IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.a

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

High-Density SRAM Read Access Yield Estimation Methodology

Gidong Baek[1] and Hanwool Jeong[1]
[1]Kwangwoon University, Seoul, South Korea.

Corresponding author: H. Jeong (e-mail: hwjeong@kw.ac.kr).

This work was supported by Research Resettlement Fund for the new faculty of Kwangwoon University in 2019 and National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2020R1G1A1009777)

ABSTRACT As high-density SRAMs must be designed to ensure a substantially small failure rate, the accurate yield estimation with practically acceptable runtime of circuit simulations is highly challenging. Here, a read access yield estimation method for high-density static random access memory (SRAM) is proposed. Instead of performing SPICE runs for the entire SRAM circuit, the proposed method partitions the SRAM into three parts—the control signal generation circuit, bitcell array, and sense amplifier (SA)—that determine three key parameters: word-line to SA enable delay, bit-line voltage difference, and SA offset voltage. Subsequently, the proposed method derives the probability density of these key parameters from each of the three partitioned circuits. Here, different methods are applied to derive the probability of the key parameters, considering the respective characteristics of each circuit part and parameter. According to our experimental results, the proposed method can accelerate the yield estimation by 500–3000×, compared with the brute-force Monte Carlo simulation method, and 10–100× compared with the other state-of-art methods. In addition, the proposed method can accelerate the circuit optimization procedure accompanied by multiple circuit revisions, that is, the circuit revisions can be reflected with SPICE runs only for the revised circuit part, unlike the previous methods that require SPICE runs for the entire SRAM.

INDEX TERMS Process variation, read access yield, sensing yield, static random access memory (SRAM), yield estimation.

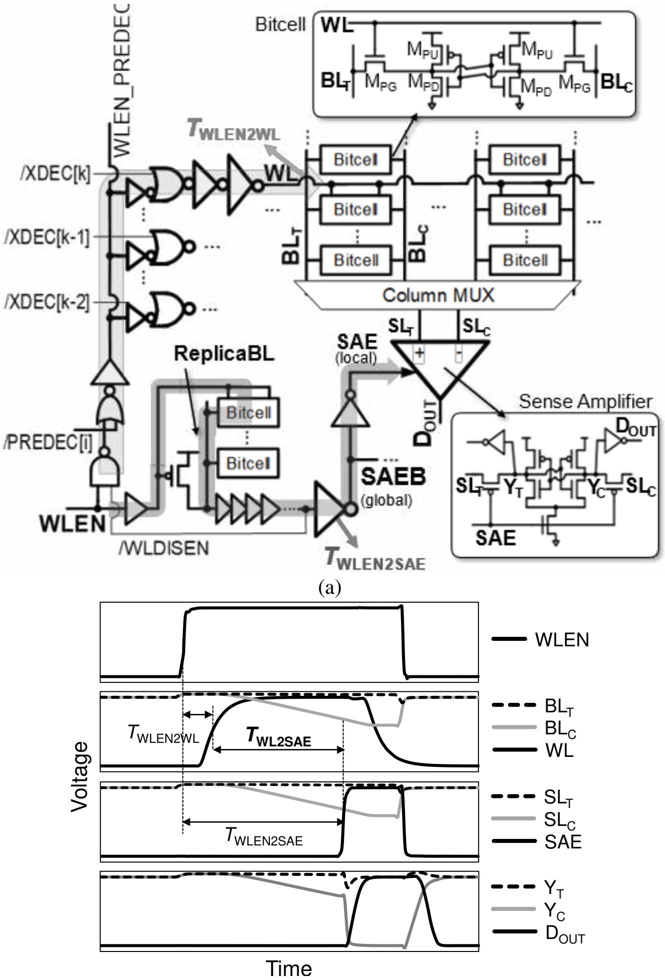
I. INTRODUCTION

Static random access memory (SRAM) is widely used as embedded memory in the recent system-on-chip (SoC) paradigm. The design of SRAM is highly important because it not only substantially affects the total power and speed of SoC but also occupies a large area. For high density integration, SRAM bitcells designed nearly minimum sized transistors, making it extremely sensitive to process variations. This means that the SRAM is highly vulnerable to operation failure, and the yield of SoC is critically determined by SRAM. Especially, the read access failure that is incorrect sensing of the stored data, is one of the most critical failures in SRAM. Thus, the SRAM design should be optimized considering the read access stability.

In several modern high-performance SoCs, millions of SRAM bitcells are implemented. Thus, a single SRAM bitcell should be designed to have an extremely low failure rate, to ensure that the entire SoC yield is within a practically acceptable range. For example, to achieve a 95% yield in a 256 Mb SRAM, the failure rate of a single SRAM bitcell, $P_{\mathrm{fail,bitcell}}$, should be less than 2×10^{-9} that can be approximated from $(1 - P_{\mathrm{fail,bitcell}})^{256M} = 95\%$. However, it is highly challenging to accurately estimate such an extremely low failure rate.

The simplest yield estimation method is the brute-force Monte Carlo (BMC) simulation. With circuit samples generated, such that the process variations are appropriately considered, circuit simulations are invoked for each sample. Thereafter, the failure rate is estimated as the fraction of the samples resulting in operation failure. The limitation of BMC is that it requires an exceedingly large number of samples for circuit simulations, when the target failure rate is extremely low. To estimate failure rate ranging from 10^{-7} to 10^{-9} with a reasonable accuracy and confidence, circuit simulations must be performed for more than 10^{9}–10^{11} samples. Due to this prohibitively large computational cost, the BMC method is impractical for SRAM yield estimation.

As an alternative to BMC simulation, Quasi-MC methods (QMC) have been used in previous studies [1]–[3]. QMC relies on the performance metric that quantifies the operation stability of a given circuit. In this method, the distribution of the performance metric is derived from a



(a)

(b)

FIGURE. 1 (a) Simplified circuits involving read access in one static random access memory (SRAM) instance and (b) operational waveforms of SRAM read access.

moderate number of MC simulations and is approximated to a known probability distribution function (PDF), usually Gaussian PDF. With the approximated PDF, the failure rate is determined as the probability that the performance metric does not meet the success criterion. Although it is efficient, most performance metrics in SRAM do not follow a known PDF in the tail region, thereby limiting the QMC accuracy.

