PAPER Special Section on VLSI Design and CAD Algorithms

Stress Probability Computation for Estimating NBTI-Induced Delay Degradation

Hiroaki KONOURA[†,††a)], *Student Member*, Yukio MITSUYAMA[†††,††b)], Masanori HASHIMOTO[†,††c)], *and* Takao ONOYE[†,††d)], *Members*

SUMMARY PMOS stress (ON) probability has a strong impact on circuit timing degradation due to NBTI effect. This paper evaluates how the granularity of stress probability calculation affects NBTI prediction using a state-of-the-art long term prediction model. Experimental evaluations show that the stress probability should be estimated at transistor level to accurately predict the increase in delay, especially when the circuit operation and/or inputs are highly biased. We then devise and evaluate two annotation methods of stress probability to gate-level timing analysis; one guarantees the pessimism desirable for timing analysis and the other aims to obtain the result close to transistor-level timing analysis. Experimental results show that gate-level timing analysis with transistor-level stress probability calculation estimates the increase in delay with 12.6% error.

key words: *NBTI, stress probability, timing analysis*

1. Introduction

In nanoscale integrated circuits design, negative bias temperature instability (NBTI) is one of the serious concerns on device reliability. NBTI is the degradation effect which causes gradual V_{th} degradation. When ΔV_{th} is defined as $\Delta V_{th} = V_{th\_aged} - V_{th\_fresh}$, ΔV_{th} of PMOS is a negative value and $|\Delta V_{th}|$ gradually increases while a negative bias is applied to PMOS, i.e. $V_{gs} = -V_{dd}$, where V_{th_aged} and V_{th_fresh} are threshold voltages of aged and fresh MOS transistors. This condition is defined as stress phase of NBTI. When NBTI stress continues for a long time, path delay increases, which may lead to a timing error. On the other hand, while PMOS is OFF, i.e. $V_{gs} = 0$, $|\Delta V_{th}|$ gradually decreases to its former value before stress impression, and PMOS degradation is relaxed. This condition is defined as recovery phase of NBTI. Repeating stress and recovery cycles, degradation caused by NBTI increases and could finally result in a timing error [1].

In order to predict NBTI effect, a long term prediction model has been proposed [1], [2]. The degradation by NBTI depends on operational parameters such as temperature, supply voltage, and stress probability [3], [4]. Stress probability is defined as time ratio of stress phase, that is (time of stress)/(time of stress + time of recovery). When the stress probability is almost 100% and PMOS is under stress for a long time, an increase in path delay becomes extremely large due to large V_{th} shift.

For accurate prediction of circuit delay degradation, appropriate stress probability estimation and consideration in timing analysis are crucially important. Reference [5] proposed a transistor-level estimation method of stress probability. Also, gate-level estimation of path delay using NBTI aware static timing analysis is proposed in [6]. Thus, there are several proposals with different granularities. However, the importance of the granularity (circuit/instance/transistor) in probability estimation and ΔV_{th} annotation for timing analysis has not been sufficiently discussed. Stress probabilities can be highly biased depending on circuit operation, and in such cases, stress probability computation with finer granularity might be necessary.

In this paper, we experimentally evaluate how the granularities (circuit/instance/transistor) of stress probability computation affects the estimation accuracy of NBTI-induced timing degradation through case studies. Our evaluation reveals that transistor-level probability computation is indispensable in cases of highly-biased circuit operations. On the other hand, transistor-level timing analysis with transistor-by-transistor ΔV_{th} variation is less compatible with gate-level timing analysis of industrial practice and it is computationally expensive. We therefore propose gate-level timing analysis in which instance-by-instance ΔV_{th} is annotated according to transistor-level stress probability estimation. Experimental results show that the proposed method significantly improves the estimation accuracy of timing degradation from timing analysis with gate-level probability computation, though the timing analysis is carried out at gate-level.

The rest of this paper is organized as follows. The prediction model of NBTI used in this work is described in Sect. 2. Section 3 presents three methods of stress probability calculation, and clarifies the importance of the granularity in stress probability computation. After that, Sect. 4 proposes instance-by-instance ΔV_{th} annotation for gate-level timing analysis, and shows its effectiveness for estimating NBTI-induced timing degradation. Finally, the discussion is concluded in Sect. 5.

Manuscript received March 26, 2011.
Manuscript revised June 20, 2011.

[†] 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.E94.A.2545

2. Prediction Model of NBTI Effect

Various prediction models of NBTI has been proposed in [1], [7]–[10]. In this paper, we use a long term prediction model which is useful for estimation of gradual V_{th} degradation by years on a basis of the stress and recovery cycles [1], [2]. With this model, path delays monotonically increase depending on V_{th} degradation, since the model gives estimates of monotonic V_{th} degradation from a long-term point of view. V_{th} degradation after time t has passed ($|\Delta V_{th}|$) is expressed as Eq. (1).

$$|\Delta V_{th}| = \left(\frac{\sqrt{K_v^2 \cdot T_{clk} \cdot \alpha}}{1 - \beta_t^{1/2n}} \right)^{2n}. \qquad (1)$$

In Eq. (1), K_v is a parameter depending on supply voltage and temperature. T_{clk} is clock period, and α is stress probability of PMOS. β_t is a parameter that has a dependence on temperature, T_{clk}, α, and t. Moreover, n is equal to 1/6 in a hydrogen molecule diffusion based model [11]. All of these parameters are important for the progress of NBTI degradation [12].

