OSFA: A New Paradigm of Aging Aware Gate-Sizing for Power/Performance Optimizations Under Multiple Operating Conditions

Subhendu Roy, *Member, IEEE*, Derong Liu, Jagmohan Singh, Junhyung Um, and David Z. Pan, *Fellow, IEEE*

***Abstract*—Modern systems-on-a-chip and microprocessors, e.g., those in smart phones and laptops, typically have multiple operating conditions, such as video streaming, Web browsing, standby, and so on. They will have different performance targets and run under different supply voltages. Gate sizing (with threshold voltage assignment) is a fundamental step for power/performance optimization. However, conventional gate sizing algorithms only consider one scenario, e.g., the performance-critical operating condition, which may be over-design for other operating conditions. In addition, reliability has become a prime concern in nanometer designs, and gate sizing has been employed to mitigate aging. However: 1) previous aging-affected delay models do not take into account more than one operating condition to estimate the aging impact and 2) earlier aging aware gate sizing algorithms only consider one operating condition at a time. In this paper, we present a new paradigm of aging aware gate sizing, one-size-fits-all (OSFA), which performs power/performance optimizations across multiple operating conditions. The existing delay model for negative bias temperature instability (NBTI) is extended to take into account multiple operating conditions, and incorporated into our OSFA framework. Based on OSFA, we also adjust the supply voltage targeting overall power optimization. A speed-up heuristic is proposed to scale our OSFA design space exploration methodology for higher number of operating conditions. Experimental results on industry-strength benchmarks demonstrate that: 1) compared with conventional approach OSFA could provide an average 6.1% reduction in power without performance loss; 2) NBTI-aware OSFA framework can provide significant improvement in comparison with guard-band based traditional NBTI-aware gate sizing approach; and 3) percentage savings compared to conventional methodology increases with the number of operating conditions.**

***Index Terms*—Gate sizing, Lagrangian relaxation (LR), negative bias temperature instability (NBTI), performance, power.**

I. Introduction

With growing design complexity of system-on-a-chip (SoC) and increasing number of cores in microprocessors, same design IP may run under different operating conditions or scenarios [1], [2]. For instance, video streaming and gaming in laptops or smart phones are high-speed applications, whereas the performance requirement for the applications such as Web-browsing or text messaging is not stringent. Consequently, supply voltage (V_{dd}) for the performance-relaxed scenarios are typically kept lower to save the dynamic and leakage power. Fig. 1 shows such an example for smart phone.

However, the physical gate sizes of the design need to be fixed and discrete across all operating conditions. A lot of work have been done in the past on simultaneous gate sizing and threshold voltage (V_{th}) assignment to perform power/performance optimization [3]–[9]. But the traditional gate-sizing algorithms consider only one scenario and then designers need to ensure that it meets the timing constraints in all scenarios.

This approach has several limitations. First, the timing models in modern cell-libraries are nonlinear, and look-up table-based [6], and in addition, the supply voltage induced delay scaling in the multithreshold cell-library depends on V_{th} as well [10]. For instance, the scaled delay at a particular lower V_{dd} would be higher for cells with higher V_{th} than the cells with lower V_{th} as CMOS gate delays depend on the over-drive voltage ($V_{dd} - V_{th}$). So it may be possible that the gate-sizes suitable for the constrained scenario do not meet the timing constraints for other scenarios under reduced voltage. As a result, designers either need to fix the timing violations incrementally for all scenarios which is tedious or boost up V_{dd} in the scenarios where timing is not met. Second, in addition to V_{dd}, the total power of the design depends on: 1) the fraction of time spent and 2) the switching activities of the nets in each scenario. Consequently, consideration of only one scenario during gate-sizing can be suboptimal in terms of power optimization. Finally, the conventional approach may need significant engineering effort,

Manuscript received August 5, 2015; revised November 14, 2015; accepted January 7, 2016. Date of publication January 29, 2016; date of current version September 7, 2016. This work was supported in part by the National Science Foundation, in part by the Semiconductor Research Corporation (SRC), and in part by the Samsung Semiconductor. This work was done when S. Roy and J. Singh were with the University of Texas at Austin, Austin, TX, USA. This paper was recommended by Associate Editor S.-C. Chang.

S. Roy and J. Singh are with Cadence Design Systems, San Jose, CA 95134 USA (e-mail: subhendu@utexas.edu; jsingh@utexas.edu).

D. Liu and D. Z. Pan are with the Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX 78712 USA (e-mail: derongliu@utexas.edu; dpan@ece.utexas.edu).

J. Um is with Samsung Semiconductor, Yongin City 17113, Korea (e-mail: junhyung.um@samsung.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2016.2523439

Fig. 1. Same IP: different applications.

thereby increasing the turn-around-time which is intrusive for today's strict time-to-market requirements.

Additionally, aging can considerably reduce the operational lifetime of an integrated circuit in the nanometer very large scale integration (VLSI) regime. It is also predicted that since supply voltage does not scale at the same pace with the device geometrics, device will degrade more in future technology nodes due to higher current density and temperature [11], [12]. To cope with the aging phenomena, such as bias temperature instability (BTI), hot carrier injection (HCI), etc., designers typically add pessimistic timing margins.

NBTI is one of the dominant reliability issues among them causing aging induced degradation in the circuits. NBTI is exhibited in pMOS devices, and it is manifested by the increase in threshold-voltage (V_{th}). It is a two-phase phenomenon, namely: 1) stress-phase when interface traps are generated under negative gate-to-source bias and 2) recovery phase under positive gate-to-source bias, annealing some of the interface traps. However, the interface traps are never annealed completely [13]. Several models [14]–[16] have been developed in the past to predict the shift in V_{th} due to NBTI considering this, causing as much as 20% degradation in circuit speed in ten years [11]. Apart from increasing the rise-delay, the increase in V_{th} can adversely affect the rise slew as well in the logic gates as shown in [17]. In [18], aging aware logic synthesis along with gate sizing is proposed tackling NBTI and HCI. Lin *et al.* [19] proposed a coordinated and scalable logic synthesis approach to combat NBTI, starting from subject graph to technology mapping and mapped netlist. Yang and Saluja [20] formulated an NBTI aware gate sizing problem where simulations are performed to compute the NBTI-induced delay degradation factors for the individual gates in the designs containing hundreds of logic gates. This approach may be accurate but computationally intensive, and so not feasible for large scale designs with millions of gates. In addition: 1) all these previous works do not consider multiple operating conditions during tackling NBTI in gate-sizing and 2) all NBTI affected delay models only consider single operating condition as well.

In this paper, we propose a new paradigm of gate-sizing one-size-fits-all (OSFA) which selects V_{th} and sizes of the logic gates in the design to optimize power meeting timing constraints across all scenarios considering NBTI. To solve this, we extend the Lagrangian relaxation (LR)-based formulation of one scenario to tackle multiple scenarios followed by sensitivity driven power recovery. NBTI-affected delay model is extended to consider multiple operating conditions. Multithreaded implementation is done to cope with the high computational need of our algorithm. A design-space exploration for power versus V_{dd} is performed to tune V_{dd} in the performance-relaxed scenarios. We also propose a speed-up technique in the design-space exploration for more than two operating conditions. The key contributions of our paper are summarized as follows.

1) To the best of our knowledge, this is the first NBTI-aware gate-sizing problem formulation considering multiple operating conditions to optimize the total power of any IP design. To tackle multiple operating conditions holistically, scenario aware LR (SALR) problem is formulated for OSFA.
2) A cross-layer methodology is developed where system and logic level specifications such as V_{dd}, scenario percentage and the switching activities in different scenarios are considered to select the gate-sizing options which can further guide design-space-exploration by providing feedback to the system level to optimize overall power consumption.
3) The existing model to characterize the rise delay and rise slew of the input-to-output arcs of the digital gates is enhanced to consider the multiple operating conditions.
4) A speed-up technique is proposed to scale this methodology for higher number of operating conditions, and the degradation in the solution quality for this speed-up is experimentally demonstrated to be small. It is also observed that the percentage savings in power compared to the conventional methodology increases with the number of operating conditions.

The rest of the paper is organized as follows. Section II motivates the problem of NBTI-aware gate-sizing under multiple operating conditions. Section III presents the problem formulation and OSFA algorithms to solve the scenario aware gate-sizing problem are described in Section IV. Section V presents the framework for NBTI-consideration in OSFA methodology. Section VI presents the experimental results for industry-strength large-scale benchmarks with the conclusion in Section VII.

