An Efficient NBTI-Aware Wake-Up Strategy for Power-Gated Designs

[1]Kun-Wei Chiu, [2]Yu-Guang Chen, and [1]Ing-Chao Lin
[1]Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan, 701
[2]Department of Computer Science and Engineering, Yuan Ze University, Taoyuan, Taiwan, 320
[1]d9372199@hotmail.com, [2]andyygchen@saturn.yzu.edu.tw, [3]iclin@mail.ncku.edu.tw

***Abstract*—The wake-up process of a power-gated design may induce an excessive surge current and threaten the signal integrity. A proper wake-up sequence should be carefully designed to avoid surge current violations. On the other hand, PMOS sleep transistors may suffer from the negative-bias temperature instability (NBTI) effect which results in decreased driving current. Conventional wake-up sequence decision approaches do not consider the NBTI effect, which may result in a longer or unacceptable wake-up time after circuit aging. Therefore, in this paper, we propose a novel NBTI-aware wake-up strategy to reduce the average wake-up time within a circuit lifetime. Our strategy first finds a set of proper wake-up sequences for different aging scenarios (i.e. after a certain period of aging), and then dynamically reconfigures the wake-up sequences at runtime. The experimental results show that compared to a traditional fixed wake-up sequence approach, our strategy can reduce average wake-up time by as much as 45.04% with only 3.7% extra area overhead for the reconfiguration structure.**

Keywords—Power gating, Wake-up sequence, NBTI

I. INTRODUCTION

In modern IC design, power gating reduces leakage power by isolating a block of unused circuits from the power supply. Fig. 1 shows a header-based coarse-grained power gating structure called a distributed sleep transistor network (DSTN) [1]. High-V_t PMOS transistors that work as switches (called *sleep transistors, STs*) are inserted between the power supply and the virtual VDD. Low-V_t transistors are used in the power-gated circuit blocks for high performance. The STs are turned off when the power-gated blocks are idle (called the *sleep mode*) and are turned on again when the blocks need to resume their normal operation (called the *active mode*). When the circuit is restored from the sleep mode to the active mode, the time between the first ST is turned on and the virtual VDD is charged to the stable supply voltage is defined as the *wake-up time* The traditional strategies to turn on STs such as proposed in [2][3][4] are intended to find a proper wake-up sequence that can minimize the wake-up time while keeping the surge current under a given constraint. A wake-up sequence (also called *wake-up scheduling*) defines the order to efficiently turn on STs. STs are divided into multiple groups, and the groups are turned on one by one based on the wake-up sequence.

On the other hand, the negative-bias temperature instability (NBTI) effect is a physical nature of the oxide wear out mechanism in PMOS devices. When a negative bias voltage is applied to PMOS gate oxide, it will induce dangling bonds (traps) at the Si-SiO_2 interface and increase the transistor threshold voltage [5], which is called the *stress phase*. The increased threshold voltage will lead to reduced current and longer propagation logic gate delays. When the negative bias voltage is removed, the traps may be disencumbered, and the threshold voltage will be partially recovered, which is called the *recovery phase*. However, the recovery phase cannot remove all the traps generated during the stress phase, which results in increases in the threshold voltage.

This work is support, in part, by the Ministry of Science and Technology of Taiwan, R.O.C., under Grant MOST 105-2221-E-006-241 and Grant 106-2221-E-006-027-MY3

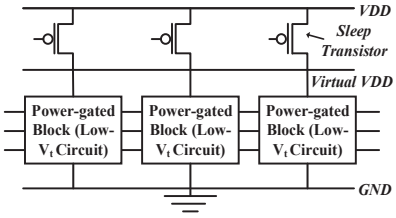

Fig. 1. A header-based power gating distributed sleep transistor network (DSTN) structure

In power-gated designs, compared to functional logics in the power-gated blocks, STs are prone to be seriously impacted by the NBTI effect [6][7]. However, the wake-up strategies proposed in [2][3][4] do not take the NBTI effect into consideration and always use the same wake-up sequence for the entire circuit lifetime, which will increase the wake-up time once the driving current is reduced after circuit aging. On the other hand, under the same given surge current constraint, the decreased driving current through PMOS STs also provides margins to turn on more STs simultaneously after circuit aging. If more STs can be turned on simultaneously after circuit aging, the wake-up time then can be reduced. Therefore, we propose a novel wake-up strategy that considers the NBTI effect to reduce the average wake-up time within a certain lifetime (called the *observation window*) while keeping the surge current under a given constraint.

