2014 27th International Conference on VLSI Design and 2014 13th International Conference on Embedded Systems

Reliability Aware Gate Sizing Combating NBTI and Oxide Breakdown

Subhendu Roy and David Z. Pan
Department of Electrical and Computer Engineering, University of Texas at Austin, USA
subhendu@utexas.edu, dpan@ece.utexas.edu

***Abstract*—Negative Bias Temperature Instability (NBTI) and Oxide Breakdown (OBD) are two key reliability concerns for nanometer VLSI circuits. Gate over-sizing has been done in the past to mitigate the effect of NBTI and aging to meet performance constraints. However, this could make the entire circuit more prone to OBD. In this paper, we propose a new gate sizing formulation that considers both NBTI-induced delay degradation and OBD-induced circuit lifetime. Since NBTI and OBD are highly sensitive to the input vectors in a conflicting way, we consider their dependencies on signal probabilities. Moreover, we take into account the degradation in rise slew due to NBTI which could affect the fall delay/slew of the inverting gates in the next stage, and this has not been considered in previous work on NBTI aware gate sizing. Experimental results on industry strength benchmarks demonstrate that by incorporating OBD into holistic gate sizing, we can achieve more reliable circuit without compromising the circuit performance and area.**

I. Introduction

In nanometer VLSI, circuit reliability has become a prime concern. Negative bias temperature instability (NBTI) and oxide breakdown (OBD) contribute significantly in worsening the circuit reliability, and as technology scales, these effects become more and more prominent. OBD is more prevalent in NMOS devices [1]. When an NMOS device is subjected to a gate voltage stress ($V_{gs} = V_{dd}$) over a long time, OBD results in a conducting path between the gate and channel. Among the possible modes of an NMOS device, $[1, 0, 0]$ configuration of $[gate, drain, source]$ causes functional failures due to OBD [2]. Thus, the probability of oxide breakdown of the device depends on the time it remains in this configuration, which has not been considered in earlier works [3][4]. On the contrary, NBTI generally occurs in PMOS devices, which is manifested by an increase in threshold voltage (V_{th}). This is due to the generation of interface traps under negative gate-to-source bias (stress phase). The positive gate-to-source bias helps in annealing some of the interface traps, thereby leading to partial recovery. This phase is known as the recovery phase. However, this recovery is never complete [5]. Several models have been developed to predict the shift in V_{th} due to NBTI considering this [6][7][8]. The increase in V_{th} results in increased delay and the amount of delay degradation depends on this stress period. For instance, NBTI can cause over 20% degradation in circuit speed in 10 years [9].

The commonly used techniques to mitigate NBTI impact include gate sizing and input vector control (IVC). Gate sizing with V_{th} selection is a very fundamental physical synthesis step to meet timing closure. With technology scaling, leakage power has become an important metric for gate sizing along with the conventional performance metric as shown by a recent work [10]. Also, reliability issues such as OBD and NBTI can be mitigated by gate sizing. For instance, [2] formulates a gate sizing problem to reduce the probability in OBD. In [11], NBTI aware gate sizing problem is formulated by adding a delay-degradation component in individual gates to the conventional gate sizing problem. A relevant work has been presented in [5] to synthesize NBTI aware digital circuits.

However, the objectives for NBTI and OBD are often conflicting with each other, as to be shown in Section II. The approaches in [5][11] try to mitigate the impact of NBTI by oversizing the gates, but that would cause more serious OBD issues [12]. Moreover, these approaches only considered the impact of NBTI on rise delay of the gates, but not the degradation in rise slew due to NBTI which would translate to a non-negligible increase in fall delay of the inverting gates (NAND, NOR, NOT etc.) in the next stage. Another effective approach to combat NBTI is IVC technique [13] due to the dependence of NBTI on duty cycle. But input vector selection (to alleviate NBTI) can make circuits more susceptible to oxide breakdown due to the conflicting stress conditions for NBTI (PMOS) and OBD (NMOS).

In this paper, we build a unified gate sizing algorithm which considers NBTI and OBD along with the traditional metric (power and performance) of gate sizing. We develop a static timing analysis (STA) engine and a discrete gate sizer which use a very realistic and accurate delay model based on the recent ISPD'12 Discrete Gate Sizing Contest. We pre-characterize the NBTI factors for input-to-output timing arcs (which correspond to stress phase of the pertaining PMOS) of the gates due to NBTI based on SP (signal probability which denotes the probability that a signal is at logic '1') and build a piecewise linear model of rise-delay/rise-slew with NBTI factors. This model is embedded in our sizer to perform NBTI aware gate sizing which degrades circuit lifetime due to OBD. We derive a metric for OBD at the circuit level and perform reliability aware gate sizing by modifying the cost/metric (that again depends on pre-characterized OBD factors) in the sizer algorithm. The key contributions of our paper are as follows:

- To the best of our knowledge, this is the first work which considers NBTI and oxide breakdown simultaneously.
- Previous work did not consider NBTI-induced rise slew degradation which can increase the fall delay of the inverting gates in the next stage. We have characterized the slew degradation due to NBTI and our gate sizing algorithm is the first to take into account this.
- We develop a holistic framework for reliability aware gate sizing which can perform smooth trade-off between circuit reliability and leakage power, without compromising circuit performance and area. However, the reliability characterization is flexible to be integrated into other discrete gate sizers.

