# BTI-Aware Sleep Transistor Sizing Algorithm for Reliable Power Gating Designs

Kai-Chiang Wu, Ing-Chao Lin, Yao-Te Wang, and Shuen-Shiang Yang

***Abstract*—Power gating is an effective way to reduce leakage power. This technique uses high $V_{th}$ transistors, called sleep transistors, to turn off the power supply. However, sleep transistors suffer from the bias temperature instability (BTI) effect, resulting in an increased $V_{th}$, and reduced reliability. This paper proposes two BTI-aware sleep transistor sizing algorithms to reduce the total width of sleep transistors based on the distributed sleep transistor network structure. The proposed algorithms reduce total width by more than 16.08%. More area can be reduced if the BTI effect on both sleep and cluster transistors is considered.**

***Index Terms*—BTI effect, power gating, reliability.**

## I. Introduction

As CMOS technology is scaled down, leakage power increases exponentially and thus has become a critical issue. An effective way to reduce leakage power is power gating [1], [2], [4]–[6]. This technique uses high-$V_{th}$ transistors, called sleep transistors, to turn off the power supply, reducing leakage power in standby mode.

There are two types of power gating designs: header- and footer-based designs, each of which use pMOS and nMOS as a sleep transistor. Because the sleep transistor behaves as a resistor, its width should be sufficiently large to avoid excessive IR-drop. Fig. 1(a) shows a footer-based design [1]. The circuit is divided into several smaller clusters, each with a pMOS header sleep transistor. However, the maximum instantaneous current (MIC) of a cluster may be large, resulting in large sleep transistor width. Long and He [6] proposed a distributed sleep transistor network (DSTN), as shown in Fig. 1(b), and sleep transistor sizing algorithms were proposed in [2]–[4] to reduce area overhead. Compared to the cluster-based design, this design connects all virtual ground ($V_{GND}$) lines together. Therefore, the current can flow from one cluster to all sleep transistors, and a discharging current can be shared among the sleep transistors, reducing sleep transistor sizes. Note that the virtual ground also has wire resistance that cannot be ignored; therefore, an empirical parameter was used to replace the effect of the virtual ground resistance on a discharging current. However, this parameter cannot accurately model the effect of the virtual ground resistance.

To obtain accurate MIC/IR-drop profiles with the virtual ground resistance in the DSTN structure, a discharging matrix method is proposed method in [2] to calculate the current more accurately. This algorithm partitions a period of time into uniform time frame partitions (UTPs) or variable-length time frame partitions (VTPs). When the number of UTPs is increased, more accurate MIC and smaller total sleep transistor widths can be obtained. However, the runtime is also increased. VTPs significantly reduce the runtime but slightly increase total width. Details on time frame partitioning can be found in [2].

NBTI is a major reliability issue in nanoscale technologies [7]. It is caused by traps at the $Si/SiO_2$ interface in pMOS transistors under a negative bias voltage ($V_{GS} = -V_{DD}$). Interface traps are generated within the $Si/SiO_2$ interface, increasing $V_{th}$ and transistor delay. The accumulation of interface traps when a pMOS transistor is under a negative biased voltage is referred to as the stress phase. When the biased voltage is removed, and the effect is reduced, this is referred to as the recovery phase. The corresponding effect on nMOS is called positive bias temperature instability (PBTI). While PBTI is normally insignificant in the SiON process, it becomes comparable to NBTI and cannot be ignored in high-k metal gate technologies [10]. Therefore, it is necessary to consider both NBTI and PBTI effects in advanced high-k metal gate technologies.

The motivation to address the BTI issue in sleep transistors is the fact that sleep transistors are always turned on and stressed when in functional mode, and thus they suffer a more significant BTI effect than do other functional gates. Since sleep transistors are on the critical path of the current flowing from the power rail to the circuit, the $V_{th}$ degradation in a sleep transistor will reduce the circuit speed as a whole and cause long-term reliability issues. In addition, both sleep transistors and cluster transistors are subject to the BTI effect, whose impact thus needs to be addressed in order to design reliable power-gated circuits. Hence, this paper proposes two sleep transistor sizing algorithms that consider the BTI effect based on DSTN structure. The contributions of this paper are summarized as follows.

1) Two sleep transistor sizing algorithms are proposed to address the BTI-aware sleep transistor sizing problem. A trade-off between runtime and total sleep transistor width can be made. The first algorithm reduces more area at the cost of a longer runtime, while the second algorithm reduces runtime but yields a slightly larger total width.
2) Experimental results show that when only the BTI effect on pMOS sleep transistors in 90 nm technology is considered, the proposed algorithms reduce total width by 17%∼31% as compared to that obtained using the method in [2]. In 32 nm technology, the proposed algorithms reduce area by 16%∼28% when nMOS transistors are used as sleep transistors. When considering the BTI effect on both sleep and cluster transistors, more area reduction can be achieved.

## II. Preliminaries

### A. BTI Model

NBTI (PBTI) occurs when a pMOS (nMOS) transistor is under a negative (positive) bias voltage. The $V_{th}$ drift of a pMOS (nMOS) transistor due to the static NBTI (PBTI) effect can be described by a direct-current (DC) reaction-diffusion (RD) framework. If a transistor is under alternating stress and recovery phases, the DC RD model

Manuscript received October 16, 2013; revised January 22, 2014 and March 31, 2014; accepted May 8, 2014. Date of current version September 16, 2014. This work was supported by the National Science Council of Taiwan under Grant NSC 100-2221-E-006-177, Grant NSC 102-2221-E-006-281, and Grant MOST 103-2221-E-006-255. This paper was recommended by Associate Editor Y. Cao.