II. Motivational Examples

In this section, the problem of scenario aware gate sizing has been motivated by examples from the perspectives of timing, power, and aging. In the example of timing perspective, we have shown that the sizing and threshold voltage assignments which meet the timing in one scenario may not meet the same for the other scenario, and vice versa. In the next example, we have illustrated that two sizing schemes may meet timing for both the scenarios, but the scheme considering both scenarios and the scenario percentages (or fraction) results in lesser power than the other scheme which considers only the constrained scenario. In the final example, we have described how NBTI-consideration across multiple scenarios can impact the selection of sizes and threshold levels for the gates.

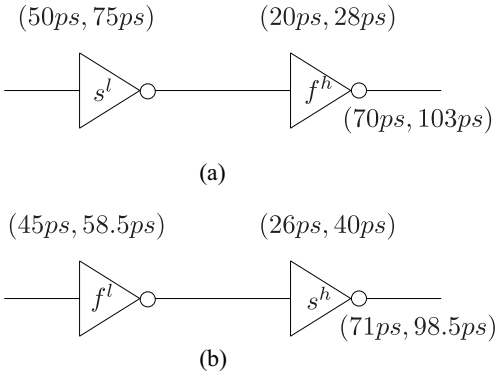

Fig. 2. Motivation for scenario aware gate sizing: timing perspective. (a) Meeting timing constraints of sc_1, but violating that of sc_2. (b) Meeting timing constraints of sc_2, but violating that of sc_1.

A. Timing Perspective

Consider a chain of two inverters with fast and slow scenarios, namely sc_1 and sc_2, respectively, with target delays of 70 and 100 ps. Since the gate-delay in the performance constrained scenario with higher V_{dd} is lesser in comparison to that in the other scenarios with lower V_{dd}, we define the delay scaling factor of a scenario with respect to the performance constrained scenario is the ratio of gate delay in that scenario to that in the performance constrained scenario. Note that this delay scaling factor also depends on V_{th} to be discussed in Section IV-A. Due to lower V_{dd} in sc_2, suppose the delay-scaling factors for sc_2 with respect to sc_1 are 1.5 and 1.3, respectively, for the slow (high-threshold) and fast (low-threshold) library cells. Fig. 2 illustrates the situation, where the symbols inside the inverters indicate the cell types. For instance, s^l represents a slow and lower size library cell and f^h represents a fast and higher size library cell. The numbers in the bracket indicate the delay of the inverters in sc_1 and sc_2, except for the output it represents the arrival times (ATs). It should be noted that the ratio of the delay values in case of second inverter is slightly higher than the scaling factors (1.5 or 1.3) in order to account for the impact of slew degradation at the output of the first inverter.

We can see that the sizing and V_{th} selection scheme shown in Fig. 2(a) can meet the target delay of sc_1, but violates the delay constraint for sc_2 and vice-versa for the scheme in Fig. 2(b). So by considering only one scenario for gate-sizing, it might not be possible to meet the timing constraints for all the scenarios.

B. Power Perspective

Consider two scenarios sc_1 and sc_2 and two sizing schemes scheme 1 and scheme 2 as shown in Fig. 3. Let the supply voltage, usage percentage, and clock frequency for sc_1 are, respectively, 1 V, 10%, and 1 GHz and those for the other scenario are, respectively, 0.8 V, 90%, and 0.7 GHz. Let sa_{ij} be the switching activity of the net n_i in scenario j and $sa_{11} = sa_{21} = sa_{31} = sa_{41} = 0.5$, $sa_{12} = sa_{22} = sa_{32} = sa_{42} = 0.32$. Suppose both the schemes meet timing in both scenarios. The leakage power (LP) and dynamic power (DP)

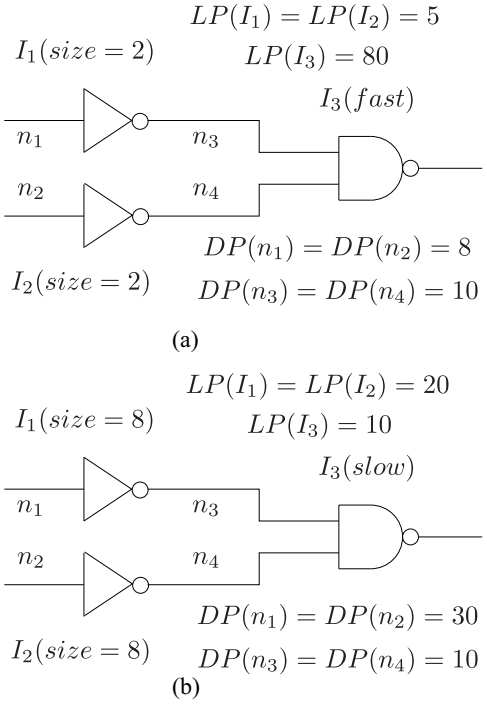

Fig. 3. Motivation for scenario aware gate sizing: power perspective. (a) Scheme 1. (b) Scheme 2.

of the cells and nets, mentioned in Fig. 3(a) and (b), are for the scenario sc_1.

If we add up the power numbers in sc_1 the total power in scheme 1 and scheme 2 are, respectively, 126 and 130. So scheme 1 is the better scheme considering only sc_1. But if we consider both the scenarios, then LP in scheme 1 is $0.1 \times (5+5+80) + 0.9 \times (5+5+80) \times (0.8/1) = 73.8$ and DP in scheme 1 is $0.1 \times (8+8+10+10) + 0.9 \times ((0.32/0.5) \times 8 + (0.32/0.5) \times 8 + (0.32/0.5) \times 10 + (0.32/0.5) \times 10) \times (0.8/1)^2 \times (0.7/1) = 12.9$ totaling 86.7. Note that the scaling factors for V_{dd}, frequency and switching activity of the nets n_i ($\forall i \in [1\ 4]$) in sc_2 with respect to sc_1 are, respectively, 0.8, 0.7, and (0.32/0.5). Similar calculation on scheme 2 gives the total power number as 69.6. So scheme 2 is actually the better option when considering both the scenarios.

C. Aging Perspective

Let us again consider two scenarios sc_1 and sc_2 for the circuit as shown in Fig. 4 and the usage percentage are, respectively, 0.2 and 0.8. Let the signal probabilities (SPs) for the nets n_1 and n_2 be, respectively, 0.7 and 0.2 in scenario sc_1, and the same in scenario sc_2 are, respectively, 0.1 and 0.8. By SP of a net we mean the probability that the signal of the net is logic "1." Suppose sc_1 is the performance-constrained scenario, and we take into account the NBTI-induced aging impact for sc_1. Then I_2 will have more degradation than I_1 as the ON-time for pMOS in I_2 $(1 - SP)$ is more. However, usage percentage of sc_2 is higher and in that scenario, pMOS for I_1 is ON more (duty cycle for pMOS of I_1 in sc_2 = $1-0.1 = 0.9$) than that for I_2. So a sizing scheme, considering aging impact for sc_1, can aggressively up-size I_2, which might

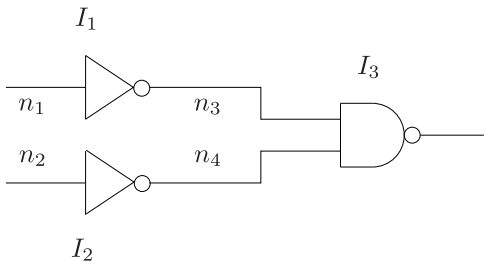

Fig. 4. Motivation for scenario aware gate sizing: aging perspective.

not be a good choice. Instead another sizing scheme, up-sizing I_1 more aggressively, may meet the timing. So qualitatively, aging impact should be determined by considering the SPs and the usage percentage in each scenario. In Section V, we present the quantitative aging estimation taking into account of these parameters.

III. OSFA Problem Formulation

Suppose there are n scenarios and each scenario $i \in [1, n]$ is characterized by the voltage level V_{dd}^i, the usage percentage and the switching activities (SA_i) of the nets in the design. The timing targets in each scenario are different and say it is T_i for the i^{th} scenario. In each scenario i, LP depends on V_{dd}^i, and the DP depends on V_{dd}^i (quadratically) and the switching activities. The formulation for our problem is as follows:

$$\begin{aligned} \text{minimize:} \quad & \sum a_i \big[\text{LP}\big(V_{\text{dd}}^i\big) + \text{DP}\big(V_{\text{dd}}^i, \text{SA}_i\big)\big] \\ \text{subject to:} \quad & \forall i \in [1, n] \quad T_{\text{delay}}^{\text{nbti}}\big(V_{\text{dd}}^i\big) \le T_i \end{aligned} \tag{1}$$

where, $T_{\text{delay}}^{\text{nbti}}(V_{\text{dd}}^i)$ is the maximum combinational delay between timing start-point to timing end-point in i^{th} scenario considering NBTI-induced delay degradation and a_i is the fractional percentage for the scenario i so that $\sum a_i = 1$.

