Reliability, Adaptability and Flexibility in Timing: Buy a Life Insurance for Your Circuits

Ulf Schlichtmann[1], Masanori Hashimoto[2], Iris Hui-Ru Jiang[3], Bing Li[1]
[1]Technische Universität München, Germany, Email: {ulf.schlichtmann, b.li}@tum.de
[2]Osaka University, Japan, Email: hasimoto@ist.osaka-u.ac.jp
[3]National Chiao Tung University, Taiwan, Email: huiru.jiang@gmail.com

Abstract— At nanometer manufacturing technology nodes, process variations affect circuit performance significantly. In addition, performance deterioration of circuits due to aging effects is also increasing. Consequently, a large timing margin is required to maintain yield. To combat the pessimism and the resulting overdesign, aging analysis with high-level models, on-chip timing margin monitoring and tuning, and flexible delay models of flip-flops can be deployed. This paper gives an overview of the state of the art of applying these techniques to improve the health of circuits.

I. Introduction

Modern Integrated Circuit (IC) design faces tremendous challenges in achieving the designated performance while guaranteeing a profitable yield. On the one hand, increasing process variations require a nearly unaffordable large timing margin. On the other hand, aging and wearout effects, e.g. Hot Carrier Injection (HCI), Bias Temperature Inversion (BTI) and Time-Dependent Dielectric Breakdown (TDDB), degrade circuit performance, thus pushing chips toward unrecoverable timing failure.

The issues of aging and reliability in digital circuits can be dealt with methods in three different, but complementary approaches. First, aging effects can be modeled, taking the stress conditions of transistors into account. Thereafter, these models can be used to assign timing margins during the design phase to counter potential aging effects. Second, post-silicon tuning components can be inserted into the circuit. After manufacturing, these components can be configured according to the aging predicted using aging models. At this stage, process variations have been fixed so that timing margins can be assigned to handle aging effects specifically. Furthermore, these components can be reconfigured online to mitigate circuit degradation according to real-time aging information from sensors. Third, more accurate timing models for flip-flops should be exploited to reduce over-optimism in the traditional static timing analysis flow. By investigating the metastable region of flip-flop transitions, this technique can provide a more accurate prediction of timing performance of the circuit. Since the above-mentioned methods work at different stages and levels, they can be combined together to form a general framework for handling aging and reliability issues in high-performance designs.

Traditionally, aging effects have been analyzed only for individual devices, resulting in aging models for transistors. From such results, overall safety factors can be determined, which are then applied to an entire IC design. This very coarse evaluation either over-estimates or underestimates aging effects, leading to overdesign or unreliable circuits, respectively.

To increase the quality of aging prediction, detailed aging analysis has become the focus of recent research. In this approach, stress profiles of circuit components are generated either from statistical analysis or simulation. With these profiles, aging models are used to predict potential aging effects. Since the direct applicability of transistor-level aging models to the whole circuit is impractical due to large simulation time, it is important to develop analysis techniques on gate and module levels, without significantly sacrificing accuracy. Since for each module only the interface simulation information needs to be retained, sizes of aging models and thus simulation complexity can be reduced significantly.

Besides aging analysis and corresponding circuit optimization, post-silicon tuning devices, e.g. reconfigurable clock tuning buffers, can be deployed into the circuit to alleviate aging effects dynamically. These devices are configured after manufacturing for aging optimization with process parameters fixed in individual chips. Furthermore, these devices can be adjusted online to counter application-dependent aging effects, by balancing timing margins between flip-flop stages.

Since aging effects and reliability degradation are affected by applications executed by the circuit, static/statistical aging models can only provide a conservative and often rough estimation of aging. To make the circuit react to real aging stress on-the-fly, special timing sensors can be integrated into the circuit to monitor timing margins of aging-/reliability-critical paths when the circuit is operating. With the *in situ* aging information, post-silicon components in the circuit can be adjusted accordingly. More importantly, a balance between timing margins and power consumption can be achieved, using techniques such as dynamic voltage/frequency scaling and body biasing. With such adaptive control techniques, timing margins reserved for aging can be reduced, thus enabling a lower supply voltage. The latter slows the aging process, and

978-1-4673-9569-4/16/$31.00 ©2016 IEEE

thus the performance degradation.

The adaptive margin control techniques above may cause unexpected timing errors due to, e.g. large supply noise. In addition, the on-chip timing margin test does not have a full coverage of all the critical paths and the test accuracy is also limited due to costs. Therefore, timing errors cannot be avoided completely. To deal with this challenge, delay testing should be performed frequently, and early as well as conservative error predictions should be produced to guarantee a large time to failure, even up to many years.

