Wisdom of Aging: New STA Solution for Non-uniform Aging

Chirayu Amin
Cadence Design Systems
Hillsboro, USA
camin@cadence.com

Shiva Raja
Cadence Design Systems
Burlington, USA
sraja@cadence.com

Igor Keller
Cadence Design Systems
San Jose, USA
ikeller@cadence.com

***Abstract* — In this paper, we introduce a comprehensive sign-off solution for aging-aware timing and optimization for digital design. Our solution is implemented in Cadence Liberate characterization flow and Tempus timing analysis flow. It is designed to accurately account for device degradation and recovery due to variation in supply voltage, temperature, and logic conditions across the chip. A unique strength of our solution is support for non-uniformity of aging stress conditions in time and across instances. We present data on a real design implementation showing that our solution is accurate within +/-3% of SPICE path delays.**

Keywords— aging aware STA, cell library characterization, timing analysis, optimization, device reliability, aging context

I. INTRODUCTION

Semiconductor devices age over time, just like their makers. Chips must continue performing at their designated frequencies or risk being unreliable and, hence, irrelevant. This is especially true for market segments such as automotive, industrial applications and IoTs that demand performance over many years.

Reliability of semiconductor devices is a well-studied topic spanning multiple decades of work to understand underlying physical phenomenon that lead to device degradation [1]. Technology scaling and introduction of new materials and manufacturing processes is producing a downward trend in expected lifetime of products [2][3]. As our experiments on 5nm CPU design show, aging of devices can lead to >5% degradation in top critical path delays, see Fig. 1. Therefore, it is imperative for a chip design company to properly account for the impact of device aging to guarantee performance for a set lifetime. Aging-aware timing and optimization is becoming a widely requested technology by design companies.

Aging of a device depends on several important instance-based factors as illustrated in Fig. 2. These factors, which we collectively refer to as *aging stress condition*, include local supply voltage, temperature, calendar age, and effective time under stress – which is mainly governed by logic conditions. We define a specific cell instance's value for the aging stress condition as the *aging context* of that instance. Different instances of the same cell, e.g. buffer in the figure, are often subject to different aging contexts in the same design. Consequently, their timing degradations due to aging are also different.

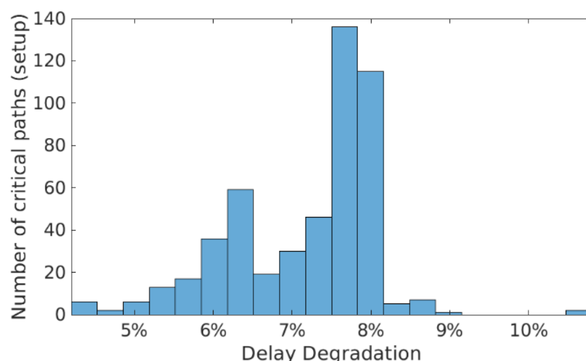

Fig. 1. Path delay degradation due to bias temperature instability (BTI) for top 500 paths on a CPU core design implemented in 5nm technology.

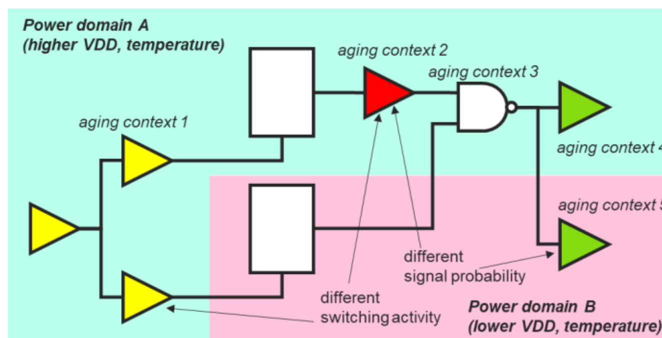

Fig. 2. Aged timing is a function of instance-based *aging context* comprising of voltage, temperature, age, switching activity, and signal probability.

Modern digital design tools such as optimizers and sign-off timers do not handle well the complexity of different aging contexts. Additionally, usage profiles for vastly different market segments such as servers, mobile communications, IoTs, automotive, aerospace, etc. lead to many different operating voltages, temperatures, and age. As an example, server market segment may demand stressing a chip at high voltage and temperature followed by timing at a lower operating voltage and temperature. On the other hand, mobile market may demand multiple segments of lower voltage and temperature stress. This in turn leads to several different aging stress conditions for the

same design. Often, such usage profiles and aging contexts are not known or predictable during cell library characterization, which is typically handled by a separate company such as foundry supplying the cell library or by a separate team within an integrated design and manufacturing company.

Given the requirements and constraints mentioned above, the only choices left to a design team when accounting for reliability in digital optimization and timing are: a) to spend extensive manhours and compute power to create design methodology and guard-banding strategies, or b) to spend extensive compute power to characterize multiple sets of cell libraries for different usage profiles. The first strategy of guard-banding can lead to overly pessimistic design since delay degradation is dependent on aging context, as is evident from the spread of critical path delays shown in Fig. 1. A worst-case guard-band of 10% would be overly pessimistic for paths that only degrade by 5%. The second strategy of using multiple fixed-stress cell libraries is too expensive and cannot cover various aging contexts. A fixed-stress, being based on a uniform degradation assumption, grossly mis-predicts timing no matter whether a cell is instantiated in a low power (i.e. lower temperature and/or supply) or a higher power domain. A fixed-stress library cannot handle various logic contexts in which a cell is instantiated.