The contributions of this work can be summarized as follows:

- We propose a reconfigurable structure that can reconfigure STs into different groups at run-time to mitigate the wake-up time increasing due to NBTI.
- We propose an NBTI-aware wake-up strategy which reconfigures the wake-up sequence at run-time by exploiting the NBTI effect to minimize average wake-up time while maintaining the surge current constraint after circuit aging.
- We develop an NBTI-aware wake-up sequence decision (NAWSD) algorithm which can determine an optimal

wake-up sequence set under a given reconfiguration constraint to minimize the average wake-up time.

- The experimental results show that compared to a fixed wake-up sequence approach, our strategy can achieve up to a 45.04% average wake-up time reduction with only 3.7% extra area overhead for the reconfiguration structure.

II. Analysis of The NBTI Effect on The Wake-Up Sequence

To simulate the NBTI effect on a power gated design, we apply an NBTI model which is used in [8]. Then we use C432 with eight STs (ST1 to ST8) as an example to demonstrate the influence of NBTI. In Fig. 2, the solid line shows the wake-up time at different aging scenarios with Seq1 = {(ST1, ST2, ST3), (ST4, ST5, ST6, ST7, ST8)}. The Seq1 is obtained before circuit aging (at year 0). The dotted line shows the wake-up time of another wake-up sequence Seq2 = {(ST1, ST2, ST3, ST4, ST5, ST6, ST7), (ST8)}, which is obtained at year 2. In both the sequences, each ST in the same group is turned on simultaneously, and each group is turned on one by one. For example, in Seq1, group (ST1, ST2, ST3) is turned on first, then group (ST4, ST5, ST6, ST7, ST8) is turned on next. It can be seen that although the wake-up time of both the wake-up sequences increase with the circuit aging, the wake-up time of Seq2 is always lower than the wake-up time of Seq1. This is because compared to Seq1, Seq2 takes the advantage of the NBTI effect to turn on more STs simultaneously in the first group. Note that the key to make Seq2 practical is the sufficient surge current margins (i.e., surge current constraint – induced surge current) after aging scenario at year 2. However, before the aging scenario at year 2, Seq2 cannot be used since the excessive surge current will lead to a surge current constraint violation.

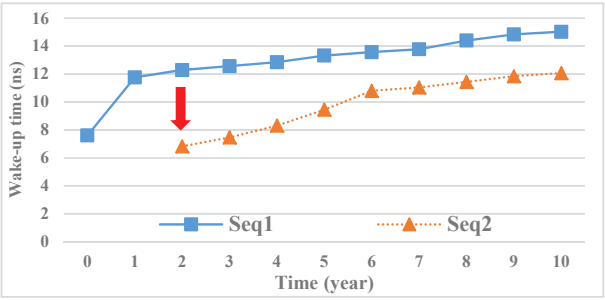

Fig. 2. Wake-up time vs. different aging scenarios of Seq1and Seq2 which are wake-up sequences used in C432. Seq1 = {(ST1, ST2, ST3), (ST4, ST5, ST6, ST7, ST8)} and Seq2 = {(ST1, ST2, ST3, ST4, ST5, ST6, ST7), (ST8)}

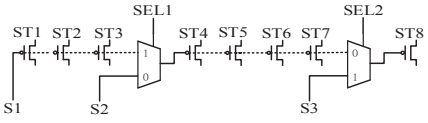

Fig. 3. An example of our proposed reconfigurable wake-up sequence structure. ST_i represents a sleep transistor i, $SELi$ represents the select signal of a $MUXi$, and Si represents the wake-up signal for group i

If we apply Seq1 to turn on STs before year 2, then we can avoid the surge current constraint violation. At year 2, we can reconfigure Seq1 to Seq2, and the wake-up time can be significantly reduced (by around 45%). To realize this idea, we propose a reconfigurable structure similar to [9] as shown in Fig. 3. In the figure, two multiplexers (MUXs) are inserted between ST3 - ST4 and ST7 - ST8, respectively. In the beginning, as Seq1 is used, SEL1 and SEL2 are both set to 0, and the STs are grouped into two groups, (ST1, ST2, ST3) and (ST4, ST5, ST6, ST7, ST8). At year 2, as signals SEL1 and SEL2 are both set to 1, the eight transistors are reconfigured into two groups (ST1, ST2, ST3, ST4, ST5, ST6, ST7), (ST8).