References [1], [2] reported that there is little relation between $|\Delta V_{th}|$ and T_{clk} [1], [2] when the operating frequency is higher than 100 Hz. In this case, Eq. (2) is used for $|\Delta V_{th}|$ estimation instead of Eq. (1).

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

In Eq. (2), parameter C is dependent on temperature, and t_{ox} denotes gate oxide thickness. Here, in Eq. (2), when α approaches to 1, $|\Delta V_{th}|$ 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].

$$|\Delta V_{th}| = \left(K_v^2 t \right)^n. \qquad (3)$$

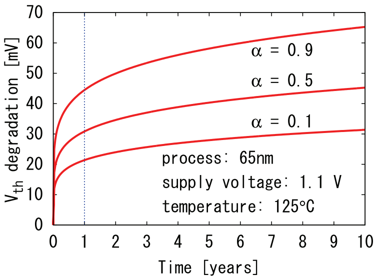


Fig. 1 Threshold voltage degradation predicted with long term model[1].

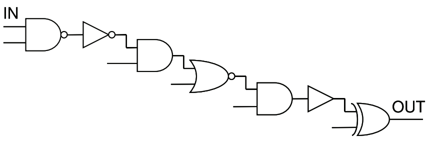


Fig. 2 An example of combinational circuit.

Let us show an example of NBTI degradation. Figure 1 shows V_{th} degradation calculated with Eq. (2), referring a parameter set of 65 nm process in [12]. A significant V_{th} degradation can be found in the first year. In addition, we can see the dependency of V_{th} degradation on α. We also applied this V_{th} degradation to all PMOSs in a small combinational circuit shown in Fig. 2. After ten years, the critical path delay increases by 4.3% ($\alpha = 0.1$), 6.7% ($\alpha = 0.5$), 11.0% ($\alpha = 0.9$), respectively. Thus, the increase in path delay depends on α.

3. Granularity of Stress Probability Computation

This section introduces three methods of stress probability computation with different granularity levels. Then, the timing analysis results based on these three methods are compared.

3.1 Stress Probability Computation

We use three stress probability calculation methods (SPCM) with different granularities as follows.

SPCM-A (Circuit level): Set stress probability of all PMOSs to 50% uniformly.
SPCM-B (Instance level): Set stress probability of all PMOSs in an instance to the state probability of the instance output. State probability means the probability of a node being high.
SPCM-C (Transistor level): Calculate stress probability for each PMOS.

SPCM-A is the simplest method, which is based on an assumption that each part of a circuit works uniformly. Reference [14] reported that 50% is a reasonable value to instantly conjecture the maximum timing degradation. In SPCM-B, the stress probability of PMOSs in an instance of standard cell is assumed to be identical to the state probability of the instance output, which can be obtained by logic simulation.

Note that other methods of state probability computation except logic simulation, such as probability propagation [15], could be used for stress probability estimation, though in this paper logic simulation is adopted as a representative method. In SPCM-C, the stress probability computation of each PMOS is individually calculated.

Table 1 lists features of SPCMs. In Table 1, with SPCM-A, logic simulation is not required. In SPCM-B, logic simulation is required, while connection of all transistors is not taken into consideration. In SPCM-C, not only logic simulation but also consideration of connection of all transistors are required. In these methods, SPCM-C is the

Table 1 Feature of each SPCM.

	logic sim.	connection of all trs.
SPCM-A	-	-
SPCM-B	required	-
SPCM-C	required	considered

finest-grained calculation method and is expected to obtain the most accurate stress probability.

In this work, with SPCM-C, stress probabilities computation is executed on the basis of logic simulation results, which means the state probabilities of all the nets are given. The computation procedure is:

1. For each cell instance, examine whether each PMOS is under stress or not for every combination of input states.
2. Calculate the probabilities of all the combinations of input states at each instance by using the state probabilities of the nets. We here call this probability as combinatorial probability.
3. Obtain the stress probability of each PMOS using the above two information.

As an example, we apply SPCM-C to 2-input AND shown in Fig. 3. Supposing the state probabilities of inputs A and B are 0.7 and 0.1 respectively, the combinatorial probabilities of {0, 0}, {0, 1}, {1, 0}, and {1, 1} are listed in Table 2. Here, the correlation between inputs is not considered in the combinatorial probability computation within an instance, though the state probabilities of inputs A and B are derived taking into account the logical correlation through logic simulation. Then, we examine whether P1, P2, and P3 are under stress for each input combination. Finally from Table 2, the stress probability of P1 is expressed as a summation of the combinational probabilities, of which P1 is under stress, $\alpha = 0.30 \times 0.90 + 0.30 \times 0.10 = 0.30$.

The stress probabilities of 2-input NAND and AND gates calculated by each SPCM using an example of Fig. 3 are listed in Tables 3 and 4. We first investigate the NAND gate having PMOSs in parallel. In Table 3, comparing SPCM-B and -C, the stress probabilities of P1 are significantly different though those of P2 are similar. SPCM-B cannot well cope with a large difference of input state probability. Moreover, another problem arises with SPCM-B when analyzing the AND gate. In Table 4, using the state probability of the instance output, the stress probability of P3 in the second stage can be well estimated. However, the stress probabilities of P1 and P2 in the first stage cannot be well estimated, because the circuit structure is not considered in SPCM-B. Thus, each method gives different stress probabilities. The impact of this probability difference will be investigated in the next section from a timing degradation point of view.