K.-C. Wu is with the Department of Computer Science, National Chao Tung University, Hsinchu 300, Taiwan (e-mail: kcw@cs.nctu.edu.tw).

I.-C. Lin, Y.-T. Wang, and S.-S. Yang are with the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan 701, Taiwan (e-mail: iclin@mail.ncku.edu.tw; p76001182@mail.ncku.edu.tw; p76021174@mail.ncku.edu.tw).

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.2014.2334331

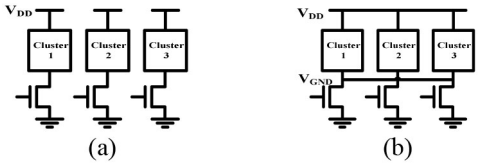


Fig. 1. Power gating structures. (a) Cluster-based. (b) DSTN designs.

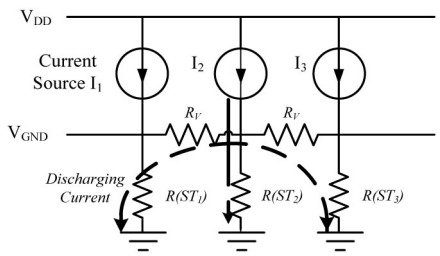


Fig. 2. Resistance network of DSTN structure.

should be modified into an alternating-current (AC) RD model [8]

$$\Delta V_{TH}(t) \cong K_{AC} \times t^n \cong \alpha(S,f) \times K_{DC} \times t^n \quad (1)$$

where $a$ is a function of stress frequency ($f$) and signal probability ($S$). Since the impact of frequency is relatively insignificant, the effect of the signal frequency is ignored. $K_{DC}$ is a technology-dependent constant. More details about this model can be found in [9].

### B. Determination of Sleep Transistor Width

To maintain circuit operation, the IR-drop should be maintained in a certain range; hence, the sleep transistor width, $W_{ST}$, can be represented as

$$W_{ST} = \left(\frac{MIC(ST)}{V_{ST}}\right) \cdot k \quad (2)$$

where $W_{ST}$ is the minimum sleep transistor width that meets the IR-drop constraint requirement, and $V_{ST}$ is the maximum IR-drop across the sleep transistor. $MIC(ST)$ is the MIC flowing through the sleep transistor, and $k$ is equal to $L/\mu C_{OX}(V_{DD}\text{-}V_{th})$. More details can be found in [2].

The DSTN structure can be represented as a resistance network, as shown in Fig. 2. Sleep transistors are replaced with resistors; thus, the resistance of sleep transistor $i$ is represented as $R(ST_i)$. Each cluster is modeled as a current source $I_i$, and the resistance of the virtual ground line is modeled as resistor $R_V$. Because the discharging current among all sleep transistors can be balanced through the DSTN structure, an accurate $MIC(ST_i)$ cannot be measured easily using (2).

Chiou *et al.* [2], [4] used Kirchhoff's current law and Ohm's law to obtain a discharging matrix in order to estimate $MIC(ST_i)$. The discharging matrix for Fig. 2 is shown in (3). $\varphi$ is a $3 \times 3$ discharging matrix constructed from the resistance network of the DSTN structure. Every entry can be replaced by $\psi_{ij}$, which is the percentage of the shared current from the current source $I_j$ to the sleep transistor $ST_i$. Every entry is positive and can be calculated from resistance values

$$\varphi = \begin{bmatrix} \psi_{11} & \psi_{12} & \psi_{13} \\ \psi_{21} & \psi_{22} & \psi_{23} \\ \psi_{31} & \psi_{32} & \psi_{33} \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} MIC(ST_1) \\ MIC(ST_2) \\ MIC(ST_3) \end{bmatrix} = \varphi \cdot \begin{bmatrix} MIC(C_1) \\ MIC(C_2) \\ MIC(C_3) \end{bmatrix}. \quad (4)$$

**Algorithm**: *Increase and Decrease Sizing Algorithm*

**Input:** $MIC(C_i, T_j), V_{DEG}$
**Output:** $R(ST_i)$ ∀i: Resistance of each sleep transistor

1. /* step 1: initialize $W_{STi}$*/
2. Find the minimum current of each cluster from $MIC(C_i, T_j)$;
3. Calculate each sleep transistor width with minimum current;
4. Update $\varphi$, $MIC(ST_i, T_j)$, $R(ST_i)$, and $Slack(ST_i, T_j)$ for all $i, j$;
5. **repeat //** step 2: increase operation
6. Find the least negative $Slack(ST_i, T_j)$ value and the corresponding sleep transistor $ST_i$;
7. Find the increase ratio and use it to increase $W_{STi}$;
8. Update $\varphi$, $MIC(ST_i, T_j)$, $R(ST_i)$, and $Slack(ST_i, T_j)$ for all $i, j$;
9. **until** $Slack(ST_i, T_j) \geq 0$ for all $i, j$
10. **repeat //** step 3: decrease operation
11. Find the largest positive $Slack(ST_i, T_j)$ value and the corresponding sleep transistor $ST_i$;
12. Find the decrease ratio and use it to decrease $W_{STi}$;
13. Update $\varphi$, $MIC(ST_i, T_j)$, $R(ST_i)$, and $Slack(ST_i, T_j)$ for all $i, j$;
14. Check whether all $Slack(ST_i, T_j)$ are greater than or equal to zero;
15. **until** no sleep transistor can be further reduced
16. **return** $R(ST_i)$ for all $i$;

Fig. 3. IDS algorithm.

