating the clock in idle circuit areas, unnecessary switching and associated power usage are eliminated. Clock gating typically involves utilizing a clock gate cell, which comprises a latch and an AND gate.

Clock gating can contribute to asymmetric aging by increasing the idleness of the clock network, as demonstrated in Figure 1 (a) and (b). In Figure 1(a), the clock gate is utilized in the launch path, leading to greater aging in this path compared to the capture path. This imbalance can result in setup-timing violations. Conversely, in Figure 1(b), employing the clock gate in the capture path intensifies its aging relative to the launch path, leading to hold-timing violations.

B. Asymmetry Between Cell and Net Delays

Asymmetric aging in clock network can also be caused by the disparity in accumulated delay between logical cells and nets. Unlike logical cells, nets are not affected by BTI. When the launch and capture paths have varying accumulated logical-cell delays, BTI can induce asymmetric aging, as depicted in Figure 1(c). If the accumulated logical-cell delay in the launch path exceeds that in the capture path, it can lead to setup-timing violations. Conversely, if the accumulated cell delay in the launch path is smaller than that in the capture path, it may result in hold-timing violations. Figure 1(c) illustrates a scenario where both the launch and capture clocks have a balanced 100 ps clock latency. However, the accumulated clock buffer delay in the capture path is 60 ps, whereas the total delay of the clock buffer in the launch path is 40 ps. Even when assuming equal aging rates for all clock buffers, the asymmetry between accumulated cell and net delays, combined with BTI, can cause hold-timing violations due to the delay shift in the capture clock.

C. Useful Skew

Useful skew is a common design method in clock tree synthesis, where intentional delays are introduced in clock paths to mitigate setup or hold timing violations. When clock skew is inserted on the capture path as illustrated in Figure 1 (d), it allows extending the design's critical timing path beyond the clock cycle time. This can only be applied if the timing path has extra positive hold slack to accommodate the delay in the capture clock. In this way, clock skew can either avoid an increase in the clock cycle time or mitigate timing violations. When useful skew is inserted in the launch path, it operates

similarly, but this time it helps extend hold margins at the expense of setup.

Useful clock skew increases the susceptibility of clock trees to asymmetric transistor aging, particularly when it is used in the presence of clock gating or when there is asymmetry between cell and net delays. Clock gating can encourage asymmetric aging of clock skew buffers, potentially contributing to timing violations when considering transistor aging. Additionally, the presence of clock skew buffers induces an intrinsic asymmetry between accumulated net delays and cell delays, which further encourages timing violations in the presence of transistor aging.

D. Timing Constraints in Presence of Asymmetric Aging

The timing constraints for a general synchronous digital circuit, as illustrated in Figure 2, are defined by Equations (1) and (2). The corresponding timing parameters of the circuit are outlined in Table I, assuming a clock cycle time of T. It is worth noting that the useful skew buffers shown in Figure 2 are positioned on the capture path with a delay parameter t_{us}. When $t_{us} > 0$, the useful skew is applied to the capture path, while it is applied to the launch path when $t_{us} < 0$.

TABLE I: Timing Parameters

Elements	Timing Parameters			
	Propagation Delay	***Containment Delay***	***Setup time***	***Hold time***
Launch clock buffers #1	t_{CL}	n/a	n/a	n/a
Launch clock nets #2	t_{NL}	n/a	n/a	n/a
Capture clock buffers #3	t_{CC}	n/a	n/a	n/a
Capture clock nets #4	t_{NC}	n/a	n/a	n/a
Flip-flops #5	t_{pdFF}	t_{cdFF}	t_s	t_h
Combinational Circuit #6	t_{pdC}	t_{cdC}	n/a	n/a
Useful skew buffers #7	t_{us}	n/a	n/a	n/a

$$\Delta slack_{setup} = T - t_{pdFF} - t_{pdC} - t_s + t_{us} + (t_{CC} + t_{NC}) - (t_{CL} + t_{NL}) \quad (1)$$

$$\Delta slack_{hold} = t_{cdFF} + t_{cdC} + (t_{CL} + t_{NL}) - (t_{CC} + t_{NC}) - t_{us} - t_h \quad (2)$$