When aging and reliability issues and process variations are considered together, timing margins in the circuit may become small enough to reach the limit of the traditional delay models of flips-flops. If the arrival time to a flip-flop is very near to its setup time boundary, the clock-to-Q propagation delay may become larger than estimated. This increased delay affects all the path delays to the fanout flip-flops, thus degrading their timing margins. The criticality-dependency effect has become worse in nanometer technology nodes and cannot be omitted further.

Recently, researchers have tried to remove the over-optimism by using different timing characterization models and/or timing analysis methods. These methods either require unbounded iterations to converge for large designs or cannot be integrated with the current STA flow easily. Consequently, a close look at the dependency between arrival times and flip-flop delays becomes necessary.

In this paper, aging models, timing monitoring and post-silicon tuning, and flip-flop delay flexibility will be discussed in detail in Section II–Section IV, respectively.

II. Aging and adaptability: How to deal with circuits getting older

Aging has become an important consideration in the design of modern, complex ICs. The dominant effect today is BTI–especially NBTI (Negative BTI) affecting PMOS transistors, but increasingly with smaller process technologies also PBTI (Positive BTI). In addition, HCI and TDDB are relevant aging effects for transistors, while electromigration (EM) is a concern for wires. In the following discussion, we will focus on aging analysis of transistors.

Traditionally, aging was the concern of technology and reliability departments. They worked on ensuring that manufactured transistors would be as reliable as reasonably possible. Their aging degradation would be characterized over time, typically as a function of supply voltage and temperature. As a result, an overall guardband factor would be recommended which designers then would take into consideration. This approach is increasingly less feasible, for two reasons: firstly, the degradation increasingly depends on other factors as well, such as the workload a specific transistor is experiencing. Secondly, as progress in manufacturing technology results in ever less performance improvements, this leads to pressure on design to reduce overall margins and perform more specific analyses to extract performance from a design.

Aging first was included in transistor models so that it could be considered during design. Transistor-level simulation obviously is limited to rather small parts of today's complex designs. Therefore, there is a need to lift aging analysis to higher levels in the design flow. Initially, aging analysis at gate-level was addressed. Later, the abstraction level was raised and module-level models were introduced to obtain further performance improvements and thus to be able to handle larger circuits.

A. Raising Aging Analysis to Gate Level

Initial approaches [1, 2, 3] considered only NBTI, addressed only delay and not slope, and considered gates monolithically without treating the different transistors of a gate individually. Then, [4] introduced an approach that removed these limitations, but required a recharacterization of the library for every new use profile, which is not realistic in industrial practice.

To relieve these limitations, [5] then introduced *AgeGate,* which still represents the state of the art in gate-level aging analysis. The approach rests on a canonical gate delay model, as it is used also in statistical timing analysis [6]:

$$q_{aged} = q_{fresh} + \Delta_q = q_{fresh} + \sum_{m \in G} \sum_{p \in P} \chi^q_{m,p} \cdot \Delta_{p_m} \qquad (1)$$

q in (1) is a timing quantity–delay or output slope; $\chi^q_{m,p}$ describes the sensitivity of a particular transistor to the drift Δ_{p_m} of a particular transistor parameter p_m. P is the set of all relevant parameters, and G denotes the set of all transistors of a considered gate.

The considered drifting parameters are V_{th} for NBTI and I_{on} for HCI. Their degradation is computed considering the structure of a gate and the time it is under stress. This time depends on the signal level for NBTI, and on the transition probability for HCI. These values can either be derived from existing input patterns for a circuit, or in their absence be estimated either probabilistically or using worst-case assumptions. The use profile (supply voltage V_{DD} and temperature T over the circuit's lifetime) is also taken into account.

The *AgeGate* approach can consider PBTI as well as NBTI. It can easily be extended to consider recovery. This is important as it has been shown on a number of standard benchmarks that neglecting the recovery effect of NBTI can lead to an overestimation of aging-induced performance degradation by up to 5 percentage points [7]. Multistage gates can be handled by *AgeGate* as well [8]. The approach is very efficient. Even large benchmark circuits such as c6288 or c7552 can be analyzed in significantly less than one minute.

Results on about a dozen benchmark circuits show that while NBTI dominates the aging effects, HCI also plays a significant role, accounting for up to 1/3 of the overall degradation. If the transistors of a gate are not considered individually, aging can be overestimated by 25% on average. Similarly, not considering the fact that output slope also ages and thereafter affects succeeding gates in a path, leads to 20% underestimation of degradation.