IV. OSFA Algorithms

In this section, we present the OSFA algorithms (a two-step approach) to solve the gate sizing problem under multiple operating conditions. In the first step, the LR-based formulation in [6] is enhanced to consider more than one scenario. This step gives a solution which meets timing in all scenarios. But since discrete gate sizing problem is NP-hard [21], LR-based solution cannot be optimal. So a scenario aware sensitivity driven power recovery technique is then applied to further optimize power. Before going into the details of these steps, we first describe the models used for delay and power to consider multiple scenarios.

A. Delay and Power Models

Modern industrial cell libraries have look-up-table based delay models for various scenarios. In our case, we have taken the industrial benchmarks and cell-library from the recent ISPD'12 contest [22]. However, it contains the delay and LP information for single V_{dd}. In Section VI-A, we have described in detail how we have generated the scenarios with different voltages. To calculate DP and LP across various scenarios, scaling factors have been introduced. Let $V_{\text{dd}}^{\text{nom}}$ and V_{dd}^i be the supply voltages at the nominal scenario and the i^{th} scenario. Assuming a first-order delay model for CMOS gate delay [10] and velocity saturation constant $\alpha \simeq 1$ the ratio of delay of the i^{th} scenario to that of the nominal scenario is given by

$$\frac{t_{\text{delay}}\big(V_{\text{dd}}^i\big)}{t_{\text{delay}}\big(V_{\text{dd}}^{\text{nom}}\big)} = \frac{1 - \frac{V_{\text{th}}}{V_{\text{dd}}^{\text{nom}}}}{1 - \frac{V_{\text{th}}}{V_{\text{dd}}^i}}. \tag{2}$$

Although V_{dd} has a second-order effect on leakage current [23], we have assumed leakage current to be independent of V_{dd} for the sake of simplicity and thus the corresponding ratio for LP is given by

$$\frac{\text{LP}\big(V_{\text{dd}}^i\big)}{\text{LP}\big(V_{\text{dd}}^{\text{nom}}\big)} = \frac{V_{\text{dd}}^i}{V_{\text{dd}}^{\text{nom}}} \tag{3}$$

where V_{dd}^i is the total LP of the design in i^{th} scenario which is computed as the sum of the LP of the individual cells of the design in that scenario. The LP of the design with n scenarios is calculated as $\sum_i^n (a_i \times \text{LP}(V_{\text{dd}}^i))$. Note that the LP numbers in the industrial library can be stateless leakage or state-dependent leakage. Stateless leakage is typically a single number for a cell, where as state-dependent leakage depends on the input state probabilities. In our case, stateless leakage has been used.

DP or specifically switching power for a net with switching activity sa is computed as $\text{DP} = \text{sa} \times f_{\text{clk}} \times C_L \times {V_{\text{dd}}}^2$, where f_{clk} is the clock frequency and C_L is the total capacitance of the net. This computation is done for each scenario with the corresponding supply voltage, switching activity etc. in that scenario. Then, it is summed over all nets to compute the total switching power in that scenario, followed by taking weighted average by the usage percentage over all scenarios (similar to that of LP) to calculate the switching power of the design. Since the internal (short-circuit) power of the library cells are not provided in the library, we have not considered it, but it can be easily added into the power component if available.

B. Scenario Aware Lagrangian Relaxation

In [24], simultaneous gate sizing and wire sizing problems are solved by LR under Elmore delay model, where the constraints of Lagrangian primal problem are specified by AT constraints. Similar formulation is adopted in [6, eq. (2)] to solve the discrete gate sizing problem, which is then translated to an unconstrained objective function, and later the ATs at the output of the intermediate gates (between the timing start points and timing end points) are omitted [6, eq. (5)] using Karush–Kuhn–Tucker (KKT) conditions. So from [6, eq. (5)], the LR-based formulation for single scenario discrete gate sizing problem can be written in the following functional form:

$$\alpha \cdot \text{power} + \sum_{u \to v} \mu_{u \to v} d_{u \to v} + \sum_{\text{po}} \mu_{\text{po}}(-r_{\text{po}}) + \sum_{\text{pi}} \mu_{\text{pi}}(a_{\text{pi}}) \tag{4}$$

where, $\mu_{u \to v}$, μ_{po}, and μ_{pi} are the Lagrange multipliers (LMs) for the timing arc $u \to v$, primary output po, and primary

Algorithm 1 SALR Optimization

```
1: Procedure SALROpt(design, library)
2: Initialize Lagrange multipliers for all scenarios;
3: while Leakage power improvement is more than a threshold do
4:    for all i ∈ Scenarios do
5:       slackFactor(i) ← T_clk,i / (T_clk,i − worstSlack(i));
6:    end for
7:    sc ← scenario with the minimum slackFactor;
8:    if worstSlack(sc) > slackThreshold(sc) then
9:       α_global ← α_global × (slackFactor(sc))^2;
10:   else
11:      α_global ← α_global × (slackFactor(sc));
12:   end if
13:   LRSOpt(design, library, α_global);
14:   for all i ∈ Scenarios do
15:      runSTA(design, library, i);
16:      updateLagrangeMultipliers(design, i);
17:   end for
18: end while
19: end Procedure
```

input pi, respectively, and $d_{u\to v}$ is the delay of the arc $u \to v$. r_{po} denotes the required time of arrival at po, a_{pi} denotes the AT at pi, and α is the tradeoff parameter between power and timing slacks.

To tackle multiple scenarios at a time, we modify eq. (4) as follows:

$$\sum_{i=1}^{n}\left(\alpha_i a_i\left[\text{LP}\left(V_{\text{dd}}^i\right)+\text{DP}\left(V_{\text{dd}}^i, SA_i\right)\right]+\sum_{u\to v}\mu_{u\to v,i}d_{u\to v,i}+\sum_{\text{po}}\mu_{\text{po},i}(-r_{\text{po},i})+\sum_{\text{pi}}\mu_{\text{pi},i}(a_{\text{pi},i})\right) \quad (5)$$

where subscript i has been added in the terms to signify the corresponding terms for i^{th} scenario. It should be stressed that the power components are weighted by the respective a_is, but no such weighing factor is added for the timing terms as the timing needs to be met for all scenarios. Note that a_i is the scenario percentage as defined in eq. (1).

Algorithm 1 presents the key steps of the SALR optimization. At first the maximum load violations are fixed by traversing the cells in reverse topological order and choosing best possible legal cell-types [5]. No max-load violation is introduced throughout the optimization procedure by checking the legality before any cell-type substitution. This is not shown in Algorithm 1. Then the LMs are initialized for all the scenarios (line 2). This is done by setting the LMs at the timing end-points (primary output/flop input) and then traversing in reverse topological order to assign the multipliers at other pins satisfying the KKT conditions [24]. Then slackFactor for all scenarios are calculated (line 5) representing the global timing picture of the design across the scenarios. Since lesser the slackFactor, more timing constrained the scenario is, the scenario sc with minimum slackFactor is selected to scale α_{global} (lines 8–12). If the worstSlack(sc) is less than 0, then α_{global} is down-scaled to impose more importance on the timing and vice-versa. If the worstSlack(sc) is greater than slackThreshold(sc), then α_{global} is up-scaled aggressively to impose more weight on LP reduction. In our algorithm, slackThreshold(i) is set to be equal to ($T_{\text{clk},i}/50$) empirically. We ran the designs with different values of slackThreshold(i), such as ($T_{\text{clk},i}/10$), ($T_{\text{clk},i}/20$), ($T_{\text{clk},i}/50$), and ($T_{\text{clk},i}/100$), and found slackThreshold(i) = ($T_{\text{clk},i}/50$) provided the best results on average among them.

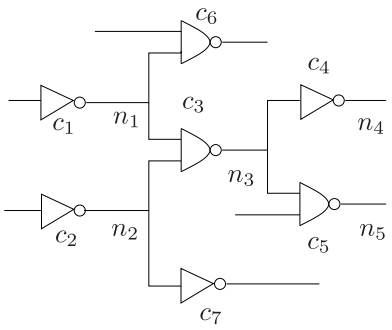

Fig. 5. Cost calculation for a cell.

