Gate Replacement Techniques for Simultaneous Leakage and Aging Optimization

Yu Wang¹, Xiaoming Chen¹, Wenping Wang², Yu Cao², Yuan Xie³, Huazhong Yang¹
¹Dept. of E.E., TNList, Tsinghua Univ., Beijing, China
²Dept. of E.E., Arizona State Univ., USA, ³Dept. of CSE, Pennsylvania State Univ., USA
¹ Email: yu-wang@mail.tsinghua.edu.cn

Abstract—As technology scales, the aging effect caused by Negative Bias Temperature Instability (NBTI) has become one of the major reliability concerns for circuit designers. On the other hand, reducing leakage power remains to be one of the design goals. Because both NBTI-induced circuit degradation and standby leakage power have a strong dependency on the input vectors, Input Vector Control (IVC) technique may be adopted to mitigate leakage and NBTI. However, IVC technique is ineffective for larger circuits. Therefore, in this paper, we propose two fast gate replacement algorithms together with optimal input vector selection to simultaneously mitigate leakage power and NBTI induced circuit degradation: Direct Gate Replacement (DGR) algorithm and Divide and Conquer Based Gate Replacement (DCBGR) algorithm. Our experimental results on 20 benchmark circuits at 65nm technology node reveal that: 1) Both DGR and DCBGR algorithms outperform pure IVC about on average 20% for three different object functions: leakage power reduction only, NBTI mitigation only, and leakage/NBTI co-optimization. 2) The DCBGR algorithm leads to better optimization results and save on average 100X runtime compared with the DGR algorithm.

I. INTRODUCTION

As technology scales, Negative Bias Temperature Instability (NBTI) is emerging as one of the major reliability degradation mechanisms [1]. NBTI occurs when PMOS transistors are negatively biased (i.e., $V_{gs} = -V_{dd}$) at elevated temperature, causing a shift in threshold voltages. Over a long period of time, such $V_{th}$ shifts can potentially cause a significant increase in the delay of PMOS devices [2], and result in about 10-20% degradation in circuit speed, thus may lead to a functional failure [3]. The impact of NBTI on circuit performance has become a key issue with technology scaling [4]. Consequently, it is important to model, analyze, and mitigate the impact of the NBTI effect on the circuit performance.

Based on the various circuit level NBTI degradation analysis models [5]–[7], previous works estimated the NBTI induced lifetime degradation with the assumption that the circuits operate all the time. However, in practical not every application requires the underlying hardware to operate at the highest performance level all the time. Modules in which the computation is burst are often idle. There are periods during which the PMOS transistors are under static stress condition. Many PMOS transistors affected by NBTI can be found in both combinational and storage blocks when the gate inputs are set to “0” during the standby time, leading to a larger degradation. Consequently, it is important to accurately estimate the NBTI-induced degradation at the standby time in order to safely guard-band the circuit performance, and to find design techniques to mitigate such degradation.

Input Vector Control (IVC) is a well-studied technique for leakage power reduction [8] at the standby time. Since NBTI also depends on the input patterns of PMOS devices, IVC can be used to mitigate the NBTI effect during the standby mode. Fig. 1 shows the relation between the circuit leakage power and the circuit delay degradation caused by NBTI under different input vectors. We can see that given the required constraint for both leakage and delay degradation (the shadow region in Fig. 1), a set of input vectors can be preselected and applied to the entire circuit at the standby mode, such that both the total leakage power and delay degradation are minimized. In this example, less than 1% of sampled input patterns provides the minimum of both circuit degradation and the leakage.

Fig. 1. Leakage power versus delay degradation for different input vectors.

Wang et al. [9] proposed a method to select the best input vectors from the minimum leakage vector set. However, the best input vectors for minimum leakage power may not be the best input vectors to minimize NBTI-induced circuit degradation and they didn’t consider the difference of NBTI effects during active and standby time, the results claimed only 3% circuit degradation saving at the 90nm technology node. Jaume et al. [10] used different input vectors to change the zero-probability of internal PMOS transistors, so that the PMOS transistors’ degradation was evenly distributed. The effect of this technique on an adder is evaluated, however, detailed research for random logic is needed.

Although pure IVC techniques have been evaluated for mitigating NBTI, they are not very effective when the circuit becomes larger. How to efficiently find the optimal results for leakage and NBTI induced circuit degradation remains a problem. There is no literature about simultaneously NBTI and leakage mitigation through Internal Node Control (INC) [11]–[13] which is proved to be more effective to reduce leakage power than pure IVC.