B. Aging Analysis on Module Level

While the *AgeGate* approach dramatically improves performance of aging analysis compared with transistor-level simulation, the abstraction level can be raised further.

For a larger module, a timing model can be built which describes only the aging information/constraints at the interface of the module [9]. Such a model has many advantages: during early design stages, timing can be estimated from timing models of modules–long before a gate-level netlist becomes available. In addition manufacturing variations can be integrated into this interface model, such as in [10].

The basic structure of such a module-level timing model is a timing graph describing the delays of gates as well as their interconnect. An analysis shows that many paths can never become critical, even if worst-case assumption for their aging are made. Using a number of dedicated reduction steps, both timing arcs and entire nodes can be removed from the timing graph, thus significantly reducing its size–and thereby the effort required to analyze the graph.

The reduction steps proposed in [9] are:

- block-based reduction step
- path-based reduction step
- reconvergent fan-out reduction step

On average over a number of benchmarks, almost 75% of all nodes and more than 80% of all edges can be removed using this approach, thus resulting in a further performance improvement of 30x compared to an analysis on gate level–without any loss in accuracy.

More refined reduction steps were added later, resulting in further improvements. More importantly, it was shown how process variations can be incorporated into module level timing models as well [11].

In [11] it was also discussed that in addition to speeding up timing analysis during design, the proposed module level techniques can also be used to develop a run-time aging monitoring system for ICs. The basic idea is that for many components of a typical design, the described reduction steps drastically reduce the number of paths which can potentially become critical (PCPs - Potentially Critical Paths). For a highly safety-critical circuit, the PCPs which remain after the reduction steps can then regularly be monitored during operation of a circuit.

C. Countering Aging by Post-Silicon Tuning

Aging effects lead to degradation of gate delays and thus the timing performance of a circuit. With analysis methods described above, aging effects can be modeled and the circuit can be optimized during the design phase. However, real applications still have a large impact, because they determine how transistors in the circuit are stressed finally. Since these applications depend on different usage scenarios, it is difficult to find a one-size-fits-all solution for aging optimization. In addition, process variations cause the same path to have different delays in different chips after manufacturing. Pre-silicon optimization, however, cannot adjust the circuit according to these specific delays, but can only handle aging statistically based on manufacturing data provided by foundries.

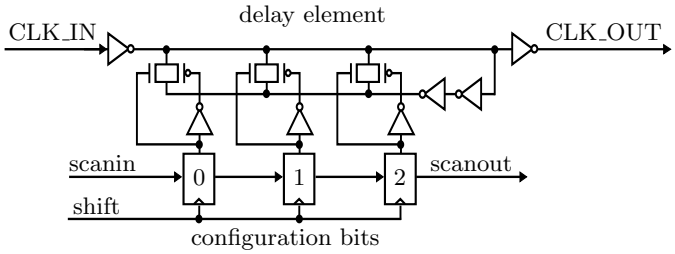

Fig. 1. Post-silicon tuning buffer in [12].

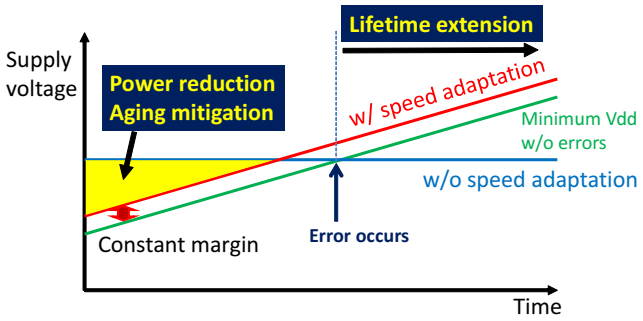

Fig. 2. Margins of circuits with and without adaptive speed control during the chip's lifetime. Supply voltage is used as a tuning knob for speed adaptation in this example.

To deal with process variations and application-dependent aging, post-silicon/online tuning components can be deployed in the circuit. An example of such tuning components is illustrated in Fig. 1, which is a delay buffer inserted into the clock path to flip-flops. The delay of such a buffer can be changed by setting the configuration bits in the three registers. After manufacturing, the clock delays are tuned to assign aging-critical paths more timing budget by shifting clock edges toward the stages with smaller combinational delays. By combining these buffers and aging models, timing margins can be balanced properly for each individual chip after manufacturing.

Since applications may affect the aging status of transistors dynamically, post-silicon tuning can be applied online according to the output of aging monitors, using a method similar to [13]. In addition to clock tuning, other techniques, such as fine tuning of supply voltages as discussed in [14], can also be applied to counter aging effects dynamically.