At the next step, the Lagrangian subproblems are solved for individual cells in topological sorted order. For each of the cell, α_i in eq. (5) is calculated by scaling α_{global} for individual cell based on the slack of that cell in the i^{th} scenario. The cost for each cell-type (ct) from the cell library is calculated according to eq. (6), and ct which minimizes the cost for that cell is selected

$$\text{cost}_c(\text{ct})=\sum_{i=1}^{n}\left(\alpha_i a_i\left[\text{LP}_{\text{ct}}(V_{\text{dd}}^i)+\sum_{net\in N}\left(\text{DP}\left(V_{\text{dd}}^i, \text{SA}_i\right)\right)\right]+\sum_{u\to v}\mu_{u\to v,i}^r d_{u\to v,i}^r+\mu_{u\to v,i}^f d_{u\to v,i}^f\right). \quad (6)$$

To illustrate this consider Fig. 5 for cost calculation of the cell c_3. For the timing part of eq. (6), the rising (r) and falling (f) timing arcs ($u \to v$) for the cells, which are immediate fan-ins (c_1, c_2), siblings (c_6, c_7), and fan-outs (c_4, c_5), are taken into account. From the power perspective, LP of the cell c_3 with type ct (LP_{ct}) and the DP of the fan-in nets (n_1 and n_2) are considered in the cost computation.

Finally, the static timing analysis (STA) engine is run and LMs are updated at the end of the iteration (lines 14–17) for all the scenarios. The update is done by first scaling the LM of individual timing arc/primary output according to the available slack. For instance, the Lagrange multiplier (for rise delay) at primary output (po) is updated as $\mu_{\text{po},i}^r = \mu_{\text{po},i}^r \times (a_{\text{po},i}^r/T_{\text{clk},i})$, where $a_{\text{po},i}^r$ represents the rise AT at po in the i^{th} scenario. Then the multipliers are updated to match the KKT conditions.

Algorithm 1 is computationally intensive as it needs to run the STA engine and calculate costs across the scenarios. So we have implemented STA (line 15), update of LMs (line 16), and cost estimation in "LRSOpt" using Intel threading building blocks [25]. Typical STA implementation involves the calculation of ATs in topological order and required time of

Algorithm 2 SDPR Optimization

```
1: Procedure SDPROpt(design, library)
2: while Leakage power improvement is more than a threshold do
3:    sc ← scenario with the minimum slackFactor;
4:    Sort cells in accordance to maximum LM in sc;
5:    Set all cell status to true;
6:    for all cell ∈ sortedCellList in increasing order do
7:       if status(cell) = false then
8:          Continue;
9:       end if
10:      Select a celltype maximizing S_factor ← ΔP / Δslack_loss;
11:      Run BFS in the fan-in/fan-out cone of cell;
12:      Set flag to false for all discovered cells;
13:   end for
14:   for all i ∈ Scenarios do
15:      runSTA(design, library, i);
16:      updateLagrangeMultiplier(design, i);
17:   end for
18: end while
19: end Procedure
```

arrival (RTA) in reverse topological order. The cells in the design are divided in accordance to the topological levels, and the computation of AT/RTA, LM update, and cost estimation in a certain topological level are done in parallel.

C. Sensitivity Driven Power Recovery

Since the problem is nonconvex, the optimal solution cannot be achieved by only solving the Lagrangian subproblems. Instead the solution obtained in the first phase is considered as a seed solution on which a sensitivity-based power recovery technique is applied to lead toward optimality by recovering more power at the non-critical paths.

In this phase, again another "while" loop is executed. Algorithm 2 shows the steps of this phase. Like Algorithm 1, the constrained scenario (sc) is determined by choosing the scenario with minimum slackFactor. Then the cells are sorted according to its criticality (line 4), determined by the maximum among the LMs (in sc) of its input pins and then these sorted cells are processed in order, i.e., the cells, which are less timing critical, are processed first (line 6). For each cell, we calculate a sensitivity factor for each of the available cell-type. The sensitivity factor is the ratio of the power gain to the loss in timing slack by substituting the cell. The cell-type which gives the maximum sensitivity factor is selected.

Consider the cell c_3 as shown in Fig. 5. Let its original cell-type be ct_1 and we want to calculate the sensitivity factor for changing its cell-type to ct_2. By changing the cell-type, the input capacitances of c_3 is modified leading to change in input-to-output delays across c_1 and c_2. For each scenario i, the AT/slew at n_3 is calculated considering this. Then we calculate the loss in timing slack $\Delta\mathrm{slack}_{\mathrm{loss},i}$ as the difference of the updated AT and the actual AT at n_3. We also consider the impact of change in slew at n_3 by taking the maximum increase in the arrival at the output nets of its fanout cells, i.e., c_4 and c_5, and add that to $\Delta\mathrm{slack}_{\mathrm{loss},i}$. If this slack loss is greater than the available slack at n_3 for any scenario, then we skip that cell-type. Otherwise, to consider various scenarios and the rise/fall slack loss, we take the worst case slack loss of the two across all scenarios in sensitivity calculation. Suppose the gain or decrease in power be $\Delta P = \sum_{i=1}^{n} a_i(P_{\mathrm{ct}_1,i} - P_{\mathrm{ct}_2,i})$ and so we calculate the sensitivity factor ($S_{\mathrm{factor}} = (\Delta P/\Delta\mathrm{slack}_{\mathrm{loss}})$) for each of the cell-types available in the library and select the cell-type with maximum S_{factor}.

Once we change the cell-type, we set a flag false corresponding to all the cells which are in fan-in and fan-out cone of c_3 and we do not try to modify the cell types of those cells in that iteration. This process is repeated by going over all cells (note the cells for which the flag becomes false are not processed in that iteration). It might be possible that the timing slack becomes negative for the design in one scenario. This is possible because when we change the cell type we do not propagate the slew impact throughout the design. In such case, we swap the cells, where we find negative timing slack, back to its earlier cell-type (not shown in Algorithm 2). The iterations are continued until we do not get any improvement in power.

V. NBTI Consideration for OSFA

The analytical model for NBTI in [16] is derived using reaction–diffusion model, and the observed phenomenon of frequency independence of NBTI has been mathematically proved. It has been shown that the impact of NBTI on a particular pMOS device depends on the duty cycle of stress, i.e., the ratio between the time of the device under stress (T_{on}) and the total aging time (T_{aging}). Based on this, the s-factor equations are developed to predict the NBTI-effect on increasing V_{th}. In [17], this duty cycle is termed as NBTI-factor, and NBTI-factors for different logic gates are computed which again vary with different inputs due to the stacking effect. For instance, consider the 2-input NOR gate as shown in Fig. 6. If SP_A and SP_B, respectively, denote the SP at the input A and B, the NBTI-factors at the inputs are given by

$$\gamma_{\mathrm{nbti},B} = (1 - \mathrm{SP}_B)$$
$$\gamma_{\mathrm{nbti},A} = (1 - \mathrm{SP}_A)(1 - \mathrm{SP}_B). \tag{7}$$

This is because for the pMOS with input A to be under stress, both pMOS should be ON [13].

In this paper, we have extended the notion of NBTI-factors for multiple operating conditions. Since the SPs for different scenarios are different, the NBTI-factors for a particular input–output timing arc are different across multiple scenarios. But we can compute the duty cycle of stress across multiple scenarios or the effective NBTI-factor as follows. Let $\gamma^{i}_{\mathrm{nbti},x}$ be the NBTI-factor in scenario i for the arc with input x. Let T_{aging} be the total aging time. Since a_i is the fractional percentage for the scenario i, time spent in scenario i is $a_i \times T_{\mathrm{aging}}$, and the stress time for scenario i for the arc with input x is $a_i \times \gamma^{i}_{\mathrm{nbti},x} \times T_{\mathrm{aging}}$. Therefore, total stress time over n operating

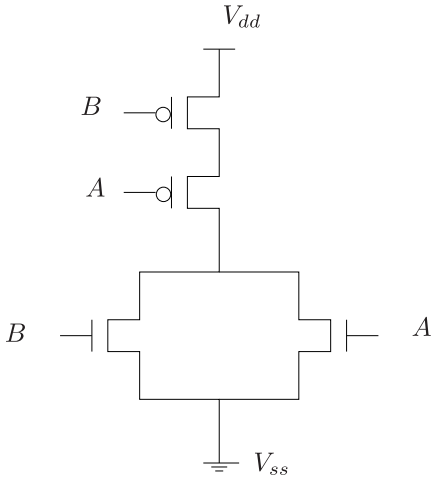

Fig. 6. Stacking effect in NOR gate.

conditions for the timing arc with input x is given by

$$T_{\text{on}} = \sum_{i=1}^{n} \left(a_i \times \gamma_{\text{nbti},x}^{i} \times T_{\text{aging}}\right). \tag{8}$$

So the effective NBTI-factor is given by

$$\gamma_{\text{nbti},x}^{\text{eff}} = \frac{T_{\text{on}}}{T_{\text{aging}}} = \sum_{i=1}^{n} \left(a_i \times \gamma_{\text{nbti},x}^{i}\right). \tag{9}$$

It should be stressed that the individual NBTI-factors for any particular scenario cannot be considered for the timing analysis for that scenario, and the effective NBTI-factors need to be used for NBTI-induced rise-delay/slew degradation for timing analysis in all scenarios. This is because the effect of NBTI is cumulative for all the scenarios, and cannot be separated out for individual scenarios.