In this paper, we propose two fast gate replacement algorithms which simultaneously mitigate the leakage power and NBTI induced circuit degradation. The contributions of this paper can be summarized in the following aspects:

1) The gate replacement techniques are for the first time used for NBTI mitigation. Based on the basic gate replacement technique for
NBTI and leakage reduction, we first propose a Direct Gate Replacement (DGR) algorithm and then propose a Divide and Conquer Based Gate Replacement (DCBGR) algorithm to further improve the NBTI/leakage reduction achievement and the optimization speed.

2) The complexity of DGR algorithm is $O(n^2)$ in the worst case and $O(n)$ on average; while the complexity of DCBGR is $O(n)$. Therefore, our algorithms will serve well when circuit scale becomes larger.

3) Our experimental results show that: for larger circuits, IVC technique is less effective, while INC through gate replacement technique is more effective for both NBTI and leakage mitigation.

4) Although the gate replacement technique is compatible with standard cell design flow, the area penalty remains a problem. Our DCBGR results for leakage only and NBTI only show that the area penalty for leakage reduction is larger: on average 3.26%, while the area penalty for NBTI mitigation is smaller: on average 3.53%.

II. PRELIMINARIES

A. Degradation Model under NBTI Effect

Depending on the bias condition of PMOS transistor, NBTI has two phases: stress phase and recovery phase. In the stress phase ($V_{gs} = 0$), the holes in the channel weaken the Si-H bonds, which results in the generation of the positive interface charges and hydrogen species, correspondingly, threshold voltage ($V_{th}$) of the PMOS transistors increases. During the recovery phase ($V_{gs} = V_{DD}$), the interface traps can be annealed by the hydrogen species and thus, $V_{th}$ degradation ($\Delta V_{th}$) is partially recovered. If a PMOS device is always under stress condition, it is referred as static NBTI. Otherwise, both stress and recovery exist during active circuit operation, it is described as dynamic NBTI.

Fig. 2. Static and dynamic NBTI degradation for different input signal probabilities.

Based on the reaction-diffusion mechanism, real time NBTI model is developed in [14], [15]. For dynamic NBTI, there is a sudden change at the beginning of the recovery phase, which has a significant impact on the estimation of NBTI degradation. A long term prediction model is derived for both static and dynamic NBTI in [15]. Fig. 2 shows $\Delta V_{th}$ prediction by using the proposed model. The big difference between the static and dynamic NBTI, has also been observed in silicon data [16], [17]. Therefore, the simple static analysis may cause an extremely pessimistic estimation of NBTI-induced degradation and consequently, results in over-margining in design stage. On the contrary, only dynamic NBTI model for the total lifetime without considering the static NBTI effect during the standby time may lead to an underestimation of NBTI-induced performance degradation. In this paper, we use dynamic NBTI model in the active time and static NBTI model in the standby time.

The delay difference due to $\Delta V_{th}$ is given by [9], [18]:

$$
\Delta d(v) = \alpha \Delta V_{th} / (V_{gs} - V_{th}) \times d(v)
$$

(1)

where $d(v)$ is the original delay of gate $v$ which can be extracted from the commercial STA tools. There could be several $\Delta V_{th}$ of different PMOS’s in one gate. In such cases, we just select the largest one to calculate the gate delay degradation, which is the worst case delay degradation.

B. NBTI/leakage co-simulation flow

Fig. 3 shows our NBTI/leakage co-simulation flow. For a given circuit, commercial static timing analysis tool is firstly used to generate the Potential Critical Paths (PCPs) using standard timing libraries. When the circuit is in the active mode, statistical information for input Signal Probability (SP) is used to generate the internal node SP. When circuit is in the standby mode, logic simulator is used to generate the voltage level of each internal node. The active time internal node SP and the standby time internal node states are used to estimate the NBTI-induced $V_{th}$ degradation through transistor level NBTI modeling. The leakage power is estimated based on the input vector aware leakage lookup tables. Based on the $V_{th}$ degradation estimation and the original timing libraries, a fast path-based NBTI-aware timing analysis is performed. We modify the input vector generation module to implement our gate replacement algorithms.