The rest of the paper is organized as follows. Section II illustrates the problem formulation. Section III presents the basic algorithm for gate sizer with the sizer result in the same section itself as this is not our main contribution. Section IV illustrates the reliability characterization for NBTI/OBD and Section V presents the reliability aware unified gate sizing algorithm. Finally, Section VI presents the experimental results with a conclusion in Section VII.

II. Problem Formulation

Besides the impact of NBTI on PMOS, the effect of Positive Bias Temperature Instability (PBTI) is not negligible for NMOS with high-k dielectric, and along with hot carrier injection, it can cause a shift in V_{th} [14]. However, in this current work, we focus on NBTI and OBD and in future, we plan to extend the gate-sizing formulation by incorporating PBTI models in the STA engine. Fig. 1 shows that the input stress conditions for NBTI (PMOS) and OBD (NMOS) are reciprocal to each other. So IVC technique to mitigate NBTI can be

1063-9667/14 $31.00 © 2014 IEEE
DOI 10.1109/VLSID.2014.14

IEEE computer society

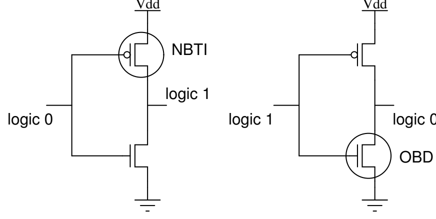

Fig. 1: Conflicting input stress conditions for NBTI in PMOS and OBD in NMOS

detrimental to OBD. The probability of OBD for an NMOS device i at time t is given by the Weibull distribution [2].

$$P^i_{BD}(t) = 1 - \exp(-(\frac{\gamma^i_{obd} t}{\alpha})^\beta a_i) \quad (1)$$

where α is a constant and β is the Weibull shape factor. γ^i_{obd} is the OBD factor corresponding to $[1, 0, 0]$ configuration of $[gate, drain, source]$ and a_i is the effective area (scaled w.r.t. the min-sized inverter) of the device. Eqn.(1) signifies that this probability increases with area, which renders oversizing to have adverse effect on OBD. To obtain the objective function of our gate sizing formulation, we first derive the OBD parameter of a circuit (F_{obd}).

Suppose a gate contains n NMOS devices and we intend to find the probability of failure for the gate. This problem can be modeled as weakest-link-failure problem, *i.e.*, the failure of any device would lead to functional failure of the logic gate. Using Eqn.(1) the probability of failure of the gate is given by

$$\begin{aligned} P_{gate} &= 1 - \prod_i (1 - P^i_{BD}(t)) \\ \Rightarrow P_{gate} &= 1 - \exp(-\sum_i (p_i a_i)) \end{aligned} \quad (2)$$

where $p_i = (\frac{\gamma^i_{obd} t}{\alpha})^\beta$.

$$\begin{aligned} \sum_i (p_i a_i) &= \sum_i ((\gamma^i_{obd})^\beta (\frac{t}{\alpha})^\beta a_i) \\ \Rightarrow \sum_i (p_i a_i) &= (\frac{t}{\alpha})^\beta A \sum_i ((\gamma^i_{obd})^\beta (\frac{a_i}{A})) \end{aligned} \quad (3)$$

where A is the area of the gate. The OBD factor (γ^{gate}_{obd}) for a gate is defined as $\sum_i ((\gamma^i_{obd})^\beta (\frac{a_i}{A}))$. It is to be noted that $(\frac{a_i}{A})$ is constant for a particular gate topology and γ^i_{obd} can be predetermined based on SP at the circuit nodes (to be illustrated later). Also, the failure of any gate would cause failure in the circuit. Therefore, the probability of circuit failure is given by the following equation.

$$\begin{aligned} P_{ckt} &= 1 - \prod_j (1 - P^j_{gate}(t)) \\ \Rightarrow P_{ckt} &= 1 - \exp[-(\frac{t}{\alpha})^\beta \sum_j (\gamma^{gate,j}_{obd} A_j)] \end{aligned} \quad (4)$$

Here A_j = area of the jth gate and $\gamma^{gate,j}_{obd}$ = OBD factor for the jth gate.

We define the metric "OBD parameter of a circuit" as

$$F_{obd} = \sum_j (\gamma^{gate,j}_{obd} A_j) \quad (5)$$

Next we will establish the dependence of circuit lifetime degradation due to OBD on F_{obd}. The Weibull distribution of time-to-failure for the logic circuit is given by:

$$\begin{aligned} W(t) &= \log(-(\log(1 - P_{ckt}(t)))) \\ &= \beta \log(\frac{t}{\alpha}) + \log(F_{obd}) \end{aligned} \quad (6)$$

The lifetime of a circuit is defined as the time corresponding to a specified failure probability W. If t_1 and t_2 are the lifetimes corresponding to F^1_{obd} and F^2_{obd} respectively, then

$$\begin{aligned} \beta \log(\frac{t_1}{\alpha}) + \log(F^1_{obd}) &= \beta \log(\frac{t_2}{\alpha}) + \log(F^2_{obd}) \\ \Rightarrow \frac{t_2}{t_1} &= \exp(\frac{1}{\beta} \log \frac{F^1_{obd}}{F^2_{obd}}) \\ &= (\frac{F^1_{obd}}{F^2_{obd}})^{\frac{1}{\beta}} \end{aligned}$$