Assuming that the launch and capture paths have undergone asymmetric aging due to different clock gating activities (or different SPs), we can represent the derate factors resulting from BTI as $d_L > 1$ for the launch path and $d_C > 1$ for the capture path. By applying d_L to the launch cells and d_C to the capture cells, as described in Equations (1) and (2), we can calculate the setup and hold slacks for an aged circuit using Equations (3) and (4) respectively.

$$\Delta slack^{aged}_{setup} = T - d_L \times (t_{pdFF} + t_{pdC} + t_{CL}) - t_{NL} + d_C \times (t_{CC} + t_{us}) + t_{NC} - t_s \quad (3)$$

$$\Delta slack^{aged}_{hold} = d_L \times (t_{cdFF} + t_{cdC} + t_{CL}) + t_{NL} - d_C \times (t_{CC} + t_{us}) - t_{NC} - t_h \quad (4)$$

Let δ_L and δ_C represent the added shift fractions in the launch and capture paths, respectively, such that $\delta_L = d_L - 1$ and let $\delta_C = d_C - 1$. By combining Equation 1 with Equation 3 and Equation 2 with Equation 4, we can derive the setup and hold slacks for an asymmetrically aged circuit, as described in Equations 5 and 6 respectively:

$$\Delta slack^{aged}_{setup} = \Delta slack_{setup} - (\delta_L \times (t_{pdFF} + t_{pdC} + t_{CL}) - \delta_C \times (t_{CC} + t_{us})) = \Delta slack_{setup} - \Delta slack^{degredation}_{setup} \quad (5)$$

$$\Delta slack^{aged}_{hold} = \Delta slack_{hold} - (\delta_C \times (t_{CC} + t_{us}) - \delta_L \times (t_{cdFF} + t_{cdC} + t_{CL})) = \Delta slack_{hold} - \Delta slack^{degredation}_{hold} \quad (6)$$

The degradation in the setup ($\Delta slack^{degredation}_{setup}$) and hold ($\Delta slack^{degredation}_{hold}$) slacks due to asymmetric aging can be expressed by Equations 7 and 8 respectively.

$$\Delta slack^{degredation}_{setup} = \delta_L \times (t_{pdFF} + t_{pdC} + t_{CL}) - \delta_C \times (t_{CC} + t_{us}) \quad (7)$$

$$\Delta slack^{degredation}_{hold} = \delta_C \times (t_{CC} + t_{us}) - \delta_L \times (t_{cdFF} + t_{cdC} + t_{CL}) \quad (8)$$

Equations 7 and 8 demonstrate two opposing forces that govern slack degradation. In setup, if the degradation in the capture path exceeds that in the launch path, it will result in setup slack degradation. On the other hand, if the degradation in the launch path is greater than that in the capture path, it will lead to hold slack degradation. It is also worth noting that even if $\delta_L = \delta_C$, indicating that both the capture and launch paths experience symmetric aging, both slacks can still be degraded. Moreover, setup timing violations can occur when $\Delta slack^{degredation}_{setup} > \Delta slack_{setup}$, while hold timing violations can happen when $\Delta slack^{degredation}_{hold} > \Delta slack_{hold}$.

IV. Experimental Analysis

Our study demonstrates the impact of asymmetric transistor aging on GPGPU processing elements (PEs). We conduct an experimental analysis using functional simulations to extract the aging profile of GPGPUs, which includes signal probability measurements. Next, we perform a comprehensive timing analysis that incorporates aging models in conjunction with the aging profile. As part of our experimental analysis, we examine the potential timing violations due to asymmetric aging when considering the impact of: 1) useful skew optimization and 2) the asymmetry between net and cell delays. Additionally, we show that certain timing paths may require extra timing margins to compensate for the degradation due to asymmetric aging. Finally, we summarize the needed extensions for the conventional physical design flow to incorporate the impact of asymmetric transistor aging.

