

NBTI Mitigation Method by Inputting Random Scan-In Vectors in Standby Time

Hiroaki KONOURA^{†,††a)}, Toshihiro KAMEDA^{†,††}, *Student Members*, Yukio MITSUYAMA^{†††,††b)}, Masanori HASHIMOTO^{†,††c)}, and Takao ONOYE^{†,††d)}, *Members*

SUMMARY Negative Bias Temperature Instability (NBTI) is one of the serious concerns for long-term circuit performance degradation. NBTI degrades PMOS transistors under negative bias, whereas they recover once negative bias is removed. In this paper, we propose a mitigation method for NBTI-induced performance degradation that exploits the recovery property by shifting random input sequence through scan paths. With this method, we prevent consecutive stress that causes large degradation. Experimental results reveal that random scan-in vectors successfully mitigate NBTI and the path delay degradation is reduced by 71% in a test case when standby mode occupies 10% of total time. We also confirmed that 8-bit LFSR is capable of random number generation for this purpose with low area and power overhead.

key words: NBTI, NBTI mitigation, performance degradation, scan path, aging, reliability

1. Introduction

In nanoscale integrated circuit design, performance degradation due to aging effects is becoming a major concern. Conventionally, to cope with the performance degradation due to aging, a design margin is given so that even an aged circuit will satisfy the performance requirement. However, as the necessary margin increases with technology scaling, the performance loss due to the margin is becoming less acceptable. Therefore, mitigating aging effects is highly demanded.

Among various aging effects, negative bias temperature instability (NBTI) is nowadays one of the most influential effects. NBTI causes a gradual increase in $|\Delta V_{th}|$ while negative bias is applied to PMOS, i.e. PMOS is ON. Here, this condition is defined as stress phase of NBTI. The $|\Delta V_{th}|$ increase reduces drain current and consequently lengthens path delay, which leads to timing failures. On the other hand, while PMOS is OFF, $|\Delta V_{th}|$ gradually decreases to its initial value before the stress enforcement. This condition is defined as recovery phase of NBTI. The PMOS degradation by NBTI gradually progresses through circuit oper-

ations that repeat the stress and recovery phases. Besides, reference [1], [2] pointed out that the ratio of stress and recovery phases, which is hereafter called stress probability, is a key parameter that determines the amount of degradation in a long-term perspective. As the stress probability gets close to 1, the degradation increases and approaches to the worst case that DC stress is given. This NBTI property of recovery can be exploited for NBTI mitigation at circuit level by controlling the stress probability.

A possible way to control the stress probability is input vector control (IVC) [3]–[5]. IVC is well known as a leakage reduction technique that exploits the dependency of leakage current on the input vector [6]. For NBTI mitigation, a pre-determined input vector is given to a combinational circuit to minimize performance degradation during standby mode. In [3], Y. Wang et al. evaluated IVC technique for NBTI mitigation taking into account the temperature that affects NBTI-induced performance degradation. The same authors also derived Pareto sets that simultaneously optimize NBTI mitigation and leakage currents in [4]. D. R. Bild et al. introduced internal node control (INC) to IVC in order to improve the controllability of internal nodes [5]. INC adds some devices to set the internal node values to 0 or 1, which involves an increase both in area and path delay.

However, only using a single input vector, a considerable number of PMOSs are in stress phase and their performance degrades in standby mode, because CMOS gates are basically inverting logic and all the logic values cannot be 1 simultaneously. This problem becomes significant as the portion of standby mode increases, because the degradation during the standby mode becomes comparable, or rather larger than that during active mode. To overcome this problem, S. Jin et al. presented a multiple input vector approach [7]. They derived a set of input vectors and allocated standby time to each of them to maximize the NBTI mitigation effect. On the other hand, the number of input vectors is thought to increase as the target circuit becomes larger, which means larger memory is necessary to store input vectors.

We then propose a random input vector approach that shifts random values through scan paths. This approach gives different input vectors for each scan-shift operation, which means the proposed random input vector approach can be regarded as one of multiple input vector approaches, though it does not prepare input vectors beforehand. In addi-

Manuscript received September 12, 2013.

Manuscript revised January 9, 2014.

[†]The authors are with the Department of Information Systems Engineering, Osaka University, Suita-shi, 565-0871 Japan.

^{††}The authors are with JST, CREST, Tokyo, 102-0075 Japan.

^{†††}The author is with the School of Systems Engineering, Kochi University of Technology, Kami-shi, 782-8502 Japan.

a) E-mail: konoura.hiroaki@ist.osaka-u.ac.jp

b) E-mail: mitsuyama.yukio@kochi-tech.ac.jp

c) E-mail: hasimoto@ist.osaka-u.ac.jp

d) E-mail: onoye@ist.osaka-u.ac.jp

DOI: 10.1587/transfun.E97.A.1483

tion, the proposed approach does not need memory to store input vectors. First, we experimentally show the difference in performance degradation with different scan-in vectors, and reveal that random scan-in vectors attain a good mitigation effect. We then evaluate how the quality of randomness affects the mitigation effect with various linear feedback shift register (LFSR) configurations.

It should be noted that another circuit level approach is power gating for NBTI mitigation [8]. As for circuits that implement power gating for leakage current reduction during standby mode, NBTI can be mitigated as a byproduct. On the other hand, it is difficult to apply power gating to high-performance design whose permissible speed degradation is highly limited, because the sleep transistors degrade the performance and increase uncertainty in performance. In addition, aging of sleep transistors could be a serious problem in some cases. The proposed NBTI mitigation is applicable to even such high-performance designs.