III. GATE REPLACEMENT (GR) TECHNIQUE

The gate replacement technique is to replace a gate $G(\overline{x})$ by another library gate $G(\overline{x}, sleep)$ [12], where $\overline{x}$ is the input vector of gate $G$, sleep is the sleep signal of the circuit, such that:

1) $G(\overline{x}, 0) = G(\overline{x})$, when the circuit is active (sleep = 0);
2) $G(\overline{x}, 1)$ has smaller leakage power or can serve as an INC point to mitigate NBTI effect when the circuit is standby (sleep = 1).

1) Gate replacement for NBTI: The NBTI effect on a PMOS transistor depends on the stress condition: $V_{gs}$ and stress time, which are both related to the input state of a gate. Consequently, all $1$’s will be the best input pattern with the smallest NBTI-induced degradation for all gate types. Fig. 4 is an example that shows how to mitigate NBTI-induced degradation by gate replacement. The NAND2 gate G2’s delay will be larger if G1’s output is 0 at the circuit standby time. Through gate replacement technique, we replace G1 by an NAND3 gate so that the output is changed to 1. Hence the NBTI effect on G2 is mitigated during the standby time.

Fig. 4. A gate replacement example for NBTI mitigation.
2) \textit{Gate replacement for leakage}: We call a gate at its WLS (worst leakage state) [12] when its input vector leads to the largest leakage power. Fig. 5 shows how to replace an NAND2 gate to reduce its leakage power. The NAND2 gate is in WLS with leakage power 454.71nW, when its input is 11. We replace it with an NAND3 gate, of which the leakage power is 249.1nW during the standby time. Then we can save up to 45.2\% of the leakage power.

Fig. 5. A gate replacement example for leakage reduction.

3) \textit{Different input vector dependency of NBTI and leakage}: All 1’s will be the best input pattern with the smallest NBTI-induced degradation for all gate types. Meanwhile, leakage power varies among different input vectors. We simulate all the cells (NAND/AND, NOR/OR, INV, BUF) in the library, and find out that the best case input patterns to mitigate the leakage for NAND/AND/INV gates are all 0’s at the inputs, while for NOR/OR/BUF gates are all 1’s at the inputs. Therefore, although NBTI and leakage both depend on the input patterns, we can see the discrepancy: for NAND/AND/INV gates, the input pattern for least leakage will lead to worst NBTI-induced delay degradation; on the contrary, for NOR/OR/BUF gates, the input pattern for least leakage will lead to best case NBTI-induced delay degradation. Consequently, if we use pure IVC technique, the best input vector for leakage may lead to worse NBTI induced degradation, and vice versa. Therefore, we have to get a thorough control of internal node state through INC techniques, such as gate replacement technique, so that the internal node state can be carefully chosen to meet both leakage power and lifetime requirements.

4) \textit{Overhead analysis of gate replacement}: Gate replacement will introduce delay and area overhead; however, these overhead can be controlled by adding delay and area constraints during the optimization algorithm, or transistor re-sizing. In this paper, the delay constraint is set to be less than 5\% of the original delay at time 0 after gate replacement. From our experimental results, although delay requirement at time 0 is relaxed, we will get a better circuit delay after 10 years. For power overhead, the dynamic power overhead is trivial, because the sleep signal remains constant at both active and standby mode; the leakage power overhead during circuit active mode caused by the leakage difference of different gate types can be neglected if the standby time is long enough.

IV. \textsc{Gate Replacement Algorithms}

In this section, we propose our two fast gate replacement algorithms: Direct Gate Replacement algorithm and Divide and Conquer Based Gate Replacement algorithm.

A. \textsc{Direct Gate Replacement (DGR) algorithm}

Similar to the previous gate replacement algorithm [12], there are also two key steps for the Direct Gate Replacement: 1) Get the optimal input vector for circuits; 2) Gate replacement based on the optimal input vector. We follow the two steps and amend the previous algorithm to further consider NBTI induced circuit degradation together with leakage power.

\textit{1) Get the optimal input vector}: An optimal input vector is chosen from 10K random input vector search. Since we are considering NBTI effect and leakage power together, the object function is as follows:

\[
F(D_{\text{circuit}}, P_{\text{leakage}}) = A \times D_{\text{circuit}} + B \times P_{\text{leakage}}
\]  

(2)

where \(D_{\text{circuit}}\) is the circuit delay after 10 years; the \(P_{\text{leakage}}\) is the circuit leakage power at time 0. \(A\) and \(B\) are two weight constants for circuit designers to balance the leakage power requirement and circuit lifetime requirement. The best leakage and circuit delay results of random search are used as our reference.