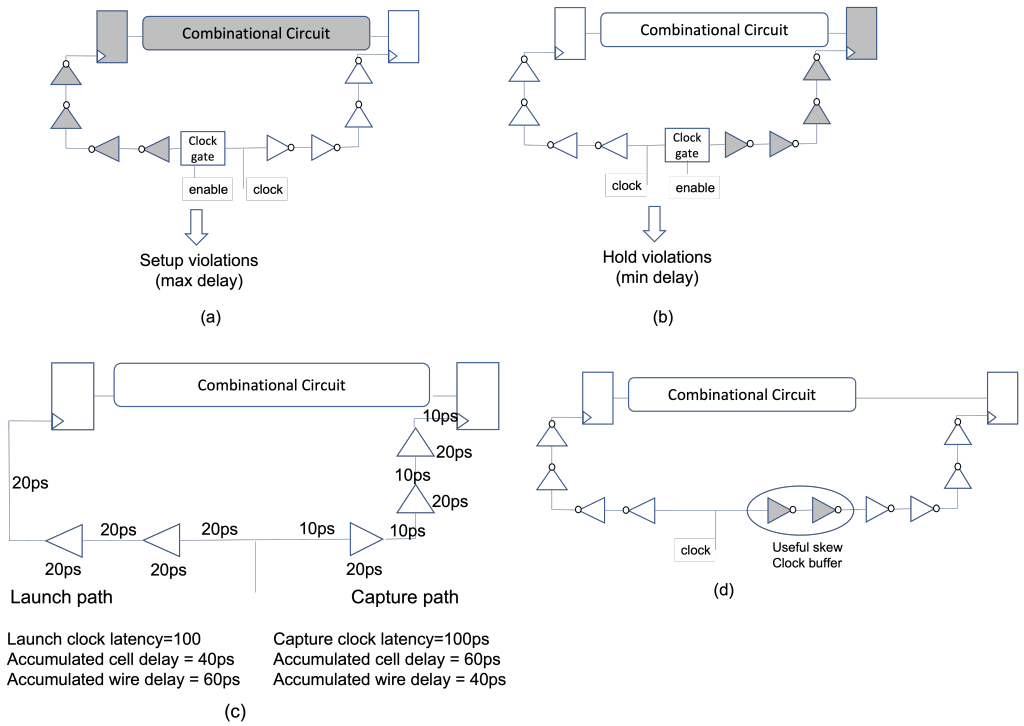

Fig. 1: Asymmetric aging in clock trees: (a) a clock gate on the launch clock, (b) a clock gate on the capture clock, (c) asymmetry between accumulated logical cells delay and net delay, and (d) useful clock skew.

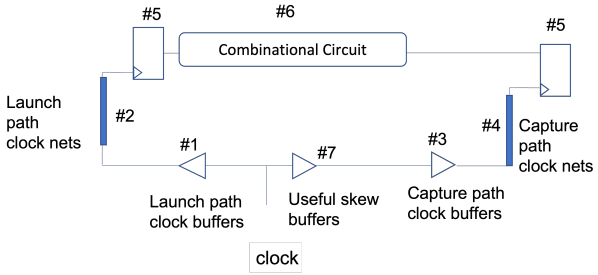

Fig. 2: Asymmetric aging in clock trees in presence of useful skew and net delays

Our functional experiments were conducted using the GPGPU simulator [11]. The simulation environment incorporates cycle-level modeling of the NVIDIA Volta V100 GPGPU [12], enabling the execution of CUDA or OpenCL computing workloads. To meet the specific requirements of our experiments, we modified the simulation platform and integrated essential mechanisms for the needed measurements. For benchmarking purposes, we employed the Neural Network (NN) benchmark from the gpgpu-sim benchmark suite of IPSS [11]. The NN benchmark encompasses training and inference of neural networks.

As part of our functional experimental analysis we measure the activity ratio and SP of the integer execution unit and the single precision floating-point unit (FPU). Figures 3 and 4 illustrate the measured activity levels in the Volta V100 Streaming Multi-Processors (SMs) during the NN benchmarks. Activity is quantified as the percentage of time the execution unit remains active relative to the total elapsed time. Our experimental findings indicate that the integer execution units within all SMs are idle approximately 80-85% of the time, while the FPUs are idle for more than 98% of the time. These observations imply that the GPGPU PEs can be susceptible to transistor aging due to their extended idle periods. The prolonged duration of being in an idle state amplifies their exposure to asymmetric aging effects, which can introduce serious reliability concerns.