The upper bound of $MIC(ST_i)$ can be calculated using a discharging matrix and $MIC(C_i)$, as shown in (4). After evaluating $MIC(ST_i)$, the required sleep transistor width can be directly calculated using (2).

## III. Proposed Sleep Transistor Sizing Algorithm

The algorithm in [2] used the worst IR-drop constraint among all the time frames under consideration to determine sleep transistor sizes. However, not all of the sleep transistors have their worst IR-drop in the same time frame, and the sizing results in [2] were usually overly pessimistic. Furthermore, the BTI effect was not considered in [2], and therefore a power-gated design could potentially fail to operate correctly after a period of time. This section proposes a modified sleep transistor sizing algorithm based on the DSTN structure to address the over-sizing problem due to the pessimistic scenario, with the BTI effect modeled for long-term reliability.

### A. Modified BTI-Aware Sleep Transistor Sizing Algorithm

The proposed algorithm, increase and decrease sizing (IDS), contains three steps, as shown in Fig. 3. $V_{\mathrm{DEG}}$ is the maximum allowable IR-drop under the BTI effect. In step 1 (lines 1–4), the minimum initial width of the sleep transistors is determined. This is done by finding the minimum current of each cluster (line 2) and using (2) to calculate the initial sleep transistor width (line 3). Then, discharging matrix $\varphi$, $MIC(ST_i, T_j)$, $R(ST_i)$, and $Slack(ST_i, T_j)$ are updated (line 4). Note that $Slack(ST_i, T_j)$ is the voltage difference between $V_{\mathrm{DEG}}$ and the voltage drop across the sleep transistor $ST_i$ in the time frame $T_j$ and is used to examine whether the $V_{\mathrm{DEG}}$ constraint is met. $MIC(ST_i, T_j)$ can be obtained using (4) by multiplying matrix $\varphi$ and $MIC(C_i, T_j)$. $R(ST_i)$ is the obtained resistance for each sleep transistor.

In step 2 (lines 5–9), the width of the sleep transistors is increased until all $Slack(ST_i, T_j)$ values are equal to or greater than zero. According to the $Slack(ST_i, T_j)$ values from step 1, the least negative $Slack(ST_i, T_j)$ value and the corresponding sleep transistor $ST_i$ are obtained (line 6). This means that the IR-drop in the case of this sleep transistor is relatively large. Thus, the width of this sleep transistor is first increased. The increase ratio is obtained from $V_{\mathrm{DEG}}$ divided by the largest IR-drop of $ST_i$ and is used to enlarge $ST_i$ (line 7). After the adjustment, the new discharging matrix $\varphi$, $MIC(ST_i, T_j)$, $R(ST_i)$, and $Slack(ST_i, T_j)$ are updated (line 8). The loop is repeated until all $Slack(ST_i, T_j)$ values are equal to or greater than zero, which implies that the $V_{\mathrm{DEG}}$ constraint is satisfied. However, the sleep transistors may be too large after the increase step. The decrease step can be used to reduce these over-sized sleep transistors.

**Algorithm**: *Dual Decrease Sizing Algorithm*

**Input:** $MIC(C_i,T_j), V_{DEG}$, $CD_i$: Current degradation of cluster $C_i$ under the BTI effect

**Output:** $R(ST_i)\forall$i: Resistance of each sleep transistor

```
1.  /* Step 1 : initialize W_STi*/
2.  Find MIC of each cluster, MIC(Ci), from MIC(Ci,Tj);
3.  Calculate W_STi with MIC(Ci)*CDi and V_DEG based on the DSTN
    algorithm;
4.  Update φ, MIC(STi,Tj), R(STi), and IR_Drop(STi,Tj) for all i, j;
5.    /*Step 2 : sizing all sleep transistors simultaneously*/
6.    while (all IR_Drop(STi,Tj) < V_DGE)
7.    Find max IR-drop from IR_Drop(STi,Tj) for all i, j;
8.      Calculate the decrease ratio with V_DGE;
9.      Reduced all ST with the decrease ratio;
10.     Update φ, MIC(STi,Tj), R(STi), and IR_Drop(STi,Tj) for all i, j;
11.     Check whether all IR-drops are smaller than V_DGE;
12.   until any IR_Drop(STi,Tj) > V_DGE
13.   /*Step 3 : sizing one sleep transistor at a time*/
14.   Find the sizing order for sleep transistors;
15.   for i=1 to NUM_TF do
16.     Calculate the decrease ratio with V_DGE;
17.     Reduced sleep transistor STi with the decrease ratio;
18.     Update φ, MIC(STi,Tj), R(STi), and IR_Drop(STi,Tj) for all i, j;
19.     Check whether all IR_Drop are smaller than V_DGE;
20.   end for
21.   go to line 15;
22.   until no STi can be further reduced
23.   return R(STi) for all i;
```

Fig. 4. DDS algorithm.