B. \textit{Direct Gate Replacement based on the optimal input vector}:

In the DGR algorithm, we first arrange all the gates in the circuit into a topological order. The topological order guarantees that when we find a gate at its WLS, all its predecessors have already been considered. Then all the gates are evaluated one by one according to this order. The detailed algorithm is shown in Fig. 6. Firstly, all the critical paths are investigated to mitigate the NBTI effect, and then we evaluate the gates in the circuits to further reduce the leakage power.

Fig. 6. Pseudo code for Direct Gate Replacement algorithm.

\begin{verbatim}

\begin{itemize}
  \item[1)] perform NBTI mitigation algorithm //NBTI mitigation Part
  \item[2)] for \(i\) to \(n\) do
  \item[3)] if \(G_i\) is WLS and not visited
  \item[4)] if \(G_i\) is not in critical path then include \(G_i\) in selection \(S\)
  \item[5)] else if \(G_i\)'s output will not be changed after replacement then include \(G_i\) in selection \(S\)
  \item[6)] while there is new addition to \(S\)
  \item[7)] for each newly selected gate \(G_i\) in \(S\) do
  \item[8)] temporarily replace \(G_i\)
  \item[9)] if \(G_i\)'s output is changed then
  \item[10)] include all \(G_i\)'s fanout gates in selection \(S\) that are unvisited and their output will not be changed after replacement
  \item[11)] calculate leakage change caused by the replacement
  \item[12)] if there is a leakage reduction then
  \item[13)] mark all the gates in \(S\) as visited
  \item[14)] make all the replacement above
  \item[15)] else mark \(G_i\) as visited only
  \item[16)] empty \(S\)
  \item[17)] else mark \(G_i\) as visited
  \item[18)] end
\end{itemize}

\end{verbatim}

\textit{B. \textsc{NBTI mitigation in the critical paths (Fig. 7)}:} The first line of DGR algorithm Fig. 6 is to perform the NBTI mitigation algorithm shown in Fig. 7. When we consider a gate \(G_c\), the critical fin-in gate \(G_c\) on the critical path is first selected (line 2). To mitigate the effect of NBTI in the critical path, the output value of \(G_c\) should be set to 1. If the output of \(G_c\) is 0 and there is a library gate \(G_c\) that can replace \(G_c\), then we replace \(G_c\) with \(G_c\) (line 3-4). After the replacement, if the output is not changed to 1, then we will try to find all the fin-in gates of gate \(G_c\), and replace them according to

Fig. 7. Pseudo code for NBTI mitigation algorithm.
is the maximum fan-in number of gates in the circuit before deleting the connections. The complexity of leakage part is input vector or an input vector that has been appeared previously. any dangling inputs that have been changed, the algorithm will be the dangling inputs again by a reverse topological order. If there are When we have got all the inputs of each gate, we assign the longest path from $G_k$ before deleting the connections. dangling inputs are always equal to the output of their fan-in gates before deleting the connections. The complexity of this algorithm is $O(n^2)$, where $n$ is the total gate number in the circuit.

B. Divide and Conquer Based Gate Replacement (DCBGR) algorithm

Although DGR algorithm described in the previous subsection can achieve better results compared with the results of pure IVC technique, the complexity is $O(n^2)$ which is not scalable when the circuit size becomes larger. On the other hand, since the DGR algorithm is performed based on an initial input vector, the final optimization results may still have a gap with the optimal ones.

We further propose a Divide and Conquer Based Gate Replacement (DCBGR) algorithm based on the improved gate replacement algorithm in [11]: 1) the circuit is divided into several trees; 2) our dynamic programming algorithm is performed on the tree circuits to achieve better results faster; 3) we adjust the dangling nodes in the whole circuit, and continue to perform the algorithm until it converges.

1) Divide the circuit into trees: At the beginning, we divide the circuit into tree circuits by deleting some connections between gates until every gate fans out to at most one gate. For example, if a gate $G$ fans out to $k$ gates $G_1, ..., G_k$, we keep one connection $G_i$ and delete other $k - 1$ connections. We keep the connection that has the longest path from $G_i$ to the outputs of the circuit. After deleting the connections, there are many dangling inputs. In this algorithm, all the dangling inputs are always equal to the output of their fan-in gates before deleting the connections.