To examine the GPGPU case study, we use the integer execution unit and FPU from an open-source Nyuzi Processor GPGPU[1] [16] to investigate the effects of asymmetric aging on timing. We conduct full synthesis and place and route on the GPGPU modules in the 28nm process node. The synthesis is performed using Cadence® $Genus^{TM}$, and the place-and-route is executed with Cadence® $Innovus^{TM}$. The clock frequency of the integer execution unit is 250MHz while the FPU is assumed to operate at 167MHz. For the timing analysis we utilize the measured aging profile of the Volta V100 in conjunction with aging-aware library models, as detailed in [9]. These models consider the impact of BTI by adjusting cell delays based on NBTI degradation factors derived from SP values extracted from the functional simulations depicted in Figures 3 and 4.

The timing results in Table II present the worst negative slack (WNS) and the number of timing violations for both

[1]https://github.com/jbush001/NyuziProcessor/tree/master/hardware/core

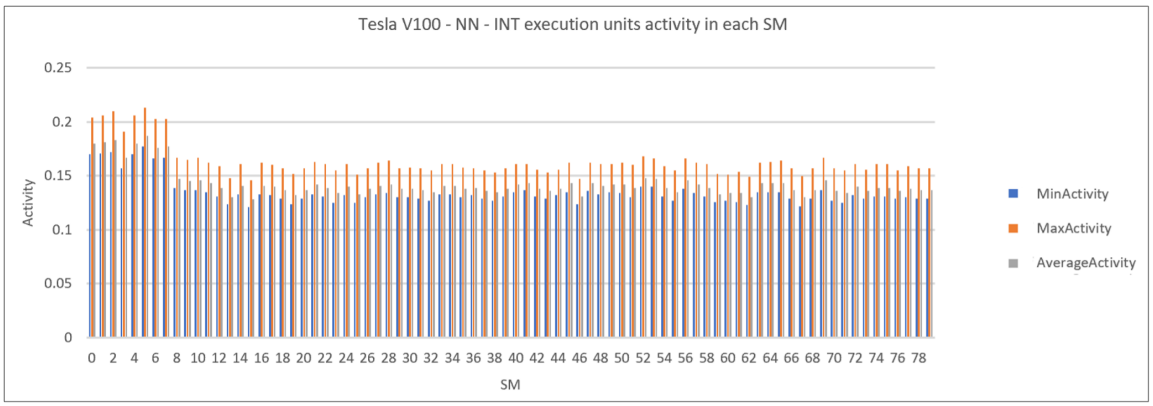

Fig. 3: Activity fraction of the integer execution units in Volta V100 Streaming Multiprocessors for NN benchmark.

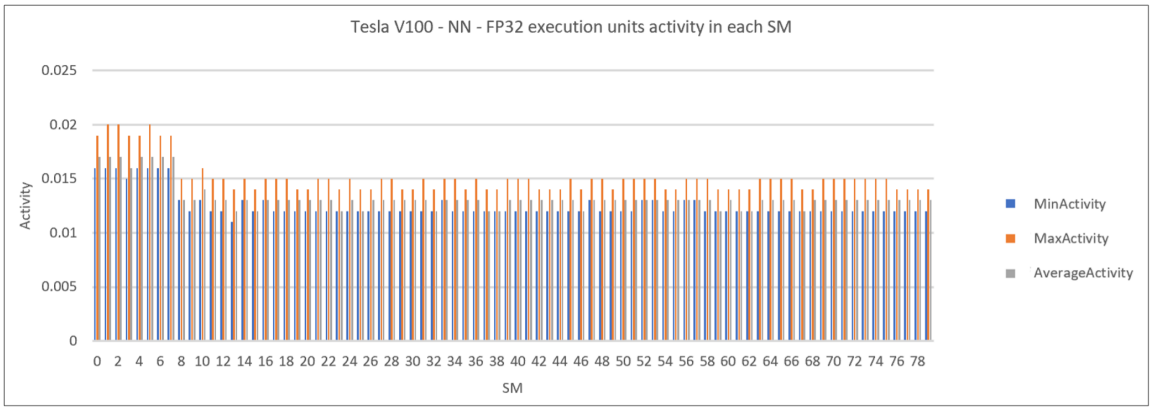

Fig. 4: Activity fraction of the single precision floating-point units in Volta V100 Streaming Multiprocessors for NN benchmark