In step 3 (lines 10–16), the sleep transistors are reduced one at a time. The sleep transistor with the largest positive slack is adjusted first (line 11) since more reduction may be obtained. Then, this sleep transistor is reduced by the decrease ratio, which is defined as one minus the ratio of the worst IR-drop of $ST_i$ across all time frames to $V_{\mathrm{DEG}}$. This ratio is used because the difference between $V_{\mathrm{DEG}}$ and the worst IR-drop of $ST_i$ across all time frames is the smallest. Hence, reducing the sleep transistor width using this ratio results in the lowest chance of violating the $V_{\mathrm{DEG}}$ constraint. After this decrease ratio is used to adjust $ST_i$, the new discharging matrix $\varphi$, $MIC(ST_i, T_j)$, $R(ST_i)$, and $Slack(ST_i, T_j)$ are updated (line 13). Then, it is determined whether all $Slack(ST_i, T_j)$ values are greater than or equal to zero (line 14). If all slacks are greater than or equal to zero, the current total width is better than the previous solution; otherwise, the previous solution is considered to be better and is therefore restored. This loop is repeated until no sleep transistor width can be further reduced, and all $Slack(ST_i, T_j)$ values are equal to or greater than zero.

### B. Enhanced BTI-Aware Sleep Transistor Sizing Algorithm

Although IDS uses the decrease step to obtain a smaller total sleep transistor width than that obtained in [3], both algorithms have a long runtime because they adjust one sleep transistor at a time. If the number of sleep transistors is large, the algorithms requires more iterations to recalculate the width of each sleep transistor. In addition, as mentioned in Section I, cluster transistors are also subject to the BTI effect, thus reducing the current. Hence, sleep transistor sizes can be further reduced if this current degradation is considered. Based on the DSTN structure, we propose an enhanced algorithm that adjusts all sleep transistors at the same time, while considering current degradation on cluster transistors.

The enhanced algorithm, dual decrease sizing (DDS), improves the total runtime while obtaining a comparable total width. It consists of three steps. Initially, the maximum initial sleep transistor width is chosen. Then, all sleep transistors are reduced simultaneously. Finally, sleep transistors are reduced one at a time for further improvement. Fig. 4 shows the enhanced algorithm in detail. $MIC(C_i, T_j)$ and $V_{\mathrm{DEG}}$, defined above, are inputs of the algorithm. $CD_i$ is the ratio of the current degradation of cluster $C_i$ under the BTI effect and is also an input of the sizing algorithm. The output is $R(ST_i)$, which contains the obtained resistance for each sleep transistor.

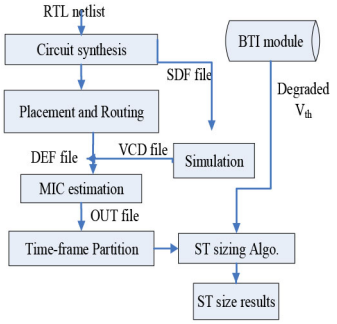


Fig. 5. Implementation flowchart.

In step 1 (lines 1–4), the MIC of each cluster $MIC(C_i)$ is found according to the values of $MIC(C_i, T_j)$ (line 2). Then, these currents are used to calculate the maximum initial sleep transistor width that meets the $V_{\mathrm{DEG}}$ constraint based on the DSTN algorithm [6] (line 3). Note that if the current degradation of each cluster is considered, $MIC(C_i)$ is multiplied by $CD_i$. If not, $CD_i$ is equal to 1. After initialization, $\varphi$, $MIC(ST_i, T_j)$, $R(ST_i)$, and $IR\_Drop(ST_i, T_j)$ are updated (line 4). The value for $MIC(ST_i, T_j)$ is the same as that in the IDS algorithm and can be obtained by multiplying $\varphi$ and $MIC(C_i, T_j)$. $R(ST_i)$ is the obtained resistance for each sleep transistor. $IR\_Drop(ST_i, T_j)$ indicates the IR-drop across the sleep transistor $ST_i$ in the time frame $T_j$.

The goal of step 2 (lines 5–12) is to find a decrease ratio to reduce all sleep transistors simultaneously. The maximum IR-drop is obtained according to all $IR\_Drop(ST_i, T_j)$ and then divided by $V_{\mathrm{DEG}}$ (lines 7 and 8) to get the decrease ratio, which is then used to reduce all sleep transistors at the same time (line 9). After that, $\varphi$, $MIC(ST_i, T_j)$, $R(ST_i)$, and $IR\_Drop(ST_i, T_j)$ are updated (line 10), and it is determined whether any new $IR\_Drop(ST_i, T_j)$ value is greater than $V_{\mathrm{DEG}}$. If all $IR\_Drop(ST_i, T_j)$ values are smaller than $V_{\mathrm{DEG}}$, this step is repeated until an $IR\_Drop(ST_i, T_j)$ value is obtained that is greater than $V_{\mathrm{DEG}}$. If any $IR\_Drop(ST_i, T_j)$ is greater than $V_{\mathrm{DEG}}$, which means that the decrease ratio is too large to maintain the $V_{\mathrm{DEG}}$ requirement. As a result, the previous sleep transistor widths are output in step 2.

After step 2, all sleep transistors cannot be adjusted simultaneously. To further reduce the total width of the sleep transistors, sleep transistors can be reduced one at a time in step 3 (lines 13–22). The main idea of step 3 is to find an appropriate sleep transistor sizing in order to further reduce individual sleep transistors (line 14). According to the $IR\_Drop(ST_i, T_j)$ values from the previous step, the worst IR-drop of each sleep transistor is found. Since the sleep transistor with the smallest worst IR-drop requires the most size reduction, the IR-drops are sorted from small to large for the purpose of sizing the sleep transistors. After the sizing order is obtained, single sleep transistors are reduced one at a time (lines 15–22). The decrease ratio used to shrink an individual sleep transistor $ST_i$ is obtained by taking the worst IR-drop of $ST_i$ and dividing it by $V_{\mathrm{DEG}}$ (line 16). After the sleep transistor $ST_i$ is adjusted (line 17), $\varphi$, $MIC(ST_i, T_j)$, $R(ST_i)$, and $IR\_Drop(ST_i, T_j)$ are updated (line 18), and it is determined whether all $IR\_Drop(ST_i, T_j)$ values are smaller than $V_{\mathrm{DEG}}$ (line 19). If all values meet the constraint, the total width is considered to be the best current width. Otherwise, the previous total width is determined to be the best width. Step 3 is repeated until no sleep transistors can be further reduced with the decrease ratio. The last sleep transistor widths are the final output of the proposed algorithm.