III. Sensor-based timing adaptation: Examining and tuning the health of circuits

To minimize design and operation margins, adaptive circuit design where each chip self-adjusts its operating condition, such as supply voltage and body bias, is promising. Let us focus on timing margin degradation due to aging. Figure 2 illustrates how the operational margin during the chip's life-

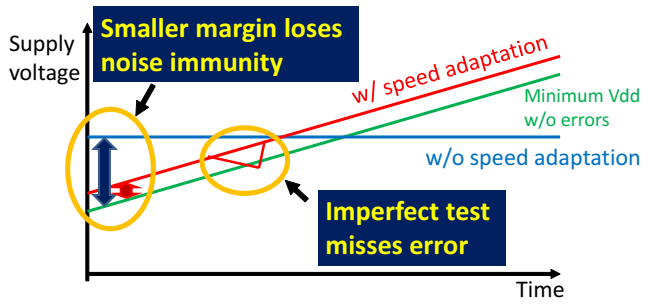

Fig. 3. Margins of circuits with and without adaptive speed control during the chip's lifetime. Smaller margin and imperfect test may cause timing errors in adaptive circuits.

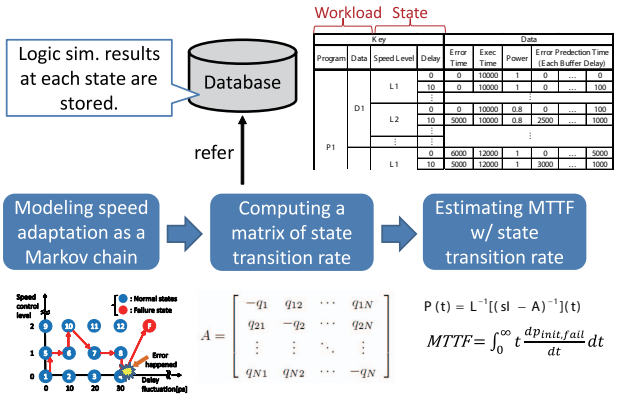

Fig. 4. Overview of stochastic error rate estimation.

time varies with and without adaptation, where supply voltage is used as a tuning knob for speed adaptation. Without adaptation, the operational margin at the beginning of the chip lifetime is large, and the margin decreases as the chip ages. If the delay increase due to aging exceeds the preallocated timing margin, timing errors occur in the chip. With adaptation, we can ideally set a constant operational margin for the entire chip lifetime by gradually increasing supply voltage. Especially, we can reduce the margin at the beginning of the chip lifetime. If the large operational margin can be translated into supply voltage reduction, the aging process, i.e. the performance degradation, can be slowed down, and the chip lifetime extends. For pursuing these advantages, performance adaptation has been widely studied. For example, [15] reported that the power dissipation with performance adaptation was smaller than that with conventional worst-case design by 46% in a 65nm subthreshold design.

On the other hand, the negative impact of performance adaptation, which is illustrated in Fig. 3, is less studied. Performance adaptation degrades noise immunity, especially at the beginning, and hence the possibility that an unexpected timing error due to, for example, unexpected large supply noise occurs becomes higher. In addition, the margin checking performed in the chip is not perfect due to the limited area and time budget for test. Therefore, there is a fundamental problem that the possibility of timing error occurrence cannot be completely reduced to zero, since, for example, a sudden delay increase larger than expected can induce a timing error without error detection or before error prediction. Similarly, offline delay testing may miss the error because delay testing is carried out with a certain time interval. It should be noted that the timing errors due to such a sudden delay increase arise even in the chips without adaptive speed control, especially at the end of the chip lifetime because the operational margin is small.

To obtain a good adaptive circuit design mitigating the above negative impact, we need to optimize the adaptive circuit. Each adaptation scheme has some design parameters to optimize and the built-in test can be tuned. However, it is difficult to evaluate how much improvement in MTTF (mean time to failure) and power is achieved by the optimization and tuning, since the device lifetime is extraordinarily longer than simulatable circuit operation time. For example, if we try to to simulate 10 year operation (3×10^{17} clock cycles) with a logic simulator processing 3×10^3 cycles per second, it takes 3×10^6 years. Therefore, another approach instead of naive simulation is indispensable. For enabling adaptive circuit optimization, we have developed a stochastic error rate estimation method [16, 17]. The necessary computation time was reduced by twelve orders of magnitude in a test case. Hereafter, let us introduce the stochastic error rate estimation method.