This paper introduces a comprehensive sign-off solution for aging-aware timing and optimization for digital designs. Our solution is designed from the ground up to:

- Provide aging library model supporting a wide range of aging stress conditions
- Handle arbitrary target usage stress profiles (segments of supply voltage, temperature, and age) during timing analysis
- Support dependency of timing degradation on aging context
- Support non-inform aging where stress condition is different than operating condition
- Have small additional cost for library characterization and STA.

The ability to handle arbitrary stress profiles during STA and the support for timing degradation's dependence on local aging context are the unique strengths of our solution.

The rest of the paper is organized as follows. In section II, we discuss semiconductor device reliability and show how aging stress conditions impact device degradation and eventually impact timing. In section III, we will present data to highlight inadequacies of existing approaches for aging-aware design. We introduce our solution in section IV and present results in section V. We end the paper with next steps and summary in section VI.

II. Background on Semiconductor Device Reliablity and its Impact on Timing

A. Semiconductor Device Reliability

Semiconductor device reliability physics is an extensive topic and not the main topic of this paper. Here, we will mainly only discuss the material as it pertains to impact on timing analysis of digital designs. We refer readers interested in a detailed study to references and other previously published literature on the topic.

There are two main physical phenomena that impact device degradation: (i) bias temperature instability (BTI) and (ii) hot carrier injection (HCI).

The first effect, BTI, occurs when unsatisfied silicon bonds at the oxide interface with silicon are passivated by annealing in hydrogen, forming Si-H bonds. Those hydrogen atoms dissociate easily and create traps when a device experiences constant field stress such as an NMOS gate terminal set to logic high (VDD) or a PMOS gate terminal set to logic low (VSS). Such scenarios are common in digital circuits due to mostly constant signals. The biasing of gate terminals of MOSFETs relative to their other terminals creates large electric fields. The rate of generation of traps is accelerated with higher supply voltage (higher fields) and higher temperature. It is these traps that lead to increase in threshold voltage (*Vt*), decrease in drain current, and decrease in channel mobility. In new technology nodes with high-K dielectrics and metal gates, PMOS devices with negative gate bias experience more degradation relative to NMOS leading to relatively stronger degradation of rise delays compared to fall delays (see Fig. 3).

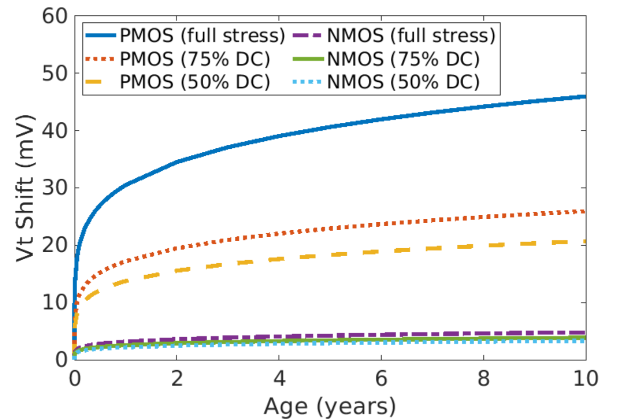

Fig. 3. PMOS devices experience much more degradation (*Vt* shift due to BTI) than NMOS devices for same stress condition (VDD: 1.0V, Temp: 125C). Aging degradation reduces when recovery is enabled with duty cycle $< 100\%$.

Some of the BTI degradation is reversible, which is referred to as the *recovery effect*. Recovery happens when a device is not under stress for some time, e.g. when NMOS gate input is set to logic low or PMOS gate input is set to logic high. Fig. 3 shows overall impact of recovery on *Vt* shift due to varying percentage of overall time spent in stress vs. recovery mode. Duty cycle value of 100% (full stress) represents a constant gate terminal input at 0V for PMOS and duty cycle value 75% represents a switching input with gate terminal at 0V for 75% of the time, i.e. 25% of the time spent in recovery with PMOS gate input equal to VDD. For NMOS, duty cycle 75% represents a switching input with gate being set to VDD for 75% of the time and 0V for 25% of the time.

HCI degradation typically occurs when signals are transitioning in digital circuits. Energetic carriers in active channel can lead to impact ionization within the substrate or they can escape into the neighboring oxide material. These injected carriers alter device characteristics such as threshold voltage and transconductance, causing *Ids* degradation as shown in Fig. 4.

Of the two physical mechanisms of aging, BTI has a much stronger effect on timing than HCI (see Fig. 5). However, unlike with BTI, devices cannot recover from HCI induced degradation. Since HCI is mainly dependent on transitions, HCI is of a higher concern for devices driven by signals that switch at high frequency such as high-speed clocks.

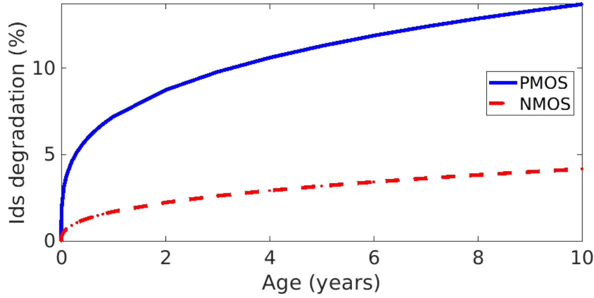

Fig. 4. *Ids* current degradation due to HCI (gate input period 5ns, VDD: 1.0V, Temp: 125C).

