An Ageing-aware Digital Synthesis Approach

Shengyu Duan, Basel Halak, Mark Zwolinski
School of Electronics and Computer Science, University of Southampton, UK
Email: {sd5g13, bh9, mz}@ecs.soton.ac.uk

***Abstract*—Due to the shrinkage of CMOS technology, wear-out mechanisms such as Bias Temperature Instability (BTI) have raised growing concerns for circuit reliability. BTI can cause a threshold voltage shift in CMOS devices and consequently increase circuit delay. This paper presents an ageing-aware gate-level optimization approach that can be used in a modern synthesis process. It aims to optimize a circuit to give improved lifetime reliability under given area and timing constraints. A new sensitivity metric is proposed as a function of area increase, delay reduction, degradation reduction and design constraints. This sensitivity metric can be adjusted to select the most favourable gates in terms of circuit timing, lifetime or both. By iteratively up-sizing the gates with the highest sensitivity, our proposed optimization flow can meet any realizable area and timing constraints, to give up to 3.3× lifetime improvement.**

I. Introduction

As CMOS technology rapidly downscales, time-dependent transistor degradation has posed a great challenge to circuit reliability [1]. One of the most common wearout mechanisms, Bias Temperature Instability (BTI), manifests itself as an increase in transistor threshold voltage (V_{th}). For combinational circuits, BTI-induced V_{th} shift (ΔV_{th}) results in path delay increase, eventually causing timing violations [2].

A practical approach to improve circuit lifetime at the design stage is to leave additional timing margins [3], which would increase the overheads of area and power. Thus, the circuit is over-designed. Alternatively, other work has proposed slowing down the actual degradation speed. A methodology to manipulate BTI stress by applying pin reordering and logic restructuring is presented in [4]. In [5], an iterative procedure is demonstrated to select and resize selected gates by considering BTI. Although these techniques can improve the lifetime reliability, the costs, in terms of other circuit characteristics like delay and area, are not really controllable. Thus an unacceptable overhead might be induced, especially for a design with significant constraints. None of these works has compared the performance penalties of their approaches with that caused by a conventional over-design technique.

In this work, we present a gate-level optimization approach to mitigate BTI degradation for given constraints. Specifically, we propose a new sensitivity metric that can be adjusted according to the design specifications, to pick the most favourable gate-level transformation in terms of timing, lifetime or both. In addition, an optimization flow employing the proposed sensitivity metric and gate up-sizing technique is demonstrated to mitigate BTI and satisfy specific timing and area constraints. We show that our strategy can be used as an alternative to over-design by having less area cost.

This paper is organized as follows. Section II describes the theory of BTI and degradation evaluation. Section III presents the proposed sensitivity metric and optimization algorithm. In Section IV, we show the results of our approach on some circuits. Finally, the paper is concluded in Section V.

II. BTI Effect and Delay Degradation

A. BTI Modelling

According to the Reaction-Diffusion (R-D) theory, BTI can be physically described as the consequence of charge generation on the transistor oxide interface [6]. The charges are produced while the transistor is stressed (i.e. turned on), and will be partially neutralized in the OFF state. Thus these charges accumulate over time. The ratio of stressed time to the total is known as the stress duty cycle, which also indicates the signal probability (SP) of a logic one/zero at a NMOS/PMOS transistor in a digital circuit.

A simplified BTI model is presented in [2] to quantify the delay shift (ΔD_{bti}) for a given set of stress duty cycles and operational times as follows:

$$\Delta D_{bti} = K.D_0.\alpha^n.t^n \tag{1}$$

where K is a constant parameter dependent on the technology node, temperature and supply voltage; α is the stress duty cycle; t represents the operational time; n is the time exponential constant equal to 0.16 [2]; and D_0 is the intrinsic delay.

Note that the proposed techniques in the remainder of this work do not necessarily rely on above model. Models proposed in other works like [7], are also compatible, since the trends in degradation will still hold.

B. BTI Evaluation on Path Delay Degradation

According to Equation 1, BTI is workload dependent (i.e. SP). Therefore, in order to evaluate path degradation due to BTI, one needs to capture the correct workload characteristics. Unfortunately, most previous work assumes that, for simplicity, the SP at each primary input is 50% [3], [4]. This assumption fails to take the unpredictability of data inputs into consideration, for instance in a general purpose processor. The results may therefore not be accurate.