Another category of SRAM yield estimation method is importance sampling (IS)-based approaches applied in [4]–[7]. In these methods, the PDF of the circuit parameters are distorted to make SRAM failure more probable. By performing circuit simulations with these distorted samples, a considerably large number of failure events are obtained, compared with BMC, implying that the required number of samples is significantly reduced. Subsequently, the resultant failure rate is mathematically adjusted to compensate for the distortion effects. Despite their effectiveness, it is highly challenging to determine the appropriately distorted PDF that could guarantee the accuracy.

In [8]–[11], the SRAM operation yield is estimated by boundary searching (BS). The BS methods determine the boundaries of the failure regions in the circuit parameter variation space. Thereafter, the hypervolume of the failure region that equals the failure rate, is calculated. The BS method is limited in that the failure regions and their boundaries are difficult to determine if the parameter variation space dimension is high. The SRAM read access failure is affected by numerous circuit components including the bitcell, sense amplifier, and control signal generation circuits. This implies that the parameter variation space dimension is high, limiting the direct application of the BS method to the estimation of the read access failure.

In this study, we propose an efficient method that can accurately estimate the SRAM read access yield. The proposed method can extract the distribution of three key parameters that determine the read access yield, exclusively on the basis of a reasonable number of circuit simulations. By analytically merging the derived distributions, the read access yield can be easily obtained.

The rest of this paper is organized as follows. In Section II, the background for the read access yield in the SRAM is covered. In Section III, the proposed read access yield estimation method is introduced. In Section IV, the experimental results are presented to evaluate the proposed method in comparison with the previous yield estimation methods. Finally, Section V concludes the paper.

II. BACKGROUND

Fig. 1(a) and (b) show the simplified circuit structure of an SRAM instance and the operational waveforms for read access, respectively. The read operation starts with the word-line enable signal (WLEN) assertion, causing one of the word lines (WLs) in the bitcell array to be selected and become high, according to the decoded row address signals XDEC[k]. After WL rises, the voltage between bitline pair (BL_T and BL_C), $V_{BL} = V_{BLT} - V_{BLC}$, is developed depending on the stored data in the bitcell. Because the small-sized bitcell has a poor current drivability, V_{BL} increases extremely slowly and highly limits the read access speed. Thus, the sense amplifier (SA) that can amplify such small V_{BL} into large digital level output (D_{OUT}) is used. Fig. 1(b) shows the case of sensing data “1,” where BL_C is discharged by the bitcell. In this case, V_{BL} is positive, and therefore, D_{OUT} becomes high. With the aid of SA, the bitcell data can be speedily detected even with a small analog value of V_{BL}. Unlike as shown in Fig. 1(b), if BL_T is discharged instead, D_{OUT} becomes low.

It should be noted that the structure of SA cannot be perfectly symmetric owing to transistor mismatch. Thus, SA has an input offset voltage V_{OS}, implying that the magnitude of V_{BL} should be larger than V_{OS} for appropriate sensing. For example, when data “1” is sensed, the $V_{BL} > V_{OS}$ condition should be met, instead of $V_{BL} > 0$. This implies that V_{BL} should be sufficiently large at the time of the SA enable signal (SAE) activation. Accordingly, the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111762, IEEE Access

time difference between the WL and SAE activation, T_{WL2SAE}, marked in Fig. 1(b), must be large. As WL and SAE are commonly triggered by WLEN signal, T_{WL2SAE} can be derived as (1) when T_{WLEN2WL} and T_{WLEN2SAE} are defined as WLEN to WL delay and WLEN to SAE delay, respectively.

$$T_{WL2SAE} = T_{WLEN2SAE} - T_{WLEN2WL} \tag{1}$$

To generate T_{WL2SAE} appropriately, the replica bitline is widely used [12], such that T_{WL2SAE} can track the global variation of the bitcell array. Furthermore, additional inverters are inserted to ensure that T_{WL2SAE} is sufficiently large, to compensate local variation effects.

The aforementioned three variables—V_{BL}, V_{OS}, and T_{WL2SAE}—are the three key random variables that determine the read access yield in one SRAM instance, Y_{R}. As the first step to derive Y_{R}, the probability that an SA succeeds should be decided. If N_{ROW} is the number of rows in a memory array and $N_{\mathrm{COL,SA}}$ is the number columns per SA, $N_{\mathrm{ROW}} \times N_{\mathrm{COL,SA}}$ bitcells share a single SA. All these bitcells should guarantee $V_{\mathrm{BL}} > V_{\mathrm{OS}}$ for a successful SA operation. Thus, the probability that an SA succeeds at the fixed $T_{\mathrm{WL2SAE}} = t$, $P(V_{\mathrm{BL}} > V_{\mathrm{OS}}|T_{\mathrm{WL2SAE}} = t)$, is derived as in (2).

$$\begin{aligned} & P\left(V_{BL} > V_{OS} \,|\, T_{WL2SAE} = t\right) \\ & = \int_{V_{OS}} \left\{ P(V_{BL} > v \,|\, T_{WL2SAE} = t) \right\}^{N_{ROW} \times N_{COL,SA}} f_{VOS}(v)\, dv \\ & = \int_{V_{OS}} \left\{ 1 - P(V_{BL} < v \,|\, T_{WL2SAE} = t) \right\}^{N_{ROW} \times N_{COL,SA}} f_{VOS}(v)\, dv \\ & = \int_{V_{OS}} \left\{ 1 - F_{VBL}(v \,|\, T_{WL2SAE} = t) \right\}^{N_{ROW} \times N_{COL,SA}} f_{VOS}(v)\, dv \end{aligned} \tag{2}$$

In (2), $F_{\mathrm{VBL}}(v|T_{\mathrm{WL2SAE}})$ is the cumulative distribution function (CDF) of V_{BL} for a given T_{WL2SAE}, and $f_{\mathrm{VOS}}(v)$ is the PDF of V_{OS}.

