Efficient Selection and Analysis of Critical-Reliability Paths and Gates

Jifeng Chen, Shuo Wang, and Mohammad Tehranipoor
University of Connecticut, Storrs, CT 06269, USA
{jic09003,shuo.wang,tehrani}@engr.uconn.edu

ABSTRACT

Aging effects such as negative bias temperature instability (NBTI) and hot carrier injection (HCI) have become major concerns when designing reliable circuits at sub-45nm technologies. It is vital to efficiently identify the paths that age at a faster rate than others in the field. Moreover, gates having the most impact on the degradation of these paths must be identified for compensation purposes. In this paper, we propose (i) a new timing analysis flow, which can quickly and accurately predict path and gate degradation due to NBTI and HCI effects, and (ii) a novel algorithm that can effectively identify the smallest set of critical-reliability gates while quantitatively evaluates their relative importance to path delay degradation. This facilitates reliability-enhancement methods to efficiently mitigate reliability threats using minimum area overhead. Our simulation results on several benchmark circuits demonstrate the efficiency of the proposed technique.

Categories and Subject Descriptors

B.8.1 [**Performance and Reliability**]: Reliability, Testing, and Fault-Tolerance

Keywords

Reliability, Aging, Timing Analysis, Gate Sizing, Optimization

1. INTRODUCTION

Aggressive scaling of CMOS technology into small feature sizes ($\leq 45nm$) worsens the degradation of performance and reliability caused by aging effects, especially NBTI [1][2] and HCI [4][5]. To compensate for aging effects and meet performance requirements throughout the lifetime operation, two categories of methods may be used: (1) Guardbanding methods [6][7][14], and (2) Adaptive control methods [8][9][11]. Due to the high complexity of aging mechanisms, it is extremely difficult to perform accurate degradation prediction. As a result, unnecessary power and area overhead are often introduced. For example, aggressive guardbanding methods may introduce an over-conservative margin by oversizing a large number of gates, especially those not critical to the overall reliability of the circuit. Meanwhile, adaptive control schemes usually introduce excessive area and power overhead, as they have to keep monitoring a large number of paths and gates for performance evaluation. A gate sizing technique was proposed in [10] to identify critical gates defined as the gates that age the most in the circuit. We shift our focus toward the identification and quantification of critical-reliability gates (CRGs), defined as the minimum number of "important" gates contributing to degradation of critical-reliability paths (CRPs), i.e., paths that are sensitive to aging and could potentially become critical in the field. By identifying CRPs and CRGs, aging compensation can be achieved with a minimal area and power overhead. As will be elaborated further in later sections, CRGs can be flexibly identified according to different workload scenarios, e.g., worst-case or actual workload conditions, which makes this technique easily compatible with conventional industry design practices.

Our work makes the following major contributions: (1) We develop an efficient aging-aware timing analysis flow to identify the CRPs in the circuit given different workloads. Rather than analyzing critical path (CP) set, which is extremely large in modern designs, we focus on selection of CRPs from CPs (CRP set is substantially smaller than CP set) to considerably reduce the computational complexity of the proposed CRG selection flow. (2) We develop a technique to quantitatively evaluate the importance of the CRGs based on their contribution to CRPs' degradation. Here, we develop a new metric to evaluate *a gate's importance to circuit CRPs degradation* making our proposed technique different from the previous ones that attempt to identify the gates with maximum aging. Our results demonstrate that the most impactful gates in the circuit are not necessarily the gates that have aged the most. (3) Lastly, we develop a novel algorithm to identify the minimum number of CRGs in the circuit for sizing to ensure that performance degradation of CRPs will not cause any failure in the field.

The remainder of the paper is organized as follows. Section 2 briefly describes our proposed CRP and CRG selection flow. The details of CRP analysis and selection are presented in Section 3. Section 4 explains CRG identification based on LP optimization technique. The simulation results and analysis are presented in Section 5. Finally, the paper is concluded in Section 6.

2. PROPOSED FLOW

Figure 1 shows our proposed flow for aging-aware path de-

GLSVLSI'12, May 3–4, 2012, Salt Lake City, Utah, USA.

lay analysis (APD) and identification of CRPs and CRGs in integrated circuits. The flow includes three major steps: (1) CP Selection, (2) CRP Selection, and (3) CRG Identification. The conventional Static Timing Analysis (STA) tool is used to extract critical paths. As investigated in [18], performance of a critical path could degrade as much as 20% over ten years operation. Thus, during the static timing analysis, if the delay of a path P_i satisfies $\Gamma_{p0i} \times 120\% \geq \Gamma_{clk} - \Gamma_m$, it is selected as a potential critical path, where Γ_{p0i} is the path delay at time 0, Γ_{clk} is the clock period and Γ_m is a safe margin (i.e., guardband) added at time zero. These paths may not be critical at time zero, but may become critical at some time point in the field. CRPs and CRGs are identified in steps 2 and 3, respectively, which will be described in details in Sections 3 and 4. Gate sizing can be applied to the CRGs in order to compensate for degradation of CRPs.