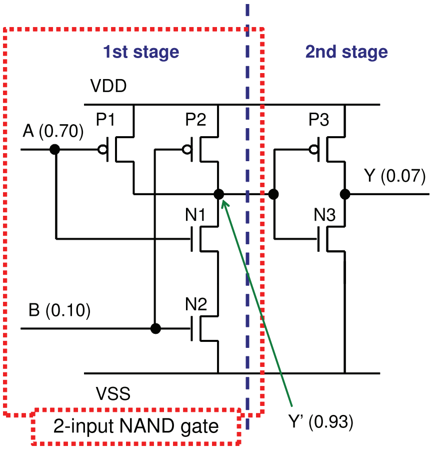


Fig. 3 2-input AND gate (including 2-input NAND gate). Values noted at input and output terminals are state probabilities of being 1.

Table 2 Combinatorial probability and PMOS stress condition.

{A, B}	combinatorial probability	P1	P2	P3
{0, 0}	0.30 × 0.90	stress	stress	-
{0, 1}	0.30 × 0.10	stress	-	-
{1, 0}	0.70 × 0.90	-	stress	-
{1, 1}	0.70 × 0.10	-	-	stress
stress probability		0.30	0.90	0.07

3.2 Experimental Setup

3.2.1 Circuits for Evaluation

In this paper, double floating point unit (D-FPU) and symmetric-key cipher algorithm AES [16] are adopted as target circuits for evaluation. The specifications and implemented results of D-FPU and AES using an industrial 65 nm standard cell library are summarized in Tables 5 and 6, respectively. Input and output operands of D-FPU are 64-bit, which consists of 1 bit sign, 11-bit exponent, and 52-bit mantissa. On the other hand, AES performs encryption of 128-bit plaintext with 128-bit key, and outputs 128-bit ciphertext.

In this work, we assume three operating situations be-

Table 3 Result of stress probability estimation (2-input NAND gate).

	P1	P2
SPCM-A [%]	50.0	50.0
SPCM-B [%]	93.0	93.0
SPCM-C [%]	30.0	90.0

Table 4 Result of stress probability estimation (2-input AND gate).

	P1	P2	P3
SPCM-A [%]	50.0	50.0	50.0
SPCM-B [%]	7.0	7.0	7.0
SPCM-C [%]	30.0	90.0	7.0

Table 5 D-FPU specification.

input/output operand	64 bits
function	four arithmetic operations
operating frequency	167 MHz
gate count	72.7 k

Table 6 AES specification.

key length	128 bits
block size	128 bits
function	encryption
operating frequency	500 MHz
gate count	20.0 k

low for D-FPU to consider the fact that stress probability depends on circuit operation.

(a) general operation with random data
Both operation and operand are randomly selected.
(b) biased operation with random data
Only addition is executed.
(c) general operation with biased data
Test vectors are generated so that operands whose 8-bit of 11-bit exponent and 40-bit of 52-bit mantissa were fixed to 0.

As for AES, the difference originating from operating conditions was limited and hence only general operation with random data will be shown in this paper. For each situation, 100,000 test vectors were prepared for D-FPU. On the other hand, for AES, 100 keys and 1,000 plain texts were prepared.

3.2.2 Procedure of Timing Degradation Analysis

In order to evaluate timing degradation due to NBTI, it is necessary to annotate ΔV_{th} to timing analysis. To annotate ΔV_{th} transistor by transistor, we adopted a commercial transistor-level static timing analyzer (Synopsys NanoTime [17]) in this work. To eliminate estimation difference originating from tool and library difference, the same timing analyzer was used for all SPCMs.

Figure 4 illustrates the overall procedure adopted for timing degradation analysis. Firstly, the state probabilities at input/output terminals of all instances are calculated from logic simulation results using given test vectors. Secondly, the stress probabilities of all PMOSs are calculated by SPCM-A through -C. After that, the stress probability for each PMOS is converted to ΔV_{th} referring the long term model (Eq. (2)), and it is applied to the transistor-level netlist. Finally, using the netlist including information of ΔV_{th}, we perform static timing analysis. In addition to these three methods, we define DC-stress, in which stress probability of all PMOSs is 100%, and evaluate the timing degradation for it. In this situation, Eq. (3) is used for ΔV_{th} prediction.