The rest of this paper is organized as follows. Section 2 refers the NBTI model used for our evaluation. Section 3 introduces the concept of NBTI mitigation by giving scan-in vectors. In Sect. 4, we evaluate the effect of NBTI mitigation assuming various operations of target circuits, and also clarify that giving random scan-in vectors generated in linear feedback shift register (LFSR) is effective to mitigate NBTI. Finally, we conclude this paper in Sect. 5.

2. NBTI Model

To characterize NBTI, a number of measurement results and closed-form models have been proposed [1], [9], [10]. Among them, long-term degradation and its dependency on stress probability are well modeled in [1]. We therefore assume the model below in this work, though other models can be used as long as the dependency on stress probability is well characterized,

$$|\Delta V_{th}(t)| = \left(\frac{\sqrt{K_v^2 \cdot T_{clk} \cdot \alpha}}{1 - \beta(t)^{1/2n}} \right)^{2n}, \quad (1)$$

where K_v is a parameter dependent on supply voltage and temperature. T_{clk} is a clock period, and α is the stress probability of PMOS. $T_{clk} \cdot \alpha$ represents a period of stress time. $\beta(t)$ is a function of time t that has a dependence on temperature and T_{clk} , and it includes a term of $(1 - \alpha)$. Both α and $\beta(t)$ are less than 1. Moreover, n is equal to 1/6 in a hydrogen molecule diffusion based model [11].

Reference [12] reported that $|\Delta V_{th}|$ is hardly dependent on T_{clk} in case of $t > 1,000$ s. In this case, Eq. (2) is more useful for $|\Delta V_{th}|$ estimation instead of Eq. (1) [12].

$$|\Delta V_{th}| \approx \left(\frac{0.001n^2 K_v^2 \alpha C t}{0.81 t_{ox}^2 (1 - \alpha)} \right)^n. \quad (2)$$

Here, in Eq. (2), when α approaches to 1, $|\Delta V_{th}(t)|$ reaches an infinite value and is not appropriate. In such a situation, as its upper limit, we use Eq. (3) which models only stress phase of NBTI [13].

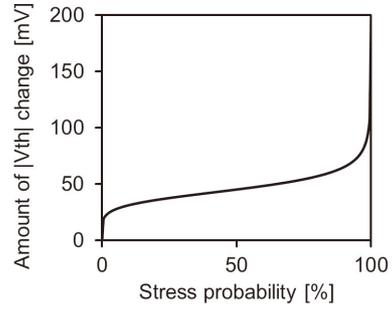


Fig. 1 Threshold voltage degradation vs. stress probability (10 years later).

$$|\Delta V_{th}(t)| = (K_v^2 t)^n \quad (3)$$

Figure 1 exemplifies the amount of threshold voltage degradation derived by Eqs. (2) and (3) after a lapse of ten years in the condition that supply voltage is 1.1 V and temperature is 125°C. The initial threshold voltage is -180 mV. Here, other parameters in the NBTI model except the initial threshold voltage were referred to [1], where [1] extracted and reported the parameters for transistors in a 65 nm process. In this graph, the degradation is quite large when the stress probability is near 100%. This result suggests that for PMOSs with near 100% stress probability, even a small reduction in the stress probability is helpful to mitigate $|\Delta V_{th}|$ increase. The proposed method aims to decrease stress probability of such PMOSs.

3. NBTI Mitigation by Giving Scan-in Vectors

For mitigating NBTI, it is effective to make PMOSs in consecutive stress experience recovery phase. For this purpose, we give a set of input vectors to combinational circuits through scan paths and switch stressed PMOSs. Note that this work assumes that the target circuit has two operation modes; active and standby modes. In standby mode, FF values are flushed and can be overwritten through scan paths. In some cases, an operation to store and restore FF values might be necessary before and after inputting vectors for NBTI mitigation. By using a set of input vectors, the multiple input vector approach solves the problem of single IVC that some of PMOSs are fixed in stress phase during standby mode. Figure 2 illustrates a conceptual diagram of the NBTI mitigation. To give a set of input vectors to combinational circuits, we use a scan path which is often embedded for design for testability (DFT). In active mode, there are PMOSs which hardly transit in recovery phase, such as a PMOS in the upper NAND gate in Fig. 2(a). The multiple input vector approach gives a chance to such PMOSs to transit in recovery phase during standby mode (Fig. 2(b)).

To apply the multiple input vector approach to a circuit, we need to determine what set of input vectors should be used as scan-in vectors. A possible way is to gather the input vectors derived by IVC techniques, which is proposed in [7]. On the other hand, because an additional memory that stores input vectors is necessary, the number of input

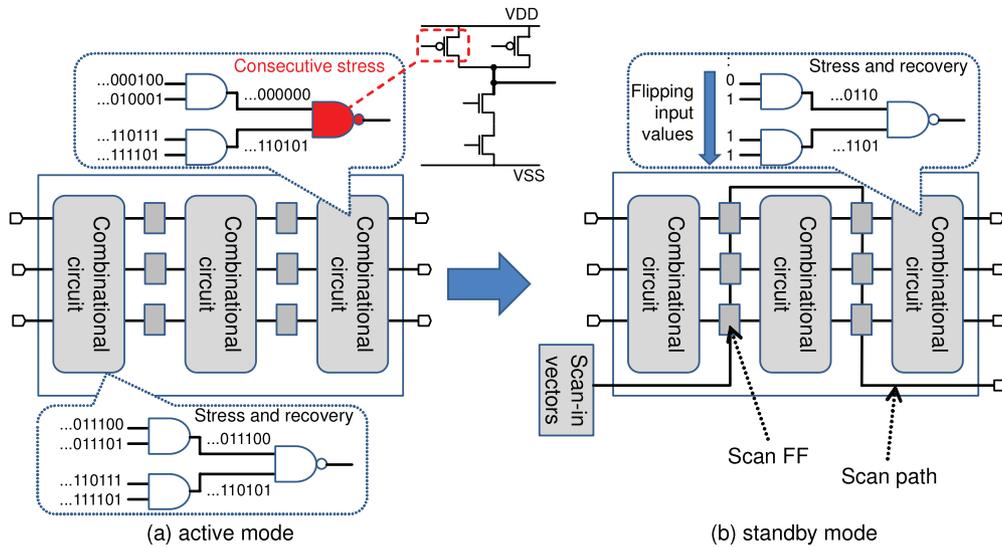


Fig. 2 Conceptual diagram of the proposed mitigation method using scan path.