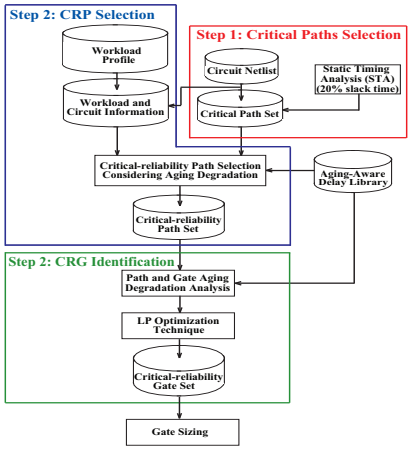


Figure 1: CRP and CRG selection flow.

3. CRP SELECTION

In this section, the conventional NBTI and HCI models are improved for CRP analysis. We then describe CRP selection and the accuracy of our CRP delay calculation flow.

3.1 Model Analysis and Improvement

The Reaction-Diffusion (R-D) model [13] is the general model used to explain and quantify the shift in threshold voltage due to NBTI and HCI effects. Also, the delay models in [16][17] are formulated to describe their dependency on threshold voltage and other parameters, such as temperature, capacitive load, stress probability, and slew rate; however, these models are incapable of incorporating complex conditions, such as multiple-input stress probabilities, which represent workload condition. First, we will briefly present and analyze these models. Then, these models will be extended for more accurate prediction based on our aging-aware delay library. Finally, the accuracy of our flow and HSpice MOSRA [12] simulation will be examined.

The R-D model formulates the increase in V_{th} for NBTI [14] and HCI [15] as:

$$\Delta V_{th\_NBTI} = A_{NBTI0} \times t_{ox} \times \sqrt{C_{ox}(V_{dd} - V_{th0})} \times e^{(\frac{V_{dd}-V_{th0}}{t_{ox}E_0})} \times (e^{-\frac{E_a}{k}})^{\frac{1}{T}} \times t_{stress}^{0.25}, \quad (1)$$

$$\Delta V_{th\_HCI} = A_{HCI0} \times \alpha \times f \times e^{(\frac{V_{dd}-V_{th0}}{t_{ox}E_1})} \times t^{0.5}, \quad (2)$$

where t is the functional time of the transistor, t_{stress} is the effective stress time for NBTI, t_{ox} is the oxide thickness, C_{ox} is the gate capacitance per unit area, α and f are the activity factor and the frequency respectively, and E_o, E_a, k and E_1 are technology-dependent constants. A_{NBTI} and A_{HCI} are constants that depend on the aging rate. Based on Alpha-power law [16] and derivation in [10], we can obtain the general form of delay estimation for gate G_j as:

$$D_j = A_j \times (1 + B_j^{\frac{1}{T}} \cdot (t_{eff})^n), \quad (3)$$

where,

$$A_j = A_1,$$
$$B_j = (A_1 B_1 A_0)^T B_0,$$

A_j and B_j are gate-dependent variables, different for NBTI and HCI. Equation (3) provides a short-range aging-degradation prediction for NBTI and HCI considering temperature T, signal probability p, time t, activity factor α and frequency f. Note that *short-range* refers to the short Euclidian distance in the parameter space between each pair of aging conditions. In [17], the authors proposed another general form for short-range gate delay calculation as a function of the input slew rate τ_j and capacitive load C_j as a two-term posynomial equation:

$$D_j = \sum_{k=1}^{2} S_k \tau_j^{m_k} C_j^{n_k}, \quad (4)$$

where τ_j is the input slew rate, C_j is the capacitive load, $S_k \geq 0$ depends on gate feature size, and m_k and n_k are real numbers and do not depend on gate feature size.

However, these short-range models are not necessarily efficient for long-range aging-degradation accurate prediction in the parameter space of aging conditions. Compared with these models, reliability analysis tools regressively approximate the result for the unknown condition on the expense of significant CPU runtime to achieve high accuracy if the Euclidian distance between the two conditions is large. In our methodology, we target a more balanced trade-off between efficiency and accuracy. An aging-aware delay library is generated to extend the models for spatially long-range aging-degradation prediction. It stores delay and output slew rate information (obtained from HSpice MOSRA simulations) for each gate according to different aging conditions including temperature T, signal probability $p_{j,m}$ (m is input pin index on a multiple-input gate), stress time t, input slew rate τ_j, and capacitive load C_j. For an aging condition not available in the library, the delay D_j is obtained by regressive approximation from the closest conditions in the library using the delay models (3) and (4). During the approximation process, once a close condition is selected as reference condition, all other available close conditions will be used as intermediate conditions during the regression process. In this way, the computational complexity will be decreased dramatically to linear level while high accuracy is maintained, which will be shown in later sections. In this paper, we improve the accuracy error to around 5% compared with direct HSpice MOSRA simulations. However, the accuracy can be further improved. For example, using smaller granularities for the parameters is one of the effective ways to improve accuracy when generating the aging-aware delay library.