TABLE I
BENCHMARK CIRCUIT INFORMATION

| CKT | # of gates | # of clusters |
|---|---|---|
| C432 | 160 | 9 |
| C499 | 202 | 11 |
| C880 | 383 | 11 |
| C2670 | 1269 | 15 |
| C5315 | 2307 | 22 |
| C6288 | 2416 | 52 |
| C7552 | 3513 | 25 |
| S15850 | 3448 | 42 |
| S35932 | 12204 | 111 |
| S38417 | 8709 | 117 |

TABLE II
TOTAL PMOS SLEEP TRANSISTOR WIDTH AND RUNTIME COMPARISONS COMPARED TO UTP IN 90 NM TECHNOLOGY

| CKT | Total width (μm) | | | | | Runtime (s) | | |
|---|---|---|---|---|---|---|---|---|
| | UTP[2] | IDS | %Impr. | DDS | %Impr. | UTP[2] | IDS | DDS |
| C432 | 793.48 | 612.18 | 22.85% | 687.07 | 13.41% | 0.02 | 1.3 | 0.01 |
| C499 | 634.05 | 515.38 | 18.72% | 629.69 | 0.69% | 0.08 | 2.3 | 0.01 |
| C880 | 732.75 | 584.15 | 20.28% | 696.74 | 4.91% | 0.08 | 1.9 | 0.07 |
| C2670 | 1231.84 | 767.29 | 37.71% | 815.46 | 33.8% | 0.15 | 7.42 | 0.34 |
| C5315 | 1607.05 | 1073.88 | 49.97% | 1388.36 | 13.61% | 1.02 | 18.46 | 0.22 |
| C6288 | 2492.39 | 1650.29 | 33.79% | 1830.96 | 26.54% | 58.73 | 277.65 | 3.14 |
| C7552 | 2820.61 | 2076.52 | 26.38% | 2552.55 | 9.5% | 1.27 | 31.33 | 0.04 |
| S15850 | 20592.1 | 14070.2 | 31.67% | 16529.9 | 19.73% | 11.61 | 155.9 | 5.24 |
| S35932 | 48597.6 | 27481 | 43.45% | 30111.9 | 38.03% | 517.75 | 1179 | 134.67 |
| S38417 | 46907.5 | 34091.2 | 27.32% | 36892.3 | 21.35% | 733.03 | 1343.4 | 315.69 |
| S38584 | 61188.2 | 39711.1 | 35.1% | 42387.5 | 30.73% | 1544.75 | 2450.4 | 580.53 |
| Avg. | 17054.32 | 11148.47 | 31.57% | 12229.31 | 19.3% | 260.77 | 497.18 | 94.542 |
| Normal | | | | | | 1 | 1.91X | 0.36X |

TABLE III
TOTAL PMOS SLEEP TRANSISTOR WIDTH AND RUNTIME COMPARISONS COMPARED WITH VTP IN 90 NM TECHNOLOGY

| CKT | Total width (μm) | | | | | Runtime (s) | | |
|---|---|---|---|---|---|---|---|---|
| | VTP[2] | IDS | %Impr. | DDS | %Impr. | VTP[2] | IDS | DDS |
| C432 | 929.4 | 842.32 | 9.37% | 875.47 | 5.8% | 0.01 | 0.22 | 0 |
| C499 | 869.61 | 709.09 | 18.46% | 865.25 | 0.5% | 0.02 | 1.27 | 0.01 |
| C880 | 894.2 | 762.9 | 14.68% | 861.97 | 3.6% | 0.01 | 1.31 | 0.04 |
| C2670 | 1402.39 | 855.38 | 39.01% | 923.63 | 34.14% | 0.05 | 3.39 | 0.15 |
| C5315 | 2117.62 | 1473.49 | 30.42% | 1631.29 | 22.97% | 0.36 | 9.25 | 0.03 |
| C6288 | 3449.91 | 2230.5 | 35.35% | 3073.33 | 10.92% | 11.44 | 144.7 | 0.27 |
| C7552 | 3056.73 | 2255.2 | 26.22% | 2527.5 | 17.31% | 0.42 | 16.36 | 0.23 |
| S15850 | 23309.6 | 15660.4 | 32.82% | 18270.4 | 21.62% | 4.55 | 86.61 | 0.93 |
| S35932 | 51847.3 | 37082.8 | 28.48% | 40706.7 | 21.49% | 170.48 | 639 | 98.15 |
| S38417 | 50989.8 | 37647.5 | 26.17% | 39585 | 22.37% | 240.49 | 753 | 40.28 |
| S38584 | 65304.6 | 41258.5 | 36.82% | 43456.3 | 33.46% | 498.39 | 1363.2 | 78.59 |
| Avg. | 18561.01 | 12798.01 | 27.07% | 13888.8 | 17.65% | 84.2 | 274.392 | 19.88 |
| Normal | | | | | | 1 | 3.26X | 0.24X |

## IV. EXPERIMENT SETUP AND RESULTS

The algorithms are implemented in C/C+, and the BTI model from Section II is used to obtain the $\Delta V_{th}$ of the sleep transistors. The benchmarks are from ISCAS 85 and 89, as shown in Table I. Fig. 5 shows the simulation framework. RTL netlists are synthesized to gate-level netlists, and the SDF file is generated using design compiler. The netlists are then simulated to obtain the VCD file with 10 000 random patterns, and placement and routing are done to obtain the location of each gate and the virtual ground resistance using SoC Encounter. Based on the gate location, the gates in a given row are grouped as a cluster. The MIC of each cluster is estimated using PrimeTime and the VCD file, and the output file containing time frames and current information is generated. Then, the time frames are partitioned based on the information contained in the output file and are designated as the inputs for the experiments.