TABLE II: Timing Analysis Summary in Presence of Asymmetric Aging

Func, Units	Setup WNS [ps] /Number of Violations			
	No Aging	*Aging*	*Aging+U.S.*	*Aging+U.S.+N.D.*
I-EXU	0 / 0	-157 / 8	-128 / 4	-181 / 22
FPU	0 / 0	-238 / 7077	-237 / 7042	-337 / 9753
Func. Units	**Hold WNS [ps] /Number of Violations**			
	No Aging	*Aging*	*Aging+U.S.*	*Aging+U.S.+N.D.*
I-EXU	+13 / 0	+9.5 / 0	-1 / 2	-1 / 10
FPU	+4.5 / 0	+4.5 / 0	+3.1 / 0	-2 / 7

setup and hold analysis in the following cases:

1) When the effect of asymmetric aging is not taken in to account (No Aging).
2) When the effect of asymmetric aging is included in timing analysis (Aging).
3) When using useful skew optimization and considering asymmetric aging impact (Aging+U.S.).
4) When using useful skew optimization and considering asymmetric aging in conjunction with the asymmetry between net delay and cell delay (Aging+U.S.+N.D.).

The results presented in Table II indicate that when asymmetric aging is taken into account with no useful skew optimization, it introduces setup violations in both integer execution unit (I-EXU) and FPU. In addition, it incurs degradation in the hold slack of the I-EXU. When useful optimization is used while considering the impact of asymmetric aging, it improves the setup WNS and reduces the number of violations since it delays the capture edge. However, useful skew further degrades the hold slack and introduces hold violation in the I-EXU. When using useful skew optimization and considering asymmetric aging in conjunction with the asymmetry between net delay and cell delay, it introduces a significant number of violations for both setup and hold. In the case of setup analysis, the I-EXU WNS is reduced from -128ps to -181ps, while the FPU WNS is reduced from -237ps to -337ps. Additionally, the number of setup violations increases from 4 to 22 in the I-EXU and from 7042 to 9753 in the FPU. In hold analysis, the WNS remains the same for I-EXU, while the number of violations increases from 2 to 10. In the FPU, the hold WNS drops from +3.1 to -2 with a total of 7 violations.

The experimental results summarized in Table II indicate that asymmetric aging can introduce severe timing violations related to clock trees. Furthermore, a combination of useful clock skew optimization and asymmetry between the net delay and cell delay can even further intensify timing violations and degrade circuit reliability. It is worth noting that hold violations are considered to be more severe than setup violations, due to the fact that the latter can be mitigated by reducing the clock frequency, while hold violations have no mitigation.

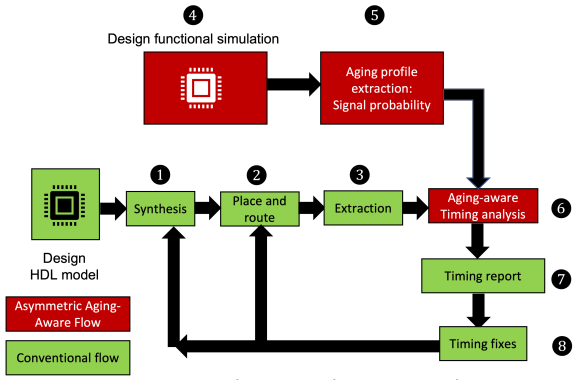

Fig. 5: Extended Physical Design Flow in Presence of Asymmetric Transistor Aging

Thereby, in presence of asymmetric transistor aging, the timing constraints should be extended as indicate by Equations 7 and 8, and extra timing margins should be added to setup and hold logical paths that exhibit marginal and negative slacks. Last, we summarize the full flow that has been used in our analysis in Figure 5. Our physical design flow extends the conventional flow, which includes 1) synthesis, 2) place-and-route, 3) extraction, 7) timing report generation, and 8) timing fixes, with new capabilities to incorporate the impact of asymmetric aging on timing. The new components added to the flow include:

4) Functional simulation of the design which is used to measure the idleness for a given workload.
5) Extraction of the aging profile in the form of signal prbability.
6) Aging-aware timing analysis which incorporate aging-aware libraries that are derated based on their corresponding signal probability.