We treat a path with q gates as a connection of gate primitives. By propagating the slew rate forward, the i-th CRP delay Γ_{CRP_i} under a specific aging condition can be obtained as:

$$\Gamma_{CRP_i} = \sum_{j=0}^{q} D_j(T, \tau_j, t, C_j, p_{j,1}, ..., p_{j,m}), \quad (5)$$

Note that, (5) applies to any path as long as its aging information is available. Since there is no analytical model for output slew rate estimation, we use the delay aging ratio, defined as $\lambda_j = D_j/D_k$, to approximate the output slew rate. In the path delay calculation, the output slew rate of one gate is propagated to the following gate and is regarded as its input slew rate. Thus, we calculate the corresponding output slew rate as:

$$S_j = S_k \times \lambda_j, \tag{6}$$

where (D_k, S_k) is the delay and output slew rate stored in the aging-aware delay library corresponding to one of the closest conditions for approximation.

3.2 CRP Delay Analysis and Selection

CRPs are paths that may violate the timing constraint at some point in the field, that is, $\Gamma_{CRP_i} > \Gamma_{clk} - \Gamma_m$, where Γ_{CRP_i} is the delay of a CRP P_i in the field. For each PCP, aging information (signal stress probability, capacitive load, and switching activity) can be obtained using our in-house tools along with existing commercial tools. For example, the switching activity for gates on the PCPs can be calculated by gate-level simulation with our developed PLI routines. By collecting the switching information, the equivalent duty ratio for signal stress probability can be calculated for NBTI analysis [3]. Moreover, the switching activity can also be studied for HCI effect.

Algorithm 1 Aging-Aware Path Timing Analysis

```
1: Initialize delay of the i-th path Γ_pi ← 0
2: Specify temperature T
3: Specify stress time t
4: Specify the 1st gate input slew rate τ_0
5: for (all logic gates index j from 0 to n in i-th path) do
6:     Aging conditions collect for gate G_j:(C_j, p_j,1, ...,p_j,m)
7:     Loading aging-aware delay library
8:     Find all close k conditions (Γ_k, t_k, τ_k, C_k, p_k,1, ...,p_k,m)
9:     Initialize temporary delay D_t ← 0
10:    Initialize temporary output slew rate S_t ← 0
11:    for each close condition (Γ_k, t_k, τ_k, C_k, p_k,1, ...,p_k,m) do
12:        Data fitting for (D_t, S_t) using the math. models
13:    end for
14:    Finalize gate G_i approximation: (D_j, S_j) ← (D_t, S_t)
15:    Γ_pi+ = D_j
16:    τ_j+1 = S_j: propagate slew rate to next gate
17: end for
```

Once all the information is obtained, timing analysis is conducted to select the CRPs from PCPs using the aging-aware delay library. Generation of the aging-aware delay library is done only once for each technology library. Note that, in this work, we assume that the on-chip PLL that provides clock frequency is designed with reliable components robust to aging. However, we believe that our proposed flow can easily adopt impact of aging on clock tree as well. Algorithm 1 shows our proposed aging-aware path timing analysis procedure. Each path is regarded as a connection of gate primitives. Path delay Γ_{p_i} is initialized in Line 1; Lines 2 and 3 specify the temperature and stress time. Based on the aging condition, every closest condition stored in the aging-aware delay library is traversed from Lines 6 to 8. Lines 6 to 14 describe that, after being initialized in Lines 8 and 9, the gate delay and output slew rate can be approximated by using interpolation and extrapolation according to the delay models described in Section 3.1; Line 15 accumulates the gate delay for path delay analysis. Line 16 propagates the output slew rate forward ensuring accurate delay calculation. The process is repeated for every gate on the path to obtain the path delay as presented from Line 5 to Line 17. Considering N paths with M gates on average for each, complexity of the algorithm is $O(N \times M)$.

To verify the accuracy of our path delay calculation, we compare the delay calculated from our flow with that from HSpice MOSRA simulations for the top 100 paths extracted from the benchmark circuit s9234. The actual circuit condition when workload $WL = 0.5$ (meaning that the signal probability of being 1 at any primary input is 50%) for all parameters is considered and extracted using commercial tools. Aging degradation measured every two years for a span of ten years is shown in Figure 2. Each circle corresponds to one path at any time point, and the temperature is set to 75^oC. In Figure 3, the computational complexity between APD flow and HSpice MOSRA is compared, where the values show the ratio of HSpice MOSRA simulation time (on average for each path) over APD flow. On average, our flow shows ≥ 0.994 correlation with $\geq 244X$ speedup over the HSpice MOSRA simulation.

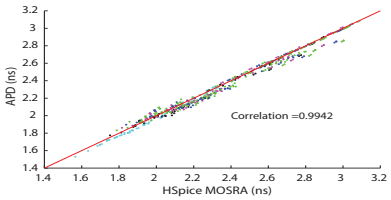


Figure 2: Accuracy analysis of our aging-aware path delay analysis flow compared with HSpice MOSRA.