vectors should be small, while targeted PMOSs should have a chance to be in recover phase. In addition, during scan-in and scan-out operations, the input vectors given to combinational circuits are different from those derived by the technique of [7]. This means that even if a good set of input vectors are available, they cannot be efficiently enforced to the combinational circuit in time when the scan paths are long and/or the standby time is short, because it takes a long time to change the input pattern.

Another approach is to give random scan-in vectors supposing that almost all PMOSs have a chance to be in recovery phase, and this is the approach proposed in this paper. In this approach, every set of input values given to a combinational circuit, which is generated each time the scan path is one-bit shifted with a random scan-in value, is regarded as an input vector in multiple input vector approach. An advantage of this approach is that an additional memory to store the input patterns is unnecessary and the number of input vectors can be easily increased by lengthening the cycle of random numbers. As weak points, this cannot guarantee that targeted PMOSs have a chance of recovery phase and additional mechanism to generate random patterns is necessary.

In the following, we focus on the approach of random scan-in vectors, and experimentally evaluate how much mitigation of performance degradation can be obtained through a case study.

4. Evaluation of NBTI Mitigation

In this section, we show a case study of NBTI mitigation and reveal that inputting random values in scan path is sufficiently effective. We also discuss the generation of random number and its cost of area and power.

4.1 Experimental Setup

As target circuits for evaluation, we selected two circuit blocks in MIPS R3000 processor; 32-bit MDU (Multiply Divide Unit) which operates multiplication and division, and 32-bit ALU (Arithmetic Logic Unit) which operates addition, subtraction and logical operations. In addition, we also target a circuit of discrete cosine transform (DCT) for image processing in MPEG-4 video standard.

These circuits were synthesized with Synopsys DFT Compiler using an industrial 65 nm standard cell library. 32-bit MDU, 32-bit ALU and DCT have 198, 68 and 677 FFs, for which the Synopsys DFT Compiler construct 20, 7 and 5 scan paths with an ordinary option, respectively. Their circuit sizes are 4,670, 3,149 and 11,591 NAND2-equivalent gates, and their increase rates from the non-scan circuits are 5.8%, 3.3% and 9.5%, respectively. In this work, we focus on long-term degradation rather than short-term one, and hence operating frequency T_{clk} both for active mode operation and scan-in operation is not explicitly considered.

Besides, the performance degradation depends on circuit operations during active mode because stress probabilities of PMOS transistors vary depending on input vectors. For taking into account different situations of circuit operation, the following five situations are considered for 32-bit MDU.

General: Both operations and operands are randomly selected and executed.

32-bit_mult: Operands are randomly selected. Only multiplication is executed.

32-bit_div: Operands are randomly selected. Only division is executed.

16-bit_mult: Higher 16-bit of operands are fixed to 0. Only multiplication is executed.

16-bit_div: Higher 16-bit of operands are fixed to 0. Only

division is executed.

Operations for 32-bit-ALU are listed below.

General: Both operations and operands are randomly selected and executed.

16-bit_add: Higher 16-bit of operands are fixed to 0. Only addition is executed.

Const_add: One operand is fixed to $00000001_{(16)}$, and the other is randomly selected. Only addition is executed.

For DCT, we change input images as follows.

Black: Pixel data of inputs is always 0.

White: Pixel data of inputs is always $11111111_{(2)}$.

Noise: Pixel data of inputs is randomly selected.

Bus: A photo image ('Bus') is given.

We gave four types of scan-in vectors ("all 1", "all 0", "0101" and "random") during standby mode. "all 0" applies 0's to scan inputs, and "all 1" applies 1's. In these two inputs, which are special cases of single-IVC, DC stress is given during standby mode. On the other hand, "0101" and "random" change input vectors to combinational circuits during standby mode. "0101" injects 1 and 0 alternately to scan inputs, in other words, the sequence stored in scan-FFs alternates "010101..." and "101010..." after the scan inputs reach the end of scan path. Moreover, "random" inputs 1 and 0 randomly, and hence the sequence stored in scan-FFs changes variously for every scan-shift. In any cases, primary inputs are supposed to remain the last input values of the previous active mode. Besides, this study does not assume reloading logic values after scan-in. Some circuits need to restore logic values of FFs after standby time, but others need not. The necessity of restoration depends on applications and circuits. In the case that the restoration is necessary, a possible approach for low latency restoration is to embed a shadow latch into each FF to store the original logic value. The area and power overhead for the shadow latches will be briefly investigated in Sect. 4.4.

We evaluated the initial circuit delay and the delay after 10 years assuming supply voltage is 1.1 V, temperature is 125°C and initial threshold voltage is -180 mV. We adopted the procedure of NBTI-aware timing analysis [14] in this paper. We first calculated stress probabilities of all PMOSs with logic simulation taking into account both active and standby modes, next annotated $|\Delta V_{th}|$ for each PMOS according to the stress probability, and carried out transistor-level static timing analysis with Synopsys Nanotime. To estimate the upper bound of NBTI mitigation effect, we also computed the circuit delay supposing that all PMOSs were always in recovery phase during the standby mode, which is called as "all recovery". Note that in complementary digital circuits, such an ideal situation is practically infeasible due to Boolean algebra, but the upper bound of NBTI mitigation effect is useful to evaluate how effectively NBTI mitigation methods work. For example, if the NBTI mitigation effect of a method is very close to this upper bound, this method is working well and there is little room for further improvement.