Therefore, increasing F_{obd} from F^1_{obd} to F^2_{obd} degrades the circuit lifetime by a factor

$$\frac{t_2}{t_1} = (\frac{F^1_{obd}}{F^2_{obd}})^{\frac{1}{\beta}} \quad (7)$$

With technology scaling beyond 32/22nm, leakage power (P_{leak}) contribution is now as significant as dynamic power and it is a prime metric for gate sizing as considered in the most recent ISPD'12/13 Discrete Gate Sizing Contest [15]. Apart from the dependence on gate sizes (like F_{obd}), P_{leak} also depends on threshold levels of the gates. Downsizing gates can reduce both F_{obd} and P_{leak}, but that would render the design to fail in meeting timing constraints. On the other-hand, low V_t cells can improve the timing at the expense of increased leakage power. In [16][17] leakage power is optimized by using multi-V_t cells along with gate sizing. So the right balance between the usage of low V_t cells and gate up-sizing is important to achieve a trade-off between F_{obd} and P_{leak} under same timing constraints. Since $\beta \simeq 1$ for lower technology node [12], by Eqn.(7), circuit lifetime due to OBD has a linear dependence on F_{obd}. So we define the objective function (F_{metric}) of our formulation as the linear combination of P_{leak} and F_{obd}, given by

$$F_{metric} = P_{leak} + wF_{obd} \quad (8)$$

where w is a parameter signifying the relative weight of F_{obd} over P_{leak}.

The problem formulation for reliability aware gate sizing is as follows:

$$\begin{aligned} \text{minimize:} \quad & F_{metric} \\ \text{subject to:} \quad & T^{nbti}_{delay}(max) \leq T_{clk} \end{aligned} \quad (9)$$

where $T^{nbti}_{delay}(max)$ is the maximum delay (NBTI affected) from timing start point to timing end point, T_{clk} is the clock period.

III. Framework for Gate Sizing

The discrete gate sizing problem is an NP-hard problem [18] and available sizers can only give a suboptimal solution [19]. This is further confirmed by the recent ISPD'12 contest [15], where none of the contestants dominated on the entire benchmark suite. We develop a gate sizer based on successive local refinement (SLR) technique and it can scale to the designs of approximately one million gates. However, the sizer is not our main contribution in this work, rather a framework built (due to the lack of high-quality open-source discrete gate-sizers) to demonstrate how NBTI/OBD can be simultaneously handled by gate-sizing.

We have used the standard cell library (32nm process technology) provided by ISPD'12 contest organizers [15]. It is a 2-D look-up-table for cell delays and output slews, as a function of input slew and output capacitance. The delay model is non-convex and very realistic, and it is built using current-source model that comprehends transition slope, threshold voltage and the ratio of driver size to the load. The cell library contains a set of cell footprints, *e.g.*, NAND2 (2 input NAND), NAND3, NAND4, NOR2, NOR3, NOR4, AOI12 (And-Or-Invert with 3 input), AOI22, OAI12 (Or-And-Invert with 3 input),

OAI22. We develop an STA engine and the rising/falling slacks/slews in ps are matched with the results of Synopsys Primetime, which are accurate to 3 decimal places. The details of the delay-model/cell-library/STA model/leakage power calculation can be found in [15].

In [20], weights are assigned to critical nets for placement driven synthesis. We have extended this notion in the frame of gate sizing by defining a parameter figure of merit $fom(n)$ for any net n. $fom(n)$ signifies the timing criticality of the net and represents the number of timing end points with negative slack, for which the critical path traverses through that net. Consider an example as shown in Fig. 2 representing a part of a circuit. In this circuit, b, c, h are not critical from timing perspective, so higher threshold level can be chosen for Gate 3 or it can be downsized to optimize leakage power or OBD metric. On the contrary, gates (like 2) driving critical nets should be up-sized and/or lower threshold level should be used for these gates to meet timing and the extent of up-sizing depends on fom value of the driven net. In Fig. 2, fom of f is 2, which means there are 2 timing end points with negative slack, for which critical path passes through f. The critical net for both the gates 4 and 5 is e, therefore total number of critical paths through e is $fom(e) = fom(f) + fom(g)$ = 2 + 4 = 6.

Algorithm 1 presents the key algorithmic steps for the discrete gate sizer. Level is assigned to each cell (Line 2), which is the logical depth of its output net. The $fom(n)$ of each net n is calculated by the subroutine 'calculateFOM' as explained earlier (Line 4). In each iteration, local refinement is done at the increasing level of cells. In Line 9, $CellSetAt(i)$ represents the set of all cells which are at level i. Then we extract the footprint fp (e.g."nand02","nor02") of the cell, and for each cell-type (ct) defined in the cell library with that footprint we calculate the cost function or metric (Line 14) which is a weighted sum of $NS(n)$ (normalized slack of the output net n w.r.t. the clock period) and $NP(ct)$ (normalized power of the cell with type ct w.r.t. the maximum power of a cell in the library). In Line 13, $T(fp)$ represents the available cell types with the same footprint fp of the original cell.