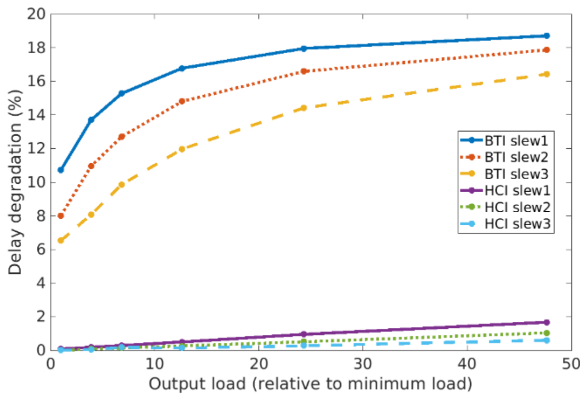

Fig. 5. Rise delay degradation due to BTI and HCI for an inverter in a 5nm technology.

B. Factors Influencing Device Degradation

Device degradation mainly depends on three factors: supply voltage, temperature, and effective time under stress which is mainly governed by logic conditions. For PMOS and NMOS devices in CMOS digital logic, effective time under stress for BTI degradation is mainly the time when gate input is set to VSS and VDD respectively. HCI degradation on the other hand is dependent on a more complex relationship between transient gate, drain, and source voltages. Those voltages are determined by input slew and output loading in the context of digital circuits.

BTI induced *Vt* shift is typically exponential with respect to supply voltage and temperature but follows a power law with respect to age time as shown in Fig. 6. Our experiments show that *Vt* degradation is stronger for low-*Vt* devices than for high-*Vt* devices. We also observe a complex behavior of degradation rate on supply voltage, temperature, and age for HCI effect as shown in Fig. 7.

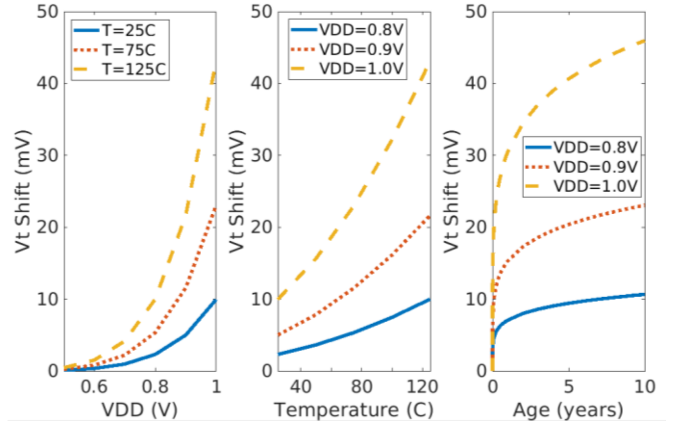

Fig. 6. BTI induced *Vt* shift's dependence on VDD, temperature, and age.

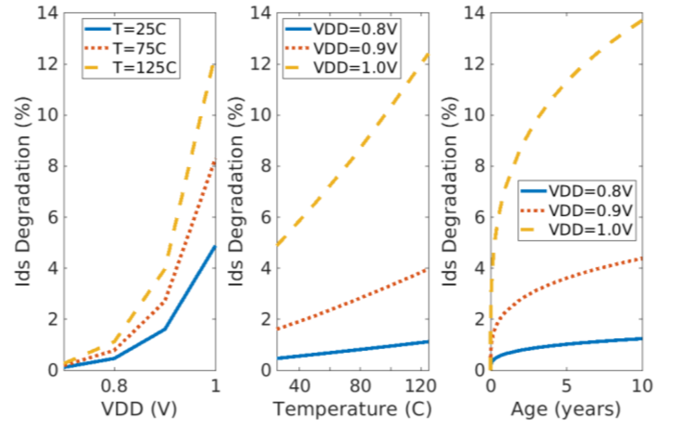

Fig. 7. HCI induced *Ids* degradation's dependence on VDD, temperature, and age.

C. Impact on Timing

In digital circuits, changes of *Vt* and *Ids* of devices within cells caused by BTI and HCI have strong impact on the cell's delay and consequently circuit's timing. For rising outputs of single stage cells, degradation of devices on the driving PMOS stack causes delays and slews to degrade. In contrast, degradation of devices on the NMOS stack causes rising delays and slews to improve. The reverse is true for falling outputs. Overall change in delay and slew for a timing arc depends on relative contributions of PMOS vs. NMOS stacks. Since PMOS devices degrade significantly more than NMOS devices as shown in previous sub-section, we observed that rise delays increase much more than fall delays as shown in Fig. 8. For cells with even number of CMOS stages such as buffers, we can expect that both rise and fall arc delays degrade significantly, with PMOS devices contributing the most to overall timing impact.