The read success probability of the entire memory instance for a given T_{WL2SAE} is the N_{SA}th power of (2), as shown in (3).

$$\left[\int_{V_{OS}} \left\{ 1 - F_{VBL}(v \,|\, T_{WL2SAE} = t) \right\}^{N_{ROW} \times N_{COL-MUX}} f_{VOS}(v)\, dv \right]^{N_{SA}}, \tag{3}$$

where the number of SAs in the array is N_{SA}. Thereafter, considering the distribution of T_{WL2SAE}, Y_{R} can be finally determined as (4).

$$Y_R = \int_{T_{WL2SAE}} \left[\int_{V_{OS}} \left\{ 1 - F_{VBL}(v \,|\, T_{WL2SAE} = t) \right\}^{N_{ROW} N_{COL,SA}} f_{VOS}(v)\, dv \right]^{N_{SA}} \times f_{TWL2SAE}(t) dt, \tag{4}$$

where f_{TWL2SAE}(t) is the PDF of T_{WL2SAE}. If the read access yield of a chip contains multiple number of instances, Y_{R} should be powered with the number of instances. Because the typical target of an entire chip is >95%, the value of $F_{VBL}(v \mid \mathrm{T}_{\mathrm{WL2SAE}} = t)$ in the range of interest is extremely small (~10^{-10}). Thus, the Taylor approximation can be used, and (4) is rewritten as (5).

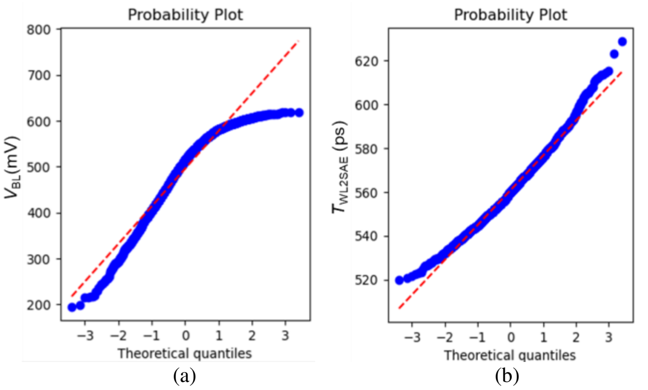


FIGURE 2. Q-Q plot example of (a) V_{BL} for given T_{WL2SAE} = 700 ps and (b) T_{WL2SAE} when V_{DD} = 0.6 V in 7 nm technology.

$$\begin{aligned} Y_R & \approx \int_{T_{WL2SAE}} \left[\int_{V_{OS}} \left\{ 1 - N_{ROW} N_{COL,SA} F_{VBL}(v \,|\, T_{WL2SAE} = t) \right\} f_{VOS}(v)\, dv \right]^{N_{SA}} f_{TWL2SAE}(t) dt \\ & = \int_{T_{WL2SAE}} \left[1 - \int_{V_{OS}} N_{ROW} N_{COL,SA} F_{VBL}(v \,|\, T_{WL2SAE} = t) f_{VOS}(v)\, dv \right]^{N_{SA}} f_{TWL2SAE}(t) dt \\ & \approx \int_{T_{WL2SAE}} \left[1 - N_{SA} N_{ROW} N_{COL,SA} \int_{V_{OS}} F_{VBL}(v \,|\, T_{WL2SAE} = t) f_{VOS}(v)\, dv \right] f_{TWL2SAE}(t) dt \\ & = 1 - N_{SA} N_{ROW} N_{COL,SA} \int_{V_{OS}} \int_{T_{WL2SAE}} F_{VBL}(v \,|\, T_{WL2SAE} = t) f_{TWL2SAE}(t) f_{VOS}(v) dt dv \end{aligned} \tag{5}$$

In the final term of (5), $N_{\mathrm{SA}}N_{\mathrm{ROW}}N_{\mathrm{COL,SA}}$ is equal to the total number of bitcells in the memory, N_{BIT}, and the integral term is considered as the bitcell failure probability, $P_{\mathrm{fail,bitcell}}$. Thus, Y_{R} can also be represented as (6) that is consistent with the results shown in [13], [14].

$$\begin{aligned} & Y_R = 1 - N_{BIT} P_{fail,bitcell} \\ & \text{where } P_{fail,bitcell} = \int_{V_{OS}} \left\{ \int_{T_{WL2SAE}} F_{VBL}(v \,|\, T_{WL2SAE} = t) f_{TWL2SAE}(t) dt \right\} f_{VOS}(v) dv \end{aligned} \tag{6}$$

There are two remarkable points in (5) and (6). First, if the target Y_{R} is sufficiently high, requiring an extremely low individual bitcell failure rate (e.g., 6 sigma yield) that is the typical case, only N_{BIT} is significant when Y_{R} is to be determined, while the individual values of N_{SA}, N_{ROW}, and N_{COL} are unimportant. For example, Y_{R} is same for (N_{SA}, N_{ROW}, N_{COL}) = (128, 512, 8) or (256, 128, 16). Second, $F_{\mathrm{VBL}}(v|T_{\mathrm{WL2SAE}}$ = t), $f_{\mathrm{VOS}}(v)$, and $f_{\mathrm{TWL2SAE}}(t)$ are required to estimate Y_{R}. However, in general, $F_{\mathrm{VBL}}(v|T_{\mathrm{WL2SAE}} = t)$ and $f_{\mathrm{TWL2SAE}}(t)$ severely deviate from the Gaussian distribution and cannot be modeled to a known distribution. Figs. 2(a) and (b) show the Q-Q plot examples of $F_{\mathrm{VBL}}(v| T_{\mathrm{WL2SAE}} = t)$ and $f_{\mathrm{TWL2SAE}}(t)$ for 7 nm technology when supply voltage V_{DD} = 0.6 V, respectively. The red dotted lines the Gaussian distribution fittings, and it can be observed that

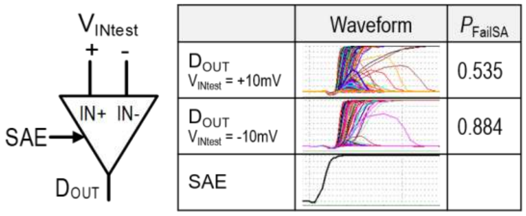


FIGURE 3. Simulation example for obtaining μ_{OS} and σ_{OS} for an SA to determine $f_{VOS}(v)$.

the data significantly deviate from the Gaussian distribution. Thus, it is highly challenging to accurately estimate Y_R, particularly for the high sigma region.