For the baseline gate sizer, w (in Eqn.(8)) is equal to 0 and this cost function/metric is later modified in Algorithm 2 to take into account of OBD. The parameter $fom(n)$ is used in the coefficient of $NS(n)$, providing more weight for slack to the critical nets. The slack is calculated assuming the required time of arrival of the output net of the cell from the previous iteration and the arrival time (AT) from the current iteration with ct. We select the cell type which maximizes the metric (M). However, oversizing a gate could adversely affect the timing as well. This is explained by considering the earlier example (Fig.2). The net e has a fan-out of two, one to the inverter 4 and another to the NAND Gate 5. Clearly, oversizing the Gate 5 will increase the capacitance of e, thereby increasing the delay of the Gate 2. This might increase the slack at g, but would make the path a-d-e-f more critical. On the contrary, the input nets of Gate 2 are either not critical (net b) or have single fan-out (net d). So oversizing the Gate 2 would not potentially worsen the circuit timing. To take care of this, slack degradation in the input nets multiplied by its fom is added to the metric (not shown in Algorithm 1).

The STA engine is run and fom for each net is updated at the end of each iteration. The iterations in Algorithm 1 are continued until there is no further improvement in terms of slack violations and leakage power minimization or the number of iteration reaches a maximum limit ($MAXITER$). The weight factor δ (Line 14) is changed according to the slack of the corresponding net. At the initial iterations, when the slack at a net is negative, the value of δ is kept low and as the net starts to attain more and more positive slack, this coefficient is increased quadratically with the slack. At the end, a post-optimization legalization step is performed to get the timing closure, if required. This subroutine traverses from timing end point to timing start point, and if it finds negative slack at a net, it changes the cell (which drives the net) from high-threshold to low-threshold type, without changing its size. The size of the cell remaining unchanged keeps the previous stage gate delays unaltered.

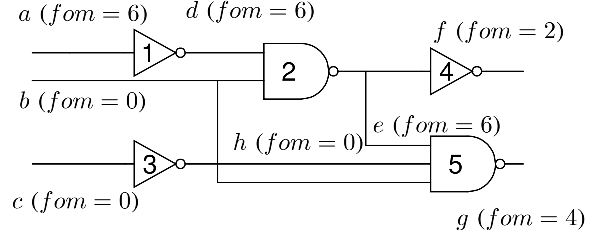

Fig. 2: Illustrative example for FOM

Algorithm 1 Successive Local Refinement (SLR)

1: //Given timing constraints, optimize Leakage Power ($w = 0$)
2: Assign level to each cell (1 to l);
3: runSTA;
4: calculateFOM;
5: $iter \leftarrow 0$;
6: **repeat**
7: $iter \leftarrow iter + 1$;
8: **for** $i = 1$ to l **do**
9: $C \leftarrow CellSetAt(i)$;
10: **for all** $c \in C$ **do**
11: $fp \leftarrow footprint(c)$;
12: $n \leftarrow outputNet(c)$;
13: **for all** $t \in T(fp)$ **do**
14: $M = fom(n) \times NS(n) + \delta \times [1 - NP(ct)]$;
15: **end for**
16: Select ct with max M;
17: **end for**
18: **end for**
19: runSTA;
20: calculateFOM;
21: **until** $iter == MAXITER$ **or** No Improvement
22: postOptimizationLegalization;

We run this sizer for all the benchmarks in [15] and it is able to size all of them with zero slack/max-load violations (worst negative slack or WNS = 0) within the hard run time limit for each benchmark, given in the contest. Table I compares the leakage power (P_{leak}) and runtime (T_{run}) of our baseline sizer with the best sizer (NTUgs) published in [15]. Column 2 represents the circuit size in terms of number of nets in the design. We can see that SLR gives competitive result for the larger benchmarks. Run time is high (it is less compared to NTUgs) but as the inner 'for' loops (Line 10 - 17) in Algorithm 1 is parallelizable, multi-core implementation has the potential to reduce the run time of the algorithm to a great extent.

TABLE I: Baseline sizer result

Design	Size	P_{leak} (W)		T_{run} (hr)	
		SLR	NTUgs	SLR	NTUgs
DMA	25,301	0.282	0.205	2.2	2.2
pci_bridge32	33,303	0.207	0.203	2.5	2.3
des_perf	111,229	1.04	0.674	3.4	7.0
vga_lcd	164,891	0.529	0.415	4.4	9.0
b19	219,268	0.99	0.627	5.3	11.0
leon3mp	649,191	1.49	1.42	10.8	22.5
netcard	958,780	1.78	1.77	14.4	29.0

IV. Reliability Characterization

As the stress time for NBTI (OBD) is different for individual PMOS (NMOS) devices within a gate, we pre-characterize NBTI factors and OBD factors for all gates in the cell library based on the signal probabilities at the gate-inputs.

NBTI Aware Timing analysis: The NBTI factors impact the rise-time delay and rise-slew of the timing arcs from gate inputs to the gate output. To capture it, we use the equations of s-factor model [5] to calculate the s values for different NBTI factors at the end of 5 years, and based on that we estimate the degradation in V_{th} with NBTI factors (γ_{nbti}). Next, we plug in the final V_{th} in HSpice simulation to get the NBTI-affected rise delay and rise slew. Then we develop a piecewise linear model (within 1% accuracy to HSpice simulation data) for rise delay/rise slew with NBTI factors, similar to [21], which is used in timing analysis. So we get rise delay/slew of gates as a function of input slew and output load from the cell library, and then scale it according to the piecewise-linear model to calculate the rise delay/slew in presence of NBTI. Fig. 3 shows the rise delay/slew characterization for an inverter and we can see that the maximum degradation in rise slew (30%) is comparable and even more than the maximum degradation in rise delay (22%). As this will affect the fall delay and fall slew of the inverting gates in the next stage, we need to consider the NBTI-induced slew degradation as well to build an accurate STA engine in presence of NBTI.

