Fast & Accurate Methodology for Aging Incorporation in Circuits using Adaptive Waveform Splitting (AWS)

S. Mishra, P. Weckx, J. Y. Lin, B. Kaczer, D. Linten, A. Spessot, F. Catthoor

IMEC vzw, Leuven 3001, Belgium
(+32-477253988, subrat.mishra@imec.be)

***Abstract*—A common approach to incorporate workload dependent aging in circuits is to use an effective stress time or so-called signal probability (SP) to calculate degradation under realistic workload scenarios. However, it turns out this approach is not fully physics-based and incurs erroneous estimation of degradation. Moreover, cycle-accurate (CA) simulations are computationally expensive. In this paper, a relatively fast yet accurate, adaptive waveform splitting (AWS) algorithm is proposed to enable fast calculation of workload-dependent device aging. The proposed algorithm has been adopted to perform aging estimation of large circuits under specific workload scenarios.**

Keywords*—Activity aware aging; Bias Temperature Instability (BTI); Circuit aging; CMOS reliability; Signal probability; Workload dependent aging.

I. Introduction

Aging related degradation in circuits due to physical mechanisms, like Bias Temperature Instability (BTI) and Hot Carrier Injection (HCI), depend on the workload profiles they are exposed to during operation. One way of incorporating aging in circuits is by introducing a fixed (worst case) delay de-rating factor as the margin so that timing violations never occur under all possible operating regimes. The major drawback of such a conservative approach is that blanket aging margins can no longer be tolerated, and the workload information is completely ignored. The other solution is to use an effective stress time or so-called signal probability (SP) while calculating degradation under realistic workload scenarios [1-3]. Though the later may be intuitive, it is not purely physics-based and incurs erroneous estimation of degradation, as will be discussed subsequently. Cycle-accurate (CA) simulations, on the other hand, are computationally expensive and hence, unsuitable for large designs.

To demonstrate this effect, Fig. 1 shows the schematic of three workload profiles applied at the gate input of a transistor under NBTI stress and the corresponding simulated degradation. BTI stress and relaxation simulations have been performed using analytical models based on CET maps [4] which are calibrated to a commercial 28nm technology. Case B and C in Fig. 1 differ in the sequence of applied active and sleep periods even though the total effective active and sleep durations are the same. The simulated degradation patterns as well as the end point degradation, however, for these two cases are not the same. This is contrary to the SP-based approach which predicts same degradation behavior since they have the same effective stress time or $t_{stress}/t_{recovery}$ ratio. This observation is in line with experiments [5] and clearly establishes the importance of preserving the activity sequence in evaluating degradation [6].

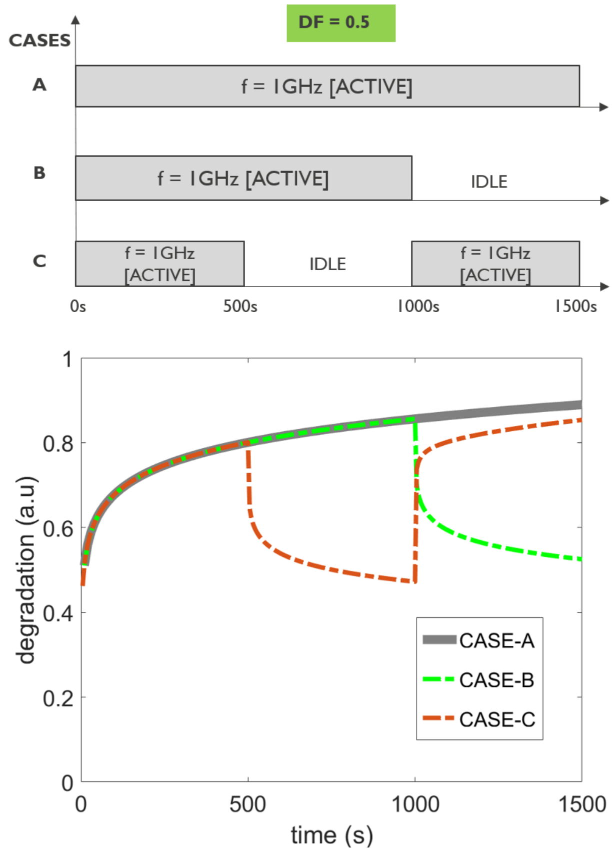

Fig. 1 Three workload profiles (cases - A, B, C) applied to a transistor under NBTI stress and the corresponding normalized degradation. During active mode, the frequency (=1GHz) and DF (=0.5) are fixed in all cases.