Using s-factor equations [16], the degradations in V_{th} are computed for different NBTI-factors, and with those V_{th}s, HSpice simulations are performed to characterize the rise delay and rise slew with NBTI-factors in the logic gates. Then a piecewise-linear model for rise delay/rise slew with NBTI-factors is developed similar to [17], and then using this model, the effective NBTI-factors are finally used to compute the rise delay/slew in presence of NBTI.

VI. Experimental Results

We have implemented the algorithms presented in this paper in C++ and run it on a Linux machine with 8-Core 2.90 GHz CPU and 72 GB RAM. In this section, first the test case generation method is described. Then we will experimentally validate that gate sizing considering one scenario may not meet the timing constraints in another scenario and vice versa. Next, power savings in our algorithm are demonstrated by performing design space exploration in the gate-sizing step by tuning the system level parameter V_{dd}. Then we propose a speed-up technique in the OSFA design space exploration. Finally, we experimentally demonstrate: 1) the impact of scenario percentage on OSFA methodology and 2) the effectiveness of OSFA considering NBTI.

A. Test Case Generation

The designs and cell-library for the experimental demonstration are taken from the recent ISPD'12 benchmark suite [22] (fast version). These benchmarks are industry-strength benchmarks and the delay model of the cell-library is nonlinear, look-up-table-based and very realistic. However, the cell-library contains the delay values under one operating condition (one V_{dd}). We have considered V_{dd} for this scenario to be 1.2 V and created another slow scenario with the timing target equal to 1.5 times that of the nominal scenario. For instance, the timing target for the benchmark "pci_bridge32_fast" is 660 ps, and so the target delay for the slow scenario is 990 ps. Equation (2) is used to compute the delay values in the slow scenario, and we need the V_{th} values of the cells for this. ISPD'12 cell-library consists of cells with three V_{th}, but the V_{th} values are not mentioned. Taking a reference from the nominal V_{th} to be around 0.46 V in [26], we assume the nominal V_{th} to be 0.45 V, and low and high V_{th}s symmetric around the nominal voltage which are 0.4 and 0.5 V, respectively. The supply voltage of the slow scenario is varied to search for power optimal solutions across the scenarios. Since the design IPs in laptops or smart phones typically run most of the times under slow operating conditions, we choose the scenario percentage for the fast and slow scenario to be 0.2 and 0.8, respectively.

For industrial designs, the switching activities are captured by Value Change Dump (VCD)/ Switching Activity Interchange Format (SAIF) files which we do not have. So we assume the SPs at the primary input (pi) to be 0.5 for the fast scenario and generate input vectors for 500 simulations. The generation of these input vectors is done by using rand() function. We generate a random number between 0 and 1, and if it is greater than 0.5, then we assign logic 1 to that pi or assign logic "0" otherwise. This is repeated for 500 times to assign logic 1 or logic 0 to pi for each of the 500 simulations. Then we run 500 Modelsim simulations [27] to compute the signal values at the internal nets and take the average over 500 simulations to obtain the SPs. Then switching activity (SA) is computed as $\text{SA} = 2 \times \text{SP} \times (1 - \text{SP})$ [28]. For the slow scenario, we assume random SP at the primary inputs and then repeat the same process (by again performing 500 Modelsim simulations) to obtain the SA of the nets. However, we believe that our algorithm will work with same efficiency in case of given switching activities, and if not good particularly for dissimilar switching activities across different operating conditions due to the switching activity driven objective function.

B. OSFA Versus Conventional

The contest held by ISPD'12 [22] has focused only on LP optimization instead of considering both LP and DP. There are several recent state-of-the-art gate sizing algorithms which used the ISPD'12 benchmarks released during this contest, and so they reported the LP numbers. In our case, it is multiscenario gate sizing framework for optimizing total power (leakage + switching). Therefore, to compare to the state-of-the-art gate sizing algorithms, we run our algorithm for single scenario. In Table I, we have presented the LP

TABLE I
COMPARISON WITH STATE-OF-THE-ART GATE SIZERS

Design	Cells	Our Approach			Contest Winner		ICCAD'12 [8]		TOADES'14 [9]	
		P_{leak} before SDPR (W)	P_{leak} (W)	T_{run} (hr)	P_{leak} (W)	T_{run} (hr)	P_{leak} (W)	T_{run} (hr)	P_{leak} (W)	T_{run} (hr)
DMA_slow	25,301	0.191	0.152	0.01	0.205	2.16	0.153	0.01	0.132	0.03
DMA_fast	25,301	0.438	0.326	0.02	0.511	1.72	0.281	0.01	0.241	0.04
pci_bridge32_slow	33,203	0.146	0.126	0.01	0.203	2.26	0.111	0.02	0.096	0.03
pci_bridge32_fast	33,203	0.217	0.180	0.01	0.512	1.79	0.167	0.02	0.139	0.03
des_perf_slow	111,229	0.734	0.625	0.05	0.674	7.0	0.671	0.10	0.586	0.25
des_perf_fast	111,229	2.229	1.810	0.08	2.390	7.0	1.930	0.11	1.429	0.34
vga_lcd_slow	164,891	0.403	0.348	0.08	0.415	9.0	0.375	0.13	0.328	0.20
vga_lcd_fast	164,891	0.604	0.459	0.08	0.758	9.0	0.460	0.17	0.419	0.27
b19_slow	212,674	0.630	0.594	0.20	0.627	11.0	0.583	0.17	0.565	0.42
b19_fast	212,674	1.046	0.904	0.28	2.710	11.0	0.771	0.20	0.724	0.46
leon3mp_slow	649,191	1.393	1.349	0.60	1.420	20.0	1.400	0.73	1.329	1.62
leon3mp_fast	649,191	1.776	1.553	1.05	-	-	1.640	0.91	1.429	1.80
netcard_slow	958,780	1.768	1.762	0.38	1.770	29.0	1.780	0.80	1.762	1.42
netcard_fast	958,780	2.969	1.881	0.91	2.010	29.0	2.180	1.48	1.840	2.14

component (from the main problem objective of total power) and run-times for all ISPD'12 contest benchmarks for our algorithm, along with those for the contest winner, the fastest gate sizer [8], and one of the best sizers in terms of LP number [9]. We have also mentioned the power numbers before sensitivity driven power recovery (SDPR) and the final power numbers. The difference in them represent the contribution of SDPR. SDPR takes around 50% of the total run-time. Note total computations or CPU time in SDPR is much lower than the first stage, but SDPR algorithm is not parallel unlike the SALR phase, thus consuming approximately half of the total run time.

On comparison, we achieve 25.8% better LP than the contest winner, 1.2% worse power than [8], and 14.3% worse power than [9] on average. For the larger benchmarks, such as "leon3mp" and "netcard," our algorithm provides better LP than the contest winner and [8], and competitive solutions with [9]. In terms of run-time, our algorithm is the fastest among all, taking 3.8 h to complete all 14 benchmarks, whereas the same for the contest winner, [8] and [9] are, respectively, 139.9 h, 4.9 h, and 9.1 h. Note that: 1) parameter tuning in our algorithm might improve the LP numbers to some extent as those have been tuned for total power optimization and 2) most of the state-of-the-art gate sizers, including [9], are LR-based. So the notion of our multiscenario LR formulation can be easily integrated into those sizers. It should be also stressed that our main novelty is not in the single scenario sizer implementation, but with the help of this, the notion of gate sizing under multiple operating conditions is demonstrated.