To illustrate this, let us consider a chain of inverters as shown in Fig. 4. We define any node in this chain as a quadruplet [rAT, rS, fAT, fS], where rAT = rise arrival time, rS = rise slew, fAT = fall arrival time and fS = fall slew. Without any NBTI impact, say all the stages have a rise/fall delay of 10ps and the nodes are given by $X_1 = [10, 10, 10, 10]$, $X_2 = [20, 10, 20, 10]$ and $X_3 = [30, 10, 30, 10]$. If we assume signal probability at the nodes as 0.5 and consider the impact of NBTI only on rise delay, the nodes will be represented as $X_1 = [11.8, 10, 10, 10]$, $X_2 = [21.8, 10, 21.8, 10]$ and $X_3 = [33.6, 10, 31.8, 10]$. Now if we consider the rise-slew degradation due to NBTI as well, the nodes will be $X_1 = [11.8, 12.3, 10, 10]$, $X_2 = [21.8, 12.3, 22.4, 10.6]$ and $X_3 = [34.4, 10.2, 32.4, 10.6]$. Please note that, these numbers have been generated using the NBTI characterization data and the cell-library delay values. As the rS at X_1 changes to 12.3ps, the fall delay in Gate 2 changes from 10ps to 10.6ps, thus making fAT at X_2 to $11.8 + 10.6 = 22.4$ps. Also, the increased fS at X_2 (10.6ps) causes the rise delay for Gate 3 to 12ps and rAT at X_3 becomes $22.4 + 12 = 34.4$ps.

Therefore when rise slew degradation is considered, the increase in delay due to NBTI is $34.4-30 = 4.4$ps. On the contrary, the increase in delay is $33.6 - 30 = 3.6$ps without this consideration and thus accuracy in estimating the impact of NBTI on circuit timing would have been worsened by around 18% by neglecting NBTI-induced rise slew increase.

The NBTI factor corresponds to the stress phase of the individual PMOS transistors. Let us take an example of "AOI12" gate (Fig. 5), for which the function signature is $F = \overline{A + BC}$. Let SP_A, SP_B and SP_C respectively denote the signal probabilities at A, B and C. γ_{nbti} at the inputs B and C are simply the probability that they are at logic '0', where as for input A to be at stress phase, apart from A being at logic '0', either the input B or input C needs to be at logic '0'. Thus γ_{nbti}'s for inputs of "AOI12" gate are given by the following equations:

$$\begin{aligned} \gamma^B_{nbti} &= (1 - SP_B) \\ \gamma^C_{nbti} &= (1 - SP_C) \\ \gamma^A_{nbti} &= (1 - SP_A)(1 - SP_B SP_C) \end{aligned} \quad (10)$$

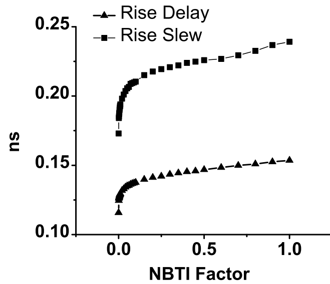

Fig. 3: Rise delay and rise slew vs. NBTI factor in an inverter

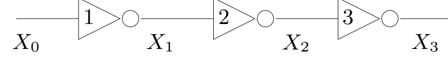

Fig. 4: Impact of NBTI-induced rise slew degradation on timing

TABLE II: NBTI factors for some gate topologies

Gate	NBTI Factor
$F = \overline{AB}$	$\gamma^A_{nbti} = (1 - SP_A)$ $\gamma^B_{nbti} = (1 - SP_B)$
$F = \overline{A + B}$	$\gamma^A_{nbti} = (1 - SP_A)(1 - SP_B)$ $\gamma^B_{nbti} = (1 - SP_B)$
$F = \overline{AB + CD}$	$\gamma^C_{nbti} = 1 - SP_C$ $\gamma^D_{nbti} = 1 - SP_D$ $\gamma^B_{nbti} = (1 - SP_B)(1 - SP_C SP_D)$ $\gamma^A_{nbti} = (1 - SP_A)(1 - SP_C SP_D)$
$F = \overline{(A + C)(B + D)}$	$\gamma^C_{nbti} = 1 - SP_C$ $\gamma^D_{nbti} = 1 - SP_D$ $\gamma^A_{nbti} = (1 - SP_A)(1 - SP_C(SP_B + SP_D - SP_B SP_D))$ $\gamma^B_{nbti} = (1 - SP_B)(1 - SP_D(SP_A + SP_C - SP_A SP_C))$

We calculate γ_{nbti} at the inputs of all gates in a similar way and develop an NBTI aware timing model. The NBTI factors for few gate-topologies are shown in Table II. The inputs A, B, C, D are ordered (for PMOS transistors) in terms of vicinity to the output.