III. NBTI-Aware Wakeup Strategy and Sequence Decision Algorithm

This section first gives an overview about our NBTI-aware wake-up strategy, and then formally formulates the NBTI-aware wake-up time minimization problem. After that, the NBTI-aware wake-up sequence decision (NAWSD) algorithm is proposed.

A. NBTI-aware Wake-up Strategy

We propose an NBTI-aware wake-up strategy to minimize the average wake-up time in the observation window. The average wake-up time is defined as the total wake-up time at all observation points divided by the number of observation points in the given observation window. Fig. 4 shows the overall flow of the proposed strategy. The strategy is a two-phase strategy that includes an off-line decision phase and an on-line reconfiguration phase.

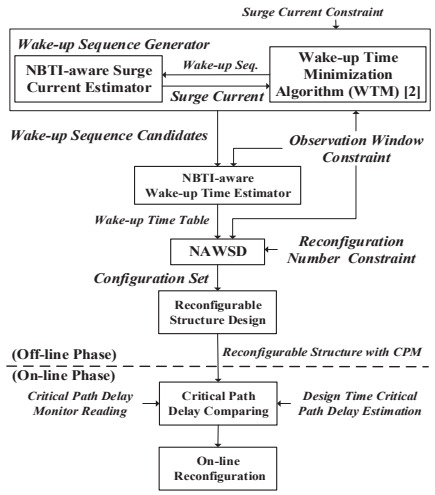

Fig. 4. Overall flow of the proposed strategy

The off-line decision phase contains two given constraints: the surge current constraint and the observation window constraint. The observation window constraint is the circuit lifetime we expected. The wake-up sequence generator generates wake-up sequence candidates for the different aging scenarios. The wake-up sequence generator iteratively executes the wake-up time minimization algorithm (WTM) [2] and NBTI-aware surge current estimator. The WTM can generate wake-up sequences, and the surge current of the generated sequences can be estimated by the NBTI-aware surge current estimator. At this step, only the wake-up sequence that has the lowest wake-up time in a specific aging scenario can be added to the wake-up sequence candidates.

After the wake-up sequences candidates are obtained, the NBTI-aware wake-up time estimator generates a wake-up time table that contains the wake-up times of each wake-up sequence candidate at all the different aging scenarios. The wake-up time of a wake-up sequence at the different aging scenarios will increase along with circuit aging, as described in Fig. 2. To consider the area overhead of MUXs, the NAWSD algorithm (which will be detailed in Section III.C) uses the wake-up time table to obtain a configuration set based on the reconfiguration number constraint. The configuration set is a subset of the wake-up sequence candidates. Each element included in the configuration set is a wake-up sequence-reconfiguration timing pair. After obtaining the configuration set, we can design the reconfigurable structure that will have the minimum area overhead and still can minimize the average wake-up time as much as possible under the reconfiguration number constraint.

In the on-line reconfiguration phase, we use the critical path delay monitor (CPM) [10] to decide the reconfiguration timing. Once the critical path delay informed by the CPM matches the design time critical path delay estimation, we reconfigure the wake-up sequence to the one that can lead to the lowest wake-up time at the aging scenario.

B. Problem formulation

The **NBTI-aware wake-up time minimization problem** is intended to pick up a number of wake-up sequences from a given group of wake-up sequence candidates as a configuration. Based on the configuration, we can minimize the average wake-up time within a given observation window.