To demonstrate the effectiveness of OSFA, we take the benchmark "pci_bridge32" and run our sizer considering only the fast scenario. Then STA is run for the slow scenario with $V_{dd} = 0.85$ V and we get 1818 timing violations (at the timing-end points, such as primary output or "D" pin of flip-flops). Then we do the opposite, i.e., size the gates considering the slow scenario and STA is run for the fast scenario. In this case, we get 11 timing violations. So if we just consider single scenario for gate-sizing like the conventional approach, it might not be possible to meet the timing constraints across all the

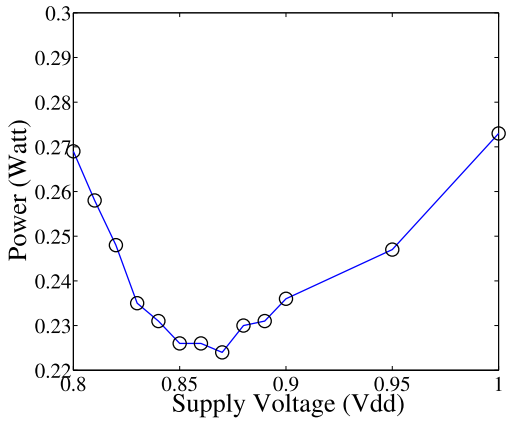

Fig. 7. OSFA design-space exploration by tuning V_{dd} of second scenario for pci_bridge32.

scenarios. This is due to the nonlinearity in delay model and nonuniform delay scaling across different V_{th} for a particular V_{dd}, as explained in Section I. The violations can be fixed by increasing V_{dd}. For instance, if we size considering only fast scenario and then raise V_{dd} of slow scenario to 0.90 V, then it meets the target delays in both scenarios. But this increases the power consumption of the design.

However, by running the OSFA considering both fast and slow scenarios, we can meet the timing constraints in both the scenarios as OSFA has the intelligence to identify which cells are critical in all scenarios and assigns sizes accordingly. This gives us the flexibility in performing design-space exploration. For instance, we set $V_{dd} = 1.2$ V for the fast scenario and then vary V_{dd} of the slow scenario from 0.8 to 1.0 V, and run OSFA. Fig. 7 shows the curve for total power of the design versus V_{dd} of the slow scenario. We can see as V_{dd} increases from 0.8 V, the power consumption initially decreases till $V_{dd} = 0.87$ V and then increases with V_{dd}.

The explanation for this behavior is as follows. When V_{dd} is increased, it has two conflicting effects: 1) increase in LP/DP due to its direct V_{dd} dependence and 2) decrease in delay of the logic gates facilitating down-sizing or high V_{th} selection resulting in lower LP/DP. If V_{dd} for the slow scenario is too low (such as 0.8 V), slow scenario becomes the

TABLE II
COMPARISON WITH CONVENTIONAL METHOD

Design	Cells	Power(W) in Conventional Methodology	Power(W) in OSFA Methodology	% Imprv.
DMA	25,301	0.399	0.375	5.9
pci_bridge32	33,203	0.242	0.226	6.4
des_perf	111,229	2.448	2.381	2.7
vga_lcd	164,891	0.651	0.633	2.7
b19	212,674	0.865	0.755	12.7
leon3mp	649,191	1.528	1.413	7.5
netcard	958,780	2.043	1.951	4.5
Average				6.1

constrained scenario and logic gates require up-sizing or low V_{th} selection, and power consumption is high. When it initially increases beyond 0.8 V, slow scenario starts to be less timing-constrained making second effect the prominent one, and decreasing power consumption. But after certain point (here $V_{dd} = 0.87$ V), the fast scenario starts to become the constrained scenario. Consequently, second effect becomes submissive because we cannot down-size the gates further or select higher V_{th} as the timing constraint of the fast scenario needs to be still met. So beyond this point, first effect plays a dominant role in increasing the power consumption.

Next, we repeat this experiment for all the benchmarks with fixed $V_{dd} = 1.2$ V for the fast scenario and varying V_{dd} of the slow scenario from 0.8 to 1.0 V in steps of 0.05 V and select the best among all solutions. To compare with the conventional methodology, the sizer is run by considering only fast scenario, and V_{dd} of the slow scenario is bumped up also in steps of 0.05 V until the timing constraint for the slow scenario is met. Then we compare the obtained power numbers with that achieved by the design-space exploration. Note that all these intermediate supply voltages may not be feasible in real industrial settings due to architectural/circuit constraints. Instead of that, there can be a few possible discrete choices for supply voltages. In that case, our OSFA methodology can be employed to select between those supply voltages.

Table II presents the comparison for all the benchmarks in terms of power. Column 2 shows the number of cells in the design. Columns 3 and 4 present the total power in the conventional and OSFA methodology, respectively. The percentage improvement in power varies with benchmarks, varying from 2.7% to 12.7% with most designs around 5%–7%. On average, OSFA methodology achieves 6.1% reduction in total power compared to conventional methodology. The run-time in OSFA is about twice that with one scenario for all the designs, and this is intuitive as OSFA needs to compute costs, run STA for 2 scenarios. The run-time for the biggest design ("netcard") of OSFA is around 1.8 h. This is comparable to the run-times in the state-of-the-art gate sizers [5], [8] even with one scenario considering only LP. More importantly, the runtime in OSFA can be further improved by running on machines with more cores. It should be noted that the power reduction in benchmarks such as "b19" or "leon3mp" are higher. This is intuitively due to the higher number of topological levels (90 for b19 and 59 for leon3mp) compared to the other benchmarks as more is the number of topological levels, more will be the effect of nonlinear delay scaling in designs. With technology scaling and more pipeline stages, the number of topological levels or depth of the design is decreasing, which may not be suitable for OSFA methodology. However, at the same time, the nonlinearity across different operating conditions is also growing at a fast rate which would make the methodology more relevant in modern designs.

C. Design Space Exploration for More Than Two Operating Conditions

When the number of operating conditions (n) is more than 2, an exhaustive way to perform the design-space exploration is to fix the voltage of one scenario and assign m voltage steps for the rest $n-1$ scenarios, and run OSFA for each case. The complexity of that approach would be $O(m^{n-1})$.

To tackle this high computational cost, we propose an alternate way of progressively selecting V_{dd} for each scenario. At the first stage, we consider two scenarios, fix V_{dd} in scenario 1 and run OSFA for m voltage steps in second scenario. V_{dd} for the second scenario is selected by taking the minimum power point in the design-space exploration curve. Next, we fix the V_{dd} of first and second scenario and run OSFA for m voltage steps in third scenario and so on. To generalize, at the i^{th} stage, we fix V_{dd} of i scenarios, and run OSFA considering $i+1$ scenarios by varying V_{dd} of $(i+1)^{th}$ scenario followed by selecting V_{dd} for the $(i+1)^{th}$ scenario. The overall complexity of this alternative method would be at most $O(m \times (n-1))$. It should be noted that for the exhaustive design space exploration, each OSFA run considers n operating conditions, where as the proposed speed-up technique considers on average $(n/2)$ operating conditions $((\sum_{i=2}^{n}(i)/(n-1)) \simeq (n/2))$. So on average, each OSFA run in the proposed technique will have half run-time in comparison to that in the exhaustive design space exploration as the run-time is proportional to the number of operating conditions. However, this approach might compromise in solution quality to some extent.

To validate this, we take the design "pci_bridge32" and create two more operating conditions with clock period twice and 2.5 times that of the fast scenario (with random switching activities). With exhaustive design-space exploration, we get 7.3% and 7.6% improvement in power with $5^{3-1} = 25$ and $5^{4-1} = 125$ OSFA runs for 3 and 4 scenarios, respectively. On the contrary, by using the second approach, the power savings reduce slightly to 6.7% and 7.2% with $5 \times (3-1) = 10$ and $5 \times (4-1) = 15$ OSFA runs, respectively for 3 and 4 scenarios. These experimental runs demonstrate the following: 1) as the number of operating conditions increases, there is generally more power savings and 2) the proposed speed-up technique can reduce the computational cost significantly with little compromise in solution quality.

D. Impact of Scenario Percentage

As mentioned in Section VI-A, the percentage for fast and slow scenario for Table II are chosen as 0.2 and 0.8, respectively. In order to explore the impact of scenario percentage on the percentage savings, we take an example benchmark

TABLE III
COMPARISON WITH THE WORST-CASE METHODOLOGY UNDER NBTI

Design	Power(W) in worst-case Methodology	Power(W) in OSFA Methodology	% Imprv.
DMA	0.375	0.244	34.9
pci_bridge32	0.226	0.156	31.0
des_perf	2.381	1.406	40.9
vga_lcd	0.633	0.510	19.4
b19	0.755	0.593	21.5
leon3mp	1.413	1.276	9.7
netcard	1.951	1.813	7.1
Average			23.5

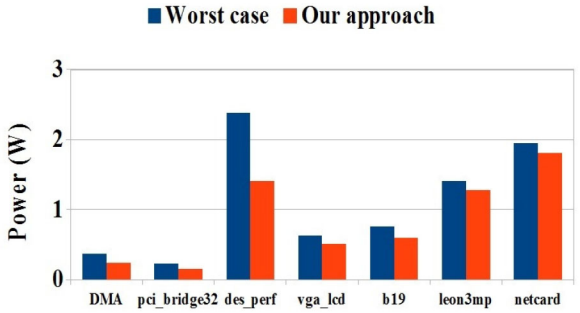

Fig. 8. Comparison with the worst case under NBTI.