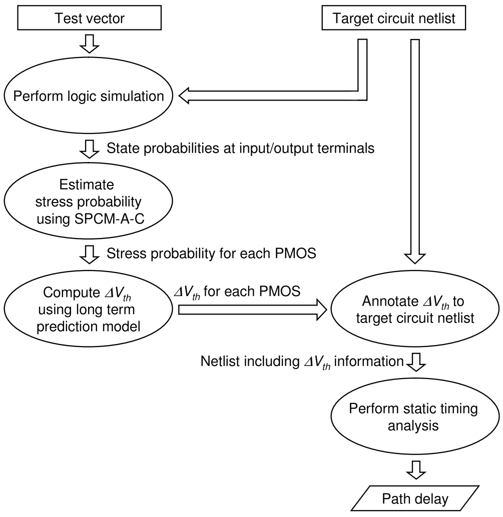

Test vector
Target circuit netlist
Perform logic simulation
State probabilities at input/output terminals
Estimate stress probability using SPCM-A-C
Stress probability for each PMOS
Compute ΔV_{th} using long term prediction model
ΔV_{th} for each PMOS
Annotate ΔV_{th} to target circuit netlist
Netlist including ΔV_{th} information
Perform static timing analysis
Path delay

Fig. 4 Estimation procedure of NBTI-induced circuit degradation.

3.3 Stress Probability Distribution for All PMOSs

We first examine the difference in stress probability distributions for three operating situations.

Stress probability distributions calculated by SPCM-B and -C for all PMOSs of D-FPU and AES are illustrated in Fig. 5 and Fig. 6, respectively. Note that PMOSs which compose feedback not related to timing in DFFs are excluded. In Fig. 5 and Fig. 6, samples of 0%/100% stress probability are

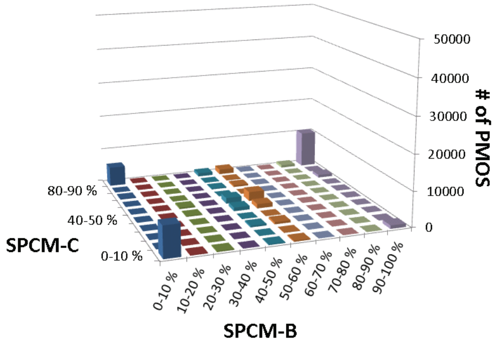
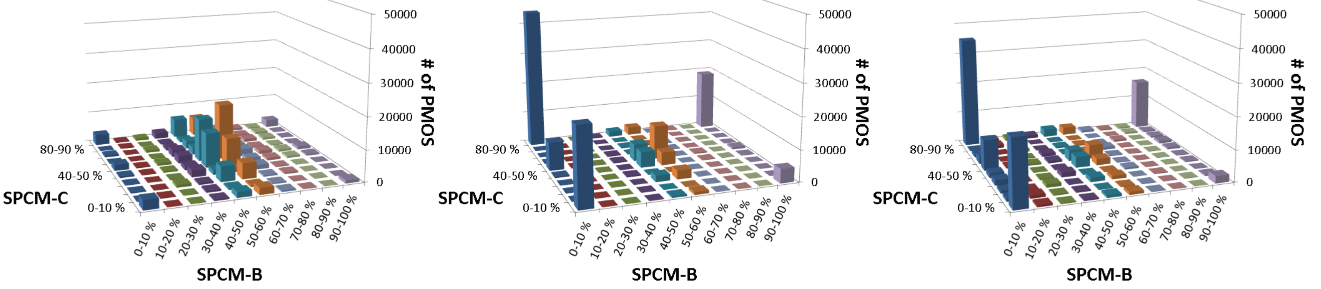


Fig. 6 Stress probability distribution (AES, SPCM-B vs. SPCM-C).



(a) General operation with random data.
(b) Biased operation with random data.
(c) General operation with biased data.

Fig. 5 Stress probability distributions (D-FPU, SPCM-B vs. SPCM-C).

Table 7 Critical path delay degradation estimated with each SPCM (D-FPU, 3 years later).

		critical path delay (degradation) [ns]				
	temperature [°C]	initial	DC-stress	SPCM-A	SPCM-B	SPCM-C
(a) general operation with random data	25	3.79	4.57 (0.77)	3.99 (0.20)	4.00 (0.20)	3.97 (0.18)
	75	3.96	5.27 (1.95)	4.27 (0.60)	4.27 (0.61)	4.25 (0.30)
	125	4.15	6.10 (1.95)	4.75 (0.60)	4.76 (0.61)	4.71 (0.57)
(b) biased operation with random data	25	3.79	4.57 (0.77)	3.99 (0.20)	4.00 (0.20)	3.97 (0.18)
	75	3.96	5.27 (1.31)	4.27 (0.31)	4.26 (0.31)	4.33 (0.38)
	125	4.15	6.10 (1.95)	4.75 (0.60)	4.76 (0.61)	5.02 (0.87)
(c) general operation with biased data	25	3.79	4.57 (0.77)	3.99 (0.20)	4.00 (0.21)	4.28 (0.48)
	75	3.96	5.27 (1.31)	4.27 (0.31)	4.27 (0.31)	4.91 (0.95)
	125	4.15	6.10 (1.95)	4.75 (0.60)	4.75 (0.60)	5.69 (1.54)

Table 8 Critical path delay degradation with each SPCM (AES, 3 years later).

		critical path delay (degradation) [ns]				
	temperature [°C]	initial	DC-stress	SPCM-A	SPCM-B	SPCM-C
general operation with random data	25	1.14	1.33 (0.19)	1.18 (0.04)	1.19 (0.05)	1.21 (0.07)
	75	1.20	1.53 (0.33)	1.29 (0.09)	1.29 (0.09)	1.34 (0.15)
	125	1.25	1.76 (0.51)	1.43 (0.18)	1.44 (0.19)	1.51 (0.26)

included in the range of below 10%/over 90%.