Fig. 5 shows the NBTI-aware wake-up time minimization problem formulation. There are four inputs: (1) wake-up sequence candidate set, W. Each element (w_i, t_i) in W is a wake-up sequence-reconfiguration timing pair. The wake-up sequence w_i is determined at aging scenario t_i, and only can be used at t_i as early as possible. These elements are the candidates to be chosen in order to reduce the wake-up time after circuit aging. (2) wake-up time table, WT. Each entry in WT is the wake-up time of a wake-up sequence at different aging scenarios. (3) reconfiguration number constraint, $CSTR_{reconfig}$. This constraint stands for the maximum number of reconfigurations we can do and the maximum number of wake-up sequence-reconfiguration timing pairs we can pick up from W. (4) observation window, OW. OW is the circuit lifetime we consider.

Under the initial conditions, the configuration set C only contains (w_0, t_0) pair. w_0 is the wake-up sequence obtained before circuit aging. The objective function is used to minimize the average wake-up time during OW while the number of wake-up sequence-reconfiguration timing pairs picked up from W cannot exceed $CSTR_{reconfig}$.

Inputs: (1) *Wake-up sequence candidate set*, $W = \{(w_1, t_1), (w_2, t_2), \ldots, (w_n, t_n)\}$ (2) *Wake-up Time Table*, *WT* (3) $CSTR_{reconfig}$ (4) *Observation Window*, *OW*
Initial Condition: *Configuration Set* $C = \{(w_0, t_0)\}$,
Constraint: $(|C|\text{-}1) \leq CSTR_{reconfig}$, $C\text{-}\{(w_0, t_0)\} \subseteq W$
Objective function: *Minimize the average wake-up time during OW by reconfiguration the wake-up sequences in C after circuit aging.*

Fig. 5. NBTI-aware wake-up time minimization problem

C. NBTI-Aware Wake-Up Sequence Decision Algorithm

To solve the NBTI-aware wake-up time minimization problem, we propose the NBTI-aware wake-up sequence decision (NAWSD) algorithm. The pseudo code of the proposed algorithm is shown in Fig. 6. NAWSD takes the wake-up sequence candidate set, the wake-up time table WT, the constraint of reconfiguration number $CSTR_{reconfig}$ and the observation window OW as inputs. The NAWSD output is a configuration set C. The elements included in C are the wake-up sequence-reconfiguration timing pairs. Based on the configuration set C, we can reduce the wake-up time under the reconfiguration number constraint.

【NBTI-Aware Wake-Up Sequence Decision Algorithm】
Input: (1) $W = \{(w_1, t_1), (w_2, t_2), \ldots, (w_n, t_n)\}$ (2) *Wake-up Time Table, WT* (3) $CSTR_{reconfig}$ (4) *Observation Window, OW*
Output: $C = \{(w_0, t_0), (w_{c1}, t_{c1}), (w_{c2}, t_{c2}), \ldots, (w_{ci}, t_{ci})\}$, where $(w_{ci}, t_{ci}) \in W$ *for* $\forall i = \{1, 2, \ldots, n\}$

```
1: for i ← 0 to CSTR_reconfig
2:     /* step 1: calculate avg. wake-up time in WT
3:     for j ← 0 to n
4:         Total_Wakeup_Time = 0 ;
5:         Avg_Wakeup_Time_j = 0 ;
6:         for k ← 0 to n
7:             Total_Wakeup_Time = + wakeup_time_entry_jk ;
8:         end for
9:         Avg_Wakeup_Time_j = Total_Wakeup_Time / n ;
10:    end for
11:    /* step 2: select the wake-up sequence configuration set*/
12:    Find the configuration C in WT that has the minimum average \
13:    wake-up time;
14:    /* step 3: if not done then update the entries in WT */
15:    if (i > n) return C ;
16:    else UPDATE (WT) ;
17: end for
18: return C ;
```

Fig. 6. The pseudo code of the NAWSD algorithm

In Fig. 6, lines 3-10 show the first step of NAWSD, which is used to calculate the average wake-up time for each possible configuration in WT when integer i is 0, 1, …, $CSTR_{reconfig}$. Note that n is the number of observation points, which is equal to $CSTR_{reconfig}$ by default. After the average wake-up time for each configuration are obtained, the second step is to find the configuration that has the minimum average wake-up time among all possible configurations (lines 12 and 13). Finally, if integer i is still less than $|W|$ and not equal to $CSTR_{reconfig}$, we update the entries in the wake-up time table for the next iteration (line 16); otherwise, we return C (line 15 and line 18).