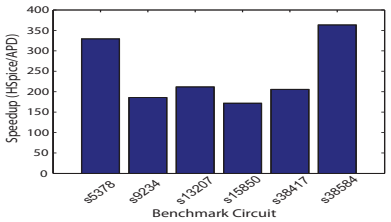


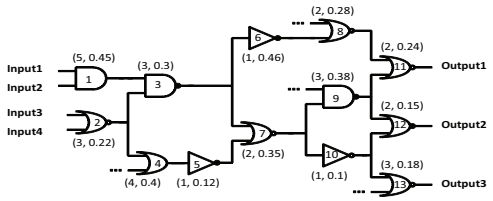
Figure 3: CPU runtime comparison between HSpice MOSRA and APD flow for different benchmark circuits.

4. CRG IDENTIFICATION TECHNIQUE

In this section, we evaluate the importance of the gates on CRPs based on their contribution to the paths' performance degradation and their relative impact on the CRPs. For the illustration purpose, Figure 4(a) shows a small circuit to demonstrate the difference between selecting gates with the largest degradation and selection of gates based on their impact on the paths. Each gate is numbered and the pair shows (d_0 and Δd_t), where d_0 is the gate delay at time zero and Δd_t shows the amount of gate degradation at time point t in the field. There are 14 paths in total in the circuit as shown in Figure 4(b). Also shown is the path delay at time zero (Γ_{t0}) and time t (Γ_{tt}). The gate and path lengths are shown using unit delay. The maximum path delay in addition to the margin (i.e. circuit timing) is set to 16 unit delays.

As seen, at time zero, all paths delay is under the timing budget; however, at time t, there are four paths (P2, P3, P11, and P12) above the timing budget, which will cause a failure in the circuit. The objective is to ensure that no path's timing become larger than the circuit timing budget.

Figure 4(c) shows the gates ordered based on their amount of aging at time t. To simplify the path delay calculation, we assume that once a gate is sized, it will not experience aging in the field. When selected based on the largest aging [10], gate G6 is chosen first. As seen in Figure 4(d), sizing this gate will impact paths P1 and P6, which are not the most critical ones. Next, gate G1 is selected, which reduces delay of paths P2 and P3. Selection of gate G4 will reduce delay of paths P11 and P12, and finally G9 will further reduce delay of paths P2, P3, P11, and P12, making all path length under the timing budget of 16 unit delay. However, using our technique, we will select only gates G9 and G7 to compensate for aging as shown in the Figure 4(d). This clearly shows that the most important gates to compensate for aging are not necessarily the gates that have aged the most.



(a)

	Path	Γ_{to}	Γ_{tt}
P_1	$G_1 \rightarrow G_3 \rightarrow G_6 \rightarrow G_8 \rightarrow G_{11} \rightarrow$ Output $_1$	13	14.73
P_2	$G_1 \rightarrow G_3 \rightarrow G_7 \rightarrow G_9 \rightarrow G_{11} \rightarrow$ Output $_1$	15	16.72
P_3	$G_1 \rightarrow G_3 \rightarrow G_7 \rightarrow G_9 \rightarrow G_{12} \rightarrow$ Output $_2$	15	16.63
P_4	$G_1 \rightarrow G_3 \rightarrow G_7 \rightarrow G_{10} \rightarrow G_{12} \rightarrow$ Output $_2$	13	14.3
P_5	$G_1 \rightarrow G_3 \rightarrow G_7 \rightarrow G_{10} \rightarrow G_{13} \rightarrow$ Output $_3$	14	15.38
P_6	$G_2 \rightarrow G_3 \rightarrow G_6 \rightarrow G_8 \rightarrow G_{11} \rightarrow$ Output $_1$	11	12.5
P_7	$G_2 \rightarrow G_3 \rightarrow G_7 \rightarrow G_9 \rightarrow G_{11} \rightarrow$ Output $_1$	13	14.49
P_8	$G_2 \rightarrow G_3 \rightarrow G_7 \rightarrow G_9 \rightarrow G_{12} \rightarrow$ Output $_2$	13	14.4
P_9	$G_2 \rightarrow G_3 \rightarrow G_7 \rightarrow G_{10} \rightarrow G_{12} \rightarrow$ Output $_2$	11	12.12
P_{10}	$G_2 \rightarrow G_3 \rightarrow G_7 \rightarrow G_{10} \rightarrow G_{13} \rightarrow$ Output $_3$	12	13.15
P_{11}	$G_2 \rightarrow G_4 \rightarrow G_5 \rightarrow G_7 \rightarrow G_9 \rightarrow G_{11} \rightarrow$ Output $_1$	15	16.71
P_{12}	$G_2 \rightarrow G_4 \rightarrow G_5 \rightarrow G_7 \rightarrow G_9 \rightarrow G_{12} \rightarrow$ Output $_2$	15	16.62
P_{13}	$G_2 \rightarrow G_4 \rightarrow G_5 \rightarrow G_7 \rightarrow G_{10} \rightarrow G_{12} \rightarrow$ Output $_2$	13	14.34
P_{14}	$G_2 \rightarrow G_4 \rightarrow G_5 \rightarrow G_7 \rightarrow G_{10} \rightarrow G_{13} \rightarrow$ Output $_3$	14	15.37

(b)