In Fig. 5(a), with both SPCM-B and -C, stress probabilities of PMOSs are largely estimated as around 50%. There is a small number of PMOS transistors whose stress probability is around 100%. In contrast, as shown in Figs. 5(b), (c), there are many PMOSs whose stress probability is nearly either 0% or 100%. Moreover, a lot of stress probabilities of PMOSs which are estimated as over 90% in SPCM-C are misestimated as below 10% in SPCM-B. When operation or data is biased, the active circuit portion is limited, and a number of nets are fixed 0 or 1. In those cases, there is a large difference in stress probability distributions between the SPCMs.

As for AES, in Fig. 6, although plain texts and keys are randomly given, there are a lot of PMOSs whose stress probabilities are under 10% or over 90%. AES has twenty substitution-boxes whose outputs are fixed to 0 or 1. Therefore, in AES, stress probability distributions are inherently biased toward both 0% and 100%.

3.4 Degradation of Critical Path Delay

Following the procedure in Fig. 4, delay increase of critical path is here estimated for each operating situation.

(1) General Operation with Random Data

Table 7(a) and Table 8 show the evaluation results of NBTI degradation after 3 years in D-FPU and AES, respectively. In D-FPU, we can see only small differences of delay increase among SPCM-A through -C. On the other hand, in AES, between SPCM-C and a group of SPCM-A and -B, the error of delay increase at 125°C is 26.9% (= (0.26 − 0.19)/0.26). It is supposed that this difference originates from the difference in a proportion of PMOSs whose stress probability is estimated at either 0% or 100% shown in Fig. 5(a) and Fig. 6.

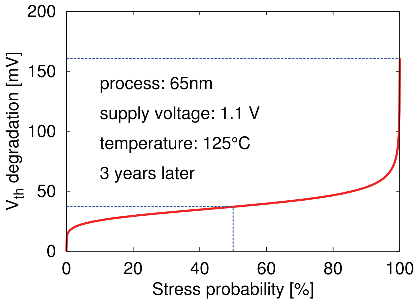


Fig. 7 V_{th} degradation versus stress probability of PMOS (65 nm, 1.1 V, 125°C, 3 years later).

(2) Biased Operation with Random Data

Table 7(b) shows the evaluation results, where D-FPU executes only addition. Comparing with Table 7(a), the amount of delay increase at 125°C in SPCM-C becomes 1.5 (= 0.87/0.57) times larger. In Table 7(b), between SPCM-C and a group of SPCM-A and -B, the error of delay increase at 125°C reaches 31.0% (= (0.87 − 0.60)/0.87). This error comes from the difference in Fig. 5(b).

(3) General Operation with Biased Data

Table 7(c) shows the result of D-FPU. Compared with Table 7(a), the amount of delay increase is 2.7 (= 1.54/0.57) times larger than that at 125°C in SPCM-C. In Table 7(c), between SPCM-C and a group of SPCM-A and -B, the error of delay increase at 125°C reaches 61.0% (= (1.54 − 0.60)/1.54) due to the mismatch in Fig. 5(c).

Meanwhile, Fig. 7 shows the V_{th} degradation predicted with Eqs. (2) and (3) as a function of stress probability assuming 125°C and 3-year operation. The stress probability difference between 0% and 50% causes 37 mV V_{th} difference. On the other hand, the difference between 50% and 100% corresponds to 124 mV shift indeed. As the stress probability approaches 100%, V_{th} degradation drastically increases. In SPCM-A, the stress probability is never estimated as 100%, which can result in an optimistic estimation. Furthermore, with SPCM-B, since the number of PMOSs whose stress probabilities are close to 100% is fewer than SPCM-C as described with Fig. 5 and Fig. 6, the estimate of path delay degradation tends to be smaller. In this way, timing degradation in an inactive circuit are often underestimated when the stress probability is estimated with circuit and instance level granularities. From another point of view, in an inactive circuit, the stress probabilities are biased toward not only 100% but also 0%, because a CMOS digital circuit basically consists of inverting logic gates, and therefore a timing analysis assuming DC-stress for all PMOSs provides extremely pessimistic results. For these reasons, the stress probability should be estimated for each PMOS.

4. Instance-by-Instance ΔV_{th} Annotation for Gate-Level Timing Analysis

Section 3 clarified that transistor-level stress probability computation is necessary for accurate timing degradation analysis. However, transistor-level timing analysis, which is necessary for transistor-by-transistor ΔV_{th} consideration, is less compatible with gate-level timing verification of industrial practice for large SoC designs. This is because ordinary gate delay models cannot give gate delay coping with transistor-by-transistor ΔV_{th} variation, though ΔV_{th} variation uniformly applied to all PMOSs is often considered in cell library characterization.

This section discusses how to exploit the transistor-level stress probability information in gate-level timing analysis, aiming to obtain the estimation results close to those by transistor-level timing analysis.

4.1 ΔV_{th} Annotation to Each Instance

For the purpose above, we introduce two instance-by-instance annotation method (IAMs) of $|\Delta V_{th}|$ to each instance.

IAM-I: Choose and annotate the largest PMOS $|\Delta V_{th}|$ within the instance of interest. This annotation always provides pessimistic results, which is a desirable property for static timing analysis.