2) Gate replacement for trees: The detailed algorithm is shown in Fig. 8, where $t_i$ denotes the $j_{th}$ input of $G_i$; $N(i)$ denotes the input number of $G_i$; $LK(i, z)$ denotes the minimum total leakage power of the subtree rooted at $G_i$, when its output is $z$; $V(i, z)$ denotes the input vector producing $LK(i, z)$; $\overline{z}$ denotes the input vector of a gate; $x^j_i$ denotes the $j^{th}$ bit of $\overline{z}$; $L(i, z)$ and $L_R(i, z)$ denote the leakage power of $G_i$ and replaced $G_i$ respectively; $Out(i, \overline{z})$ denotes the output of $G_i$ with its input vector $\overline{z}$.

Initialization is firstly performed for all the gates from line 1 to 5. Then we perform the NBTI mitigation algorithm described in Fig. 7. We modify the algorithm in [11] to serve as our leakage reduction part.

3) Adjust dangling assignments and perform the algorithm until it converges: When we have got all the inputs of each gate, we assign the dangling inputs again by a reverse topological order. If there are any dangling inputs that have been changed, the algorithm will be repeated from the most anterior gate in the topological order with new dangling input values until the algorithm generates the same input vector or an input vector that has been appeared previously.

4) Complexity: The complexity of NBTI part is $O(Kn)$ where $K$ is the maximum fan-in number of gates in the circuit before deleting the connections. The complexity of leakage part is $O(n^2)$ [11].

C. C17 circuit as an example of DCBGR algorithm

Fig. 9 shows an example of the DCBGR algorithm for circuit C17. At the beginning, we divide the circuit into trees by deleting connections in Fig. 9(1), $G_1, G_4$, and $G_6$ have dangling inputs. If we set the input vector of the circuit to all 0’s then the value of these dangling inputs are 011, we set 011 to these dangling inputs as their initial values in Fig. 9(2).

Then we run the algorithm for two different object functions: leakage power reduction only and NBTI mitigation only. The critical paths are marked in red in Fig. 9(5). If NBTI mitigation is considered, the internal node values along these paths should be 1 as more as possible. Then the dynamic algorithm will generate optimal input vectors for different object functions (Fig. 9(3)). We calculate all the logic values in the circuit and assign new dangling inputs in Fig. 9(4). With the new dangling inputs, the algorithm is repeated again until the algorithm converges to optimal input vectors for different object functions as shown in Fig. 9(5). Hence, the optimal input vector for leakage is 00010 while the optimal input vector for NBTI is 11000.

The detailed results are listed in Table I. $D_{nbti}$ is the original delay at time 0. $D_{d}$ is the circuit delay after 10 years. $L_K$ is the leakage power at time 0. The optimal result for NBTI can save the circuit degradation from 8.79% to 1%, since the NBTI effect are eliminated.

![Fig. 8. Pseudo code for Divide and Conquer Based Gate Replacement algorithm for a tree circuit.](image1)

![Fig. 9. An example of DCBGR algorithm for NBTI and leakage mitigation.](image2)
TABLE I

DCBG algorithm can outperform the pure IVC about 10% and 16.56% for leakage only and NBTI only respectively. For larger circuits, we have slightly more leakage saving but potentially larger NBTI effect mitigation because the critical paths may be longer in larger circuits. We also evaluate the runtime and area penalty $\alpha_{inc}$. The runtime grows fast when the circuit becomes bigger. For C6288, we may need several minutes.

Table IV shows the optimization results for simultaneous leakage and NBTI mitigation. $D_{imp}$ is the delay improvement after 10 years. $L_{K_{imp}}$ is the leakage improvement at time 0. These improvements are compared with the best results of Random Search in Table I. Our

DGR algorithm can outperform the pure IVC about 10% and 16.56% for leakage only and NBTI only respectively. For larger circuits, we have slightly more leakage saving but potentially larger NBTI effect mitigation because the critical paths may be longer in larger circuits. We also evaluate the runtime and area penalty $\alpha_{inc}$. The runtime grows fast when the circuit becomes bigger. For C6288, we may need several minutes.
VI. CONCLUSIONS

Table V shows the optimization results of DCBGR algorithm. All the results are compared with previous DGR algorithm. For leakage only, DCBGR can achieve on average 31.73% leakage power saving while DGR can achieve 10%. For NBTI only, DCBGR can compensate on average 23.65% NBTI induced circuit degradation, while DGR result is 16.56%. The best results in Table II are better than the weighted results in Table IV, hence DCBGR algorithm can achieve better results than the DGR algorithm for co-optimization.