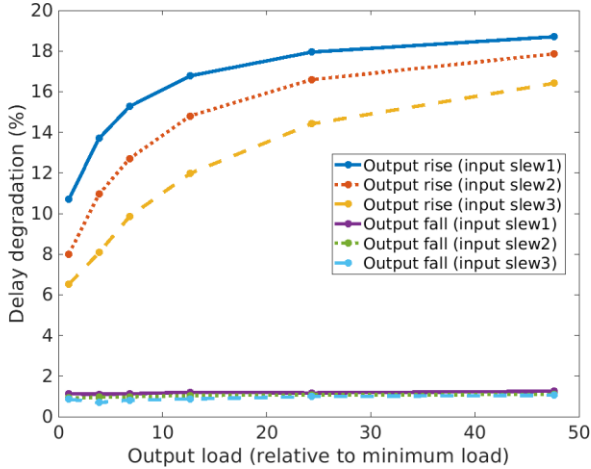

Fig. 8. Inverter delay degradation due to BTI. Rise delays degrade much more than fall delays since PMOS devices degrade much more than NMOS devices.

As we mentioned before, delay of an arc depends on amount of stress applied to devices on the arc. For multi-input cells, a device may belong to multiple arcs and hence is influenced by activity at the corresponding input pins. For example, aging effect on A→Y rising output arc for a 2-input AND depends on activities at both inputs A and B. In other words, aging effect on a cell is a function of activity of its fan-in cone making graph-based STA necessary for accurate modeling of aging effects on timing. In addition, the same cell instantiated in different power domains will experience different stresses due to voltage and temperature differences (see Fig. 2). Therefore, overall calculation of timing degradation must consider power domain and fan-in logic of the cell instance.

III. Limitations of Existing Approaches

Aging-aware digital timing analysis and optimization is a complex topic, dependent on awareness of aging context. Previous approaches to address the problem include [5]-[13]. In [5], an approach to handle switching probability in library design is presented but it is not scalable for handling full range of activity or aging context without leading to exponential increase in library size. The works presented in [6]-[10] rely on modeling based on analysis of selected circuits or pre-determined workloads and therefore require significant effort to create models appropriate for different design types and aging contexts. Moreover, as we described earlier in the introduction, workloads are better determined by design teams and are typically not available during commercial library characterization. Approaches [11], [12], and [13] depend on circuit simulation. Even with fast-SPICE simulators, they are prohibitively expensive from a runtime perspective of timing analysis of designs with millions of instances.

None of the existing approaches provide *aging context* dependent comprehensive digital design solution ranging from library characterization to sign-off timing. Design teams have been relying on methods mentioned in the introduction such as guard-banding strategies or using fixed-stress libraries, both of which suffer from inaccuracies and resourcing problems.

For example, if we apply a worst-case guard-band to a design to cover top 500 critical paths for setup timing, we must then pick a guard-band of ~10% delay degradation. This results in path delay errors >5% as shown in Fig. 9.

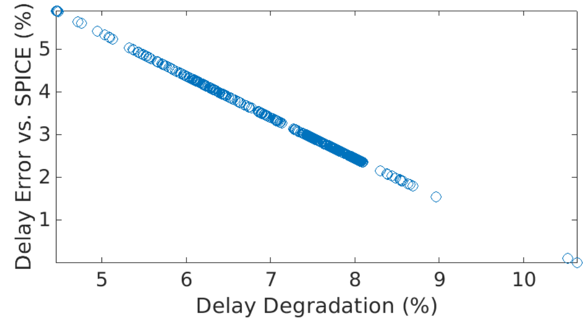

Fig. 9. Path delay errors in a guard-banding approach for aging-aware timing.

If we use fixed-stress library approach, where aging stress condition parameters such as supply voltage, temperature, and age are selected during library characterization, we end up with a library that provides only uniform aging regardless of the aging context of the cell instance. A design methodology team with more resourcing on hand may be able to develop a strategy of using different fixed-stress libraries depending on context, but such solution quickly becomes intractable for designs with tens of power domains and millions of instances. Moreover, such a library cannot easily account for logical portion of aging context. Today, characterization must assume a fixed signal probability (typically 0.5) for input signals. As a result, stress used in such characterization would differ significantly from actual stress in a real design. Consequently, such approach can result in large errors as shown in Fig. 10.

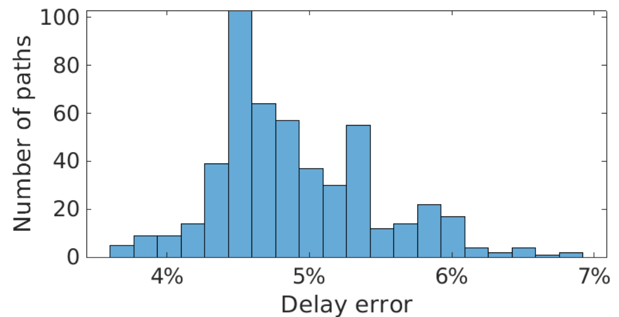

Fig. 10. Path delay pessimism of fixed-stress aging methodology as compared to actual (instance specific) stress in a real design.

IV. Comprehensive Solution for Aging-Aware Design

We have implemented a comprehensive sign-off solution for aging-aware timing analysis (including characterization and STA) and optimization for digital design in Cadence Liberate characterization flow and Tempus timing analysis flow.

Our cell library characterization flow (Fig. 11) captures aging-aware timing degradation across a wide range of stress voltages, temperatures, and ages with small incremental runtime cost and library footprint. Unlike previous aging-aware characterization approaches, our patented solution generates a generic *age.lib* capable of providing values of delays, slews, and

constraints for all possible aging contexts the cell may experience in a real design. Our solution presently supports Cadence Spectre® [14] compatible SPICE device models in combination with Si2 OMI [15], Cadence RelXpert URI [16], or TSMC TMI [17] SPICE reliability models. Our solution's stress and recovery models are in accordance with behavior of respective foundry's SPICE reliability models. Our flow provides additional features such as aging-aware cell library validation to provide confidence in accuracy of results.