IAM-II: Choose and annotate the largest $|\Delta V_{th}|$ of PMOS on the most timing-critical path within the instance of interest. It is expected to estimate the timing degradation close to that by transistor-by-transistor annotation, because $|\Delta V_{th}|$s which are large yet less related to the circuit critical path are not annotated. Therefore, this annotation cannot guarantee the pessimism, but helps eliminate the excessive pessimism of IAM-I.

Let us apply IAM-I and -II to 2-input AND shown in Fig. 8 as an example. With IAM-I, 42.6 mV is the largest $|\Delta V_{th}|$ of PMOSs in this instance and we annotate it as the $|\Delta V_{th}|$ of this instance. On the other hand, with IAM-II, $|\Delta V_{th}|$ of P1 (32.2 mV) on the most timing-critical path is selected. This difference between 42.6 mV and 32.2 mV corresponds to reduction in the pessimism of estimated timing degradation.

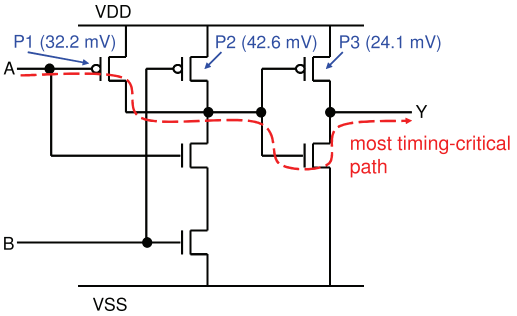


Fig. 8 2-input AND gate. Values shown in parentheses are $|\Delta V_{th}|$ of each PMOS.

4.2 Experimental Results of Timing Degradation

We evaluated the timing degradation of D-FPU with IAM-I and -II using transistor-level stress probabilities obtained by SPCM-C. The evaluation setup is the same with Sect. 3.3.

Table 9 shows the evaluation results of NBTI-induced timing degradation after 3 years in D-FPU. In this table, timing degradation estimated by transistor-by-transistor annotation method (TAM), which is equivalent to SPCM-C in Table 7, in addition to that under DC-stress are listed as well.

With IAM-I, the amount of delay increase is larger than that with TAM shown in Table 9(a) through Table 9(c), which means IAM-I preserved the pessimism in estimation as expected. The introduced pessimism is the largest in biased operation with random data (Table 9(b)), and the amount of delay increase at 125°C in IAM-I is 2.2 (= 1.93/0.87) times larger than that with TAM.

On the other hand, with IAM-II, the amount of delay increase is close to that with TAM under Table 9(a) through Table 9(c), though a small optimism is introduced. In Table 9(b), the error of delay increase at 125°C estimated with IAM-II is 12.6% (= (0.87 − 0.76)/0.87), which is smaller than those of SPCM-A and SPCM-B (31.0% and 29.9%).

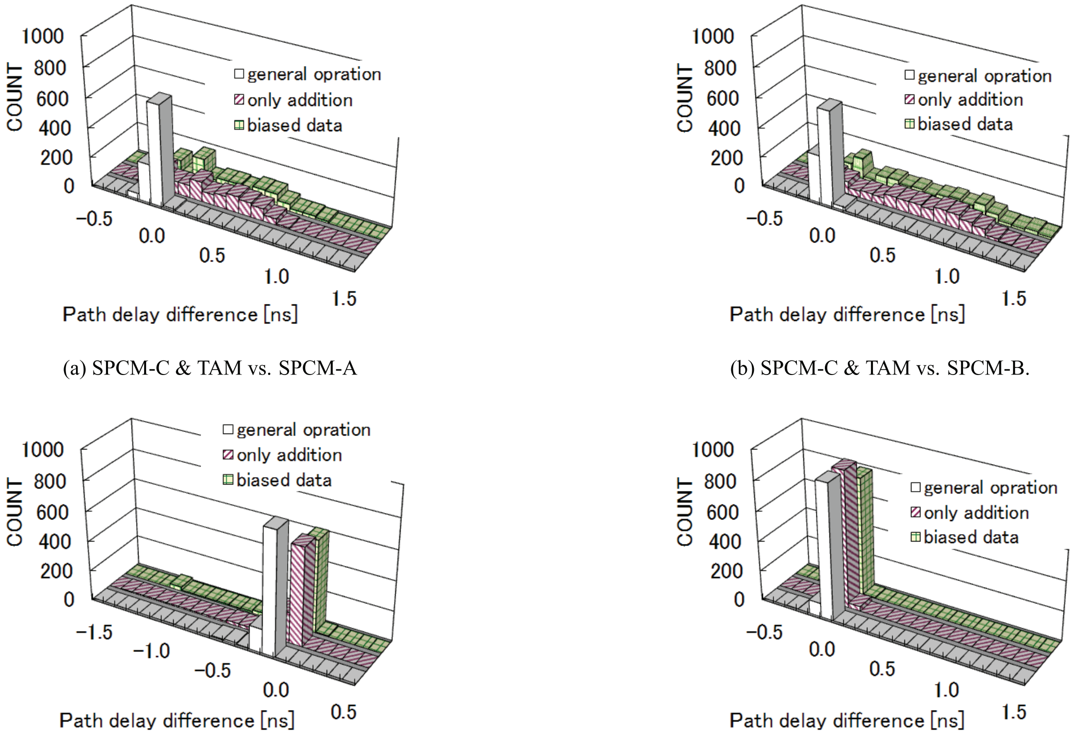
4.3 Discussion

We have discussed the granularity of stress probability computation and ΔV_{th} annotation, and now have five estimation methods of NBTI-induced timing degradation; SPCM-A, SPCM-B, SPCM-C & TAM, SPCM-C & IAM-I and SPCM-C & IAM-II. Here, we summarize the features of each method regarding SPCM-C & TAM as reference.