"DMA" and run our OSFA methodology for other combinations such as (0.1+0.9), (0.3+0.7), (0.4+0.6), and (0.5+0.5) for, respectively, fast and slow scenario. As the scenario percentage for fast scenario increases, the percentage savings in power monotonically decreases from 5.9% → 4.4% → 4.0% → 3.4%. This is intuitive since we have considered fast scenario as the base scenario for the conventional approach, and as the percentage for the fast scenario increases, the savings of OSFA compared to conventional approach decreases. However, it should be stressed that since the design IPs in laptops or smart-phones run in the fast operating conditions for a very small fraction of time, the benefit of OSFA methodology would be more prominent for these applications.

E. OSFA Considering NBTI

Next, NBTI-impact is considered in our OSFA algorithms as described in Section V, and NBTI-driven gate-sizing framework is compared with the worst-case methodology. We have assumed a circuit-speed degradation of 20% due to NBTI. To compare with the worst-case methodology in our experimental set-up, we run our NBTI-aware OSFA methodology with 20% relaxed timing constraints and have compared the achieved power with the power which would have been achieved with the worst-case methodology under that relaxed timing constraints. The comparison of total power between our approach and the worst-case methodology for the designs is shown in Table III and Fig. 8. Overall, our approach improves the total power by 23.5% on average over all designs.

Note that the power consumption with NBTI (Table III) is less than that without NBTI (Table II). This is because we have allowed a 20% timing margin in presence of NBTI. For instance, the clock period for "des_perf" is 735 ps in Table II. Now, when we tabulate the worst-case methodology power number in Table III, we take the power numbers from OSFA methodology in Table II. But assuming 20% delay degradation, the clock period of the worst-case methodology in presence of NBTI would be 20% higher, and so when we run our OSFA methodology in presence of NBTI, we increase the clock period to $735 + 735 \times 0.2 = 882$ ps to match the same timing constrains with the worst-case methodology. Consequently, OSFA methodology, by embedding the multi-scenario NBTI model into the sizer, is able to exploit this pessimistic guard band to provide better power numbers.

We made several observations in these runs. First, improvement in our approach is more for more time-constrained designs. For instance, we achieve, respectively, 34.9% and 40.9% improvements in DMA and des_perf which are more time-constrained, where as comparatively smaller improvements, such as 9.7% and 7.1% for leon3mp and netcard, which are less timing constrained. This is because more the design is time-constrained, the pessimism in the worst-case-methodology increases the use of higher size and/or lower V_{th} cells more aggressively. As a result, improvement in our approach is more for more time-constrained designs.

Second, as shown in Fig. 7, there is some optimum voltage point in the design space exploration curve which provides the minimum power of the design. For our OSFA runs considering NBTI, this optimum supply voltage value for the second scenario increases for the timing constrained designs. For instance, this voltage increases from 0.85 to 0.9 V for DMA and des_perf. This can be explained as NBTI causes increase in threshold voltage, and for $V_{\text{dd}}^{\text{nom}} > V_{\text{dd}}^{i}$, the delay scaling factor in the performance relaxed scenario increases. Mathematically, from (2) delay scaling factor is given by

$$DF = \frac{V_{\text{dd}}^{i}}{V_{\text{dd}}^{\text{nom}}} \frac{\left(V_{\text{dd}}^{\text{nom}} - V_{\text{th}}\right)}{\left(V_{\text{dd}}^{i} - V_{\text{th}}\right)}. \tag{10}$$

Differentiating both sides by V_{th}, we get

$$\frac{d(\text{DF})}{dV_{\text{th}}} = \frac{V_{\text{dd}}^{i}}{V_{\text{dd}}^{\text{nom}}} \frac{\left(V_{\text{dd}}^{\text{nom}} - V_{\text{dd}}^{i}\right)}{\left(V_{\text{dd}}^{i} - V_{\text{th}}\right)^{2}}. \tag{11}$$

So $(d(\text{DF})/dV_{\text{th}}) > 0$ for $V_{\text{dd}}^{\text{nom}} > V_{\text{dd}}^{i}$. It causes the shift of optimum V_{dd}^{i} so that the overdrive voltage $(V_{\text{dd}}^{i} - V_{\text{th}})$ becomes sufficient to render the optimum voltage V_{dd}^{i} compatible with the nominal supply voltage in the performance-constrained scenario.

VII. CONCLUSION

This paper introduces a novel problem formulation of NBTI-aware gate-sizing under multiple operating conditions. We present our OSFA algorithms and a design-space exploration methodology to optimize power of any IP-design without affecting the performance at different operating conditions. Compared with conventional methodology, our approach has achieved an average power improvement of 6.1% in industry-strength large-scale benchmarks. We have also proposed a faster yet efficient design-space exploration technique for more

than two scenarios and demonstrated its effectiveness. The impact of scenario percentage on OSFA algorithms is also studied. By extending the existing NBTI-affected delay model across multiple operating conditions and incorporating this into our OSFA framework, a significant improvement in total power is achieved in comparison with the conventional guard-band-based methodology. We also experimentally observe that the power savings increases with the number of operating conditions. In future, we plan to enhance our OSFA methodology to consider other aging issues, such as positive bias temperature instability (PBTI) and HCI. With aggressive technology scaling, the number of operating conditions will further increase, and we believe the OSFA methodology will become more and more relevant in the VLSI industry.

REFERENCES

[1] S. Roy, D. Liu, J. Um, and D. Z. Pan, "OSFA: A new paradigm of gate-sizing for power/performance optimizations under multiple operating conditions," in *Proc. Design Autom. Conf.*, San Francisco, CA, USA, 2015, pp. 1–6.

[2] Y. Liu, J. Hu, and W. Shi, "Multi-scenario buffer insertion in multi-core processor designs," in *Proc. Int. Symp. Phys. Design*, Santa Rosa, CA, USA, pp. 15–22, 2008.

[3] Y. Liu and J. Hu, "A new algorithm for simultaneous gate sizing and threshold voltage assignment," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 29, no. 2, pp. 223–234, Feb. 2010.

[4] H. Chou, Y. H. Wang, and C. C. P. Chen, "Fast and effective gate-sizing with multiple-V_t assignment using generalized Lagrangian relaxation," in *Proc. Asia South Pac. Design Autom. Conf.*, Shanghai, China, 2005, pp. 381–386.

[5] J. Hu, A. B. Kahng, S. Kang, M. Kim, and I. L. Markov, "Sensitivity-based metaheuristics for accurate discrete gate sizing," in *Proc. Int. Conf. Comput.-Aided Design*, San Jose, CA, USA, 2012, pp. 233–239.

[6] M. M. Ozdal, S. Burns, and J. Hu, "Gate sizing and device technology selection algorithms for high-performance industrial designs," in *Proc. Int. Conf. Comput.-Aided Design*, San Jose, CA, USA, 2011, pp. 724–731.

[7] H. Ren and S. Dutt, "A network-flow based cell sizing algorithm," in *Proc. Int. Workshop Logic Synth.*, 2008, pp. 7–14.

[8] L. Li, P. Kang, Y. Lu, and H. Zhou, "An efficient algorithm for library-based cell-type selection in high-performance low-power designs," in *Proc. Int. Conf. Comput.-Aided Design*, San Jose, CA, USA, 2012, pp. 226–232.

[9] V. S. Livramento, C. Guth, J. L. Guntzel, and M. O. Johann, "A hybrid technique for discrete gate sizing based on Lagrangian relaxation," *ACM Trans. Design Autom. Electron. Syst.*, vol. 19, no. 4, 2014, Art. ID 40.

[10] A. Ramalingam, S. V. Kodakara, A. Devgan, and D. Z. Pan, "Robust analytical gate delay modeling for low voltage circuits," in *Proc. Asia South Pac. Design Autom. Conf.*, Yokohama, Japan, 2006, pp. 61–66.

[11] W. Wang *et al.*, "The impact of NBTI on the performance of combinational and sequential circuits," in *Proc. Design Autom. Conf.*, San Diego, CA, USA, 2007, pp. 364–369.