4.2 Evaluation Results

4.2.1 Evaluation on Standby Time Ratio

When implementing the proposed NBTI mitigation method, standby time ratio can be a key parameter that determines the amount of NBTI mitigation. Here, the ratio is defined as (time of standby mode)/(time of active mode + time of standby mode). If the ratio necessary for mitigating NBTI sufficiently is quite large, the designs that can use the proposed method become limited. Therefore, we evaluate the relationship between standby time ratio and mitigated critical path delay.

Figures 3–5 show the dependency of critical path delay on the ratio of standby mode in each case of *16-bit_mult* of 32-bit MDU, *16-bit_add* of 32-bit ALU, and *black* of DCT. Here, the ratio of 0% means that the circuit is in active mode all the time, and 100% corresponds to the case that the circuit is always in standby mode. The initial critical path delays of 32-bit MDU, 32-bit ALU and DCT are 0.978, 3.424 and 2.441 ns, respectively.

First, we can see from Figs. 3(a) and (b) that the critical path delays of "all 0" and "all 1" are larger than those of "0101" and "random". A single input vector during standby mode significantly degrades the circuit delay. While other single-IVC input vectors might give better results, it should be noted that even such single-IVC input vectors increase the circuit delay as the ratio of standby time approaches to 100%, because almost DC stress is given to the circuit. On the other hand, "random" attained the minimum delay increase, which indicates that giving a set of input vectors to combinational circuits is helpful to mitigate NBTI. Another important observation is that even a short ratio of standby time, such as 2%, greatly contributed to NBTI mitigation (Fig. 3(b)), which suggests that our approach is applicable for busy circuits operating with little idle time.

Figures 4(a) and (b) show the result of *16-bit_add*. Similarly to Fig. 3, a small standby ratio reduced the delay increase. Below 90% of the ratio, the critical path delays are almost the same for "all 1", "0101" and "random", but above 90%, "all 0" and "all 1" induced a significant delay increase, whereas "random" mitigated NBTI further. This is because the degradation of PMOS under static stress during standby mode becomes quite large along the increase of the standby time ratio. Besides, under the DC stress (standby time ratio is 100%), there is 1 ns difference between critical path delay of "all 0" and that of "all 1". This large difference is explained as follows. In Fig. 4, the critical paths consist of carry propagation logics in full adders. Figure 6 illustrates a part of full adder including the path from carry-in (*CIN*) to carry-out (*COU*T). When inputs $\{A, B, CIN\}$ are "all 0", then *COU*T becomes '0'. In this case, PMOSs $\{P1, P2, P3\}$ are under stress. On the other hand, when inputs $\{A, B, CIN\}$ are "all 1", then *COU*T becomes '1'. In this case, a PMOS *P4* is under stress. Here, the path delays from *CIN* to *COU*T of "all 0" and "all 1" after 10 years are evaluated at 0.108 ns

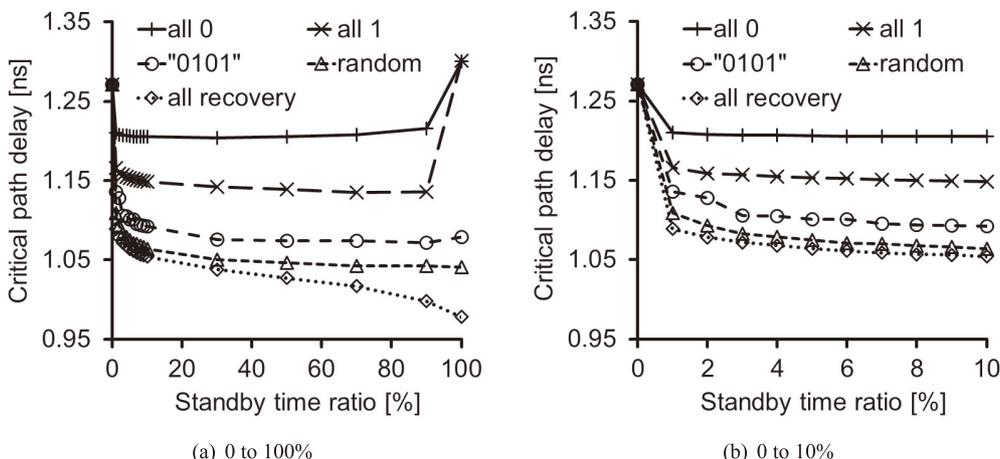


Fig. 3 Critical path delay vs. standby time ratio (16-bit_mult of 32-bit MDU).