The proposed method models adaptive speed control under dynamic delay variation as a continuous-time Markov process (Fig. 4). Markov process is a stochastic process having a Markov property that the next state is determined by only the current state and is independent of the previous states. Especially, continuous-time Markov process is a special Markov process whose time parameter is continuous.

We assign states as follows. The circuit delay temporally fluctuates due to unintentional temperature change, power supply noise and aging. By sensing such temporal delay fluctuation with online/offline delay testing, the performance of the circuit under adaptive speed control is intentionally tuned by supply voltage scaling and/or body biasing. We define states in Markov process such that each state is associated with a pair of unintentional delay variation and levels of intentional speed control. We often prepare several discrete values for supply voltage scaling and body biasing. On the other hand, the unintentional delay variation is continuous in nature, but for model simplicity, we discretize the unintentional delay variation into several representative values. We call these states as normal states. On the other hand, we add an additional failure state which indicates a timing error happened in the past.

In a continuous-time Markov process, transition rate of going from state i to state j, is the key parameter that characterizes the process behavior (Fig. 4). Given a matrix of the transition rates, we can obtain closed-form expressions of state probability as a function of time t. This means that once the matrix of transition rates is given, the MTTF computation can be carried out with a constant time, and the computation time is independent of the timing error rate and MTTF of the circuit under evaluation. Note that the above computation is

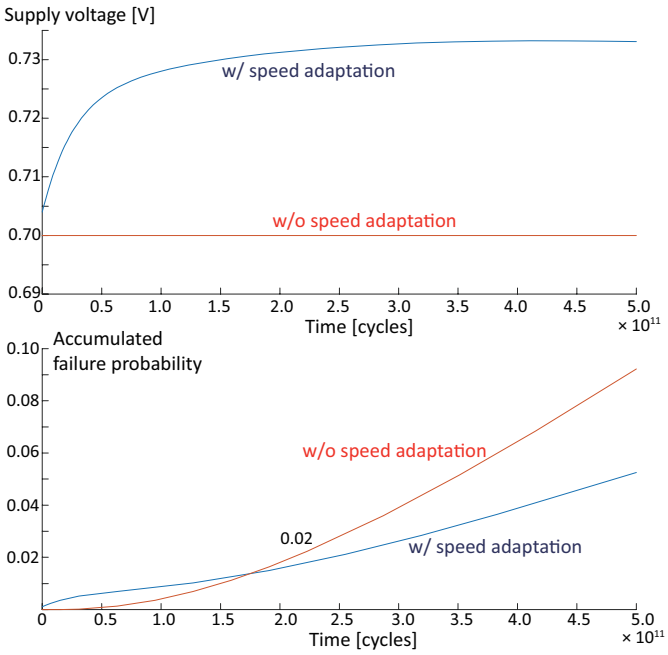

Fig. 5. Supply voltage and accumulated failure probability under an aging process.

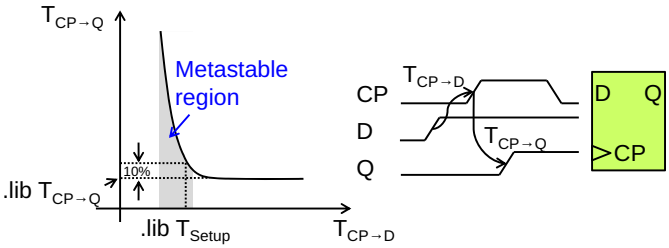

Fig. 6. The criticality effect.

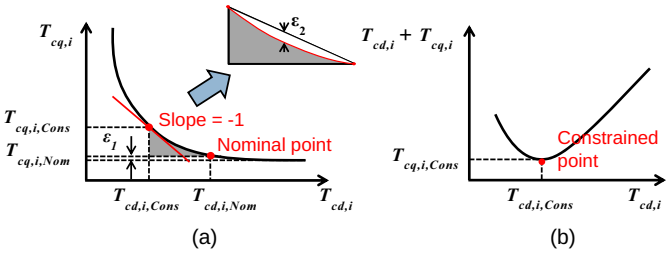

Fig. 7. (a) The nominal point characterization. (b) The constrained point characterization.

applicable to any type of adaptive speed control, since the state assignment explained above is independent of the implementation of adaptive speed control. To construct the transition rate matrix, we developed a similarity database and a direct derivation method of the matrix using the database (Fig. 4). Thanks to this development, the proposed method computes MTTF 10^{12} times faster than a logic simulator in a test case [16]. In addition, [17] extends the state definition for coping with the inter-die and within-die process variations. The extended state definition includes the distributions of gate delay for within-die variation at each state, and then the probability distributions of path delay violation can be considered. Also, a method of fast state transition rate calculation is developed.