For a modern digital circuit with a large number of primary inputs, it is impractical to apply all the combinations of input SPs, as the number is vast. Nevertheless, Bian et al, [8], applied a finite number of SPs to all primary inputs, and showed that the overall path delay degradation has a Gaussian-like distribution. The Central Limit Theorem under weak dependence [9], suggests that the sum of an infinite number

of weakly-dependent variables tends to have a Gaussian distribution. This condition is generally true for a combinational path: the total delay degradation is the sum of ΔD_{bti} for all transistors constructing the path, and the dependence of ΔD_{bti} on two transistors far enough apart from one another is negligible. In fact, path degradations can only approximate a Gaussian distribution, since a real path consists of finite transistors. But path degradations for all combinations of SP should be generally convergent in a limited interval.

Motivated by this assumption, a dataset consisting of 2,500 combinations of SP to all primary inputs was used to evaluate the path degradations. The size of the dataset, 2,500, was found in [8] to result in as consistent a distribution as those with more samples. We denote the mean of the distribution as the nominal path degradation ($\Delta D^{nominal}$). Figure 1 shows $\Delta D^{nominal}$ due to Negative BTI (NBTI) for the critical path of an *int to float converter* from the EPFL combinational benchmark suite, [10]. The overall shape approximates a Gaussian distribution, if we consider both NBTI and Positive BTI (PBTI). The circuit is constructed from standard cells of a 65-nm technology node.

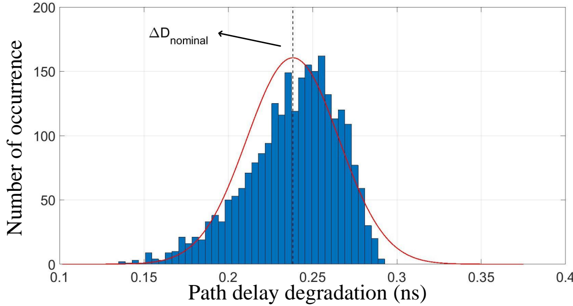

Fig. 1. Nominal case critical path degradation in *int2float converter*

III. Proposed BTI-aware Gate-level Optimization

A. Problem Formulation

Gate-level optimization in a modern synthesis process is performed as the final optimization phase following logic-level optimization and technology mapping. It attempts to meet the exact timing/area constraint by selecting the optimal cells from the library [11]. Nevertheless, current optimization techniques in commercial synthesis tools are blind to ageing effects. Thus, one can only over-design the circuit by putting in pessimistic timing margins to compensate for the impact of ageing.

The technique proposed in this work applies one of the most commonly used gate-level optimization transformations – gate up-sizing, but other techniques like load isolation are also applicable. The objective is not only to meet the design demand for timing and area, but also to minimize the maximal $\Delta D^{nominal}$ in the entire circuit. Thus we can formulate the problem for a circuit with N paths as follows:

$$\begin{aligned} \textbf{Minimize} \quad & \Delta D_{max} = \max_{1 \le i \le N} (\Delta D_i^{nominal}) \\ \textbf{Subject to} \quad & D_{max} = \max_{1 \le i \le N} (D_i) \le D_{cons} \\ & A \le A_{cons} \end{aligned} \quad (2)$$

where D_i and $D_i^{nominal}$ denote the intrinsic delay and delay degradation of path i; D_{max} and A indicate the maximal delay and total area of current design; D_{cons} and A_{cons} represent timing and area constraints respectively.

B. A Constraint-related Sensitivity

Delay-area sensitivity (S_D), represented by the fraction of delay reduction over area increase, is used as a standard metric for gate sizing in commercial synthesis tools [12]. The transformation with the biggest delay-area sensitivity is performed until the timing constraints are met. The optimization aims to add as small as possible area to satisfy the timing specifications. While this technique does not account for the impact of BTI, degradation-area sensitivity ($S_{\Delta D}$) is proposed to identify the transformation that can reduce the greatest degradation per unit area increase [5]. Since the degradation is proportional to the intrinsic delay according to Equation 1, up-sizing the gate with the highest $S_{\Delta D}$ would also be favourable for circuit delay reduction. Thus the timing constraint would be met eventually, with the cost of more area.

Obviously, the circuit area is not really controllable by using either of the above sensitivities. We therefore propose a new sensitivity metric combining both to realize any customized area/timing constraints, as given by:

$$S_{prop} = C_{cons}.w.S_{\Delta D} + S_D \quad (3)$$

$$where$$

$$C_{cons} = \max(0,\ \frac{\frac{A_{cons}}{A} - 1}{1 - \frac{D_{cons}}{D_{max}}})$$

and C_{cons} is a constraint-related coefficient. w is a weight, indicating the influence of $S_{\Delta D}$ on the proposed sensitivity, S_{prop}. The value of w is adjustable to satisfy any given constraints, as described in next subsection.