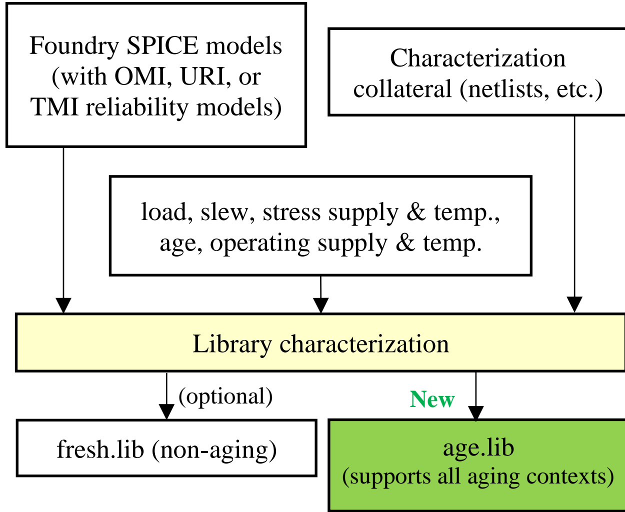

Fig. 11. Aging-aware cell library characterization flow. Aging data is output in age.lib that supports all aging contexts.

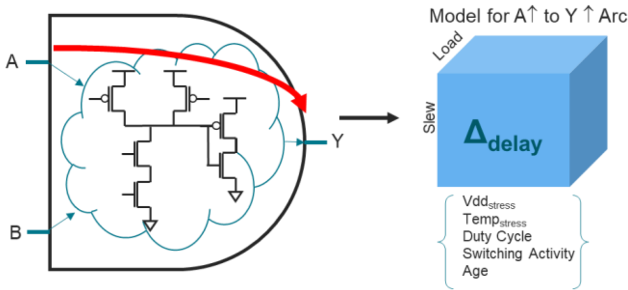

Fig. 12. Example of library aging model for A to Y arc of 2-input *and*.

Inputs to characterization flow consist of ranges of stress VDD, stress temperature, and age as well as relevant SPICE device models and SPICE reliability models. SPICE reliability model parameters, such as failure rate (ppm) which is typically market segment dependent, are also allowed to be set by characterization flow. For every arc of a cell, characterization flow generates aging library models which facilitate calculation of all timing quantities (such as arc delays and slews) for specific stress VDD, stress temperature, age, input waveform, and output load combinations as illustrated in Fig. 12. Moreover, the model also contains information to handle the impact of input pin duty cycles and switching activity on arc timing.

Our timing analysis and optimization solution (Fig. 13) consumes the aging libraries produced by the characterization step along with user-specified stress profile(s). Modification to existing timing analysis setup is minor: (i) provide aging libraries, and (ii) specify stress profile for design using special TCL command.

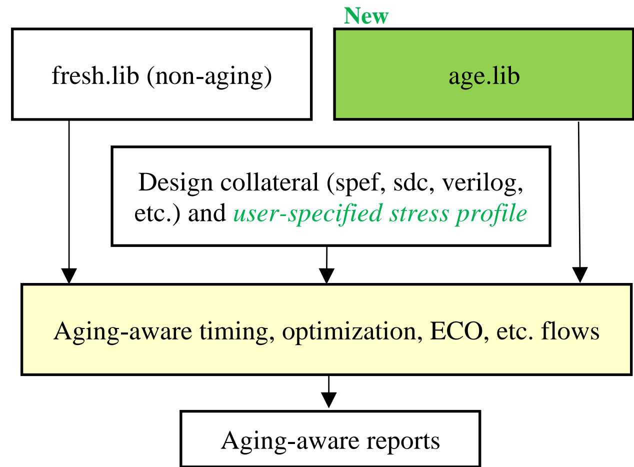

Fig. 13. Aging-aware timing analysis flow.

The user-specified stress profile consists of one or more segments of *supply voltage*, *temperature*, and *age* as defined by the design teams and based on target market requirements. As the example in Fig. 14 illustrates, server market segment may demand stressing a chip at high voltage and temperature followed by timing at a lower operating voltage and temperature. As approximate examples, the relevant TCL commands for specifying the two stress profiles would be:

Server: `set_aging_profile {VDDA=1.1,VDDB=1.05,T=125,age=10}`

Mobile: `set_aging_profile {VDDTOP=0.9,T=75,age=2} {VDDTOP=0.8, T=75, age=4}`

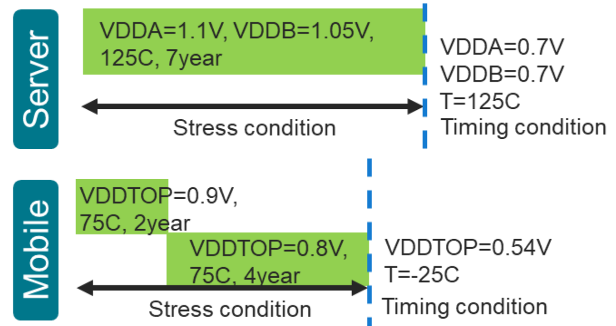

Fig. 14. *Aging stress conditions* depend on usage profiles for different markets.