In this paper, we propose AWS - a relatively fast workload dependent aging analysis methodology which can tackle the complexities and run-time issues associated with considering "true" workload, at the same time preserving accuracy.

II. The AWS Approach

Fig. 2 presents another set of workload profiles which all have a fixed frequency f=1GHz for the entire stress duration. Cases B, C and D comprise of alternating segments of DF (0.2 and 0.8) but have a common mean DF=0.5 over the stress period; note case A also has fixed DF of 0.5. Simulation results indicate that the average DF can project the overall

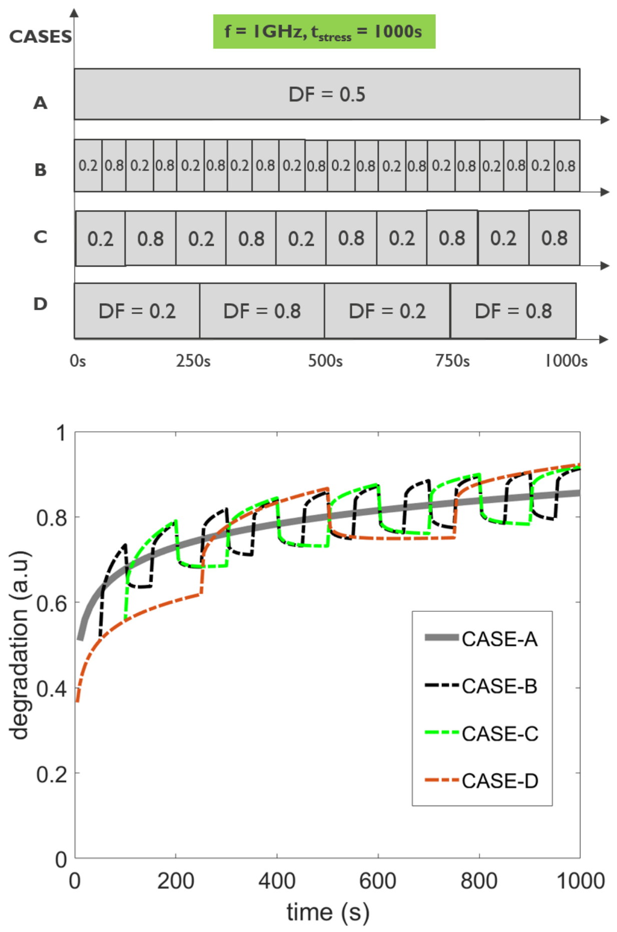

Fig. 2 Schematic of different workload cases to demonstrate DF averaging effect. The frequency and mean DF over the stress duration are fixed at 1GHz and 0.5 respectively, in all cases.

trajectory for degradation [5], though the transient behavior of actual degradation is different. It is important to note, the envelope of the fluctuations is around the mean DF case. It can be expected that, *with the duration of DF segments getting smaller in the scale of ~µs and ~ns, typical in real circuit conditions, the average behavior is a good estimator of actual degradation.* Based on the workload averaging effect demonstrated in Fig. 2, the following algorithm is adopted to perform real workload simulations: (i) split the waveform into segments which have high toggle rate and segments with lower toggle rate (ii) simulate the high toggling segments by using the AVERAGE workload within the segment and simulate the low toggling segments in CA manner. In this way, by adaptively splitting the stress waveform based on the toggling behavior, fewer simulations are needed by the aging-based compact models to reach to the end-point degradation as shown in Fig. 3. It is to be noted that the concept of workload splitting has been used in [7-9] which grouped consecutive signal regions into segments that feature similar f and DF numbers. In the previous works [7-9], the segments were based on the numerical value of the signal characteristics (f, DF) which suffer from limited compressibility, hence scalability issues for large designs. For example, if two cycles with different DFs (say, 0.3 and 0.55) are separated beyond the set accuracy limit (say, 0.1), those 2