III. PROPOSED METHOD

To accurately estimate Y_R, in this section, we propose an efficient method to obtain accurate forms of $F_{VBL}(v|T_{WL2SAE} = t)$, $f_{VOS}(v)$ and $f_{TWL2SAE}(t)$. For the sake of quantitively demonstrating and evaluating the proposed method, ASAP7 PDK model [15] is used for HSPICE simulation. Unless otherwise specified, V_{DD} = 0.6 V, and the number of rows and columns per array are 256 and 128, respectively. First, $f_{VOS}(v)$, $F_{VBL}(v|T_{WL2SAE})$, and $f_{TWL2SAE}(t)$ are separately obtained in the following subsections. Thereafter, Y_R is determined using (5).

A. Determination of $f_{VOS}(v)$

As the first step, $f_{VOS}(v)$ is determined through MC simulation of SA. According to [12], [16], [17], V_{OS} of SA can be assumed to follow Gaussian distribution, $N(\mu_{OS}, \sigma_{OS}^2)$. Thus, obtaining $f_{VOS}(v)$ is equivalent to determining μ_{OS} and σ_{OS}. For the simulation setup, a fixed initialized input difference of SA, V_{INtest}, is applied, and SA is operated. Here, V_{INtest} is set to a sufficiently small value, such that it may result in a considerable number of failures. Thereafter, the failure rate of SA, P_{FailSA}, can be derived as the ratio of the number of failures to the total number of MC simulations.

With P_{FailSA} known, the relation of (7) is used to determine μ_{OS} and σ_{OS}.

$$\begin{aligned} P_{FailSA} &= P\left(V_{IN,test} - V_{OS} < 0\right) = P\left(V_{OS} > V_{INtest}\right) \\ &= P\left(\frac{V_{OS}-\mu_{OS}}{\sigma_{OS}} > \frac{V_{IN,test}-\mu_{OS}}{\sigma_{OS}}\right) = P\left(Z > \frac{V_{INtest}-\mu_{OS}}{\sigma_{OS}}\right) \end{aligned} \tag{7}$$

Here, Z is the standard Gaussian random variable (RV) that follows zero mean unit variance Gaussian distribution. Denoting the standard Gaussian CDF as $\Phi(z)$, (7) can be reduced to (8).

$$P_{FailSA} = 1 - \Phi\left(\frac{V_{INtest} - \mu_{OS}}{\sigma_{OS}}\right) \tag{8}$$

Using the inverse of $\Phi(z)$, $\Phi^{-1}(\cdot)$, (8) can be rearranged as (9).

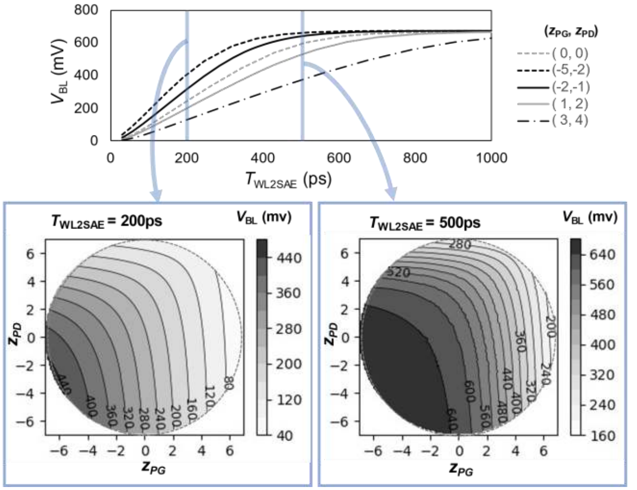


FIGURE 4. V_{BL} vs. T_{WL2SAE} for a few examples of (z_{PG}, z_{PD}) and derivation of V_{BL} vs (z_{PG}, z_{PD}) when T_{WL2SAE} is given as 200 ps and 500 ps.

$$\mu_{OS} + \sigma_{OS}\Phi^{-1}\left(1 - P_{FailSA}\right) = V_{INtest} \tag{9}$$

Because there are two variables to be determined, μ_{OS} and σ_{OS}, two equations are required. These are obtained by performing two runs of MC simulations under two different conditions of V_{INtest}—$V_{INtest1}$ and $V_{INtest2}$—that result in two different P_{FailSA}—$P_{FailSA1}$ and $P_{FailSA2}$, respectively. This results in (10).

$$\begin{aligned} \mu_{OS} + \sigma_{OS}\Phi^{-1}\left(1 - P_{FailSA1}\right) &= V_{INtest1} \\ \mu_{OS} + \sigma_{OS}\Phi^{-1}\left(1 - P_{FailSA2}\right) &= V_{INtest2} \end{aligned} \tag{10}$$

Combining the two equations of (10), μ_{OS} and σ_{OS} can be determined as (11).