Gate	G_6	G_1	G_4	G_9	G_7	G_3	G_8	G_{11}	G_2	G_{13}	G_{12}	G_5	G_{10}
Δd_t	0.46	0.45	0.40	0.38	0.35	0.30	0.28	0.24	0.22	0.18	0.15	0.12	0.10

(c)

Selection Based on the Largest Aging					Selection Using Our Algorithm		
	6 → 1 → 4 → 9					9 → 7	
P_2	16.72	16.27	16.27	15.89	P_2	16.34	15.99
P_3	16.63	16.18	16.18	15.80	P_3	16.25	15.90
P_{11}	16.71	16.71	16.31	15.93	P_{11}	16.33	15.98
P_{12}	16.62	16.62	16.22	15.84	P_{12}	16.24	15.89

(d)

Figure 4: Demonstrating the CRG selection using the concept of importance to CRGs: (a) an illustrative circuit, (b) paths in the circuit and their timing, (c) order of gate aging at time t, and (d) the gate selection and sizing to compensate aging.

Based on above, a fast linear programming (LP) optimization technique is developed to identify the CRGs. Aging compensation will ensure that $\Gamma_{CRP_i} \leq \Gamma_{clk} - \Gamma_m$ throughout the chip's lifetime. In the LP, the predicted degradation of each gate is represented as a coefficient of the variable representing the importance of each gate. The optimal solution evaluates and quantifies the importance of each gate's contribution to the CRP degradation. Then, by ranking the importance of each gate, a minimum number of gates on each path can be selected to receive compensation for the lifetime degradation. Once the importance of each gate is identified, any pre-tapeout aging compensation techniques (e.g., gate sizing) can be implemented with a minimum area overhead. We formulate the aging compensation problem into an LP optimization function to quantify the importance of each gate on the CRPs:

$$\begin{aligned} minmax \quad &: \quad F(X) \\ subject\ to \quad &: \quad \Gamma_{CRP} - G \times X \leq \Gamma_{spec} \end{aligned} \tag{7}$$

where,

$$F(X) = \begin{bmatrix} f_{1,1}(x_1) & f_{1,2}(x_2) & \dots & f_{1,m}(x_m) \\ f_{2,1}(x_1) & f_{2,2}(x_2) & \dots & f_{2,m}(x_m) \\ & & \vdots & \\ f_{n,1}(x_1) & f_{n,2}(x_2) & \dots & f_{n,m}(x_m) \end{bmatrix}_{n \times m},$$

$$f_{i,j}(x_j) = \begin{cases} 1, & \text{if } G_j \text{ is in path } P_i \text{ and } x_j > 0 \\ 0, & \text{otherwise} \end{cases}$$

$$\Gamma_{spec} = \begin{bmatrix} \Gamma_{clk} - \Gamma_m \\ \Gamma_{clk} - \Gamma_m \\ \vdots \\ \Gamma_{clk} - \Gamma_m \\ \Gamma_{clk} - \Gamma_m \end{bmatrix}_{n \times 1}, \quad G = \begin{bmatrix} g_{1,1} & g_{1,2} & \dots & g_{1,m} \\ g_{2,1} & g_{2,2} & \dots & g_{2,m} \\ & & \vdots & \\ g_{n,1} & g_{n,2} & \dots & g_{n,m} \end{bmatrix}_{n \times m},$$

$$\Gamma_{CRP} = \begin{bmatrix} \Gamma_{CRP_1} & \Gamma_{CRP_2} & \dots & \Gamma_{CRP_n} \end{bmatrix}_{1 \times n},$$

$$X = \begin{bmatrix} x_1 & x_2 & \dots & x_m \end{bmatrix}_{1 \times m},$$

n is the number of CRPs and m is the number of unique gate instances on the n CRPs. Each row of matrix F and G corresponds to one CRP, while each column corresponds to one gate instance of all the non-duplicated gate instances on the CRPs. Γ_{CRP_i} is the i-th CRP's degraded delay, $g_{i,j}$ is gate degradation of the j-th gate on the i-th path with a stress time of t. Based on different lifetime timing specifications, $g_{i,j}$ is obtained from the CRP selection procedure discussed earlier. By solving the optimization problem, x_j $(j = 1, \dots, m)$ will be obtained. The larger the x_j value is, the more important the j-th gate instance is. X is initialized to be $X_0 = [0, 0, \cdots, 0]'_{1 \times m}$ to make all gates equally important at the beginning. After solving the LP optimization problem, the gates can be ranked according to their x values rather than their degradation values. By ranking the x values from high to low, each CRP P_i can find the smallest set of gates U_i whose compensation, once is applied, can make up for the paths degradation over the lifetime. Assume that P_i has M gates, i.e., $G_1, G_2, \dots, G_M$. Their delay degradations are $g_{i,1}, g_{i,2}, \dots, g_{i,M}$ and corresponding x values are $x_1 > x_2 > \dots > x_M$, respectively. Assume that the path delay Γ_{CRP_i} does not satisfy the timing requirement, i.e., $\Gamma_{CRP_i} > \Gamma_{clk} - \Gamma_m$ at some time point in the field. There are several methods [9][7], including gate sizing, that we can use to compensate for this degradation. For simplicity, we assume that once a gate is chosen for applying compensation on, its degradation is fully offset to 0 as $g_{i,j} = 0$. Then, we need to find the *critical length* l_i for the P_i to meet the specification. Note that the *critical length* l_i is the minimum number of gates necessary for aging compensation. As a result, the gate set $U_i = (G_1, G_2, \dots, G_{l_i})$ out of $U_{P_i}(G_1, G_2, \dots, G_M)$ is the i-th CRG set for P_i. The gate set $U = \bigcup_{i=1}^{n} U_i$ is the CRG set for the design.