To demonstrate that the proposed method can consider gate-by-gate aging processes, we exemplified an analysis [17]. The upper figure of Fig. 5 shows the supply voltages of the circuit with and without adaptive speed control under the aging. The supply voltage of the circuit without adaptive speed control is fixed to 0.7V. We can see that the average supply voltage of the circuit with adaptive speed control increases as the time elapses and the aging effect proceeds. The bottom figure of Fig. 5 shows the accumulated failure probability. At the beginning, the supply voltage of 0.7V gives enough timing margin, and hence the increase in the accumulated failure probability is smaller in the case without adaptive speed control. However, as the time elapses, the timing margin of the circuit without adaptive speed control becomes smaller, and the accumulated failure probability increases faster. On the other hand, with the adaptive speed control, the timing margin is kept almost constant, and hence the increasing rate of the accumulated failure probability, i.e. the failure rate, is almost constant. For real products, the failure rate needs to be reduced further via design optimization, but the important point here is that such an analysis can be performed with the proposed estimation method.

We are now developing a design optimization framework that uses the stochastic error rate estimation method as an engine of performance evaluation. Our future work includes the establishment of design optimization and test advancement.Then, we would like to provide a design methodology for adaptive circuits.

IV. Flexible Flip-flops: False alarm and false negative in timing

In this session, we will discuss how to reduce over-optimism in static timing analysis by considering the criticality dependency effect and unnecessary pessimism incurred in common clock paths.

A. The Criticality Dependency Effect

The criticality-dependency effect indicates that a late arrival at the data input of a flip-flop lengthens the propagation delay from the clock pin to the data output of this flip-flop, thus degrading the timing margins of paths launching from this flip-flop. The criticality-dependency effect keeps deteriorating and cannot be omitted as technology advances into the nanometer era. Fig. 6 shows the transfer curve of the clock-to-Q propagation delay versus the input slack between the data input arrival and the clock edge of a flip-flop. In a conventional standard cell library, the clock-to-Q propagation delay is characterized by assuming an infinite input slack [18, 19], while the setup time is defined as the input slack at the 10%

degradation point. The defined setup time, however, may fall into the metastable region for advanced technology; thus when the input slack of a flip-flop is close to the setup time, the increased propagation delay largely degrades the timing margins of paths launching from this flip-flop.

Let $T_{cq,i}$ indicate the clock-to-Q propagation delay of a flip-flop FF_i, and $T_{cd,i}$ indicate its data input slack. Fig. 7 shows that a triangle model can be used to characterize delay degradation [20], which consists of two essential points: the nominal point and the constrained point for FF_i. The nominal point $(T_{cd,i,Nom}, T_{cq,i,Nom})$ is defined as the point where $T_{cq,i,Nom}$ is equal to the clock-to-Q propagation delay in the stable region $T_{cd,i,Stable}$ plus an optimistic error ε_1 . The constrained point $(T_{cd,i,Cons}, T_{cq,i,Cons})$ is defined as the point where its tangent slope is -1. The pessimistic error ε_2 of the triangle model is defined as the maximum difference between the clock-to-Q propagation delay calculated by the triangle model and the simulated value.

The nominal point and the constrained point can be characterized simultaneously because they are independent. For the nominal point characterization [see Fig. 7(a)], given an upper bound and a lower bound of the data input slack, we can easily find the degraded propagation delay that equals $T_{cd,i,Stable} + \varepsilon_1$ by simulation. For the constrained point characterization, however, without an analytical formula for the transfer curve, it is hard to calculate the tangent slope at a specific data input slack. Hence, the constrained point characterization can be reduced to finding the minimal point on a modified transfer curve [see Fig. 7(b)], where the x axis represents the data input slack, and the y axis represents the sum of the data input slack and the clock-to-Q delay. It can be shown that in the modified transfer curve, the minimal point represents the point where the propagation delay increasing rate equals the data input slack decreasing rate, i.e., the constrained point on the original transfer curve.

Based on the triangle model, we can determine timing-risky flip-flops, including critical $(T_{cd,i,Cons} \leq T_{cd,i} \leq T_{cd,i,Nom})$ and violating $(T_{cd,i} < T_{cd,i,Cons})$.