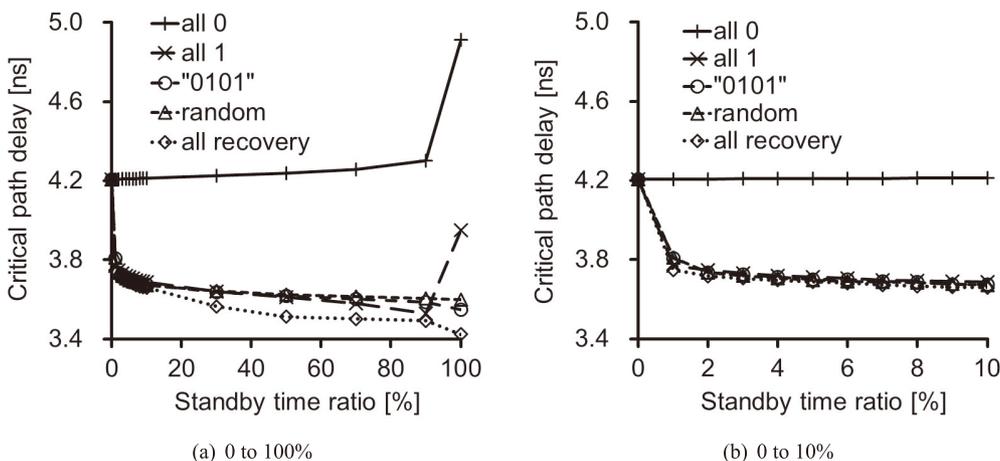


Fig. 4 Critical path delay vs. standby time ratio (16-bit_add of 32-bit ALU).

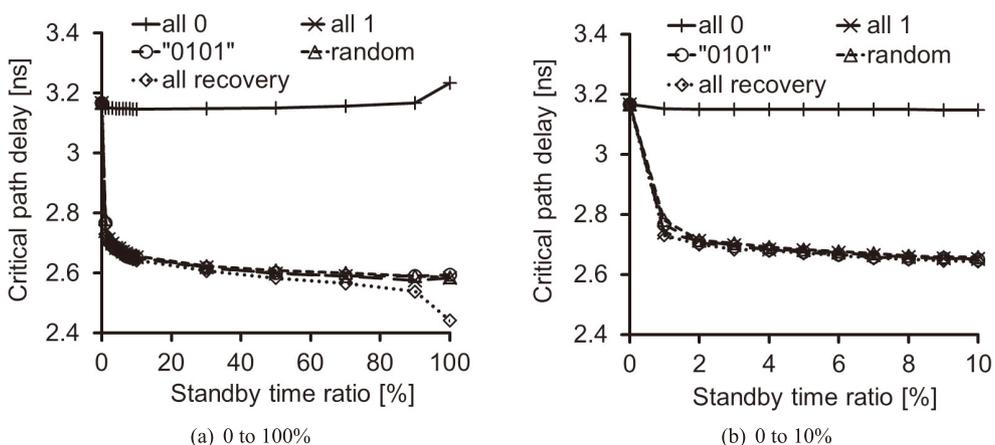


Fig. 5 Critical path delay vs. standby time ratio (black of DCT).

and 0.056 ns. Note that for both cases, inputs and outputs of n-bit full adders are kept the same, and thus stressed PMOSs are also the same. Therefore, the difference of path delays between “all 0” and “all 1” of 16-bit full adders is estimated as $(0.108 - 0.056) \times 16 = 0.832$ ns, which is mostly correlated

with the results of Fig. 4. This result indicates that even under DC stress, path delay degradation can vary significantly depending on scan-in vectors.

Also, Figs. 5(a) and (b) show the result of *black* of DCT. Similarly to Figs. 3 and 4, a small standby ratio suc-

successfully mitigated the delay increase. The critical path delay of “all 0” is larger than those of other scan-in vectors. Especially, there is a large difference between “all 0” and “all 1”. We confirmed that this can be explained by the carry propagation logics in full adders similarly to Fig. 4.

Here, to investigate the impact of the number of scan-chains on the mitigation effect of NBTI-induced delay degradation, we evaluated an additional circuit of DCT having only one scan path as an example. This evaluation result is shown in Fig. 7. Compared with Fig. 5 where the number of scan-chains is 5, we could not see a visible difference and the delay differences were within a few pico-seconds in this test case.

4.2.2 Evaluation on Various Operations

As NBTI model of Eq. (1) indicates, degradation amount due to NBTI is expressed as a function of PMOS usage, i.e.

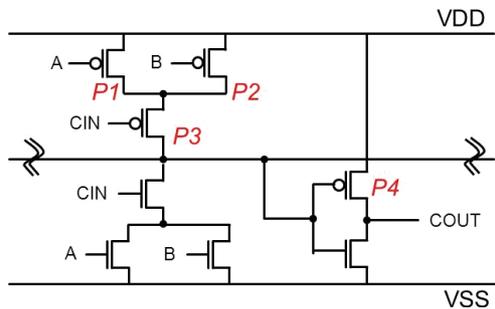


Fig. 6 A schematic of carry logic in full adder.

circuit operation. Therefore, we here evaluate NBTI mitigation assuming various circuit operations. Table 1 lists the delay increases of 32-bit MDU in the case of 10% standby ratio. In the situations of *16-bit_mult* and *16-bit_div*, the amounts of mitigation with “random” are large, and especially in *16-bit_mult* situation, “random” reduced the delay increase by 71% ((30.0%–8.79%)/30.0%).

Table 2 lists the evaluation results of 32-bit ALU. Similarly to 32-bit MDU, except for *general*, “random” attained large NBTI mitigation effects, while “0101” and “all 1” also mitigated NBTI significantly. It is due to that the degradation amount of “all 1” is smaller than “all 0”, as explained in the previous section. It is also surmised that “all 1” achieves large mitigation because operands in which higher bits are

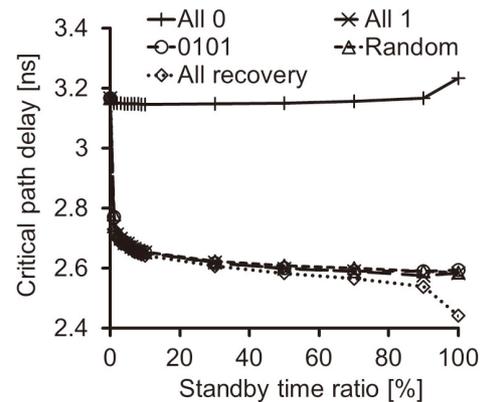


Fig. 7 Critical path delay vs. standby time ratio in using 1 scan-chain (black of DCT).