The tool automatically determines aging context for all instances using the user-specified stress profile and design netlist. First, the stress voltages specified in stress profile are translated into cell instance stress voltages using power net connectivity information. Then aging context's logic information such as duty cycle and switching activity is computed for input pins for all instances in the design. Along with instance-specific stress voltages and stress temperature,

local information such as input pin duty cycles, switching activity, input slew, and output load are used to retrieve context-specific aging sensitivities of delays and other timing quantities from the library model corresponding to the timing arc. Aging sensitivities are then used to degrade timing of the specific arcs on the cell instances.

The presented aging aware STA solution is equally applicable to standard cells and MACROs in both GBA and PBA. For MACROs that require transistor-level timing and for non-digital portions of the design such as analog IP blocks, we recommend that a proper margining approach based on circuit simulation or other applicable methodology is used for building block-level timing models with aging-aware delays. For hierarchical timing analysis, our solution automatically builds aging-aware block-level models for stress profile that was used in timing analysis of the block. If models for different stress profiles are necessary, then we recommend that a model is built for each such profile.

Our solution, by design, also supports non-uniform aging where stress condition may be different than operating condition at which timing analysis is run. Such a use-case is common for many designs especially with dynamic voltage and frequency scaling schemes. In such scenarios, the design typically operates in a high-performance mode and therefore devices degrade rapidly at higher supplies and temperatures. However, the part must still be guaranteed to function in a low-power mode for lighter workloads, necessitating timing analysis at a lower supply voltage and temperature.

The unique strengths of our solution to handle arbitrary stress profiles during STA and support for timing degradation's dependence on local aging context allow for extensive design space exploration. It is often not clear which stress profile is the worst for timing. Our solution employs a comprehensive exploration of parameter space. With our aging solution and efficient infrastructure to run multiple views concurrently, users are able to address potential optimism and identify failure scenarios by running multiple profiles efficiently.

We have also implemented support for aging-aware SPICE deck generation so that aging-aware timing analysis can be compared easily with accurate SPICE analysis on a path-by-path basis in PBA mode, providing confidence in the accuracy of timing results.

V. Results

In this section, we will present results of aging-aware timing analysis in comparison with SPICE. STA path delay accuracy vs. SPICE is typically influenced by many factors other than aging such as nonlinearity of drivers and receivers, inaccuracies associated with waveform models and driver models, as well as inaccuracies associated with interconnect analysis. We must remove the impact of those inaccuracies from the impact of aging in order to do a proper comparison between aging-aware STA and SPICE. Therefore, we propose the following error metric when comparing aging-aware path delays from STA vs. SPICE:

$$Error\ \% = 100 * \frac{\Delta Delay_{STA} - \Delta Delay_{SPICE}}{Delay_{SPICE,Fresh}}$$

where $\Delta Delay_{STA} = Delay_{STA,Aging} - Delay_{STA,Fresh}$ and $\Delta Delay_{SPICE} = Delay_{SPICE,Aging} - Delay_{SPICE,Fresh}$.

We are thus comparing the change in delay due to aging between STA and SPICE as opposed to comparing total (fresh plus aging delta) delay from STA and SPICE. We normalize the results by fresh delay to provide a better referencing of error from a design perspective. Aging data from SPICE is collected by running full reliability simulation using Cadence Spectre®, including stress simulation to degrade devices and a subsequent post-degradation aging simulation to capture impact on timing.

We characterized a library with ~750 cells covering a broad range of combinational and sequential cells of various drive strengths and *Vt* types. Aging data is captured using models like the one shown in Fig. 12. Cost of aging characterizing is about 15-30% of regular timing library characterization and we continue to improve efficiency. In order to validate accuracy of cell library models, we collected aging-aware delay errors relative to SPICE simulation at a given aging stress condition for all the library cell arcs. Results of cell-level validation for BTI degradation are shown in Fig. 15. The data demonstrates that our characterization flow provides good cell-level accuracy.

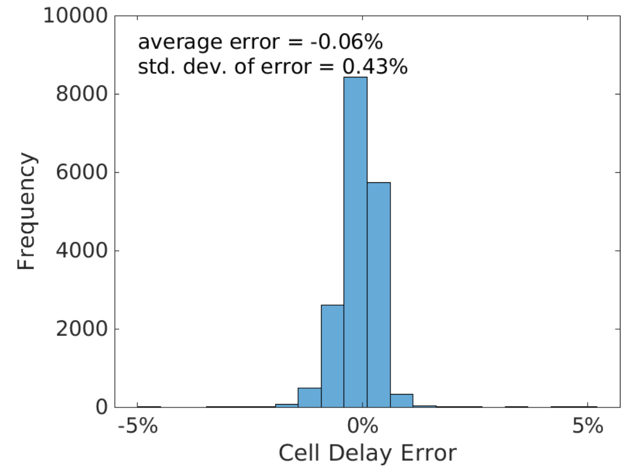

Fig. 15. Aging-aware cell library delay errors.

Using above library, we ran aging-aware STA on a CPU core design implemented in a 5nm technology and collected path delay errors relative to SPICE for critical paths. Our solution predicts aging degradation accurately as shown in Fig. 16 for above discussed error metric and in Fig. 17 for aged path delay from STA vs. SPICE. Errors are computed by comparing STA vs. SPICE in PBA mode. Incremental cost of runtime and memory for aging-aware STA is generally less than 5%.