B. Common Path Pessimism Removal

Excess pessimism in clock network skews the actual timing of a circuit. Conventional static timing analysis (STA) considers different delay values for the common part of launching and capturing clock paths. Nevertheless, along the common segment of the launching and capturing clock paths, the clock signal cannot simultaneously experience different operating conditions. To avoid over-optimization, the artificial pessimism induced by the delay difference along the common clock path segment should be removed. The challenge of common path pessimism removal (CPPR) is that the amount of pessimism is path dependent. Although CPPR credits are path dependent, credits can be pre-computed and stored at each point in clock network during block-based STA delay propagation. Once the launching and capturing flip-flop pair is identified, the stored credits can be used for post-CPPR slack calculation.

By identifying the common path between launching and capturing clock paths (see Fig. 8), CPPR credits for hold and

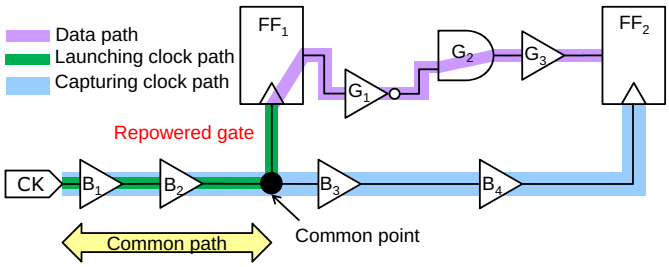

Fig. 8. Common path pessimism removal.

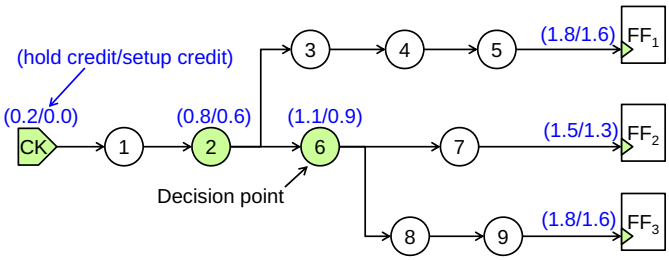

Fig. 9. Decision points on a clock tree and credits. Assume early/late delay for each gate is 0.1/0.3, while 0.1/0.2 for each wire. The shaded nodes indicate decision points. The common path between FF_1 and FF_2 ends at g_2 , and the hold and setup credits are 0.8 and 0.6, respectively.

setup checks can be approximated as follows.

$$credit^{hold} = at^{late}_{common\_point} - at^{early}_{common\_point}; \quad (2)$$

$$credit^{setup} = credit^{hold} - (at^{late}_{CK} - at^{early}_{CK}). \quad (3)$$

Thus, post-CPPR slacks can be obtained as follows.

$$post\_slack^{hold}_{path} = pre\_slack^{hold}_{path} + credit^{hold}; \quad (4)$$

$$post\_slack^{setup}_{path} = pre\_slack^{setup}_{path} + credit^{setup}. \quad (5)$$

To remove the pessimism between two flip-flops, we have two tasks: One is to find the common path of the launch and capture clock paths, and the other is to compute the amount of pessimism. For the first task, the common path finding can be efficiently done by iteratively comparing decision points. A decision point on a clock tree is the clock source, a clock sink, or a clock buffer with multiple fanouts [21]. (See Fig. 9.) For the second task, once the last node of the common path (common point) is identified, the setup and hold credits can then be directly determined. Therefore, instead of computing the setup/hold credits after each path retrieval, we can precompute these credits accumulated for each node in the clock tree during block-based STA.

V. Conclusion

In this paper, we have discussed techniques to raise aging analysis from transistor level to gate and module level. With these models, aging analysis can be performed more efficiently

while still maintaining a good accuracy. We have also discussed timing margin monitoring and on-chip dynamical tuning. This technique can translate large operational margins into supply voltage reduction so that aging effects can be reduced. Furthermore, flexible delay models of flip-flops have also been discussed and the resulting timing analysis method can reduce the over-optimism in conventional static timing analysis effectively. With these methods, the lifetime of a circuit can be extended significantly and its health consolidated steadily.

Acknowledgments

The work in Section II was partly supported by the German Research Foundation (DFG) as part of the Transregional Collaborative Research Centre Invasive Computing (SFB/TR 89) and as part of the priority program "Dependable Embedded Systems" (SPP 1500 - spp1500.itec.kit.edu). The work in Section III was partly supported by NEDO and STARC. The work in Section IV was partly supported by Taiwan MOST.

References

[1] B. C. Paul, K. Kang, H. Kufluoglu, M. Alam, and K. Roy. Temporal performance degradation under NBTI: Estimation and design for improved reliability of nanoscale circuits. In *Proc. Design, Autom., and Test Europe Conf.*, pages 169–174, 2006.