In summary, by incorporating these new capabilities into our physical design flow, we can effectively address the challenges posed by asymmetric aging, ensuring improved timing analysis and design reliability as demonstrated on for GPGPU processing elements.

V. Conclusions

This study has investigated the impact of asymmetric aging on clock tree design considerations. Clock trees are vital circuit resources responsible for distributing a balanced clock signal across the chip die, and their reliability is critical to overall IC operation. Our study has revealed that clock trees are highly susceptible to reliability issues arising from asymmetric aging. While previous studies primarily focused on the effects of asymmetric aging on gated clocks, our work expands the scope of such investigations, highlighting the significance of factors like useful skews and the asymmetry between net and cell delays in causing substantial timing violations in clock trees, potentially leading to IC failure. Utilizing General Purpose Graphic Units (GPGPUs) as a case study, our experimental analysis has demonstrated the timing violations resulting from the consideration of asymmetric transistor aging. In response to the challenges posed by asymmetric aging, we have introduced an aging-aware design flow, which encompasses new extensions to the timing constraints. These extensions enforce adding extra timing margins to setup and hold logical paths that exhibit marginal and negative slacks. Understanding the impact of asymmetric aging on clock tree design is crucial for ensuring the reliability and performance of ICs. By incorporating aging-aware design flow and timing constraints, potential timing violations can be mitigated, contributing to enhanced overall IC operation and longevity.

References

[1] 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.
[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. https://doi.org/10.1016/j.microrel.2006.10.012
[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] 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.
[5] A. Ricketts, J. Singh., K. Ramakrishnan, N. Vijaykrishnan, and D. K. Pradhan, “Investigating the impact of NBTI on different power saving cache strategies,” in Proc. DATE, Mar. 2010, pp. 592–597.
[6] 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.
[7] 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.
[8] 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.
[9] F. Gabbay and A. Mendelson, “Asymmetric aging effect on modern microprocessors,” Microelectronics Reliability, vol. 119, pp. 71–81, 2021, https://doi.org/10.1016/j.microrel.2021.114090.
[10] S. Arasu, M. Nourani, F. Cano, J. M. Carulli and V. Reddy, ”Asymmetric aging of clock networks in power efficient designs,” Fifteenth International Symposium on Quality Electronic Design, Santa Clara, CA, USA, 2014, pp. 484-489, doi: 10.1109/ISQED.2014.6783365.
[11] M. Khairy, Z. Shen, T. M. Aamodt, T. G. Rogers, ”Accel-Sim: An Extensible Simulation Framework for Validated GPU Modeling”, In proceedings of the 47th IEEE/ACM International Symposium on Computer Architecture (ISCA), May 29 - June 3, 2020.
[12] A.A. Awan, H. Subramoni, and D.K. Panda. An In-depth Performance Characterization of CPU- and GPU-based DNN Training on Modern Architectures. Proceedings of the Machine Learning on HPC Environments, 2017.
[13] Component Technical Committee Automotive Electronics Council. Failure mechanism based stress test qualification for integrated circuit. AEC – Q100 – REV-G standard.
[14] J. Huang, J. Chai, and S. Cho, “Deep learning in finance and banking: A literature review and classification,” Frontiers of Business Research in China, vol. 14, 2020, https://doi.org/10.1186/s11782-020-00082-6.
[15] M. Macas, C. Wu, and W. Fuertes, “A survey on deep learning for cybersecurity: Progress, challenges, and opportunities,” Computer Networks, vol. 212, 2016, https://doi.org/10.1016/j.comnet.2022.109032.
[16] Bush, J., TaheriNejad, N., Willegger, E., Wojcik, M., Kessler, M., Blatnik, J., Daktylidis, I., Ferdigg, J. and Haslauer, D., 2021. Nyuzi: An Open Source GPGPU for Graphics, Enhanced with OpenCL Compiler for Calculations. In IEEE Design, Automation & Test in Europe (p. 1). IEEE.