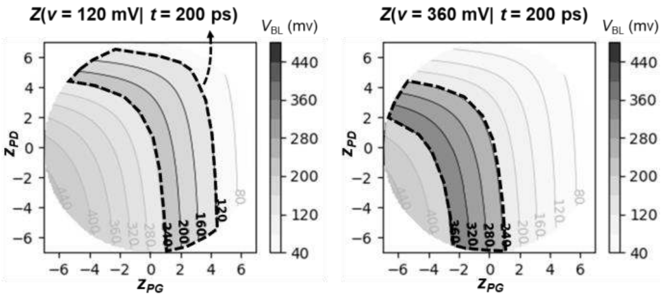
$$\begin{aligned} \mu_{OS} &= \frac{\Phi^{-1}\left(1-P_{FailSA2}\right)V_{INtest1} - \Phi^{-1}\left(1-P_{FailSA1}\right)V_{INtest2}}{\Phi^{-1}\left(1-P_{FailSA2}\right) - \Phi^{-1}\left(1-P_{FailSA1}\right)} \\ \sigma_{OS} &= \frac{V_{INtest1} - V_{INtest2}}{\Phi^{-1}\left(1-P_{FailSA1}\right) - \Phi^{-1}\left(1-P_{FailSA2}\right)} \end{aligned} \tag{11}$$

Fig. 3 shows an example of the simulation results for SA to determine $f_{VOS}(v)$ by setting V_{INtest} to 10 mV and -10 mV. The corresponding P_{FailSA} are 0.535 and 0.884, and μ_{OS} = 11.6 mV and σ_{OS} = 18.1 mV can be obtained using (11).

B. Determination of $F_{VBL}(v_{bl}|T_{WL2SAE} = t)$

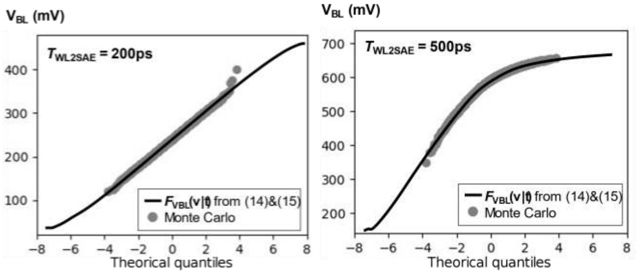
Subsequently, to obtain $F_{VBL}(v|T_{WL2SAE} = t)$, we make use of the fact that V_{BL} at a certain T_{WL2SAE} is determined by the V_{th} of the pass-gate and pull-down transistors (M_{PG} and M_{PD} in Fig. 1(a)) in the selected bitcell, $V_{th,PG}$, and $V_{th,PD}$, respectively. V_{th} of a transistor can be considered as Gaussian RV [18]–[20], and $V_{th,PG}$ and $V_{th,PD}$ can be converted to z_{PG} and z_{PD}, respectively, as shown in (12), to follow the standardized Gaussian distribution $N(0, 1^2)$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111762, IEEE Access



(a) (b)

FIGURE 5. Graphical descriptions of (a) Z(v = 120mV | t = 200ps) and (b) Z(v = 360mV | t = 200ps) that is derived from the left bottom of Fig. 4.



(a) (b)

FIGURE 6. Q-Q plots $F_{VBL}(v|t = T_{WL2SAE})$ derived using (14) and (15) based on the Monte Carlo (MC) results when T_{WL2SAE} is (a) 200 ps and (b) 500 ps.

$$z_{PG}=\frac{V_{th,PG}-\mu(V_{th,PG})}{\sigma(V_{th,PG})},\; z_{PD}=\frac{V_{th,PD}-\mu(V_{th,PD})}{\sigma(V_{th,PD})}, \tag{12}$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ are the mean and standard deviation of an RV, respectively.

V_{BL} for a given T_{WL2SAE} is a function of z_{PG} and z_{PD}, implying that a function $V_{BL}(z_{PG}, z_{PD}|T_{WL2SAE} = t)$ can be defined. Although the analytical form of $V_{BL}(z_{PG}, z_{PD}|T_{WL2SAE} = t)$ cannot be determined, its numerical form can be obtained through transient simulations by measuring V_{BL} for a range of T_{WL2SAE}, with 2D sweep of (z_{PG}, z_{PD}).

For example, V_{BL} versus T_{WL2SAE} can be obtained from the transient simulations with different conditions of (z_{PG}, z_{PD}), as shown at the top of Fig. 4. By repeating this form of simulations with the 2D sweep of (z_{PG}, z_{PD}), V_{BL} versus T_{WL2SAE} is obtained for a wide range of (z_{PG}, z_{PD}). Thereafter, $V_{BL}(z_{PG}, z_{PD} \mid T_{WL2SAE} = t)$ is obtained by extracting V_{BL} for various (z_{PG}, z_{PD}), fixing T_{WL2SAE} as t, as shown at the bottom of Fig. 4. The left and right bottom of Fig. 4 show $V_{BL}(z_{PG}, z_{PD} \mid T_{WL2SAE} = 200\text{ ps})$ and $V_{BL}(z_{PG}, z_{PD} \mid T_{WL2SAE} = 500\text{ ps})$ as examples.

From the obtained $V_{BL}(z_{PG}, z_{PD}|T_{WL2SAE} = t)$, $F_{VBL}(v|T_{WL2SAE} = t)$ can be derived. First, the median of V_{BL} at $T_{WL2SAE} = t$, $M_{VBL}(t)$, can be used as the reference point for $F_{VBL}(v|T_{WL2SAE} = t)$ as in (13).

$$F_{VBL}(v = M_{VBL}|T_{WL2SAE} = t) = 0.5, \tag{13}$$

where $M_{VBL}(t)$ can be easily obtained from a moderate number of MC simulations. Before deriving $F_{VBL}(v|T_{WL2SAE} = t)$ for an arbitrary value of v other than M_{VBL}, it is effective to define the set of (z_{PG}, z_{PD}), $Z(v|t)$, that is defined as (14) for v and t.

$$Z(v\,|\,t)=\begin{cases}\{(z_{PG},z_{PD})\,|\,M_{VBL}(t)\le V_{BL}(z_{PG},z_{PD}\,|\,T_{WL2SAE}=t)\le v\} \\ \qquad\qquad\qquad\qquad\qquad \text{if } v>M_{VBL} \\ \{(z_{PG},z_{PD})\,|\,v\le V_{BL}(z_{PG},z_{PD}\,|\,T_{WL2SAE}=t)\le M_{VBL}(t)\} \\ \qquad\qquad\qquad\qquad\qquad \text{if } v<M_{VBL}\end{cases} \tag{14}$$

Although (14) is apparently complex, it becomes evident if $Z(v|t)$ is graphically demonstrated. Fig. 5(a) and (b) depict $Z(v = 120\text{ mV} \mid t = 200\text{ ps})$ and $Z(v = 360\text{ mV} \mid t = 200\text{ ps})$, respectively, derived in the left bottom of Fig. 4, where $M_{VBL} = 239.2$ mV. That is, $Z(v|t)$ is nothing but the set of (z_{PG}, z_{PD}), such that V_{BL} is in the range between v and M_{VBL} for given $T_{WL2SAE} = t$.