Figure 9 gives histograms of difference between path delays estimated by SPCM-C & TAM and SPCM-A, SPCM-C & TAM and SPCM-B, SPCM-C & TAM and SPCM-C & IAM-I, and SPCM-C & TAM and SPCM-C & IAM-II, respectively. Here, top 1,000 paths in delay are selected in the initial circuit for evaluation.

Table 9 Critical path delay degradation estimated with TAM, IAM-I and IAM-II (D-FPU, 3 years later).

		critical path delay (degradation) [ns]				
	temperature [°C]	initial	DC-stress	TAM	IAM-I	IAM-II
(a) general operation with random data	25	3.79	4.57 (0.77)	3.97 (0.18)	4.02 (0.23)	3.97 (0.18)
	75	3.96	5.27 (1.31)	4.25 (0.30)	4.44 (0.48)	4.23 (0.27)
	125	4.15	6.10 (1.95)	4.71 (0.57)	4.94 (0.79)	4.69 (0.54)
(b) biased operation with random data	25	3.79	4.57 (0.77)	3.97 (0.18)	4.55 (0.76)	3.97 (0.18)
	75	3.96	5.27 (1.31)	4.33 (0.38)	5.25 (1.29)	4.29 (0.33)
	125	4.15	6.10 (1.95)	5.02 (0.87)	6.08 (1.93)	4.90 (0.76)
(c) general operation with biased data	25	3.79	4.57 (0.77)	4.28 (0.48)	4.42 (0.62)	4.26 (0.46)
	75	3.96	5.27 (1.31)	4.91 (0.95)	5.04 (1.09)	4.86 (0.90)
	125	4.15	6.10 (1.95)	5.69 (1.54)	5.80 (1.65)	5.56 (1.41)



(a) SPCM-C & TAM vs. SPCM-A

(b) SPCM-C & TAM vs. SPCM-B.

(c) SPCM-C & TAM vs. SPCM-C & IAM-I.

(d) SPCM-C & TAM vs. SPCM-C & IAM-II.

Fig. 9 Distribution of differences of path delay (D-FPU).

In Figs. 9(a) and 9(b), path delay differences both between SPCM-C & TAM and SPCM-A, and SPCM-C & TAM and SPCM-B are close to 0 ns when four arithmetic operations are done randomly.

These results demonstrate that, when all functions are randomly performed, SPCM-A and -B are effective for delay prediction as shown in Table 7. Meanwhile, in case that operation or operand is biased, the accuracies of SPCM-A and -B significantly degrade and the error reaches 2.0 ns. We can find that instance-level stress probability computation might induce unacceptable error in timing estimate, even though actual input patterns are used for evaluation.

On the other hand, as shown in Fig. 9(c) and Fig. 9(d), path delay differences between SPCM-C & TAM and SPCM-C & IAM-I/II are small even if the operation or data is biased.

NBTI V_{th} degradation models have a certain amount of error, and it causes delay estimation uncertainty. If the uncertainty is much larger than the difference of delay degradation discussed above, the accurate stress probability computation and annotation becomes less important. On the other hand, the models are still actively studied, and hence a typical value of their accuracy is not available. We therefore show, as a reference data, V_{th} degradation values that correspond to the difference of estimated delay degradation in Table 10. This degradation values were obtained by changing V_{th} of all PMOSs in the circuit uniformly so that the critical path delay became the same with one of the estimated delay

Table 10 Critical path delays and corresponding V_{th} degradation (D-FPU, general operation with biased data, 125°C).

	initial	DC-stress	SPCM-A	SPCM-B	SPCM-C & TAM	SPCM-C & IAM-I	SPCM-C IAM-Ⅱ
critical path delay [ns]	4.15	6.10	4.75	4.75	5.69	5.80	5.56
corresponding V_{th} degradation [mV]	0	161	68	68	133	140	124

Table 11 Comparison of CPU times for NBTI-aware timing analysis (D-FPU, general operation with random data, Inter(R) Xeon(R) CPU X5680 @ 3.33 GHz).

	CPU time [s]				
	SPCM-A	SPCM-B	SPCM-C & TAM	SPCM-C & IAM-I	SPCM-C & IAM-Ⅱ
logic simulation	—	6,747			
stress probability calculation & $\lvert\Delta V_{th}\rvert$ annotation	1	1	135	101	151
preliminary tr.-level timing analysis	—	—	—	—	570
tr.-level timing analysis	647	1,103	1,078	970	943
total execution time	648	7,851	7,960	7,818	8,411

degradations. In Table 10, the difference of V_{th} degradation between SPCM-C & TAM and a group of SPCM-A and -B is significantly large (= 65 mV). Meanwhile, comparing SPCM-C & TAM with SPCM-C & IAMs, the differences of V_{th} degradation are 9 mV at most. Taking into account the accuracy of NBTI V_{th} degradation model, we will need to select an appropriate estimation method of timing degradation.