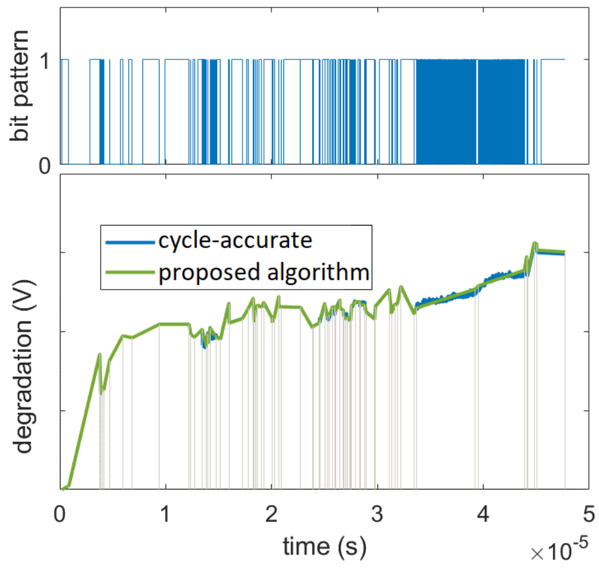

Fig. 3 Typical workload trace for a transistor in a 64-bit adder block obtained by simulating an application (fast Fourier transform from [10]) run on an ARM core using an instruction set simulator [12], followed by gate level simulations to get the binary waveform at the gate input of a transistor. AWS algorithm is applied to this real workload case. In this example, CA simulation requires 538 points vs. 80 in AWS case, i.e. 6.7x reduction in run time compared to CA.

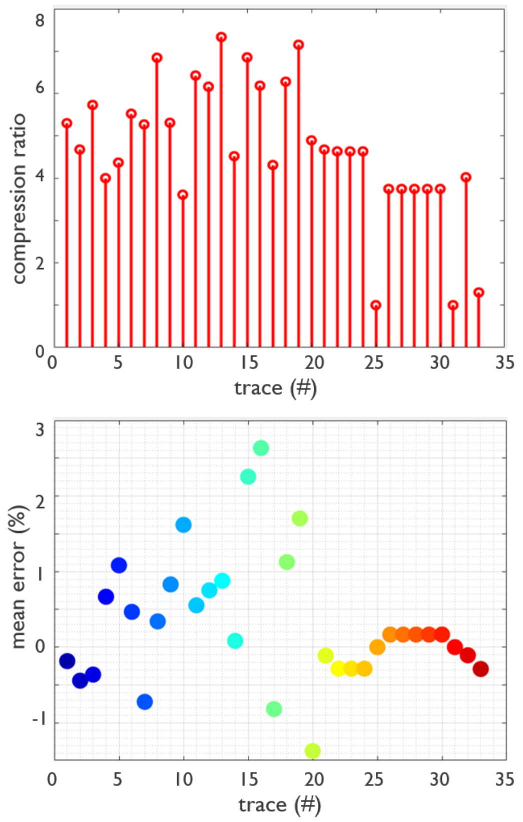

Fig. 4 Compression ratio (top) and % mean error (bottom) for a bunch of 33 traces in the 64-bit adder block. Average compression ratio per program is found to be 4.7 here with below +/- 3% error limit, keeping CAS as the reference.

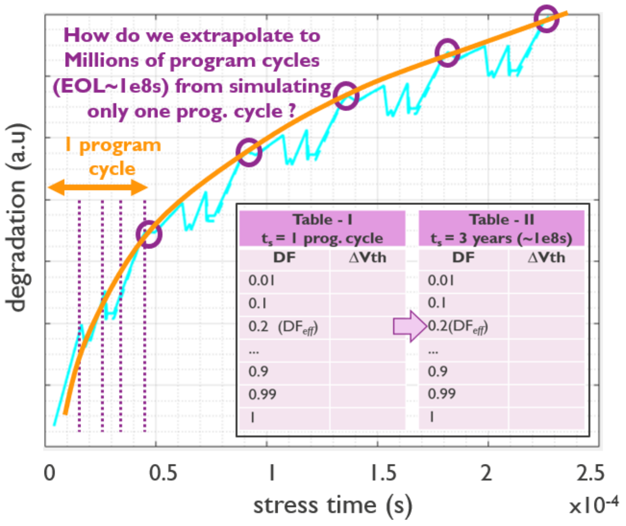

Fig. 5 Long-term extrapolation strategy for degradation by simulating only one or a few programs. An effective DF (DF_{eff}) is extracted from simulations using table-I and then mapped to long-term degradation with the DF_{eff} entry in table-II.