Fig. 18 shows change in slack due to aging. Data path delays degraded more than clock paths and hence slack on setup critical paths has reduced relative to fresh design. Hold paths on the other hand, experience an improvement in slack on average due to relatively larger aging-induced delay increase on data paths as compared with clock paths.

Fig. 19 shows the impact of BTI recovery on path delay degradation. Data shows that it is important to model BTI recovery to recover pessimism related to aging. Moreover,

notice how recovery amount is not similar across all paths. Our solution models that dependence of recovery on aging context.

Fig. 20 shows the impact of stress profile differences on path delay degradation. Stress profile A represents 10 years of stress at 1.1V and 125C. Stress profile B represents a two-segment stress with 2 years of aging at 1.1V and 125C followed by additional 8 years of aging at 1.0V and 125C. As expected, and confirmed with data, delays degrade less with the profile B. This analysis shows the power of our solution in handling different usage profiles with ease during timing analysis.

We also applied our solution to timing ECO and compared it with a fixed-stress library approach. Since fixed-stress library characterization cannot handle the non-uniform aging context, we artificially constrained our solution to have uniform aging context for all instances. This avoids comparison discrepancies related to mismatching aging contexts between our generic solution and the old fixed-stress library approach. Results using our solution show good match in worst negative slack (WNS) and total negative slack (TNS) with solution from ECO with fixed-stress library. WNS and TNS differences were 1.2% and 1.6% respectively.

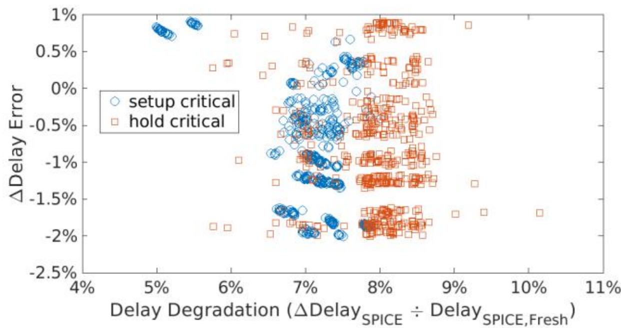

Fig. 16. BTI aging-aware path ΔDelay errors vs. delay degradation for a CPU core design in 5nm (PBA mode). *Stress*: (1.1V, 125C, 10 years), *Timing*: (1.1V, 125C).

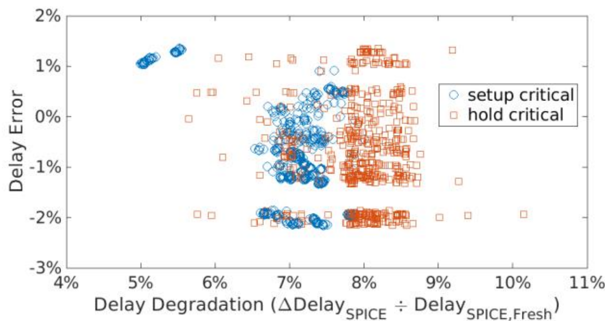

Fig. 17. BTI aging-aware path delay errors vs. delay degradation for a CPU core design in 5nm (PBA mode). *Stress*: (1.1V, 125C, 10 years), *Timing*: (1.1V, 125C).

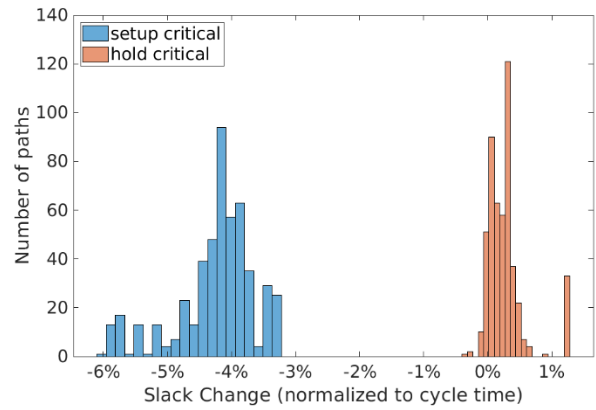

Fig. 18. Change in slack due to aging.

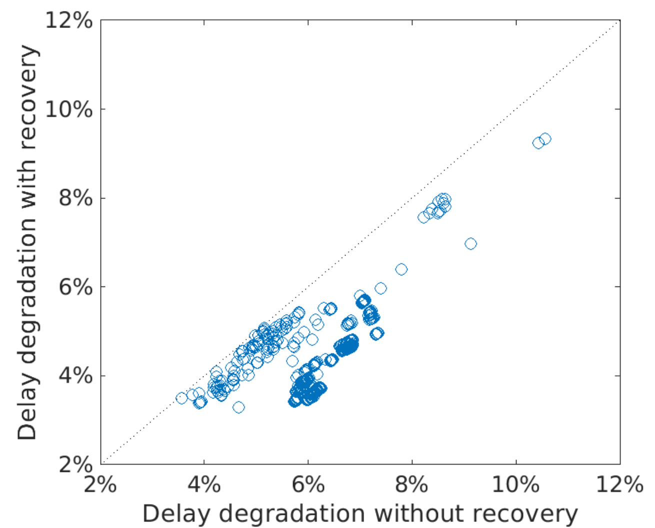

Fig. 19. Impact of BTI recovery on path delay degradation.