IV. Simulaiton Setup and Experimental Results

We use ISCAS-85 benchmarks to demonstrate the efficacy and effectiveness of our proposed strategy. NAWSD and WTM are implemented with Perl. All experiments are performed on a 6-core, 2.10GHZ, Intel Xeon CPU, with 8GB memory, and CentOS release 6.5 machine. All the experimental benchmarks are implemented with PTM 32nm CMOS technology [11]. We use the PTM customized model cards [11] to simulate the different PMOS ST aging scenarios, and then obtain the surge current and NBTI-induced delay using HSPICE simulations. For the aging simulation, we set the observation window to 10 years. Note that to balance simulation accuracy and simulation time, we set the observation point monthly for the first year and yearly for

the following 9 years. We also set the supply voltage to 0.9V, the operating temperature to 45°C, and the size of all the inserted STs to identical. The surge current constraint was set to half of the maximum surge current induced by the all-on wake-up sequence before circuit aging.

Our experimental results are shown in Table 1, where column 1 and column 2 show the benchmark information; column 3 shows the number of STs inserted; column 4 shows the surge current constraint (in μA), and columns 5, 6 are the average wake-up time (in *ns*) of a fixed wake-up sequence approach [2] and our method, respectively. The fixed wake-up sequence approach always turns on STs by using a wake-up sequence obtained before circuit aging. Finally, we compare our method to the fixed wake-up sequence approach in terms of average wake-up time reduction (in %). Column 7 shows the results. It can be seen that our method can achieve a reduction in average wake-up time of up to 45.04% for C1908. Even for C6288, our method can still achieve to a 33.41% wake-up time reduction. Note that among these benchmarks, the least runtime (about a half hour) experiment is for C432, the largest runtime (about 46 hours) experiment is for C7552.

TABLE 1. BENCHMARK INFORMATION, SURGE CURRENT CONSTRAINT, AVERAGE WAKE-UP TIME FOR FIXED WAKE-UP SEQUENCE APPROACH AND OUR APPROACH, AVERAGE WAKE-UP TIME REDUCTION IN OUR APPROACH

Circuit	Gate Count	# STs	Surge Current Constraint (μA)	Fixed [2]	Our Method	
				Avg. Wake-up Time (ns)	*Avg. Wake-up Time (ns)*	*Avg. Wake-up Time Reduction (%)*
C432	160	8	38.00	12.23	7.37	39.71
C880	383	20	96.00	11.74	6.60	43.75
C1355	546	20	98.00	12.05	6.92	42.56
C1908	880	40	187.00	11.87	6.52	45.04
C2670	1269	70	334.85	11.94	6.85	42.58
C3540	1669	80	425.49	11.83	6.82	42.39
C5315	2307	120	573.45	12.10	7.04	41.79
C6288	2416	90	418.65	12.50	8.32	33.41
C7552	3513	170	807.90	12.27	7.47	39.09

TABLE 2 PARASITIC MULTIPLEXER AREA OVERHEAD INDUCED BY WAKE-UP SEQUENCE RECONFIGURABLE STRUCTURE FOR ISCAS-85 BENCHMARKS

Circuit	#Reconf	#MUX	Original	Our Method Parasitic	
			Cell Area (μm) ^{2}	*MUX Area (μm)* ^{2}	*Area Overhead (%)*
C432	4	4	87.82	5.23	6.0
C880	8	8	186.38	10.45	5.6
C1355	7	7	246.18	9.14	3.7
C1908	11	11	392.20	14.37	3.7
C2670	12	12	606.45	15.68	2.6
C3540	14	14	799.64	18.29	2.3
C5315	14	14	1183.71	18.29	1.5
C6288	12	12	1084.58	15.68	1.4
C7552	12	12	1668.81	15.68	0.9

To estimate the MUXs area overhead, we first analyze the reconfigurable structure and then count the number of the MUXs. Second, we refer to a given TSMC 40nm technology data book to obtain the cell area information for each different cell in the benchmarks. Then, we estimate the cell area in 40nm technology to the one in 32nm technology using constant voltage scaling. After that, we can calculate the total cell area and the area overhead increase after inserting the MUXs for each benchmark.