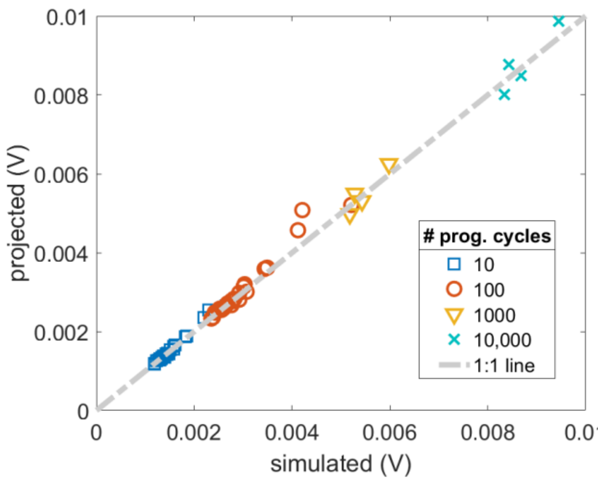

Fig. 6 Projection up-to 10k program cycles using LUT-based method shows good agreement with CA simulation results. Each symbol represents one transistor in the design.

cycles will have to be simulated in cycle-accurate manner. In this work, we improve run-time significantly by further grouping signal regions beyond the boundary of numerical values only. The efficiency and accuracy of the proposed approach has been demonstrated in Fig. 4 for an ensemble of 33 traces in an adder block inside an ARM core by running a benchmark program (fft1 from [10]) on an operating system (OS). Here, Compression Ratio is defined as the ratio of number of simulation points in original waveform for CA simulation and the number of simulation points in the compressed waveform. An average 4.7X improvement in run-time per program is observed with less than +/-3% error compared to CA simulation.

Many application programs have typical workload profiles which are repeated periodically over the lifetime (e.g. ASICs). We propose a long-term extrapolation method (see schematic in Fig. 5) under such scenarios by constructing look-up tables (LUTs) – one for short-term (one or a few program cycles) and another for long-term degradation (e.g. 3 years) as a function of uniform activity or DF. Then the simulated short-term degradation under real workload is interpolated to get the corresponding effective DF (DF_{eff}) in table-I which is mapped to long-term degradation with the corresponding DF_{eff} entry in table-II. Excellent projection accuracy can be seen in Fig. 6 which plots the cycle-accurate simulated degradation vs the proposed LUT based projection for up-to 10,000 program cycles.

III. Circuit Aging Analysis Flow

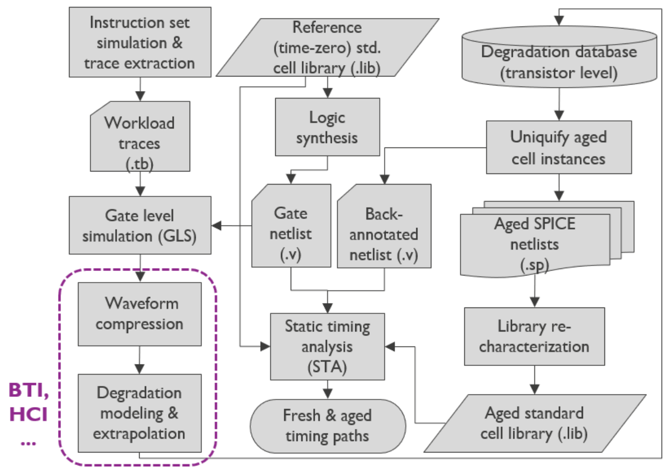
Instruction set simulation & trace extraction
Reference (time-zero) std. cell library (.lib)
Degradation database (transistor level)
Workload traces (.tb)
Logic synthesis
Uniquify aged cell instances
Gate level simulation (GLS)
Gate netlist (.v)
Back-annotated netlist (.v)
Aged SPICE netlists (.sp)
Waveform compression
Static timing analysis (STA)
Library re-characterization
BTI, HCI ...
Degradation modeling & extrapolation
Fresh & aged timing paths
Aged standard cell library (.lib)

Fig. 7 Proposed circuit level simulation flow for workload (WL) dependent aging analysis.