From Table V, the results show that the area overhead for leakage reduction is larger than that for NBTI mitigation, since leakage power saving is much larger than that of NBTI mitigation. Hence, for future work, co-optimization techniques will be further explored during logic synthesis combined with NBTI optimization phase. The area overhead for NBTI optimization phase is much less than that of leakage optimization phase.

Power and reliability become two key design goals with technology scaling down. In this paper, we have proposed two gate reconfiguration techniques for leakage power and NBTI-induced aging effect mitigation. Both DGR and pure IVC technique are capable to achieve better results than the overhead of circuit delay at time 0. The pure IVC technique is much faster than the DGR algorithm. We also analyze the overhead of circuit delay at time 0 caused by gate replacement technique. The area overhead for leakage power reduction is much larger than that of NBTI mitigation, since leakage power saving is much larger than that of NBTI mitigation.

For leakage only, DCBGR can achieve on average 31.73% leakage power saving while DGR can achieve 10%. For NBTI only, DCBGR can compensate on average 23.65% NBTI induced circuit degradation, while DGR result is 16.56%. The best results in Table II are better than the weighted results in Table IV, hence DCBGR algorithm can achieve better results than the DGR algorithm for co-optimization.

From Table V, the results show that the area overhead for leakage reduction is larger than that for NBTI mitigation, since leakage power saving is much larger than that of NBTI mitigation. Hence, for future work, co-optimization techniques will be further explored during logic synthesis combined with NBTI optimization phase. The area overhead for NBTI optimization phase is much less than that of leakage optimization phase.

Power and reliability become two key design goals with technology scaling down. In this paper, we have proposed two gate reconfiguration techniques for leakage power and NBTI-induced aging effect mitigation. Both DGR and pure IVC technique are capable to achieve better results than the overhead of circuit delay at time 0. The pure IVC technique is much faster than the DGR algorithm. We also analyze the overhead of circuit delay at time 0 caused by gate replacement technique. The area overhead for leakage power reduction is much larger than that of NBTI mitigation, since leakage power saving is much larger than that of NBTI mitigation.

For leakage only, DCBGR can achieve on average 31.73% leakage power saving while DGR can achieve 10%. For NBTI only, DCBGR can compensate on average 23.65% NBTI induced circuit degradation, while DGR result is 16.56%. The best results in Table II are better than the weighted results in Table IV, hence DCBGR algorithm can achieve better results than the DGR algorithm for co-optimization.

From Table V, the results show that the area overhead for leakage reduction is larger than that for NBTI mitigation, since leakage power saving is much larger than that of NBTI mitigation. Hence, for future work, co-optimization techniques will be further explored during logic synthesis combined with NBTI optimization phase. The area overhead for NBTI optimization phase is much less than that of leakage optimization phase.

Power and reliability become two key design goals with technology scaling down. In this paper, we have proposed two gate reconfiguration techniques for leakage power and NBTI-induced aging effect mitigation. Both DGR and pure IVC technique are capable to achieve better results than the overhead of circuit delay at time 0. The pure IVC technique is much faster than the DGR algorithm. We also analyze the overhead of circuit delay at time 0 caused by gate replacement technique. The area overhead for leakage power reduction is much larger than that of NBTI mitigation, since leakage power saving is much larger than that of NBTI mitigation.

For leakage only, DCBGR can achieve on average 31.73% leakage power saving while DGR can achieve 10%. For NBTI only, DCBGR can compensate on average 23.65% NBTI induced circuit degradation, while DGR result is 16.56%. The best results in Table II are better than the weighted results in Table IV, hence DCBGR algorithm can achieve better results than the DGR algorithm for co-optimization.

From Table V, the results show that the area overhead for leakage reduction is larger than that for NBTI mitigation, since leakage power saving is much larger than that of NBTI mitigation. Hence, for future work, co-optimization techniques will be further explored during logic synthesis combined with NBTI optimization phase. The area overhead for NBTI optimization phase is much less than that of leakage optimization phase.

Power and reliability become two key design goals with technology scaling down. In this paper, we have proposed two gate reconfiguration techniques for leakage power and NBTI-induced aging effect mitigation. Both DGR and pure IVC technique are capable to achieve better results than the overhead of circuit delay at time 0. The pure IVC technique is much faster than the DGR algorithm. We also analyze the overhead of circuit delay at time 0 caused by gate replacement technique. The area overhead for leakage power reduction is much larger than that of NBTI mitigation, since leakage power saving is much larger than that of NBTI mitigation.