These results indicate that the combination of SPCM-C and IAMs gives reasonably accurate estimation in NBTI-induced timing degradation using an ordinary framework of gate-level timing analysis.

Finally, Table 11 shows the CPU times for operations accompanied with each SPCM and each annotation method. In this work, stress probability calculation and $|\Delta V_{th}|$ annotation were executed by ruby scripts, and hence CPU time could be reduced by implementing with other languages, such as C and C++. In Table 11, logic simulation notably consumes larger CPU time than other operations. Because of not performing logic simulation in SPCM-A, there is a large difference in total execution time between SPCM-A and a group of SPCM-B and -C. Furthermore, with SPCM-C & IAM-Ⅱ, preliminary transistor-level timing analysis is required to obtain the order of paths in delay, which is referred in the annotation process. Owing to this extra timing analysis, CPU time with SPCM-C & IAM-Ⅱ is certainly larger than those with SPCM-C & TAM/IAM-I, while this CPU time increase is still much smaller than that of logic simulation.

5. Conclusion

This paper evaluated how much stress probability consideration in estimation of NBTI-induced delay degradation impacts the accuracy, focusing on analysis granularity. Stress probability calculation was performed for two circuits at three granularity levels: circuit-level, instance-level, and transistor-level. Evaluation results showed that considerable path delay difference may arise even though instance-level stress probability calculation is performed. We demonstrated that distribution of stress probabilities heavily depends on operation of circuits and affect prediction of path delay. Moreover, in order to enable gate-level timing analysis even with transistor-level stress probability calculation, instance-by-instance $|\Delta V_{th}|$ annotation is considered. Using the proposed $|\Delta V_{th}|$ annotation, accurate prediction of delay degradation due to NBTI can be performed with ordinary gate-level timing analysis.

Acknowledgement

The authors would like to thank the project members of JST CREST of Kyoto University, Kyoto Institute of Technology, Nara Institute of Science and Technology, and ASTEM RI for their discussions. This study was partly supported by NEDO.

References

[1] 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.

[2] W. Wang, S. Yang, S. Bhardwaj, S. Vrudhula, F. Liu, and Y. Cao, “The impact of NBTI effect on combinational circuit: modeling, simulation, and analysis,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol.18, no.2, pp.173–183, Feb. 2010.

[3] W. Wang, S. Yang, S. Bhardwaj, R. Vattikonda, S. Vrudhula, F. Liu, and Y. Cao, “The impact of NBTI on the performance of combinational and sequential circuits,” Proc. ASP-DAC, pp.364–369, Jan. 2007.

[4] B. Zhang and M. Orshansky, “Modeling of NBTI-induced PMOS degradation under arbitrary dynamic temperature variation,” Proc. ISQED, pp.748–779, March 2008.

[5] A. Stempkovsky, A. Glebov, and S. Gavrilov, “Calculation of stress probability for NBTI-aware timing analysis,” Proc. ISQED, pp.714–718, March 2009.

[6] W. Wang, S. Yang, and Y. Cao, “Node criticality computation for circuit timing analysis and optimization under NBTI effect,” Proc. ISQED, pp.763–768, March 2008.

[7] S. Mahapatra, M.A. Alam, P. Bharath, T.R. Dalei, and D. Saha, “Mechanism of negative bias temperature instability in CMOS devices: Degradation, recovery and impact of nitrogen,” Proc. IEDM,

pp.105–108, Dec. 2004.

[8] M.A. Alam and S. Mahapatra, “A comprehensive model of PMOS NBTI degradation,” Microelectronics Reliability, vol.45, pp.71–81, 2005.

[9] 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 2008.

[10] T. Grasser, B. Kachzr, W. Goes, T. Aichinger, P. Hehenberger, and M. Nelhiebel, “A two-stage model for negative bias temperature instability,” Proc. IRPS, pp.33–44, April 2009.

[11] A.T. Krishnan, C. Chansellor, 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] 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 and Material Reliability, vol.7, no.4, pp.509–517, Dec. 2007.

[13] Y. Cao, “Reliability mechanisms and the impact on IC designs,” Proc. ASP-DAC, Tutorial 4, pp.342–372, Jan. 2009.

[14] W. Wang, Z. Wei, S. Yang, and Y. Cao, “An efficient method to identify critical gates under circuit aging,” Proc. ICCAD, pp.735–740, Sept. 2007.

[15] F.N. Najm, “Transition density, a stochastic measure of activity in digital circuits,” Proc. DAC, pp.644–649, June 1991.

[16] Open Cores, “http://www.opencores.org/”

[17] Synopsys, Inc.: NanoTime and NanoTime Ultra User Guide, June 2009.

Hiroaki Konoura received the B.E. degree in Information Systems Engineering from Osaka University, Osaka, Japan, in 2009. He is currently pursuing the M.E. degree in the Department of Information Systems Engineering at Osaka University. His major interest is evaluation and mitigation of circuit degradation. He is a student member of IEEE.

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. Since 2011, he has been an Assistant 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, ASP-DAC, ICCD and ISQED. He is a member of IEEE 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.