OBD Factor Characterization: The OBD factor for a NMOS corresponds to $[1, 0, 0]$ configuration for $[gate, drain, source]$. We characterize it for a gate based on SP. Let us illustrate this with the same example of AOI gate. We can see that this factor for A and C are simply the signal probability of logic '1' for the inputs A and C respectively, where as for the NMOS with input B to be at $[1, 0, 0]$, B should be at logic '1' and either of A or C needs to be at logic '1'. Therefore, the individual NMOS OBD factors (γ^i_{obd}) are given by the following equations.

$$\gamma^A_{obd} = SP_A \quad (11)$$

$$\gamma^C_{obd} = SP_C \quad (12)$$

$$\gamma^B_{obd} = SP_B(SP_A + SP_C - SP_A SP_C) \quad (13)$$

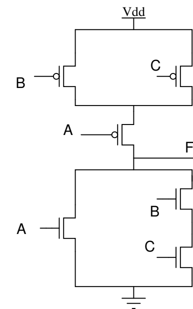

Fig. 5: AOI Gate

TABLE III: OBD factors for some gate topologies

Gate	OBD Factor
$F = \overline{AB}$	$\gamma_{obd}^{A} = SP_A SP_B$ $\gamma_{obd}^{B} = SP_B$
$F = \overline{A+B}$	$\gamma_{obd}^{A} = SP_A$ $\gamma_{obd}^{B} = SP_B$
$F = \overline{AC+BD}$	$\gamma_{obd}^{C} = SP_C$ $\gamma_{obd}^{D} = SP_D$ $\gamma_{obd}^{B} = SP_B(SP_D$ $+SP_A SP_C(1-SP_D))$ $\gamma_{obd}^{A} = SP_A(SP_C$ $+SP_B SP_D(1-SP_C))$
$F = \overline{(A+B)(C+D)}$	$\gamma_{obd}^{C} = SP_C$ $\gamma_{obd}^{D} = SP_D$ $\gamma_{obd}^{A} = SP_A(SP_C + SP_D(1-SP_C))$ $\gamma_{obd}^{B} = SP_B(SP_C + SP_D(1-SP_C))$

We follow the same technique to characterize OBD factors for all gates and use this in our gate-sizing approach. Table III shows the OBD factors for some gate topologies. The inputs A, B, C, D are ordered (for NMOS transistors) in terms of vicinity to the output.

V. Reliability Aware Gate Sizing

Algorithm 2 presents the steps of the reliability aware gate sizing. The estimation of signal probability (SP) is crucial in predicting aging effect (NBTI) and OBD. The SPs at different nodes of a circuit depend on circuit topology and application. However, we assume SP at primary inputs (PI) to be uniform (0.5) and propagate SP in accordance to the increasing level of logical depth of the gates [22] in "PropagateSP" (Line 1). Alternatively, logic simulation can be performed or methods such as cutting-algorithm [23] could be employed for more accurate SP estimation taking care of fanout re-convergence. Next, we pre-characterize γ_{obd}^{c} for each cell c in the design and γ_{nbti} for each timing arc of the cells from input to output as described earlier. γ_{nbti} is used in the piecewise-linear model developed to get the NBTI influenced rise delay/slew. γ_{obd}^{c} is used in line 4 to modify the metric to take account OBD. The area of the cell type (ct) chosen is represented by A_{ct}. The parameter w used in Algorithm 2 is the relative weight introduced in Eqn.(8) and it is a critical parameter to obtain trade-off between leakage power and circuit lifetime to be discussed in Section VI.

Importantly, the reliability characterization is flexible to be integrated into other discrete gate sizers, because (i) NBTI-induced delay/slew degradation can be readily introduced to any commercial STA tool, as we did in our developed STA engine and (ii) a weighted component of OBD can be incorporated to the metric of any cost-based or sensitivity-guided sizer. For instance, the power-delay sensitivity for greedy iterative approach such as TILOS [24] can be calculated with weighted OBD cost included in power and the impact of NBTI included in the delay.

Algorithm 2 Reliability Aware Gate Sizing (RAGS)

1: PropagateSP;
2: Precharacterize γ_{nbti} for each timing arc from input to output of a gate;
3: Precharacterize γ_{obd} for each gate;
4: Call SLR (Algorithm 1) with the modified metric. Use γ_{nbti} to estimate NBTI impact on rise-delay and rise-slew. $M_{modified} = fom(n) \times NS(n) + \delta \times [1 - NP(ct)] - \delta \times w \times [\gamma_{obd}^{c} A_{ct}]$;

VI. Experimental Results and Discussions

We have implemented our technique in C++ and executed on a linux machine with 4 GB memory and 2.9 GHz CPU. NBTI aware STA engine gives timing violations for the solution provided by the baseline sizer which is quite intuitive. So the sizer is run again with NBTI impact on it (sizer now takes care of NBTI while estimating rise delay and rise slew) and it generates solution with no violation ($WNS = 0$). In Table IV, we have compared F_{obd} for the solutions provided by baseline gate sizer (Column 2) and NBTI aware gate sizer (Column 3) meeting the same timing constraint. Then we calculate the lifetime degradation due to OBD for NBTI aware gate sizing using Eqn.(7) and assuming $\beta = 1$ for the lower technology node [12]. We can see that the lifetime degradation is more in time-constrained designs compared to that in the less stringent designs like 'leon3mp' or 'netcard'. To illustrate this, we pick one benchmark ('pci_bridge32') and run NBTI-aware gate-sizer by changing the clock period in steps of 10ps and observe that the circuit lifetime degrades more under stringent timing constraints (Fig. 6), which is due to the aggressive gate up-sizing to meet timing.