Table 1 Critical path delay with and without NBTI mitigation (32-bit MDU). The initial delay is 0.978 ns. Numbers in parentheses mean the ratios of delay increase from the initial circuit.

Operation	Critical path delay [ns] (degradation)					
	No standby time	All recovery	w/mitigation			
			All 0	All 1	0101	Random
<i>16-bit_mult</i>	1.271 (30.0%)	1.055 (7.9%)	1.205 (23.2%)	1.148 (17.4%)	1.092 (11.7%)	1.064 (8.8%)
<i>32-bit_mult</i>	1.131 (15.6%)	1.040 (6.3%)	1.116 (14.1%)	1.109 (13.4%)	1.083 (10.7%)	1.051 (7.5%)
<i>16-bit_div</i>	1.262 (29.0%)	1.059 (8.3%)	1.198 (22.5%)	1.097 (12.2%)	1.081 (10.5%)	1.070 (9.4%)
<i>32-bit_div</i>	1.077 (10.1%)	1.043 (6.6%)	1.077 (10.1%)	1.069 (9.3%)	1.060 (8.4%)	1.052 (7.6%)
<i>General</i>	1.049 (7.3%)	1.037 (6.0%)	1.049 (7.3%)	1.048 (7.2%)	1.049 (7.3%)	1.047 (7.1%)

Table 2 Critical path delay with and without NBTI mitigation (32-bit ALU). The initial delay is 3.424 ns. Numbers in parentheses mean the ratios of delay increase from the initial circuit.

Operation	Critical path delay [ns] (degradation)					
	No standby time	All recovery	w/mitigation			
			All 0	All 1	0101	Random
<i>16-bit_add</i>	4.206 (22.8%)	3.659 (6.9%)	4.211 (23.0%)	3.687 (7.7%)	3.673 (7.3%)	3.671 (7.2%)
<i>Const_add</i>	4.565 (33.3%)	3.740 (9.2%)	4.579 (33.7%)	3.735 (9.1%)	3.739 (9.2%)	3.744 (9.4%)
<i>General</i>	3.599 (5.1%)	3.581 (4.6%)	3.611 (5.5%)	3.590 (4.9%)	3.599 (5.1%)	3.599 (5.1%)

Table 3 Critical path delay with and without NBTI mitigation (DCT). The initial delay is 2.441 ns. Numbers in parentheses mean the ratios of delay increase from the initial circuit.

Operation	Critical path delay [ns] (degradation)					
	No standby time	All recovery	w/mitigation			
			All 0	All 1	0101	Random
<i>Black</i>	3.167 (22.9%)	2.643 (7.6%)	3.147 (22.4%)	2.652 (8.0%)	2.650 (7.9%)	2.657 (8.1%)
<i>White</i>	3.170 (23.0%)	2.639 (7.5%)	3.152 (22.6%)	2.652 (8.0%)	2.648 (7.8%)	2.657 (8.1%)
<i>Noise</i>	2.594 (5.9%)	2.581 (5.4%)	2.595 (5.9%)	2.583 (5.5%)	2.586 (5.6%)	2.587 (5.6%)
<i>Bus</i>	2.594 (5.9%)	2.580 (5.4%)	2.596 (6.0%)	2.584 (5.5%)	2.587 (5.6%)	2.588 (5.7%)

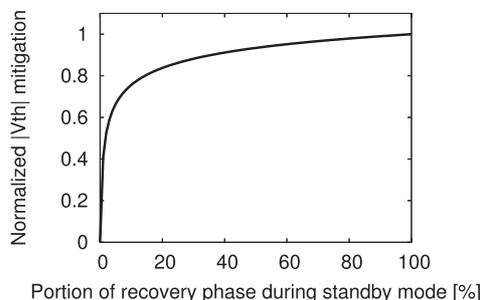


Fig. 8 $|V_{th}|$ mitigation as a function of the portion of recovery phase during standby mode.

fixed to 0 s experience 1 s during standby mode.

Table 3 lists the evaluation results of DCT. In the situations of *black* and *white*, not only “random” but also “0101” and “all 1” well mitigated the NBTI degradation. In both the situations, 8×8 input pixels are all the same, and thus all coefficients for AC components after DCT are always 0s. Scan-FFs that store AC coefficients hold 0s for a long period. Therefore, changing the values of scan-FFs with “0101” and “all 1” is highly effective to mitigate NBTI degradation. On the other hand, in the situations of *noise* and *bus*, degradation amount of NBTI is fairly small and its further mitigation is not necessary. It is because each of 8×8 input pixels varies and consequently the values of scan-FFs storing AC coefficients change.

We can also see that the circuit delay mitigated by “random” is close to that of “all recovery” in all situations. This result indicates that random scan-in vectors attained the mitigation effect close to the upper limit. Please recall that this upper limit cannot be actually obtained because no input vectors to make all PMOSs be in recovery phase simultaneously.

Let us investigate the reason why the NBTI mitigation by “random” is close to that of “all recovery”. In this evaluation, we assumed standby time occupies 10% of the total time, which means stress probabilities can be reduced by 10% with “all recovery”. On the other hand, the variation in stress probability due to “random” is less than 10%. Figure 8 shows $|V_{th}|$ mitigation for a PMOS whose stress probability in active mode is 100% as a function of the portion of recovery phase during standby mode. Note that under the long-term prediction model of NBTI used in this paper, all

Table 4 Critical path delay with different LFSR configurations (10 years later).