Note that: (1) in the LP constraint function, $g_{i,j}$ represents the degradation for gate instance G_j. Thus, by replacing $g_{i,j}$ with the worst-case degradation, our flow is applicable to worst-case analysis; (2) our method of critical length l_i calculation for P_i is based on the assumption that $g_{i,j} = 0$ once the gate G_j is chosen for applying compensation. This is essentially assuming that the degradation of gate G_i will decrease at a scale of $g_{i,j}$ benefiting from the compensation scheme. Thus, by substituting the value $g_{i,j}$ with an actual

compensation value from a different compensation scheme, the LP optimization method is scalable to any aging compensation schemes for CRG identification. For example, for compensation using gate sizing, $g_{i,j}$ is calculated in the following way:

$$g_{i,j} = D_j(t) - D_{sub}(t), \quad (8)$$

where $D_j(t)$ represents the delay of the j-th gate at time t and $D_{sub}(t)$ represents that of the new gate. In addition, the gate sizing impact on the previous and next gates has to be considered as well. For example, when a gate is substituted for a new gate, its capacitive load has impact on the gate in the previous stage. Increasing gate size also impacts the output slew rate, which needs to be propagated forward. However this does not influence the flow of our method where gates are treated as a primitive unit and paths are treated as connection of gate primitives. Thus, the delay and degradation approximation can still be obtained from the flow with the capacitive load C_{j-1} changed in previous stage and output slew rate S_j changed in current stage.

To find the most critical gates for each CRP and obtain the entire CRG set U, [10] sorts all the gates on the CRPs according to their degradation from high to low as $G_1^*, G_2^*, \ldots, G_M^*$ with $g_{i,1}^* > g_{i,2}^* > \ldots > g_{i,M}^*$. The minimum l_i^* gates are found to meet the criterion $\Gamma_{CRP_i} - (g_{i,1}^* + \ldots + g_{i,l_i^*}^*) < \Gamma_{clk} - \Gamma_m$ and the l_i^* gates $(G_1^*, G_2^*, \ldots, G_{l_i^*}^*)$ are identified as CRGs for path P_i, and the united set of all the CRGs is the CRG set U^* for the design. In contrast, our LP technique offers several advantages over [10]: (1) Our flow is extendable to other aging effects beyond NBTI and HCI, (2) Our analysis of CRGs evaluates not only their own aging degradation, but also their contribution to CRPs degradation, and (3) Our aging analysis flow also enables the x_j to incorporate the topology-related information, such as capacitive load.

5. RESULTS

This section presents results obtained using our proposed flow on several ISCAS'89 benchmark circuits and Nangate *45nm* standard-cell library for CRP and CRG analysis. Experiments are run on a 64-bit Windows Desktop with a 3 GHz Intel Core 2 Duo CPU and 4 GB of memory. Each benchmark circuit is processed using commercial ASIC design tools (Design Compiler, Prime Time, VCS, etc.) to extract the aging information. For the first simulation, the CRGs in the benchmark s38417 are identified for different timing margins (Γ_m) over ten years considering NBTI and HCI effects at three workload scenarios (WL = 0.25, 0.5, and 0.75). The proposed flow is run to find the CRG set for the s38417 benchmark circuit; $g_{i,j} = 0$ is still held while deciding the critical length l_i for CRP P_i, for simplicity. The results are shown in Figure 5, which indicate that a more stringent timing margin will lead to a larger CRG set. For example, when the timing margin Γ_m is equivalent to 10% of the largest delay at time 0 when WL = 0.5, 176 gates are identified as CRG; while at the same WL value, when the timing margin Γ_m is equivalent to 5%, 493 gates are identified as CRG.

The two timing margin values 5% and 10% are also used to repeat the simulations with the stress time changing from $0.6E+08s$ ($\approx 2\,years$) to $3.0E+08s$ ($\approx 10\,years$). Results in Figure 6 indicate that more gates are identified as CRGs when the target lifetime is longer. Meanwhile, to satisfy the same lifetime specification, a smaller timing margin leads to more identified CRGs. The results also demonstrate that circuit lifetime can be extended by only introducing a small area overhead as the minimum gate set is selected for aging compensation, because resources are efficiently allocated to CRGs only.

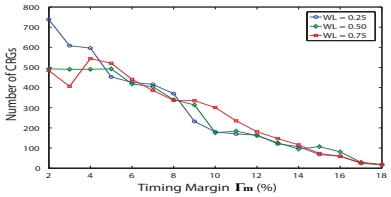


Figure 5: Number of CRGs for different timing margins (when $T = 75^oC, WL = 0.5$, and $t = 10$ years).