[12] Y. Wang *et al.*, "Temperature-aware NBTI modeling and the impact of input vector control on performance degradation," in *Proc. Design Autom. Test Europe*, Nice, France, 2007, pp. 1–6.

[13] S. V. Kumar, C. H. Kim, and S. S. Sapatnekar, "NBTI-aware synthesis of digital circuits," in *Proc. Design Autom. Conf.*, San Diego, CA, USA, 2007, pp. 370–375.

[14] R. Vattikonda, W. Wang, and Y. Cao, "Modeling and minimization of PMOS NBTI effect for robust nanometer design," in *Proc. Design Autom. Conf.*, San Francisco, CA, USA, 2006, pp. 1047–1052.

[15] M. A. Alam and S. Mahapatra, "A comprehensive model of PMOS NBTI degradation," *Microelectron. Rel.*, vol. 45, no. 1, pp. 71–81, 2005.

[16] S. V. Kumar, C. H. Kim, and S. S. Sapatnekar, "An analytical model for negative bias temperature instability," in *Proc. Int. Conf. Comput.-Aided Design*, San Jose, CA, USA, 2006, pp. 493–496.

[17] S. Roy and D. Z. Pan, "Reliability aware gate sizing combating NBTI and oxide breakdown," in *Proc. VLSI Design*, Mumbai, India, 2014, pp. 38–43.

[18] M. Ebrahimi, F. Oboril, S. Kiamehr, and M. B. Tahoori, "Aging-aware logic synthesis," in *Proc. Int. Conf. Comput.-Aided Design*, Austin, TX, USA, 2013, pp. 61–68.

[19] C.-H. Lin, S. Roy, C.-Y. Wang, D. Z. Pan, and D. Chen, "CSL: Coordinated and scalable logic synthesis techniques for effective NBTI reduction," in *Proc. Int. Conf. Comput. Design*, New York, NY, USA, 2015, pp. 236–243.

[20] X. Yang and K. Saluja, "Combating NBTI degradation via gate sizing," in *Proc. Int. Symp. Qual. Electron. Design*, San Jose, CA, USA, 2007, pp. 47–52.

[21] W. N. Li, "Strongly NP-hard discrete gate sizing problems," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 13, no. 8, pp. 1045–1051, Aug. 1994.

[22] M. M. Ozdal *et al.*, "The ISPD-2012 discrete cell sizing contest and benchmark suite," in *Proc. ISPD*, Santa Rosa, CA, USA, 2012, pp. 161–164.

[23] B. J. Sheu, D. L. Scharfetter, P.-K. Ko, and M.-C. Jeng, "BSIM: Berkeley short-channel IGFET model for MOS transistors," *IEEE J. Solid-State Circuits*, vol. 22, no. 4, pp. 558–566, Aug. 1987.

[24] C.-P. Chen, C. C. N. Chu, and D. F. Wong, "Fast and exact simultaneous gate and wire sizing by Lagrangian relaxation," in *Proc. Int. Conf. Comput.-Aided Design*, Austin, TX, USA, 1998, pp. 617–624.

[25] (Aug. 2015). *Intel Threaded Building Blocks*. [Online]. Available: https://www.threadingbuildingblocks.org/

[26] (Aug. 2015). *45 nm Predictive Technology High Performance Model*. [Online]. Available: http://ptm.asu.edu/modelcard/HP/45nm_HP.pm

[27] (Aug. 2015). *Mentor Graphics Functional Verification Modelsim*. [Online]. Available: http://www.mentor.com/products/fv/modelsim/

[28] Q. Wu, M. Pedram, and X. Wu, "A note on the relationship between signal probability and switching activity," in *Proc. Asia South Pac. Design Autom. Conf.*, Chiba, Japan, 1997, pp. 117–120.

Subhendu Roy (S'13–M'16) received the B.E. degree in electronics and telecommunication engineering from Jadavpur University, Kolkata, India, in 2006, the M.Tech. degree in electronic systems from the Indian Institute of Technology Bombay, Mumbai, India, in 2009, and the Ph.D. degree in electrical and computer engineering from the University of Texas at Austin, Austin, TX, USA, in 2015.

He is currently a Principal Software Engineer with Cadence Design Systems, San Jose, CA, USA. He has three years of full-time industry experience in an EDA company, Atrenta, India, where he was involved in developing tools in the architectural power domain and RTL domain. He was an Intern with IBM T. J. Watson Research Center, Yorktown Heights, NY, USA, in 2012, and Mentor Graphics, Fremont, CA, USA, from 2013 to 2014. He holds one patent and has first-authored papers in major EDA conferences/journals, such as Design Automation Conference (DAC), International Symposium on Physical Design (ISPD), Asia and South Pacific Design Automation Conference (ASPDAC), and the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS (TCAD). His current research interests include design automation for logic synthesis, physical design, and cross-layer reliability.

Mr. Roy was a recipient of the Best Paper Award at ISPD'14.

Derong Liu received the B.S. degree in microelectronics from Fudan University, Shanghai, China, in 2011. She is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX, USA, under the supervision of Prof. D. Z. Pan.

Her current research interests include physical design and design automation for logic synthesis.

Jagmohan Singh received the B.Tech. degree in electronics and communication engineering from the National Institute of Technology Jalandhar, Jalandhar, India, in 2006, and the M.S. degree in electrical and computer engineering from the University of Texas at Austin, Austin, TX, USA, in 2013, with focus on discrete gate sizing for power optimization under performance constraints.

Since 2013, he has been a Software Engineer with Cadence Design Systems, San Jose, CA, USA. From 2006 to 2011, he was with STMicroelectronics, Greater Noida, India. He was an Intern with Motorola, India, in 2005, and Advanced Micro Devices, Austin, TX, USA, in 2012. His current research interests include discrete optimization, algorithms, and automation for physical design of integrated circuits.

Junhyung Um received the B.S. degree from Seoul National University, Gwanak, Korea, in 1996, and the M.S. and Ph.D. degrees from the Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 1999 and 2003.

She is currently a Principal Engineer with Samsung Electronics, Suwon, Korea.

David Z. Pan (S'97–M'00–SM'06–F'14) received the B.S. degree from Peking University, Beijing, China, and the M.S. and Ph.D. degrees from the University of California at Los Angeles, Los Angeles, CA, USA.

From 2000 to 2003, he was a Research Staff Member with IBM T. J. Watson Research Center, Yorktown Heights, NY, USA. He is currently the Engineering Foundation Professor with the Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX, USA. He has published over 240 papers in refereed journals and conferences, and holds eight U.S. patents. He has graduated 20 Ph.D. students. His current research interests include cross-layer nanometer IC design for manufacturability and reliability, new frontiers of physical design, and CAD for emerging technologies such as 3-D-IC, biochip, and nano photonics.

Prof. Pan was a recipient of a number of awards for his research contributions and professional services, including the SRC 2013 Technical Excellence Award, the DAC Top 10 Author in Fifth Decade, the DAC Prolific Author Award, the ASP-DAC Frequently Cited Author Award, 12 Best Paper Awards, such as ISPD 2014, International Conference on Computer-Aided Design (ICCAD) 2013, ASPDAC 2012, ISPD 2011, IBM Research 2010 Pat Goldberg Memorial Best Paper Award in EE/CS/Math, ASPDAC 2010, Design, Automation and Test in Europe (DATE) 2009, International Conference on IC Design and Technology (ICICDT) 2009, SRC Techcon in 1998, 2007, 2012, and 2015, and 11 other Best Paper Award Nominations at DAC/ICCAD/ASPDAC/ISPD, Communications of the ACM Research Highlights in 2014, the ACM/SIGDA Outstanding New Faculty Award in 2005, the NSF CAREER Award in 2007, the SRC Inventor Recognition Award three times, the IBM Faculty Award four times, the UCLA Engineering Distinguished Young Alumnus Award in 2009, the UT Austin RAISE Faculty Excellence Award in 2014, the ISPD Routing Contest Awards in 2007, the eASIC Placement Contest Grand Prize in 2009, and the ICCAD CAD Contest Awards in 2012 and 2013, among others. He has served as a Senior Associate Editor of *ACM Transactions on Design Automation of Electronic Systems*, an Associate Editor of the IEEE TRANSACTIONS ON COMPUTER AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION SYSTEMS, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS PART I, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS PART II, the IEEE DESIGN AND TEST, *Science China Information Sciences*, the *Journal of Computer Science and Technology*, and the IEEE CAS Society Newsletter. He has served on the Executive/Program Committees of many major conferences, including DAC, ICCAD, ASPDAC, and ISPD.