Configuration	Critical path delay [ns]		
	32-bit MDU	32-bit ALU	DCT
Random	1.064	3.671	2.657
4-bit LFSR	1.104	3.676	2.655
8-bit LFSR	1.064	3.672	2.655
12-bit LFSR	1.064	3.672	2.655
16-bit LFSR	1.064	3.676	2.655

PMOSs commonly follow this tendency. Here, $|V_{th}|$ mitigation is normalized by that of “all recovery”. This figure indicates that if we could allocate 3% of standby time to recovery phase, 60% of mitigation effect can be obtained. When 15% of standby time can be given to recovery phase, we can have 80% of mitigation effect. We think this is the reason why “random” attained a good NBTI mitigation effect.

4.3 Random Number Generation

The previous section showed that random scan-in vectors are useful to mitigate NBTI. On the other hand, when applying this scheme to a circuit on a chip, we need an on-chip random number generator. We here focus on LFSR which is a popular circuit to generate pseudo random numbers, and evaluate the relation between the quality of random number and NBTI mitigation effect. As the number of registers in LFSR increases, the cycle of random number becomes longer and the quality of random number improves, while the area needed for implementation becomes large.

We here evaluate critical path delays when LFSRs with different bits are used. Here, we assumed that operating frequency for LFSR is the same as that for active mode. Note that even LFSR and scan-chain are operating fast, the stress probability changes depending on the scan-in vectors, and hence the NBTI mitigation effect can be obtained. The results of 4-bit, 8-bit, 12-bit, and 16-bit LFSRs are shown in Table 4. When 8-bit or larger-bit LFSRs are used, there is little difference from “random” in mitigation efficiency. For 32-bit MDU, 32-bit ALU, and DCT used for experiments in this work, 8-bit LFSR can generate a sufficiently random scan-in vectors for NBTI mitigation. As the number of LFSR bits increases, the cycle of random number becomes

Table 5 Area and static power consumption.

	Area [# of NAND2 gates]		
	32-bit MDU	32-bit ALU	DCT
Original	4,415 (–5.5%)	3,048 (–3.2%)	10,583 (–8.7%)
Scan design	4,670	3,149	11,591
Scan design w/8-bit LFSR	4,722 (+1.1%)	3,202 (+1.7%)	11,644 (+0.5%)
Scan design w/shadow latch & 8-bit LFSR	5,366 (+14.9%)	3,423 (+ 8.7%)	13,844 (+19.4%)
	Static power consumption [μ W]		
	32-bit MDU	32-bit ALU	DCT
Original	1.49 (–2.6%)	1.04 (–1.0%)	3.20 (–8.3%)
Scan design	1.53	1.05	3.49
Scan design w/8-bit LFSR	1.55 (+ 1.3%)	1.07 (+ 1.9%)	3.51 (+ 0.6%)
Scan design w/shadow latch & 8-bit LFSR	1.81 (+18.4%)	1.16 (+10.5%)	4.40 (+26.2%)

longer, i.e. the number of input vectors for a combinational circuit increases. To give all the input vectors in a limited time, scan-shift must be performed faster when the number of LFSR is large. On the other hand, in the case of smaller LFSR, such as 8-bit, slow shifting is sufficient, which is desirable in terms of power dissipation because fast shifting involves large power dissipation.

4.4 Area and Power Estimation

Thus far, we revealed that NBTI mitigation can be performed by inputting the random scan-in vectors generated in LFSR during a short standby time. We finally discuss the cost of the proposed method, i.e. area and power overhead.

Table 5 lists the area and static power consumption of 32-bit MDU, 32-bit ALU and DCT. Original means non-scan design. Here, we consider that scan design is a common practice for DFT and then is regarded as a baseline for comparison. In this table, the area increase from original circuits to scan designs is in the range of 3.2–8.7%. On the other hand, the differences between scan designs and scan designs with 8-bit LFSR are less than 2.0%. Furthermore, when shadow latches are embedded in scan design for logic value restoration with 8-bit LFSR, the area increase from scan designs is in the range of 8.7–19.4%. As for static power consumption, the differences between original and scan designs are in the range of 1.0–8.3%. Meanwhile, the differences between scan designs with and without 8-bit LFSR are below 2.0%. Besides, as for scan design with shadow latch and 8-bit LFSR, static power consumption increases by 10.5% to 26.2% compared with scan design. These results lead to a deduction that adding LFSR only has a quite limited impact on area and static power consumption.

In addition, Fig. 9 shows the relationship between clock frequency for scan-shift and dynamic power consumption. Here, we assumed that “0101” was given to scan paths. This figure shows that when the clock frequency is kept below 1 MHz, the dynamic power consumption is at most comparable to static power consumption. Through these evaluations, we confirmed that giving random scan-in vectors generated in 8-bit LFSR with up to 1 MHz clock helps the mitigation of NBTI degradation with low area and power

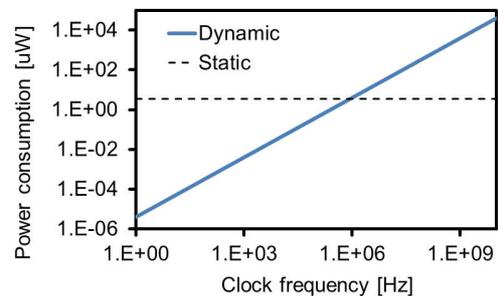


Fig. 9 Relationship between clock frequency and dynamic power consumption.

overhead.