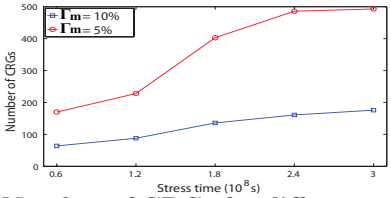


Figure 6: Number of CRGs for different stress times (when $T = 75^oC, WL = 0.5$).

Further simulations are conducted for several benchmark circuits with three timing margin values (2%,7%,12%). The results are shown in Table 1. Columns 2 and 3 list the total gate and flip-flop count. Under each timing margin category, the first and the second columns list the PCPs and CRPs, respectively. The CRGs in the third column are obtained using our flow. Finally, the fourth column shows the area overhead due to sizing the CRGs. Figures 7 and 8 demonstrate the effectiveness of our CRP and CRG identification. Gate sizing is conducted on the CRGs identified by our flow. In each figure, at each time point, the number of CRPs and the number of CRGs identified are presented in the form of CRP/CRG. The horizontal line (in green) is the timing specification for the design at that time point including the timing margin. The longer the stress time is, the more CRPs and CRGs are identified, indicating that a longer lifetime requirement leads to more CRGs. In Figure 7, $\Gamma_m = 5\%$; while in Figure 8, $\Gamma_m = 10\%$. The CRP delays before gate sizing are checked and presented (in red) above the horizontal line (in green), without exception that their delays violate the timing specification at different time point. After sizing the CRGs, the CRP delays are effectively reduced and are below the horizontal line accordingly (in blue). In the figures, the CRP delays before and after gate sizing are not displayed vertically on a same line at each time point just for demonstration purpose. In Figure 9, the stress time is fixed with a varying Γ_m, which shows that a stringent timing margin leads to more CRGs. From the results, we can see that aging compensation on the CRGs can effectively reduce the aging degradation and satisfy the design with the timing specification for the lifetime operation.

These experimental results demonstrate that circuit performance degradation can be minimized if early prediction and optimization is conducted during the design stage, and that circuit aging degradation can be compensated for by focusing on a minimum set of CRGs. The experimental results also show that we can efficiently identify a small number of CRPs and CRGs. After identifying the CRGs, other aging compensation mechanisms can also be utilized more effectively with lower area and power overheads.

Table 1: CRP and CRG identification for three timing margins with 10 years NBTI and HCI degradation ($T = 75^oC, WL = 0.5$)

Bench. Circuit	# of Gates	# of FFs	$\Gamma_m = 2\%$ PCP	CRP	CRG	$Area_o$(%)	$\Gamma_m = 7\%$ PCP	CRP	CRG	$Area_o$(%)	$\Gamma_m = 12\%$ PCP	CRP	CRG	$Area_o$(%)
s5378	1324	153	94	87	106	1.1479	83	52	41	0.5000	45	16	19	0.2592
s9234	1120	125	85	49	29	0.4934	50	26	24	0.4464	25	10	10	0.1645
s13207	1477	245	19	13	49	0.5550	7	4	21	0.1682	4	2	5	0.0168
s15850	3338	452	890	865	21	0.1936	289	273	19	0.2005	34	33	11	0.1037
s38417	10221	1523	9671	5616	493	1.1252	4927	1922	404	0.6881	1689	485	162	0.2677
s38584	12482	1246	293	267	137	0.3349	164	139	67	0.1708	55	39	15	0.0225

PCP: potential critical path; CRP: critical-reliability path; CRG: critical-reliability gate;
$Area_o$: area overhead; FF: flip-flop; Γ_m: timing margin.

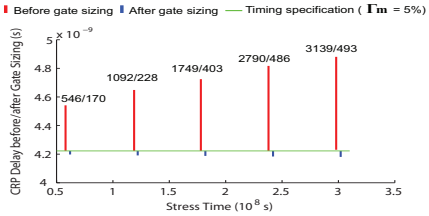


Figure 7: CRP delay comparison before/after gate sizing on the CRGs with varying stress time (when $\Gamma_m = 5\%, WL = 0.5$, and $T = 75^oC$).

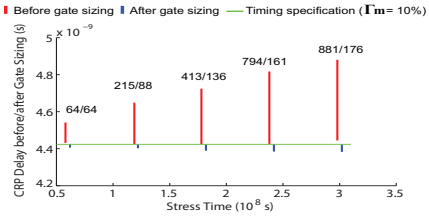


Figure 8: CRP delay comparison before/after gate sizing on the CRGs with varying stress time (when $\Gamma_m = 10\%, WL = 0.5$, and $T = 75^oC$).

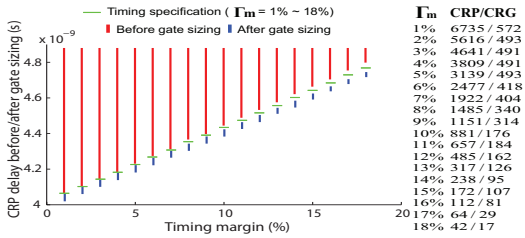


Figure 9: CRP delay comparison before/after gate sizing on the CRGs with varying Γ_m (when $WL = 0.5, T = 75^oC$, and $t = 10$ years).