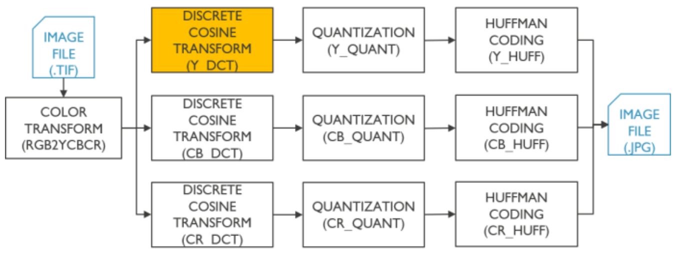

Fig. 8 Workload-dependent aging analysis flow is used to evaluate timing path degradation for a Discrete Cosine Transform (DCT) block inside a jpeg encoder design.

Based on the above aging simulation strategy, we propose in Fig. 7, an activity-aware reliability analysis flow for large systems under real workload scenarios. *This flow is instance-based and considers aging of individual transistor as per the workload it sees.* The design under test (DUT) for our analysis, in Fig. 8, is a Discrete Cosine Transform (DCT) block inside a jpeg encoder design [11]. The circuit accepts a 24-bit raw image file in RGB format to perform image compression. The design has ~ 140k gates when mapped using a commercial 28nm standard cell library. Aging analysis for over 30k NOR2 gates in the design shows the expected correlation between the degradation of top and bottom transistors, see Fig. 9.

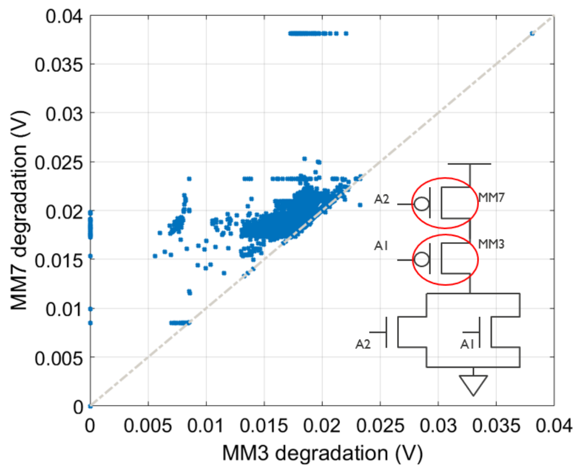

Fig. 9 Top and bottom transistor degradations inside the 2-input NOR gates (over 30k cells). Correlation is observed due to stacking effect.

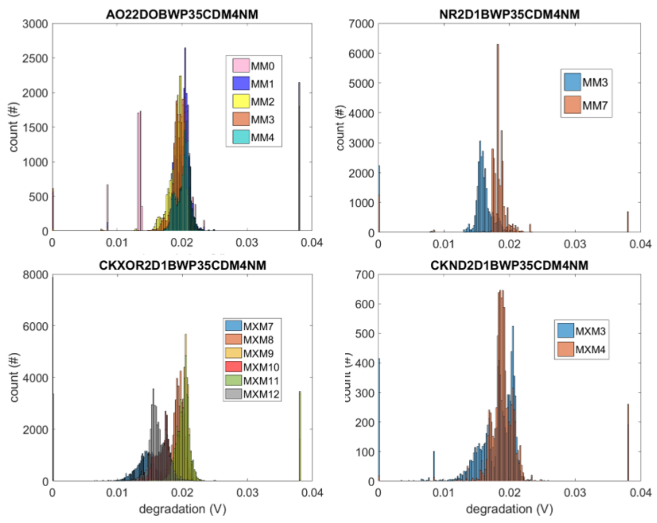

Fig. 10 Histograms for transistor degradation inside different cells. Most transistors degrade in a cluster and a small fraction of transistors is degraded to the maximum limit.