For leakage only, DCBGR can achieve on average 31.73% leakage power saving while DGR can achieve 10%. For NBTI only, DCBGR can compensate on average 23.65% NBTI induced circuit degradation, while DGR result is 16.56%. The best results in Table II are better than the weighted results in Table IV, hence DCBGR algorithm can achieve better results than the DGR algorithm for co-optimization.

From Table V, the results show that the area overhead for leakage reduction is larger than that for NBTI mitigation, since leakage power saving is much larger than that of NBTI mitigation. Hence, for future work, co-optimization techniques will be further explored during logic synthesis combined with NBTI optimization phase. The area overhead for NBTI optimization phase is much less than that of leakage optimization phase.

Power and reliability become two key design goals with technology scaling down. In this paper, we have proposed two gate reconfiguration techniques for leakage power and NBTI-induced aging effect mitigation. Both DGR and pure IVC technique are capable to achieve better results than the overhead of circuit delay at time 0. The pure IVC technique is much faster than the DGR algorithm. We also analyze the overhead of circuit delay at time 0 caused by gate replacement technique. The area overhead for leakage power reduction is much larger than that of NBTI mitigation, since leakage power saving is much larger than that of NBTI mitigation.

For leakage only, DCBGR can achieve on average 31.73% leakage power saving while DGR can achieve 10%. For NBTI only, DCBGR can compensate on average 23.65% NBTI induced circuit degradation, while DGR result is 16.56%. The best results in Table II are better than the weighted results in Table IV, hence DCBGR algorithm can achieve better results than the DGR algorithm for co-optimization.

From Table V, the results show that the area overhead for leakage reduction is larger than that for NBTI mitigation, since leakage power saving is much larger than that of NBTI mitigation. Hence, for future work, co-optimization techniques will be further explored during logic synthesis combined with NBTI optimization phase. The area overhead for NBTI optimization phase is much less than that of leakage optimization phase.

Power and reliability become two key design goals with technology scaling down. In this paper, we have proposed two gate reconfiguration techniques for leakage power and NBTI-induced aging effect mitigation. Both DGR and pure IVC technique are capable to achieve better results than the overhead of circuit delay at time 0. The pure IVC technique is much faster than the DGR algorithm. We also analyze the overhead of circuit delay at time 0 caused by gate replacement technique. The area overhead for leakage power reduction is much larger than that of NBTI mitigation, since leakage power saving is much larger than that of NBTI mitigation.

For leakage only, DCBGR can achieve on average 31.73% leakage power saving while DGR can achieve 10%. For NBTI only, DCBGR can compensate on average 23.65% NBTI induced circuit degradation, while DGR result is 16.56%. The best results in Table II are better than the weighted results in Table IV, hence DCBGR algorithm can achieve better results than the DGR algorithm for co-optimization.

From Table V, the results show that the area overhead for leakage reduction is larger than that for NBTI mitigation, since leakage power saving is much larger than that of NBTI mitigation. Hence, for future work, co-optimization techniques will be further explored during logic synthesis combined with NBTI optimization phase. The area overhead for NBTI optimization phase is much less than that of leakage optimization phase.

Power and reliability become two key design goals with technology scaling down. In this paper, we have proposed two gate reconfiguration techniques for leakage power and NBTI-induced aging effect mitigation. Both DGR and pure IVC technique are capable to achieve better results than the overhead of circuit delay at time 0. The pure IVC technique is much faster than the DGR algorithm. We also analyze the overhead of circuit delay at time 0 caused by gate replacement technique. The area overhead for leakage power reduction is much larger than that of NBTI mitigation, since leakage power saving is much larger than that of NBTI mitigation.

For leakage only, DCBGR can achieve on average 31.73% leakage power saving while DGR can achieve 10%. For NBTI only, DCBGR can compensate on average 23.65% NBTI induced circuit degradation, while DGR result is 16.56%. The best results in Table II are better than the weighted results in Table IV, hence DCBGR algorithm can achieve better results than the DGR algorithm for co-optimization.

From Table V, the results show that the area overhead for leakage reduction is larger than that for NBTI mitigation, since leakage power saving is much larger than that of NBTI mitigation. Hence, for future work, co-optimization techniques will be further explored during logic synthesis combined with NBTI optimization phase. The area overhead for NBTI optimization phase is much less than that of leakage optimization phase.