Given $Z(v|t)$, the probability that (z_{PG}, z_{PD}) belongs to $Z(v|t)$ is added to or subtracted from 0.5 as in (15), according to whether v is larger or smaller than M_{VBL}. The resultant probability is equal to $F_{VBL}(v|T_{WL2SAE} = t)$, considering the definition of $Z(v|t)$.

$$F_{VBL}(v\,|\,T_{WL2SAE}=t)=\begin{cases}0.5+\iint_{Z(v|t)} f_{PG-PD}(z_{PG},z_{PD})dz_{PG}dz_{PD} \\ \qquad\qquad\qquad\qquad \text{if } v>V_{BL0} \\ 0.5-\iint_{Z(v|t)} f_{PG-PD}(z_{PG},z_{PD})dz_{PG}dz_{PD} \\ \qquad\qquad\qquad\qquad \text{if } v<V_{BL0}\end{cases} \tag{15}$$

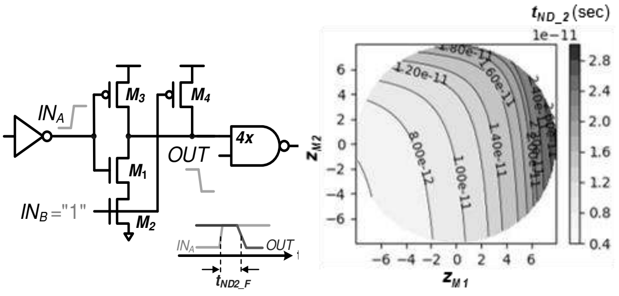
As z_{PG} and z_{PD} both follow $N(0,1^2)$, with the assumption that z_{PG} and z_{PD} are independent, $f_{PG\text{-}PD}(z_{PG}, z_{PD})$ can be written as (16).

$$\begin{aligned}&f_{PG-PD}(z_{PG},z_{PD})\\&=1\big/\sqrt{2\pi}\exp\left(-\tfrac{1}{2}z_{PG}^2\right)\times 1\big/\sqrt{2\pi}\exp\left(-\tfrac{1}{2}z_{PD}^2\right)\\&=1\big/2\pi\exp\left\{-\tfrac{1}{2}\left(z_{PG}^2+z_{PD}^2\right)\right\}\end{aligned} \tag{16}$$

Substituting $f_{PG\text{-}PD}(z_{PG}, z_{PD})$ in (15) with (16), the value of $F_{VBL}(v|T_{WL2SAE} = t)$ can be determined. As the integral is calculated numerically, an infinite range of (z_{PG}, z_{PD}) cannot be covered. Instead, only the region of $z_{PG}^2 + z_{PD}^2 \le R^2$ is considered, while R is chosen to be sufficiently large, guaranteeing an accurate estimation of the $P_{fail,bitcell}$ close to 10^{-10}.

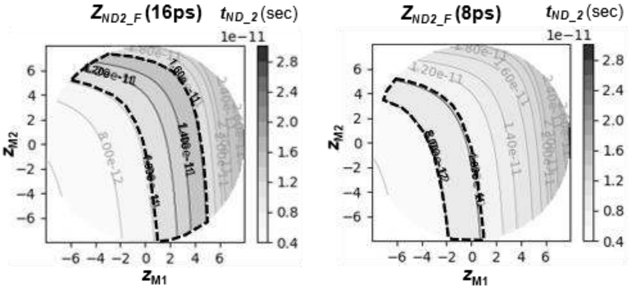
Fig. 6(a) and (b) show the obtained Q-Q plots of $F_{VBL}(v|T_{WL2SAE} = 200\text{ ps})$ and $F_{VBL}(v|T_{WL2SAE} = 500\text{ ps})$, respectively, obtained using (14) and (15). In addition, V_{BL} distributions obtained from 20k runs of MC simulations are shown for comparison. It is evident that the proposed method can appropriately estimate the V_{BL} distribution near the center region. Moreover, the proposed method can estimate the V_{BL} distribution even for a high sigma region

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111762, IEEE Access



(a) (b)

FIGURE 7. **(a) Transistor-level schematic of an NAND2 gate and (b) t_{ND2_F} vs. (z_{M1}, z_{M2}).**



(a) (b)

FIGURE 8. **Graphical representation of (a) Z_{ND2_F}(16ps) and (b) Z_{ND2_F}(8 ps) when M_{tND2_F} is 10.3 ps.**

that cannot be characterized through MC simulations.

It is noticeable that V_{BL} deviates from the Gaussian distribution, particularly when T_{WL2SAE} is large, and thus, QMC cannot be used to estimate the V_{BL} distribution. This is because when T_{WL2SAE} is large, the increase in V_{BL} according to T_{WL2SAE} becomes extremely slow due to a decreased read current flowing through the bitcell. Moreover, when V_{BL} becomes close to V_{DD}, it does not increase further. This causes V_{BL} to have a skewed probability distribution that becomes denser near V_{DD} for a large T_{WL2SAE}.

C. Determination of $f_{TWL2SAE}(t)$

As the final step, $f_{TWL2SAE}(t)$ is determined. According to (1), T_{WL2SAE} can be considered as the difference of two combinational digital logic circuit delays: $T_{WLEN2WL}$, the delay for the path comprising a row decoder and WL driver, and $T_{WLEN2SAE}$, the delay for the path comprising the replica BL, delay buffer, and global and local SAE drivers.