TABLE IV
TOTAL NMOS SLEEP TRANSISTOR WIDTH AND RUNTIME COMPARISONS COMPARED WITH UTP IN 32 NM TECHNOLOGY

| CKT | Total sleep transistor width (μm) | | | | | Runtime (s) | | |
|---|---|---|---|---|---|---|---|---|
| | UTP[2] | IDS | %Impr. | DDS | %Impr. | UTP[2] | IDS | DDS |
| C432 | 498.13 | 441.92 | 11.28% | 473.88 | 4.87% | 0.02 | 1.73 | 0.06 |
| C499 | 411.4 | 333.26 | 18.99% | 393.89 | 42.56% | 0.08 | 2.56 | 0.01 |
| C880 | 465.02 | 353.17 | 24.05% | 444.77 | 43.55% | 0.09 | 2.96 | 0.05 |
| C2670 | 774.29 | 438.51 | 43.37% | 543.71 | 29.78% | 0.23 | 7.53 | 0.19 |
| C5315 | 1056.92 | 657.45 | 37.96% | 755.33 | 28.54% | 1.37 | 18.7 | 0.26 |
| C6288 | 1529.72 | 935.24 | 38.86% | 1164.91 | 23.85% | 39.55 | 301.49 | 2.72 |
| C7552 | 1753.51 | 1185.65 | 32.38% | 1494.51 | 14.77% | 1.24 | 31.96 | 1.26 |
| S15850 | 13192.2 | 9489.57 | 28.07% | 10764.6 | 18.4% | 9.9 | 157.43 | 7.97 |
| S35932 | 31393.8 | 21571.4 | 31.29% | 26365.3 | 16.02% | 443.85 | 1171.5 | 136.22 |
| S38417 | 31927.4 | 21614.9 | 32.3% | 24302.3 | 23.88% | 665.05 | 1414.8 | 347.98 |
| S38584 | 38957.3 | 31022.7 | 20.37% | 35791 | 8.13% | 1333.75 | 2478 | 648.71 |
| Avg. | 11087.24 | 8003.98 | 28.98% | 9317.65 | 16.08% | 226.83 | 508.06 | 104.13 |
| Normal | | | | | | 1 | 2.24X | 0.46X |

TABLE V
TOTAL NMOS SLEEP TRANSISTOR WIDTH AND RUNTIME COMPARISONS COMPARED WITH VTP IN 32 NM TECHNOLOGY

| CKT | Total sleep transistor width (μm) | | | | | Runtime (s) | | |
|---|---|---|---|---|---|---|---|---|
| | VTP[2] | IDS | %Impr. | DDS | %Impr. | VTP[2] | IDS | DDS |
| C432 | 572.75 | 534.13 | 6.74% | 548.13 | 4.3% | 0.01 | 0.72 | 0.01 |
| C499 | 549.16 | 442.88 | 19.35% | 506.83 | 7.7% | 0.02 | 1.28 | 0.01 |
| C880 | 554.99 | 512.98 | 7.57% | 540.1 | 2.68% | 0.03 | 1.2 | 0.01 |
| C2670 | 872.85 | 527.01 | 39.62% | 581.27 | 33.41% | 0.05 | 3.42 | 0.03 |
| C5315 | 1361.95 | 763.61 | 43.93% | 822.51 | 39.61% | 0.22 | 9.33 | 0.03 |
| C6288 | 2060 | 1346.31 | 34.65% | 1758.43 | 14.64% | 10.42 | 145.04 | 0.21 |
| C7552 | 1884.21 | 1283.12 | 31.9% | 1643.53 | 12.77% | 0.42 | 16.6 | 0.03 |
| S15850 | 14418.2 | 10052.7 | 30.28% | 11843.2 | 17.56% | 3.46 | 86.88 | 0.78 |
| S35932 | 32621.1 | 23464.3 | 28.07% | 27422.3 | 15.94% | 162.35 | 664.5 | 14.9 |
| S38417 | 39689 | 22299.1 | 43.82% | 25878.9 | 34.8% | 205.64 | 759.3 | 32.38 |
| S38584 | 41071.5 | 32211.1 | 21.57% | 36577.9 | 10.94% | 441.46 | 1389.3 | 63.55 |
| Avg. | 12332.34 | 8494.29 | 27.95% | 9829.372 | 17.7% | 74.916 | 279.779 | 10.176 |
| Normal | | | | | | 1 | 3.74X | 0.14X |

Note that $V_{\mathrm{DEG}}$ is set to 10% of the supply voltage. The probability of sleep transistors being turned on is set to 0.5, and $\Delta V_{th}$ is estimated for 10 years. The number of variable-length time frames is equal to the cluster numbers based on [2], and the number of uniform time frames is set to ten times the cluster number. TSMC 90 nm and the PTM 32 nm high-k metal-gate technology model [11] are used. TSMC 90 nm technology considers only NBTI because PBTI is relatively insignificant, and 32 nm technology considers both NBTI and PBTI to demonstrate that the proposed sizing algorithms can be applied for both header-based and footer-based designs. The current inputs and cluster numbers of the 90 nm experiment are used in the 32 nm experiment. Both UTP and VTP [2] algorithms are compared with the IDS and DDS algorithms. Since DSTN, UTP, and VTP algorithms do not consider the BTI effect, in order to compare these works with the proposed algorithms, the maximum allowable IR-drop with the BTI effect, the $V_{\mathrm{DEG}}$ constraint, is used to replace the worst IR-drop constraint.