[2] S. V. Kumar, C. H. Kim, and S. S. Sapatnekar. An analytical model for negative bias temperature instability. In *Proc. Int. Conf. Comput.-Aided Des.*, pages 493–496, 2006.

[3] Y. Wang, H. Luo, K. He, R. Luo, H. Yang, and Y. Xie. Temperature-aware NBTI modeling and the impact of input vector control on performance degradation. In *Proc. Design, Autom., and Test Europe Conf.*, pages 546–551, 2006.

[4] J. Chen, S. Wang, N. Bidokhti, and M. Tehranipoor. A framework for fast and accurate critical-reliability paths identification. In *IEEE North Atlantic test workshop(NATW)*, 2011.

[5] D. Lorenz, G. Georgakos, and U. Schlichtmann. Aging analysis of circuit timing considering NBTI and HCI. In *IEEE international on-line testing symposium (IOLTS)*, pages 3–8, 2009.

[6] C. Visweswariah, K. Ravindran, K. Kalafala, S.G. Walker, S. Narayan, D.K. Beece, J. Piaget, N. Venkateswaran, and J.G. Hemmett. First-order incremental block-based statistical timing analysis. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 25(10):2170–2180, 2006.

[7] V. B. Kleeberger, M. Barke, C. Werner, D. Schmitt-Landsiedel, and U. Schlichtmann. A compact model for NBTI degradation and recovery under use-profile variations and its application to aging analysis of digital integrated circuits. *Microelectronics Reliability*, 54(6-7):1083–1089, 2014.

[8] D. Lorenz, M. Barke, and U. Schlichtmann. Efficiently analyzing the impact of aging effects on large integrated circuits. *Microelectronics Reliability*, 52(8):1546–1552, 2012.

[9] D. Lorenz, M. Barke, and U. Schlichtmann. Aging analysis at gate and macro cell level. In *Proc. Int. Conf. Comput.-Aided Des.*, pages 77–84, 2010.

[10] B. Li, N. Chen, Y. Xu, and U. Schlichtmann. On timing model extraction and hierarchical statistical timing analysis. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 32(3):367–380, 2013.

[11] D. Lorenz, M. Barke, and U. Schlichtmann. Monitoring of aging in integrated circuits by identifying possible critical paths. *Microelectronics Reliability*, 54(6-7):1075–1082, 2014.

[12] S. Naffziger, B. Stackhouse, T. Grutkowski, D. Josephson, J. Desai, E. Alon, and M. Horowitz. The implementation of a 2-core, multi-threaded Itanium family processor. *IEEE J. Solid-State Circuits*, 41(1):197–209, January 2006.

[13] R. Ye, F. Yuan, and Q. Xu. Online clock skew tuning for timing speculation. In *Proc. Int. Conf. Comput.-Aided Des.*, pages 442–227, 2011.

[14] R. Kumar, B. Li, Y. Shen, U. Schlichtmann, and J. Hu. Timing verification for adaptive integrated circuits. In *Proc. Design, Autom., and Test Europe Conf.*, pages 1587–1590, 2015.

[15] H. Fuketa, M. Hashimoto, Y. Mitsuyama, and T. Onoye. Adaptive performance compensation with in-situ timing error predictive sensors for subthreshold circuits. *IEEE Trans. VLSI Systems*, 20(2):333–343, 2012.

[16] S. Iizuka, M. Mizuno, D. Kuroda, M. Hashimoto, and T. Onoye. Stochastic error rate estimation for adaptive speed control with field delay testing. In *Proc. Int. Conf. Comput.-Aided Des.*, 2013.

[17] S. Iizuka, Y. Masuda, M. Hashimoto, and T. Onoye. Stochastic timing error rate estimation under process and temporal variations. In *Proc. Int. Test Conf.*, 2015.

[18] W. Roethig. Library characterization and modeling for 130 nm and 90 nm SoC design. In *Proc. Int. SoC Conf.*, pages 383–386, 2003.

[19] D. Patel. CHARMS: Characterization and modeling system for accurate delay prediction of ASIC designs. In *Proc. Custom Integr. Circuits Conf.*, pages 9.5/1–9.5/6, 1990.

[20] Y.-M. Yang, K. H. Tam, and I. H.-R. Jiang. Criticality-dependency-aware timing characterization and analysis. In *Proc. Design Autom. Conf.*, pages 1–6, 2015.

[21] Y.-M. Yang, Y.-W. Chang, and I. H.-R. Jiang. iTimerC: Common path pessimism removal using effective reduction methods. In *Proc. Int. Conf. Comput.-Aided Des.*, pages 600–605, 2014.