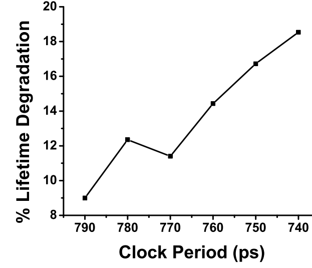

Fig. 6: NBTI aware sizing impact on circuit lifetime degradation due to OBD increase for faster circuits

Next, we run the reliability aware gate sizing algorithm (Algorithm 2) with $w = 0.1$. Table V presents the result for this with P_{leak}, F_{obd} and area for each benchmark, along with run time. WNS for each solution is 0. Area is represented w.r.t. the area of the min-sized inverter. The multi-columns tabulate the data for the two runs, the first being the NBTI aware sizing and the second is reliability aware gate sizing considering both NBTI and OBD. We also calculate the percentage lifetime improvement ($LT_{improve}$) in the 2nd run in comparison to the 1st run. Table V shows that we have achieved significant improvement in lifetime (an average of 24.9%) with leakage power overhead (an average of 11.7%) under same timing constraint. It is important to note that $LT_{improve}$ is more in time-constrained designs compared to less timing-stringent designs ('leon3mp' and 'netcard').

For most of the benchmarks, we get even lesser F_{obd} than nominal case as in Table IV, leading to the recovery of lifetime degradation due to OBD. This depends on the value of w in Line 4 (Algorithm 2). More is the weight w, lesser will be F_{obd}, but the leakage power overhead becomes more. To illustrate this, we run Algorithm 2 on a particular benchmark ('pci_bridge32') for $w = 0.04, 0.08, 0.12, 0.16$ and plotted the trade-off curve of % increase in P_{leak} vs. % $LT_{improve}$ as shown in Fig. 7. Intuitively, higher value of w assigns more weight on OBD (Eqn.(8)) and low V_t cells are used more extensively than up-sizing the gates to meet the performance constraint which leads to increase in P_{leak}. On the contrary, gate up-sizing is done more often rather than using low V_t

TABLE IV: Impact of NBTI aware sizing on circuit lifetime degradation due to OBD

Design	F_{obd} (Nom.)	F_{obd} (NBTI)	% increase	% Lifetime degradation
DMA	78676	88524	12.5	11.1
pci_bridge32	40936	49528	21.0	17.4
des_perf	304670	344171	13.0	11.5
vga_lcd	144366	214862	48.8	32.4
b19	423873	490153	15.6	13.5
leon3mp	631206	653511	3.5	3.4
netcard	752034	792549	5.4	5.1

TABLE V: Reliability aware gate sizing

Design	NBTI				NBTI + OBD				
	P_{leak} (W)	F_{obd}	Area	T_{run} (hr)	P_{leak} (W)	F_{obd}	Area	$LT_{improve}$	T_{run} (hr)
DMA	0.335	88524	107944	2.3	0.367 (+9.55%)	66403 (-24.87%)	87794 (-18.67%)	33.31%	2.5
pci_bridge32	0.245	49528	64380	2.8	0.252 (+2.85%)	36136 (-27.03%)	49510 (-23.09%)	37.06%	2.8
des_perf	1.180	344171	456150	3.6	1.450 (+22.88%)	291928 (-15.18%)	408418 (-10.46%)	17.90%	3.7
vga_lcd	0.608	214862	304677	4.7	0.685 (+12.66%)	148799 (-30.75%)	205912 (-32.42%)	44.40%	4.9
b19	1.516	490153	564932	5.8	1.757 (+15.89%)	389416 (-20.55%)	462901 (-18.06%)	25.87%	5.9
leon3mp	2.050	653511	700316	10.9	2.410 (+17.56%)	599194 (-8.31%)	708257 (+1.13%)	9.07%	11.0
netcard	1.808	792549	945945	14.6	1.819 (+0.60%)	743146 (-6.23%)	937048 (-0.94%)	6.65%	14.8

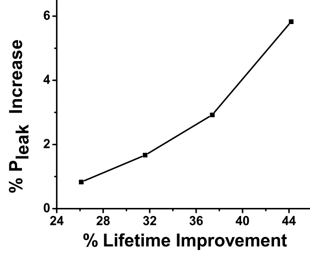

Fig. 7: % P_{leak} increase vs. % $LT_{improve}$ in pci_bridge32 under same timing constraints

cells for lower value of w and circuit lifetime is compromised. Also, there is no area overhead in the optimization process, except a slight increase in area for one benchmark ('leon3mp'). The area reduction (an average of 14.6%) in most of the benchmarks is anticipated as a weighted area component is subtracted in the modified metric for OBD.

Besides P_{leak} vs. $LT_{improve}$ trade-off for any design under same timing constraint, our holistic framework can achieve similar lifetime improvement as well without any significant leakage power overhead by a little sacrifice in timing. To elucidate this, we increase the clock-period in one design 'DMA' by 2% in the 2nd run (Table V) and adjusted w to 0.08 to obtain similar lifetime improvement (34%) with only 1% of leakage power overhead. Reducing the weight-parameter w further (setting to 0.05) even results decrease in leakage power (2%), changing lifetime improvement to 28%. So reliability, leakage power and timing can be traded off one against another through our unified gate-sizing approach by adjusting w and clock-period.