It can be noted that the degradation values for most transistors in all gates are localized and clustered around a certain mean value and only a small fraction of transistors degrade to the maximum limit. Histograms in Fig. 10 highlights this effect for different gates in the design. Based on the exact degradation numbers for transistors, standard cell characterization is performed to do timing analysis. For the number of cell characterization instances to not explode, post-aging cell instances are uniquified by grouping cells with similar combination of transistor degradation within. With uniform binning of 7 levels and maximum degradation of 38mV, the peak quantization error is within +/- 2.7mV for any transistor. Fig. 11 shows dramatic reduction in the number of cell characterization to be performed after binning. Based on the instance-based library, the gate level netlist is back-annotated with the uniquified aged instances.

Static timing analysis is performed to compare the path delay distribution after 3 years of aging under worst case (WC) i.e., a continuous DC stress condition vs. workload dependent (WL) aging scenario. A significant reduction in number of timing path violations compared to the worst-case aging can be seen in Fig. 12.

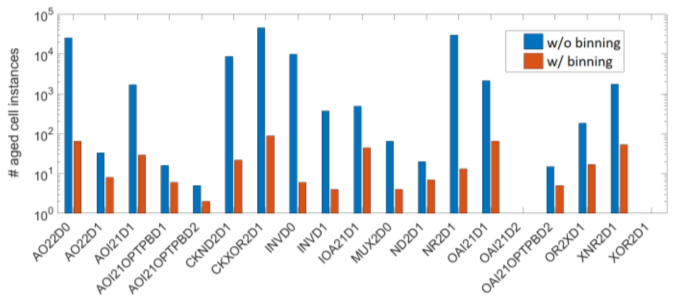

Fig. 11 Instance based library characterization for aged cells is performed by uniquifying the cells which have similar combination of transistor degradation inside the cell. Only the instances with unique bins are characterized resulting in orders of magnitude improvement in characterization efficiency.

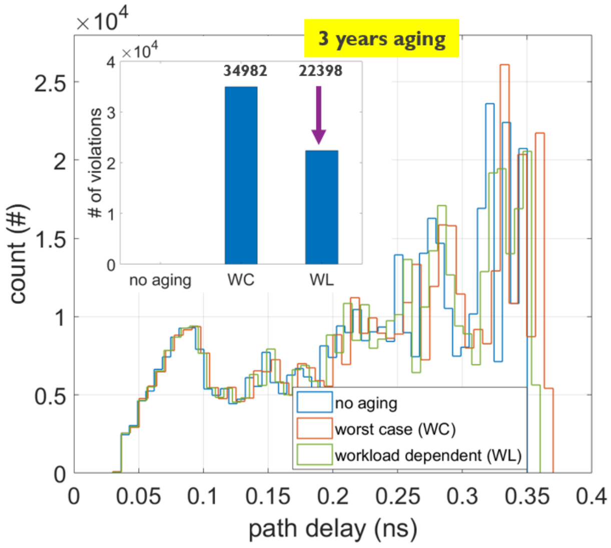

Fig. 12 Timing analysis performed using the back-annotated gate level netlist and WL dependent aged cell library. Path delay distribution shows significant reduction in number of timing violations in case of the proposed WL dependent aging analysis approach.

IV. Conclusion

In conclusion, we proposed AWS algorithm that can be used for fast calculation of transistor degradation under real workload scenarios. An overall improvement of 4.7x in run-time per program is observed compared to cycle accurate simulations at the expense of only +/-3% peak-error in degradation. Using the AWS, together with a LUT-based long-term extrapolation methodology, a circuit level aging estimation framework was demonstrated in terms of timing path violations.

ACKOWLEDGMENT

This project was funded by EU’s Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement.

REFERENCES

[1] Mintarno et al., IRPS 2013
[2] Sivadasn et al., IRPS, 2017
[3] Bian et al., ISQED 2016
[4] Grasser et al., IEDM 2011
[5] Chen et al., IRPS 2011
[6] Krishnan et al., IEDM 2010
[7] Loris Duch, EPFL, Ph.D thesis 2018
[8] Stamoulis et al., GLSVLSI 2016
[9] Rodopoulos et al., TDMR 2014
[10] Gustafsson et al., WCET 2010
[11] https://opencores.org/
[12] Binkert et al., ACM SIGARCH 2011