5. Conclusion

In this paper, we proposed a NBTI mitigation method that injects random patterns to scan paths, and evaluated the NBTI mitigation effect through case studies. The experimental results with 32-bit MDU and ALU in MIPS R3000 processor and DCT circuit for MPEG-4 showed that the proposed method well mitigated NBTI. The reduction of delay increase reached 71% in a test case and was comparable to the upper limit assuming all PMOSs were in recovery phase. We also confirmed that 8-bit LFSR is capable of random number generation for NBTI mitigation with low area and power overhead.

References

- [1] W. Wang, V. Reddy, A.T. Krishnan, R. Vattikonda, S. Krishnan, and Y. Cao, “Compact modeling and simulation of circuit reliability for 65 nm CMOS technology,” *IEEE Trans. Device Material Reliability*, vol.7, pp.509–517, Dec. 2007.
- [2] S. Ramey, C. Prasad, M. Agostinelli, S. Pae, S. Walstra, S. Gupta, and J. Hicks, “Frequency and recovery effects in high- κ BTI degradation,” *Proc. IRPS*, pp.1023–1027, April 2009.
- [3] Y. Wang, H. Luo, K. He, R. Luo, H. Yang, and Y. Xie, “Temperature-aware NBTI modeling and the impact of input vector control on performance degradation,” *Proc. DATE*, pp.546–551, April 2007.
- [4] Y. Wang, X. Chen, W. Wang, V. Balakrishnan, Y. Cao, Y. Xie, and H. Yang, “On the efficacy of input vector control to mitigate NBTI effects and leakage power,” *Proc. ISQED*, pp.19–26, March 2009.
- [5] D.R. Bild, G. Bok, and R.P. Dick, “Minimization of NBTI performance degradation using internal node control,” *Proc. DATE*,

- pp.148–153, April 2009.
- [6] A. Abdollahi, F. Fallah, and M. Pedram, “Leakage current reduction in CMOS VLSI circuits by input vector control,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol.12, no.2, pp.140–154, Feb. 2004.
- [7] S. Jin, L. Zhang, H. Li, X. Li, and G. Yan, “M-IVC: Using multiple input vectors to minimize aging-induced delay,” *Proc. ATS*, pp.437–442, Nov. 2009.
- [8] A. Calimera, E. Macii, and M. Poncino, “NBTI-aware sleep transistor design for reliable power-gating,” *Proc. GLSVLSI*, pp.333–338, May 2009.
- [9] M.A. Alam and S. Mahapatra, “A comprehensive model of PMOS NBTI degradation,” *Microelectronics Reliability*, vol.45, pp.71–81, 2005.
- [10] J.H. Lee, W.H. Wu, A.E. Islam, M.A. Alam, and A.S. Oates, “Separation method of hole trapping and interface trap generation and their roles in NBTI reaction-diffusion model,” *Proc. IRPS*, pp.745–746, April–May 2008.
- [11] A.T. Krishnan, C. Chancellor, S. Chakravarthi, P.E. Nicollian, V. Reddy, A. Varghese, R.B. Khamankar, and S. Krishnan, “Material dependence of hydrogen diffusion: Implications for NBTI degradation,” *Proc. IEDM*, pp.688–691, Dec. 2005.
- [12] S. Bhardwaj, W. Wang, R. Vattikonda, Y. Cao, and S. Vrudhula, “Predictive modeling of the NBTI effect for reliable design,” *Proc. CICC*, pp.1047–1052, Aug. 2006.
- [13] Y. Cao, “Reliability mechanisms and the impact on IC designs,” *Proc. ASP-DAC*, Tutorial 4, pp.342–372, Jan. 2009.
- [14] H. Konoura, Y. Mitsuyama, M. Hashimoto, and T. Onoye, “Stress probability computation for estimating NBTI-induced delay degradation,” *IEICE Trans. Fundamentals*, vol.E94-A, no.12, pp.2545–2553, Dec. 2011.



Hiroaki Konoura received B.E. and M.E. degrees in Information Systems Engineering from Osaka University, Japan, in 2009 and 2011, respectively. He is currently a doctoral student in the Department of Information Systems Engineering at Osaka University. His research interest is development of high reliable reconfigurable architecture. He is a student member of IEEE.



Toshihiro Kameda received the B.E. and M.E. degrees in Information Systems Engineering from Osaka University, Osaka, Japan, in 2010 and 2012, respectively. He is currently with NTT communications. His research interest includes testing and fault avoidance for reconfigurable devices.



Yukio Mitsuyama received B.E., M.E., and Ph.D. degrees in Information Systems Engineering from Osaka University, Japan, in 1998, 2000, and 2010, respectively. He is currently an Associate Professor in School of Engineering, Kochi University of Technology. His research interests include reconfigurable architecture and its VLSI design. He is a member of IEEE and IPSJ.



Masanori Hashimoto received the B.E., M.E. and Ph.D. degrees in Communications and Computer Engineering from Kyoto University, Kyoto, Japan, in 1997, 1999, and 2001, respectively. Since 2004, he has been an Associate Professor in Department of Information Systems Engineering, Graduate School of Information Science and Technology, Osaka University. His research interest includes computer-aided-design for digital integrated circuits, and high-speed circuit design. Dr. Hashimoto served on the technical program committees for international conferences including DAC, ICCAD, ITC, ASP-DAC, DATE, ISPD, and Symposium on VLSI Circuits. He is a member of IEEE, ACM, and IPSJ.



Takao Onoye received the B.E. and M.E. degrees in Electronic Engineering, and Dr.Eng. degree in Information Systems Engineering all from Osaka University, Japan, in 1991, 1993, and 1997, respectively. He is currently a professor in the Department of Information Systems Engineering, Osaka University. His research interests include media-centric low-power architecture and its SoC implementation. He is a member of IEEE, IPSJ, and ITE-J.