For better understanding of Equation 3, we analyse the characteristics of S_{prop} under three scenarios:

Scenario 1: Consider a circuit with area A very close to or bigger than A_{cons}, while the delay D_{max} is much greater than D_{cons}. In such a case, S_{prop} is dependent mainly on S_D, since C_{cons} tends to be zero. This situation can be interpreted as follows: if little area is left to accomplish a relatively big delay decrease, the optimization should be dedicated to reducing the delay and inducing as small an area increase as possible;

Scenario 2: Consider another extreme case, where D_{max} nearly reaches D_{cons} but the difference between A and A_{cons} is big. Then C_{cons} tends to infinity, and S_{prop} is thereby determined by $S_{\Delta D}$. This case indicates the remaining area should be used to mitigate the degradation, if it is more than sufficient to meet the timing constraint;

Scenario 3: If neither of the above holds, C_{cons} is a finite number. Thus S_{prop} is determined by both S_D and $S_{\Delta D}$, representing both delay and degradation in the transformation.

In summary, we can rewrite Equation 3 as follows:

$$S_{prop} = \begin{cases} S_D, & \frac{A_{cons}}{A} - 1 \ll 1 - \frac{D_{cons}}{D_{max}} \\ C_{cons}.w.S_{\Delta D}, & \frac{A_{cons}}{A} - 1 \gg 1 - \frac{D_{cons}}{D_{max}} \\ C_{cons}.w.S_{\Delta D} + S_D, & otherwise \end{cases} \quad (4)$$

C. Optimization Algorithm for BTI Minimization

Figure 2 shows the proposed optimization algorithm to minimize BTI degradation under given constraints. Firstly, the weight w from Equation 4 is initialized to a finite number. The initial value of w does not really matter, since it can be adjusted in the remaining steps. However, a greater w indicates $S_{\Delta D}$ has a stronger influence on S_{prop}, and the gate with high $S_{\Delta D}$ is more likely to be up-sized. In such a case, the overall optimization would bring about less degradation but more area according to last subsection. Secondly, the critical path, where the path delay is D_{max}, needs to be identified. In addition, because S_{prop} can be determined by both S_D and $S_{\Delta D}$, the path with a degradation equal to ΔD_{max}, or the ageing-critical path, is identified as well. Afterwards, S_{prop} on all gates of the critical/ageing-critical path is computed, and the gate with the highest S_{prop} is sized up. The above steps are repeated until D_{cons} is met. Finally, we check whether A_{cons} is satisfied. A pre-defined constant ϵ is used to restrict the acceptable range of area: if the actual area does not lie in $[A_{cons} - \epsilon, A_{cons}]$, w would increase or decrease depending on exact value of A, and the design is restored to the initial state, where any gate-level transformations have not yet been applied. A maximum number of iterations can be set to give reasonable runtime.

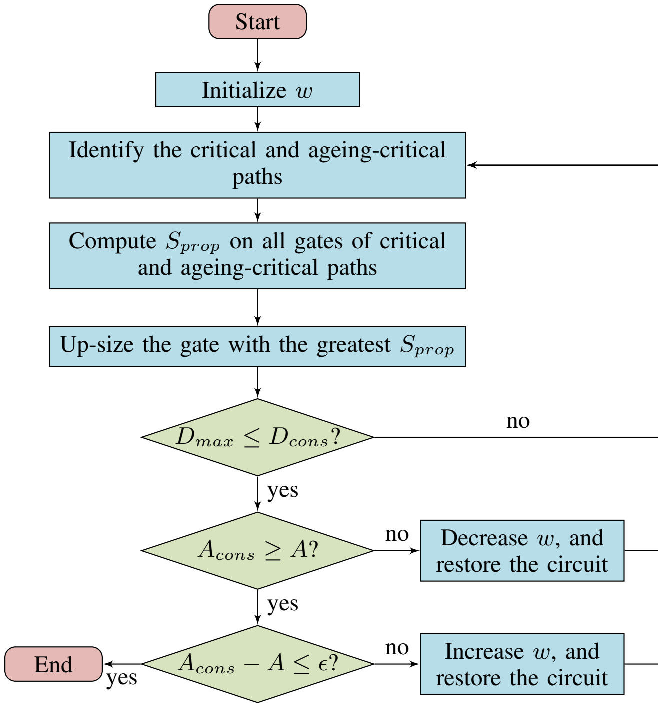

Fig. 2. The optimization flow to minimize BTI effect under given constraints

IV. Experimental Results

The proposed approach has been validated by applying it to several circuits from the EPFL combinational benchmark suite [10] for the case of NBTI only. For each circuit, we used Synopsys Design Compiler for logic-level optimization and our approach for gate-level optimization to optimize away the worst-case negative slack of 150ps. The area constraints were given by the equivalent gate count, with respect to the area of the smallest NAND cell. Referring to the three scenarios of Section III-B, we set three area constraints as follows:

1) $A_{cons} = 0$, indicating using the least possible area as in Scenario 1. This constraint is set as a standard to be compared with the two cases below, since S_D is applied as a general metric in current synthesis tools. We denote the final circuit area in this case as A_{min};

2) $A_{cons} = \infty$, meaning there is no upper limit for the area as in Scenario 2. The final area is denoted as A_{max};

3) $A_{cons} = A_{cus}$, where A_{cus} represents the customized maximal allowed area. By definition, our approach can find the circuit with the optimal lifetime for any A_{cus} within the range of A_{min} and A_{max}. Here we show the case as A_{cus} equalling to the average of A_{min} and A_{max}, Scenario 3.

Figure 3 shows the NBTI lifetimes of the *int to float converter* under the three area constraints after removing all negative slacks. As can be seen, the lifetime is extended as more area is used. Compared with the case of minimal area, the lifetime increases 184.48% and 252.95% for A_{cons} are equal to A_{cus} and ∞ respectively. This result indicates that the lifetime can be significantly extended by a relatively small amount of degradation reduction, because of the exponential behaviour of BTI degradation with respect to the operational time, as in Equation 1.

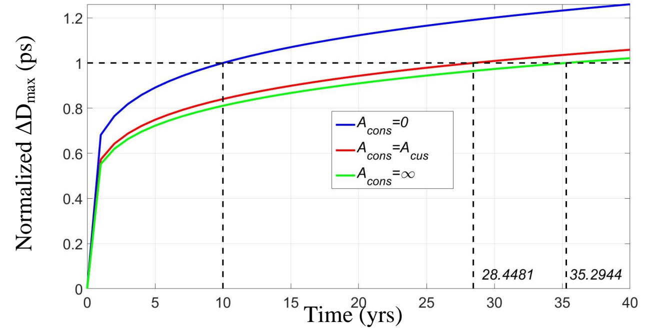

Fig. 3. Degradation changes and lifetime increases of *int2float converter*

Table I presents the optimization results for the experimental circuits. As can be seen, the minimal slacks of all circuits are positive, revealing that the timing constraints are satisfied by our proposed approach. In addition, the NBTI lifetime is optimized within the given amount of area overhead. For the case of $A_{cons} = 0$, where the allowed area increase is too small to remove all negative slacks, the proposed approach can find the smallest design that satisfies the timing constraint. Comparing to the case of minimal area, the lifetime is improved by up to 337.48%.

We compare the optimized circuits by our approach with the ones that are over-designed for the case of $A_{cons} = \infty$, in Table II. Equation 1 is used to compute the post-ageing delay. Assume both techniques can assure the circuits working

TABLE I
CIRCUIT PERFORMANCES BY THE PROPOSED ALGORITHM (WORST SLACKS FOR ALL ORIGINAL CIRCUITS ARE -150PS)

Circuit	A_{cons} = 0		A_{cons} = ∞				$A_{cons} = A_{cus} = (A_{min} + A_{max})/2$				
	Area (A_{min})	Min slack (ps)	Area (A_{max})	Min slack (ps)	Degradation reduction (%)	Lifetime increase (%)	A_{cons}	Area	Min slack (ps)	Degradation reduction (%)	Lifetime increase (%)
Int to float converter	238.5	3.27	246.5	7.99	18.96	252.95	242.5	242.5	2.95	15.99	184.48
Alu control unit	141.5	0.33	174.25	9.84	21.8	337.48	157.88	153	6.6	10.93	100.25
Lookahead XY router	174	1.05	217.75	0.41	12.93	107.48	195.38	190.75	0.37	7.01	54.74
Coding-cavlc	694.5	11.34	705.5	29.28	9.14	77.75	699.5	698.25	11.64	2.37	15.56

properly in 10 years lifetime. 20% guardbands in terms of timing is required by conventional over-designing for a 65-nm technology from emperical study [3]. The guardbands are realized by gate-level optimization (i.e. incremental mapping) of Design Compiler. As can be noted, our approach can save 4.6% area on average to guarantee the same lifetime reliability. Thus, the proposed approach can work as an alternative to conventional over-design resulting in less area cost.