To obtain the PDF of the path delay, such as T_{WLEN2E} or $T_{WLEN2SAE}$ in (1), composed of multiple stages of logic gates, the PDF of single logic gate delay is first determined. For example, the falling delay in a NAND2 gate shown in Fig. 7(a), t_{ND2_F}, is determined. Thereafter, by merging the PDF of each single gate delay, the PDF of the path delay can be finally obtained.

The procedure of deriving the PDF of t_{ND2_F} is similar to the procedure of deriving the PDF of V_{BL}, discussed in the previous subsection. First, the relationship between t_{ND2_F} and V_{th} variations is obtained through simulation. Thereafter, by merging the obtained relationship with the PDF of V_{th}s, the CDF or PDF of t_{ND2_F} can be derived.

As the pull-down path of a NAND2 gate is composed of two stacked nFETs, M_1 and M_2, t_{ND2_F} is predominantly determined by V_{th}s, V_{thM1}, and V_{thM2}. Similar to (11), the standard Gaussian RV z_{M1} and z_{M2} are defined as in (17).

$$z_{M1}=\frac{V_{th,M1}-\mu(V_{th,M1})}{\sigma(V_{th,M1})},\quad z_{M2}=\frac{V_{th,M2}-\mu(V_{th,M2})}{\sigma(V_{th,M2})} \tag{17}$$

Further, the relation of t_{ND2_F} versus z_{M1} and z_{M2} is obtained from 2D sweep transient simulation, in terms of a contour, as shown in Fig. 7(b). It is observed that t_{ND2_F} is monotonically increased with z_{M1} or z_{M2}. It is assumed that fan-out is 4, and the rising edge of IN_1 is arrived later than that of IN_2 that is a typical case. Consequently, the effect of z_{M1} on the delay is much larger than that of z_{M2}. The case when IN_2 arrives later than IN_1 can also be easily considered by the repeating the procedure with fixing IN_1 high while applying rising signal to IN_2.

With the median of t_{ND2_F}, M_{tND2_F}, the CDF of t_{ND2_F} $F_{ND2\_F}(t)$ satisfies (18).

$$F_{ND2\_F}(M_{tND2\_F})=0.5 \tag{18}$$

For an arbitrary value of $t_{ND2\_F}=t$ other than M_{tND2_F}, inside the sufficiently large circle in Fig. 7(b), the set of (z_{M1}, z_{M2}) that result in $M_{tND2\_F}\le t_{ND2\_F}\le t$ or $t\le t_{ND2\_F}\le M_{tND2\_F}$, is defined as $Z_{ND2\_F}(t)$ as in (19).

$$Z_{ND2\_F}(t)=\begin{cases}\{(z_{M1},z_{M2})\,|\,M_{tND2\_F}\le t_{ND2\_F}\le t\} & \text{if } t>M_{tND2\_F}\\ \{(z_{M1},z_{M2})\,|\,t\le t_{ND2\_F}\le M_{tND2\_F}\} & \text{if } t<M_{tND2\_F}\end{cases} \tag{19}$$

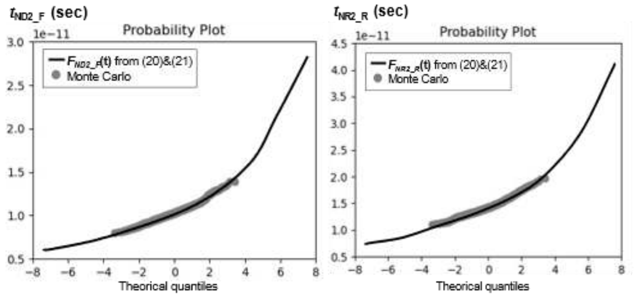
Fig. 8(a) and (b) depict Z_{ND2_F}(16 ps) and Z_{ND2_F}(8 ps), respectively, when M_{tND2_F} = 10.3 ps. Thereafter, $F_{ND2\_F}(t)$ is determined as in (20).

$$F_{ND2\_F}(t)=\begin{cases}0.5+\iint_{Z_{ND2\_F}(t)} f_{M1\text{-}M2}(z_{M1},z_{M2})\,dz_{M1}dz_{M2} & \\ \qquad\qquad\text{if } t>M_{tND2\_F} & \\ 0.5-\iint_{Z_{ND2\_F}(t)} f_{M1\text{-}M2}(z_{M1},z_{M2})\,dz_{M1}dz_{M2} & \\ \qquad\qquad\text{if } t<M_{tND2\_F} & \end{cases} \tag{20}$$

The joint PDF $f_{M1\text{-}M2}(z_{M1}, z_{M2})$ in (20) can be substituted with (21), and $F_{ND2\_F}(t)$ can be numerically obtained.

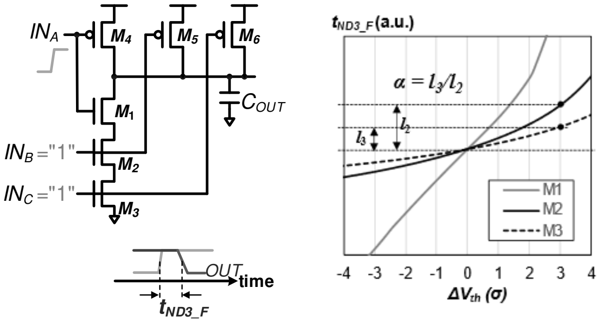
$$\begin{aligned}f_{M1\text{-}M2}(z_{M1},z_{M2})&=1\big/\sqrt{2\pi}\exp\left(-1\big/2\,z_{M1}^2\right)\times 1\big/\sqrt{2\pi}\exp\left(-1\big/2\,z_{M2}^2\right)\\&=1\big/2\pi\exp\left\{-1\big/2\left(z_{M1}^2+z_{M2}^2\right)\right\}\end{aligned} \tag{21}$$

Substituting $f_{M1\text{-}M2}(z_{M1}, z_{M2})$ in (20) with (21), F_{ND2_F}(t) can be numerically obtained as shown in Fig. 9(a), where MC simulation results are shown alongside. The same procedure can be applied to the rising delay of NOR2 gate,



(a) (b)

FIGURE 9. Q-Q plots of (a) $F_{ND2\_F}(t)$ and (b) $F_{NR2\_R}(t)$ obtained from (20) and (21). MC simulation results are shown alongside for comparison.