VII. Conclusion

NBTI and OBD are two key reliability concerns for nanometer circuits. In this paper, we propose a unified gate sizing algorithm to tackle NBTI-induced delay degradation while minimizing circuit lifetime degradation due to OBD. To our best knowledge, this is the first gate sizing work that considers both NBTI and OBD together. We develop a discrete gate sizer using ISPD'12 benchmarks/library with accurate timing analysis. We derive a circuit level OBD metric and use accurate models to guide holistic gate sizing optimization. Our experimental results show an average improvement of 24.9% in circuit lifetime with an average overhead of 11.7% in leakage power and a smooth trade-off between reliability and leakage power under same timing constraints. We also show how we can explore our holistic framework to achieve a trade-off in timing, reliability and leakage power, one against another. As technology moves down to 22nm and further, more and more reliability issues are becoming prominent, which need to be considered in classical physical synthesis.

Acknowledgment

This work is supported by the Semiconductor Research Corporation under Task 2419.001.

References

[1] F. Crupi *et al*. "A Comparative Study of the Oxide Breakdown in Short-channel nMOSFETs and pMOSFETs Stressed in Inversion and in Accumulation regimes", *IEEE Trans. Device and Mater.*, pp. 8–13, 2003.

[2] J. Fang and S. Sapatnekar, "Scalable Methods for the Analysis and Optimization of Gate Oxide Breakdown", *International Symposium on Quality of Electronic Design*, 2010.

[3] Y.H. Lee *et al*. "Prediction of Logic Product Failure due to Thin-gate Oxide Breakdown", *IRPS.*, pp. 18-28, 2006.

[4] K. Chopra *et al*. "A Statistical Approach for Full-chip Gate-oxide Reliability Analysis", *ICCAD*, pp. 698-705, 2008.

[5] S. V. Kumar *et al*. "NBTI aware Synthesis of Digital Circuits",*Design Automation Conference*, pp. 370–375, 2007.

[6] R Vattikonda *et al*. "Modeling and Minimization of PMOS NBTI Effect for Robust Nanometer Design", *Design Automation Conference*, pp. 1047–52, 2006.

[7] M. A. Alam and S. Mahapatra, "A Comprehensive Model of PMOS NBTI Degradation", *Microelectronics Reliability*, pp 71–81, 2005.

[8] S. V. Kumar *et al*. "An Analytical Model for Negative Bias Temperature Instability", *International Conference on Computer-Aided Design*, pp. 493–496, 2006.

[9] W. Wang *et al*., "The Impact of NBTI on the Performance of Combinational and Sequential Circuits", *Design Automation Conference*, pp. 364–369, 2007.

[10] M. M. Ozdal *et al*. "Gate Sizing and Device Technology Selection Algorithms for High-performance Industrial Designs",*International Conference on Computer-Aided Design*, pp. 724–731, 2011.

[11] X. Yang and K. Saluja, "Combating NBTI Degradation via Gate Sizing", *International Symposium on Quality Electronic Design*, pp. 47–52, 2007.

[12] E.Y. Wu *et al*., "CMOS Scaling beyond the 100-nm Node with Silicon-dioxide-based Gate Dielectrics", *IBM Journal of Research and Development*, 2002.

[13] Y. Wang *et al*., "On the Efficacy of Input Vector Control to Mitigate NBTI Effects and Leakage Power", *International Symposium on Quality of Electronic Design*, 2009.

[14] K. T. Lee *et al*. "PBTI-Associated High-Temperature Hot Carrier Degradation of nMOSFETs With Metal-Gate/High-k Dielectrics", *Electron Device Letters*, pp. 389–391, 2008.

[15] M. M. Ozdal *et al*., "The ISPD-2012 Discrete Cell Sizing Contest and Benchmark Suite" *Proc. ACM International Symposium on Physical Design*, pp. 161–164, 2012.

[16] Y. Liu and J. Hu, "A New Algorithm for Simultaneous Gate Sizing and Threshold Voltage Assignment", *IEEE Trans. on CAD*, pp–223-234, 2010.

[17] T. Luo *et al*. "Total Power Optimization Combining Placement, Sizing and Multi-Vt through Slack Distribution Management", *Asia Pacific Design Automation Conference*, 2008.

[18] W.N. Li, "Strongly NP-Hard Discrete Gate Sizing Problems", *International Conference on Computer Design*, 1993.

[19] P. Gupta *et al*. "Eyecharts: Constructive Benchmarking of Gate Sizing Heuristics", *Design Automation Conference*, 2010.

[20] H. Ren *et al*. "Sensitivity Guided Netweighting for Placement Driven Synthesis", *International Symposium on Physical Design*, 2004.

[21] A. Chakraborty and David Z. Pan, "Skew Management of NBTI Impacted Gated Clock Trees", *International Symposium on Physical Design*, 2010.

[22] F. Brglez *et al*. "Application of Testability Analysis: from ATPG to Critical Delay Path Tracing", *International Test Conference*, 1984.

[23] J. Savir *et al*. "Random Pattern Testability", *IEEE Transaction on Computer*, 1984.

[24] J. P. Fishburn and A. E. Dunlop, "TILOS: A Posynomial Programming Approach to Transistor Sizing", *International Conference on Computer Aided Design*, pp. 326–28, 1984.