### A. Width and Runtime Comparisons When Considering the BTI Effect on Sleep Transistors

Table II compares the UTP, IDS, and DDS algorithms with uniform time frames in 90 nm technology when only the NBTI on pMOS sleep transistors is considered. Uniform time frames are used in UTP, IDS, and DDS algorithms. It can be seen that DDS and IDS reduce transistor width by 31.57% and 19.3%, on average, respectively. The runtimes of the IDS and DDS algorithms are 1.91X and 0.36X of the runtime required by UTP. This is because the UTP and IDS adjust one sleep transistor at a time, and the DDS algorithm adjusts all transistors simultaneously. Similarly, Table III compares

TABLE VI
COMPARISONS OF TRANSISTOR WIDTHS CONSIDERING THE BTI EFFECT ON PMOS SLEEP AND CLUSTER TRANSISTORS IN 90 NM TECHNOLOGY

| CKT | Uniform time frames | | | | | | | Variable-length time frames | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | UTP [2] | DDS | | | | | | VTP [2] | DDS | | | | | |
| | | (0%) | (10%) | (15%) | (20%) | (25%) | (30%) | | (0%) | (10%) | (15%) | (20%) | (25%) | (30%) |
| C432 | 793.48 | 687.07 | 635.54 | 629.63 | 613.75 | 575.78 | 535.92 | 929.4 | 875.49 | 801.47 | 753.75 | 708.92 | 670.47 | 623.64 |
| C499 | 634.05 | 629.69 | 557.89 | 523.02 | 485.73 | 454.07 | 417.58 | 869.61 | 865.25 | 771.5 | 733.01 | 703.02 | 699.17 | 691.61 |
| C880 | 732.75 | 696.74 | 649.09 | 606.62 | 557.64 | 525.17 | 482.92 | 894.2 | 861.97 | 793.19 | 746.14 | 689.63 | 651.02 | 604.6 |
| C2670 | 1231.84 | 815.46 | 731.29 | 730.46 | 635.8 | 600.14 | 557.58 | 1402.39 | 923.63 | 858.95 | 789.51 | 747.09 | 700.4 | 644.16 |
| C5315 | 1607.05 | 1388.36 | 1282.66 | 1127.84 | 1011.85 | 1037.29 | 928.55 | 2117.62 | 1631.29 | 1450.59 | 1384.23 | 1270.06 | 1190.79 | 1115.02 |
| C6288 | 2492.39 | 1830.96 | 1609.66 | 1526.21 | 1410.5 | 1323.59 | 1223.87 | 3449.91 | 3073.33 | 2292.31 | 2168.39 | 2021.4 | 1894.23 | 1848.81 |
| C7552 | 2820.61 | 2685.48 | 2515.79 | 2286.5 | 2237.17 | 2195.14 | 2184.07 | 3056.73 | 2826.14 | 22680.9 | 2410.96 | 2334 | 2315.95 | 2015.3 |
| S15850 | 20592.1 | 16529.9 | 14600.5 | 13496.5 | 12596.6 | 11711.5 | 10756.3 | 23309.6 | 18270.4 | 11679.8 | 15662 | 14672.7 | 13727 | 12720.5 |
| S35932 | 48597.6 | 30111.9 | 27031.1 | 25893.9 | 24373.8 | 22660.8 | 20711.9 | 51847.3 | 40706.7 | 36704.7 | 34678.3 | 32561.9 | 30576.2 | 28346.6 |
| S38417 | 46907.5 | 36892.3 | 32972.4 | 31153.2 | 29248.1 | 27275.9 | 25317.4 | 50989.8 | 39585 | 35350.8 | 33437.7 | 31276.4 | 29272.3 | 27223.4 |
| S38584 | 61188.2 | 42387.5 | 35021.5 | 33095.1 | 31124 | 29275.6 | 24906.3 | 65304.6 | 43456.3 | 38850.8 | 36320.3 | 32974.4 | 30389.1 | 28831.4 |
| Avg. | 17054.32 | 134655.36 | 10691.56 | 10097.18 | 9481.36 | 8875.91 | 8007.49 | 18561.01 | 13915.95 | 12475.91 | 11734.94 | 10905.41 | 10189.69 | 9515 |
| Normal | 1 | 0.72X | 0.63X | 0.59X | 0.56X | 0.52X | 0.47X | 1 | 0.75X | 0.67X | 0.63X | 0.59X | 0.55X | 0.51X |

the VTP, IDS, and DDS algorithms with variable-length time frames in 90 nm technology when only the NBTI on pMOS sleep transistors is considered. The proposed algorithms reduce sleep transistor width by 27.07% and 17.65%, on average, respectively. The runtimes of IDS and DDS are 3.26X and 0.24X of the runtime required by VTP, respectively. Compared to UTP and VTP, the proposed algorithms reduce the sleep transistor width more in 90 nm technology header-based designs.