In Table 2, column 1 shows the benchmark name. Column 2 shows the reconfiguration number which is the total number of times we reconfigure the wake-up sequences. Column 3 shows the number of inserted MUXs. Column 4 shows the original total cell area (in $(\mu m)^2$) of each benchmark. Column 5 shows the MUX area (in $(\mu m)^2$) we added. Column 6 shows the area overhead after adding the MUXs. It can be seen that the area overhead is no more than 6%, and the overhead is decreased with increased total cell area..

V. CONCLUSIONS

In this paper, we propose a novel NBTI-aware wake-up strategy to minimize the average wake-up time for a power-gated design. The proposed strategy dynamically reconfigures wake-up sequences at different aging scenarios so more STs can be turned on at earlier stage. We also bring up a MUX-based reconfigurable structure, and develop the NAWSD algorithm which can find a wake-up sequence configuration set under a given reconfiguration constraint to minimize the average wake-up time. The experimental results show that our method can achieve up to 45.04% average wake-up time reduction compared to a fixed wake-up sequence approach with only 3.7% extra area overhead for the reconfiguration structure.

REFERENCES

[1] C. Long and L. He, "Distributed ST network for power reduction," in *IEEE Transactions on Very Large Scale Integration Systems* (*TVLSI*), vol. 12, no. 9, pp. 937–946, Sep. 2004.

[2] D.-C. Juan, Y.-T. Chen, M.-C. Lee, and, S.-C. Chang, "An efficient wake-up strategy considering spurious glitches phenomenon for power gating designs," in *IEEE transactions on very large scale integration systems* (*TVLSI*), vol. 18, no. 2, pp. 246-255, 2010.

[3] M.-C. Lee, Y. Shi, and S.-C. Chang, "Efficient wakeup scheduling considering both resource usage and timing budget for power gating designs," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (*TCAD*), vol. 31, no. 7, pp.1041-1049, 2012.

[4] Y.-T. Shyu, J.-M. Lin, C.-C. Lin, C.-P., Huang, S.-J. Chang, "An Efficient and Effective Methodology to Control Wake-up sequence of Power Switches for Power Gating Designs," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (*TCAD*), vol. 35, no. 10, pp. 1730-1743, 2016.

[5] Neil H. E. Weste and David Money Harris, Integrated Circuit Design, Fourth Edition. PEARSON, 2011.

[6] K.-C. Wu, I.-C. Lin, Y.-T. Wang, and S.-S. Yang, "BTI-aware ST sizing algorithm for reliable power gating designs," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (*TCAD*), vol. 33, no. 10, pp. 1591–1595, Oct. 2014.

[7] M.-C. Lee, Y.-G. Chen, D.-K. Huang, and S.-C. Chang, "NBTI-Aware Power Gating Design," in *Proc. of Asia and South Pacific Design Automation Conference* (*ASP-DAC*), pp.609-614, Jan. 2011.

[8] D. Rossi, V. Tenentes, S. Khursheed, and B. M. Al-Hashimi, "NBTI and leakage aware ST design for reliable and energy efficient power gating," in *Proc. of IEEE European Test Symposium* (*ETS*), pp.1-6, May 2015.

[9] Y.-G. Chen, W.-Y. Wen, Y.-T. Wang, Y.-L. Lee, and S.-C. Chang, "A Novel Low-Cost Dynamic Logic Reconfigurable Structure Strategy for Low Power Optimization," in *Proc. of Asia and South Pacific Design Automation Conference* (*ASP-DAC*), pp.250-255, Jan. 2016.

[10] Drake, R. Senger, H. Deogun, G. Carpenter, S. Ghiasi, T. Nguyen, N. James, M. Floyd, and V. Pokala, "A Distributed Critical-Path Timing Monitor for a 65nm High-Performance Microprocessor," in *Proc. of IEEE International Solid- State Circuits Conference* (*ISSCC*), pp.398–399, 2007.

[11] Predictive Technology Model (PTM). [Online]. Available http://www.ptm.asu.edu, accessed Mar. 2013.