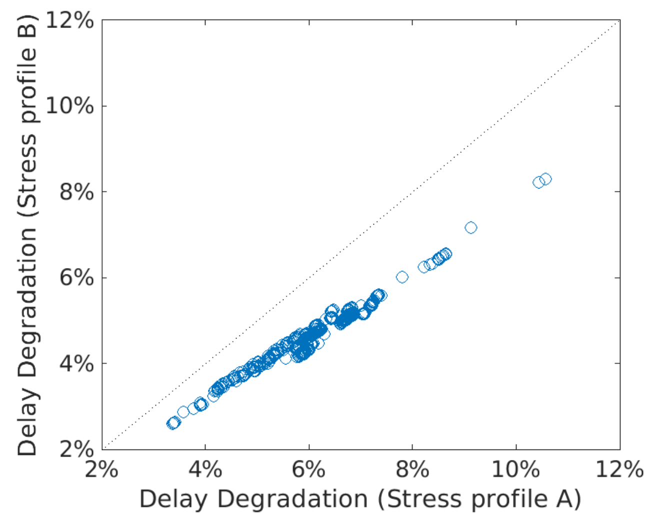

Fig. 20. Impact of stress profile differences on path delay degradation. *Stress profile A*: (1.1V, 125C, 10 years), *Stress profile B*: (1.1V, 125C, 2 years) followed by (1.0V, 125C, 8 years).

VI. Conclusion

In this paper we have presented data showing the importance of aging-aware timing closure for digital designs. We presented a comprehensive sign-off solution for aging-aware timing and optimization implemented in Cadence Liberate characterization flow and Tempus timing analysis flow. Unlike previous approaches, our solution is designed to accurately account for device degradation and recovery due to variation in supply voltage, temperature, and logic conditions across the chip. A unique strength of our solution is support for non-uniformity of aging stress conditions in time and across instances. We demonstrated that our solution is accurate within +/-3% of SPICE in terms of path delay. We are continuing to further improve our solution by implementing full support for HCI degradation.

Acknowledgment

The authors would like to thank Sandhya Karanam, Praveen Patel, Swamy Lokanadham, and Sant Saran for their support to validate and qualify the solution.

References

[1] J. W. McPherson, Reliability Physics and Engineering, Springer International Publishing, 2019.

[2] J. W. McPherson, "Reliability trends with advanced CMOS scaling and the implications for design," IEEE Custom Integrated Circuits Conference (CICC), pp. 405-412, 2007.

[3] V. Huard, F. Cacho, X. Federspiel, W. Arfaoui, M. Saliva, and D. Angot, "Technology scaling and reliability: challenges and opportunities," IEEE International Electron Devices Meeting (IEDM), pp. 551-556, 2015.

[4] S. Ramey et al., "Frequency and recovery effects in high-K BTI degradation," IEEE International Reliability Physics Symposium (IRPS), pp. 1023-1027, 2009.

[5] S. Kiamehr, F. Firouzi, M. Ebrahimi, and M. B. Tahoori, "Aging-aware standard cell library design," IEEE Design, Automation, & Test in Europe (DATE), pp. 1-4, 2014.

[6] K-C Wu and D. Marculescu, "Aging-aware timing analysis and optimization considering path sensitization," IEEE Design, Automation, & Test in Europe (DATE), 2011.

[7] E. Mintarno, V. Chandra, D. Pietromonaco, R. Aitken, and R. W. Dutton, "Workload dependent NBTI and PBTI analysis for a sub-45nm commercial microprocessor," IEEE International Reliability Physics Symposium (IRPS), pp. 3A.1.1-3A.1.6, 2013.

[8] M. Katoozi et al., "An age-aware library for reliability simulation of digital ICs," IEEE International Reliability Physics Symposium (IRPS), pp. 3A.3.1-3A.3.5, 2013.

[9] S. Karapetyan and U. Schlichtmann, "Integrating aging aware timing analysis into a commercial STA tool," IEEE VLSI Design, Automation and Test (VLSI-DAT), 2015.

[10] S. Kiamehr, F. Firouzi, and M. B. Tahoori, "Aging-aware timing analysis considering combined effects of NBTI and PBTI," IEEE Symposium on Quality Electronic Design (ISQED), 2013.

[11] S. Raja, J. Geada, Q. Shen, and N. Rethman, "Multiple personality disorder of cell models – a cure by transistor," TAU 2017.

[12] T. An, H. Cai, L. A. de Barros Naviner, "Simulation study of aging in CMOS binary adders," International Convention on Information and Communication Technology, Electronics and Microelectronis (MIPRO), 2014.

[13] E. Maricau and G. Gielen, "Efficient variability-aware NBTI and hot carrier circuit reliability analysis," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29. no. 12, pp. 1884-1893, Dec. 2010.

[14] Cadence Spectre® User Guide, https://support.cadence.com/apex/techpubDocViewerPage?path=spectreuser/spectreuser20.1/spectreuserTOC.html

[15] *Open Model Interface*, https://si2.org/2018/05/08/omi_model/

[16] Spectre® RelXpert Reliability Simulator User Guide, https://support.cadence.com/apex/techpubDocViewerPage?path=rxuser/rxuser20.1/rxuserTOC.html

[17] M. Jeng, C. Hsiao, K. Su and C. Lin, "Circuit reliability simulation using TMI2," IEEE Custom Integrated Circuits Conference (CICC), pp. 1-7, 2013.