Since PBTI becomes significant in 32 nm process technology, experiments are done for footer-based designs with the PTM 32 nm technology model. Table IV compares UTP, IDS, and DDS in uniform time frames. Compared to UTP, DDS, and IDS reduce sleep transistor width by 28.98% and 16.08%, on average, respectively. The runtimes of the IDS and DDS algorithms are 2.24X and 0.46X of the runtime required by UTP, respectively. Table V compares the VTP, IDS, and DDS algorithms in a variable-length time frame. Compared to VTP, IDS, and DDS reduce sleep transistor width by 27.95% and 17.7%, on average, respectively. The runtimes of IDS and DDS are 3.74X and 0.14X of the runtime required by VTP. It can be seen that the proposed algorithms reduce sleep transistor width more in both header-based and footer-based designs.

### *B. Width Comparisons When Considering the BTI Effect on Both Sleep Transistors and Cluster Transistors*

Since the BTI effect occurs on sleep and cluster transistors, its influence on both needs to be considered. Table VI compares the UTP, VTP, and DDS algorithms in 90 nm when the BTI effect occurs on both pMOS sleep and cluster transistors. Five current degradation ratios are used in the cluster transistors: 10%, 15%, 20%, 25%, and 30%. The circuits are assumed to remain reliable when these current degradation ratios occur on the cluster transistors. It can be seen that when current degradation on the cluster transistors is not considered in uniform time frames, the sleep transistor width obtained by DDS is 72% of that obtained by UTP. However, when 10%, 15%, 20%, 25%, and 30% of current degradation on the cluster transistors are considered, the sleep transistor width is 63%, 59%, 56%, 52%, and 47% of that obtained by UTP. Similar results are obtained when variable time frames are used. As the current degradation ratio increases, DDS can reduce more sleep transistor width. This is because the current flowing through sleep transistors decreases due to the current degradation of the cluster transistors, and thus, sleep transistors need less width to satisfy the IR-drop constraints.

## V. CONCLUSION

This paper proposed two sleep transistor sizing algorithms to reduce the total sleep transistor width under the BTI effect. A trade-off between runtime and sizing results can be made by choosing the proper algorithm. The total sleep transistor width obtained using the proposed algorithms when only the BTI effect on sleep transistors is considered was reduced by 25%∼35% in 90 nm technology and by 13%∼31% in the PTM 32 nm technology model, on average. Therefore, when the BTI effect on both sleep and cluster transistors are considered, greater sleep transistor width reduction can be achieved.

## ACKNOWLEDGMENT

The authors would like to thank Y.-C. Phong and Prof. L.-Y. Chiou of Electronic Engineering, National Cheng Kung University for their help on power gating simulation.

## REFERENCES

[1] M. Anis, S. Areibi, and M. Elmasry, "Design and optimization of multithreshold CMOS (MTCMOS) circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 22, no. 10, pp. 1324–1342, Oct. 2003.

[2] D. S. Chiou, D. C. Juan, Y. T. Chen, and S. C. Chang, "Fine-grained sleep transistor sizing algorithm for leakage power minimization," in *Proc. 44th ACM/IEEE Design Autom. Conf. (DAC)*, San Diego, CA, USA, 2007, pp. 81–86.

[3] D. S. Chiou, Y. T. Chen, D. C. Juan, and S. C. Chang, "Sleep transistor sizing in power gating designs," in *Proc. 7th Int. Conf. ASICON*, Guilin, China, 2007, pp. 1326–1331.

[4] D. S. Chiou, Y. T. Chen, D. C. Juan, and S. C. Chang, "Sleep transistor sizing for leakage power minimization considering temporal correlation," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 29, no. 8, pp. 1285–1289, Aug. 2010.

[5] J. Kao, S. Narendra, and A. Chandrakasan, "MTCMOS hierarchical sizing based on mutual exclusive discharge patterns," in *Proc. Design Autom. Conf. (DAC)*, San Francisco, CA, USA, 1988, pp. 495–500.

[6] C. Long and L. He, "Distributed sleep transistor network for power reduction," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst*, vol. 12, no. 9, pp. 937–946, Sep. 2004.

[7] N. Kimizuka *et al.*, "The impact of bias temperature instability for direct-tunneling ultrathin gate oxide on MOSFET scaling," in *Proc. Symp. VLSI Technol. Dig. Tech. Papers*, Kyoto, Japan, 1999, pp. 73–74.

[8] B. C. Paul, K. Kang, H. Kufluoglu, M. A. Alam, and K. Roy, "Negative bias temperature instability: Estimation and design for improved reliability of nanoscale circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 4, pp. 743–751, Apr. 2007.

[9] R. Vattikonda, W. Wang, and Y. Cao, "Modeling and minimization of pMOS NBTI effect for robust nanometer design," in *Proc. 43rd ACM/IEEE Design Autom. Conf. (DAC)*, San Francisco, CA, USA, 2006, pp. 1047–1052.

[10] S. Zafar *et al.*, "A comparative study of NBTI and PBTI (charge trapping) in $SiO_2$/$HfO_2$ stacks with FUSI, TiN, Re gates," in *Proc. VLSI Technol. Dig. Tech. Paper*, Honolulu, HI, USA, 2006, pp. 23–25.

[11] (2013, Aug.). *Predictive Technology Model (PTM) and NBTI Model* [Online]. Available: http://www.eas.asu.edu/~ptm