TABLE II
CIRCUIT PERFORMANCES OF OUR APPROACH AND OVER-DESIGNING

Circuit	Over-design		Our approach		
	10 yrs D_{max} (ps)	Area	10 yrs D_{max} (ps)	Area	Area saving (%)
Int to float converter	496.65	261.76	496.22	246.5	5.82
Alu control unit	315.08	182.25	308.4	174.25	4.39
Lookahead XY router	577.03	221	577.18	217.75	1.47
Coding-cavlc	756.95	756.25	764.18	705.5	6.71
Average					4.60

V. CONCLUSION

In this paper, we present a novel gate-level optimization approach for digital synthesis considering ageing. Unlike state-of-the-art techniques, the objective of the proposed approach is not only to improve the lifetime reliability, but also to make sure the design is always within given specifications in terms of timing and area. A constraint-related sensitivity metric considering both delay and degradation reductions is applied. The sensitivity is adjustable to identify the most favourable transformation with regard to circuit timing, lifetime or both. Our optimization flow iteratively up-sizes the most favourable gate, which has the highest sensitivity according to the proposed metric. The results show for any realizable customized design specifications, our approach manages to find the design within the constraints, and with up to 337.48% lifetime improvement for the case of NBTI. By definition, this approach is also valid if considering both NBTI and PBTI. The cost of area of our approach is 4.6% less compared with a conventional over-design technique. Thus our approach is controllable and applicable to trade off between circuit delay, area and lifetime.

ACKNOWLEDGEMENTS

This work was supported by Cisco Research Center Grant No. 593688.

REFERENCES

[1] B. Kaczer, T. Grasser, J. Franco, M. T. Luque, P. Weckx, P. J. Roussel, and G. Groeseneken, "Assessing reliability of nano-scaled cmos technologies one defect at a time," in *Emerging Electronics (ICEE), 2012 International Conference on*. IEEE, 2012, pp. 1–2.

[2] K.-C. Wu and D. Marculescu, "Aging-aware timing analysis and optimization considering path sensitization," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2011*. IEEE, 2011, pp. 1–6.

[3] M. Ebrahimi, F. Oboril, S. Kiamehr, and M. B. Tahoori, "Aging-aware logic synthesis," in *Proceedings of the International Conference on Computer-Aided Design*. IEEE Press, 2013, pp. 61–68.

[4] K.-C. Wu and D. Marculescu, "Joint logic restructuring and pin reordering against nbti-induced performance degradation," in *Proceedings of the Conference on Design, Automation and Test in Europe*. European Design and Automation Association, 2009, pp. 75–80.

[5] A. Gomez and V. Champac, "A new sizing approach for lifetime improvement of nanoscale digital circuits due to bti aging," in *Very Large Scale Integration (VLSI-SoC), 2015 IFIP/IEEE International Conference on*. IEEE, 2015, pp. 297–302.

[6] A. Campos-Cruz, E. Tlelo-Cuautle, and G. Espinosa-Flores-Verdad, "Review: Advances in bti modeling for the design of reliable ics," in *Electrical Engineering, Computing Science and Automatic Control (CCE), 2016 13th International Conference on*. IEEE, 2016, pp. 1–4.

[7] E. Afacan, G. Dündar, A. E. Pusane, and F. Başkaya, "Semi-empirical aging model development via accelerated aging test," in *Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD), 2016 13th International Conference on*. IEEE, 2016, pp. 1–4.

[8] S. Bian, M. Shintani, S. Morita, H. Awano, M. Hiromoto, and T. Sato, "Workload-aware worst path analysis of processor-scale nbti degradation," in *Proceedings of the 26th edition on Great Lakes Symposium on VLSI*. ACM, 2016, pp. 203–208.

[9] R. C. Bradley, "Central limit theorems under weak dependence," *Journal of Multivariate Analysis*, vol. 11, no. 1, pp. 1–16, 1981.

[10] L. Amarú, P.-E. Gaillardon, and G. De Micheli, "The epfl combinational benchmark suite," in *Proceedings of the 24th International Workshop on Logic & Synthesis (IWLS)*, no. EPFL-CONF-207551, 2015.

[11] *Design compiler optimization reference manual*, Synopsys Inc., 2010.

[12] D. G. Chinnery and K. Keutzer, "Linear programming for sizing, vth and vdd assignment," in *Proceedings of the 2005 international symposium on Low power electronics and design*. ACM, 2005, pp. 149–154.