6. CONCLUSIONS

In this paper, a new technique is proposed to quickly identify CRPs and CRGs. Aging compensation can therefore be focused only on the minimum number of CRGs as they contribute the most to path delay degradation. The flow is easy to be integrated into conventional industry IC design flow. Experimental results indicate that our technique is able to identify the minimum number of CRGs, minimize area overhead for design margining, and ensure performance throughout its lifetime operation.

7. ACKNOWLEDGMENTS

This work was supported in part by Semiconductor Research Corporation (SRC) under grants 2053 and 2094, and a gift from Cisco.

8. REFERENCES

[1] T. Nigam, "Impact of Transistor Level Degradation on Product Reliability," *Custom Integrated Circuits Conference*, pp. 431-438, September 2009.

[2] G. Chen, M. Li, C. Ang, J. Zheng, and D. Kwong, "Dynamic NBTI of p-MOS Transistors and Its Impact on MOSFET Scaling," *IEEE Electronic Device Letter*, vol. 23, pp. 734-736, December 2002.

[3] S. V. Kumar, C. H. Kim, and S. S. Sapatnekar, "An Analytical Model for Negative Bias Temperature Instability (NBTI)," in Proceeding of the *IEEE/ACM International Conference for Computer Aided Design*, pp. 493-496, November 2006.

[4] L. F. Wu, J. K. Fang, H. Yonezawa, Y. Kawakami, N. Iwanishi, H. Yan, P. Chen, A. Chen, N. Koike, Y. Okamoto, C. Yeh, and Z. Liu, "GLACIER: A Hot Carrier Gate Level Circuit Characterization and Simulation System for VLSI Design," *ISQED*, pp. 73-79, 2000.

[5] M. Dai, C. Gao, K. Yap, Y. Shan, Z. Cao, K. L, L. Wang, B. Cheng, S. Liu, "A Model With Temperature-Dependent Exponent for Hot-Carrier Injection in High-Voltage nMOSFETs Involving Hot-Hole Injection and Dispersion," *IEEE Transactions on Electron Devices*, pp. 1255-1258, April 2008.

[6] B. Paul, K. Kang, H. Kufluoglu, M. Alam, and K. Roy, "Temporal Performance Degradation under NBTI: Estimation and Design for Improved Reliability of Nanoscale Circuits," *Design Automation and Test in Europe*, pp. 1-6, March 2006.

[7] S. Kumar, C. Kim, S. Sapatnekar, "NBTI-Aware Synthesis of Digital Circuits," *Design Automation Conference*, pp. 370-375, June 2007.

[8] S. Kumar, C. Kim, and S. Sapatnekar, "Adaptive Techniques for Overcoming Performance Degradation due to Aging in Digital Circuits," *Design Automation Conference*, pp. 284-289, 2009.

[9] E. Mintarno, J. Skaf, R. Zhang, J. Velamala, Y. Cao, S. Boyd, R. Dutton, and S. Mitra, "Optimized Self-Tuning for Circuit Aging," *Design, Automation & Test in Europe Conference & Exhibition DATE*, pp. 586-591, April 2010.

[10] W. Wang, Z. Wei, S. Yang, and Y. Cao, "An efficient Method to Identify Critical Gates under Circuit Aging," *International Conference on Computer Aided Design*, pp. 735-740, 2007.

[11] N. Shah, R. Samanta, M. Zhang, J. Hu, and D. Walker, "Built-In Proactive Tuning System for Circuit Aging Resilience," *IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems*, pp. 96-104, 2008.

[12] Synopsys, *http://www.synopsys.com/*

[13] S. Ogawa and N. Shiono, "Generalized Diffusion-Reaction Model for the Low-Field Charge Build up Instability at the $Si - SiO_2$ Interface," in *Phys. Rev. B.*, Vol. 51, Issue 7, pp. 4218-4230, February 1995.

[14] R. Vattikonda, W. Wang, and Y. Cao, "Modeling and Minimization of PMOS NBTI Effect for Robust Nanometer Design," In *Design Automation Conference*, pp. 1047, 2006.

[15] E. Takeda, C. Yang, and A. Miura-Hamada, "Hot-Carrier Effects in MOS Devices," *Academic Press*, 1995.

[16] T. Sakurai, et. al., "Alpha-Power Law MOSFET Model and Its Application to CMOS Logic," *JSSC*, pp. 548-594, 1990.

[17] H. Tennakoon, and C. Sechen, "Efficient and Accurate Gate Sizing with Piecewise Convex Delay Models," *Design Automation Conference*, pp. 807-812, June 2005.

[18] W. Wang, S. Yang, S. Bhardwaj, R. Vattikonda, S. Vrudhula, F. Liu, and Y. Cao, "The Impact of NBTI on the Performance of Combinational and Sequential Circuits," *Design Automation Conference*, pp. 364-369, Jun. 2007.