(a) (b)

FIGURE 10. (a) Schematic of NAND3 and (b) the sensitivity of Vth of M1, M2, and M3 transistors on t_{ND3_F} for determination of α.

t_{NR_R} that is predominantly determined by two stacked pFETs of the logic gate. The resultant CDF $F_{NR2\_R}(t)$ can also be derived as shown in Fig. 9(b). It is observed that the CDFs obtained by the proposed method appropriately fit the MC simulation results in the center region; moreover, the proposed method can characterize the high sigma region that cannot be characterized through MC simulations.

For logic gates with a large number of inputs, such as NAND3 or NOR4, the derivation of Z(t) in (19) to derive CDF becomes highly complex. This is because they necessitate a higher dimension simulation, 3D or 4D, that leads to a significant increase in the simulation time. To reduce the required simulation dimension while retaining the accuracy, the effects of non-critical transistors can be merged. For example, to characterize the probability distribution of NAND3 that is shown in Fig. 10(a), the variation effects of M_3 are merged to the variation of M_2 by increasing the V_{th} variation of M_2 by $(1+\alpha^2)^{1/2}$ times, while making the V_{th} variation of M_3 zero. Here, α implies the sensitivity ratio of V_{th} on delay in M2 and M3 that can be easily determined by the 1D sweep circuit simulation as shown in Fig. 10(b). In this manner, the circuit simulation is limited to 2D sweep, while the variations of three transistors can be considered.

Once the CDF for logic delay is determined using (20) for the one fixed circuit condition of input slope, output load, and transistor width, the CDF of delay for the same type of logic, but different circuit condition (input slope, output load and transistor width), can be easily derived through a simple conversion process. That is, if the CDF of a logic delay for one condition is obtained as F_{logic1}(t) using (20) as a reference, then the CDF of a logic delay for any other target condition, F_{logic2}(t) is determined using (21).

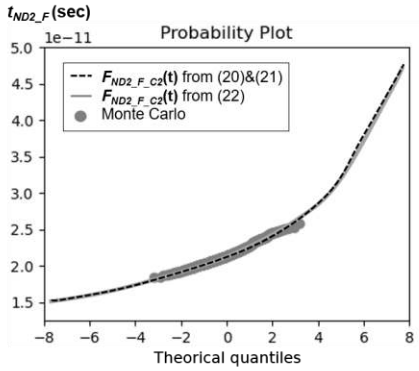


FIGURE 11. Q-Q plot comparison of FND2_F_C2(t) from (20) and (22).

$$F_{logic2}(t) = F_{logic1}\left(\frac{M_1}{M_2}\sqrt{\frac{W_2}{W_1}}(t-M_2)+M_1\right) \qquad (22)$$

In (22), M_1 and M_2 are the median values of the logic gate delay under the reference condition and target condition, respectively, while W_2/W_1 is the transistor width ratio of the target condition to the reference condition.

To delay the deviation of (t-M_2), M_1/M_2 and $(W_2/W_1)^{1/2}$ are multiplied to reflect the condition difference. For example, a larger input slope, larger output capacitance, or smaller transistor width reducing the on-current, could result in a larger delay. These effects that change the delay magnitude are included in the change of the median value and therefore, can be appropriately considered by multiplying M_1/M_2. In addition, the transistor width affects not only the magnitude of the delay, but also its variation. This is because the V_{th} variation is inversely proportional to the square root of the transistor width. This effect is reflected in (22) through the term $(W_2/W_1)^{1/2}$.

M_1 and M_2 can be derived from the moderate number of MC simulation results, or more simply, can be substituted with the nominal delay obtained from the single time circuit simulation for most cases. Thus, using (22), the number of circuit simulations can be significantly reduced. Fig. 11 compares the CDF of the NAND2 fall delay under different load and transistor width conditions from Fig. 9(a), $F_{ND2\_F\_C2}(t)$, determined thorough the direct method of (20) and indirect method (22), with MC simulation results. It is observed that the two curves are almost equal, implying that (22) can successfully characterize the distribution without obtaining $Z_{ND\_F\_C2}$(t) through timely 2D circuit simulation.

Combining (20) and (22), the distribution of delay for any logic path consisting of multiple logic stages, can be obtained as shown in Fig. 12, through the following four steps.

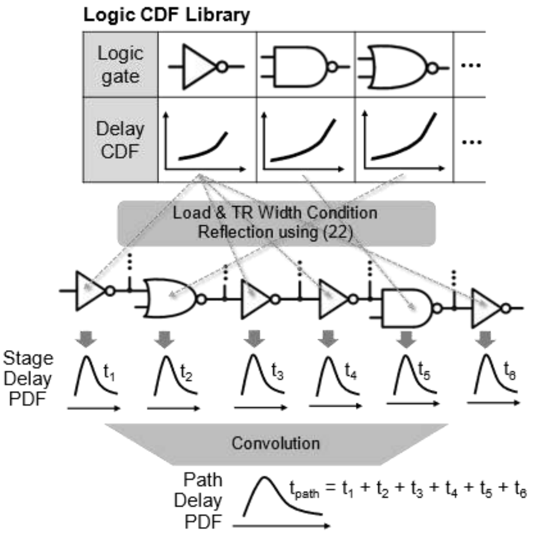


FIGURE 12. Procedure for determining PDF of path delay.

1) The CDFs of delay for the required logic types are obtained at the reference condition by (20) to form the CDF library.
2) The median value of each stage is derived.
3) With the median values, the CDF of each stage is derived using (22).
4) The PDF of each stage, the derivative of the CDF, is merged through convolutions.

Through the above procedure, the PDF of T_{WLEN2E} or $T_{WLEN2SAE}$ in (1) that are the delay incurred in the circuit shown in Fig. 1, can be obtained. For instance, $T_{WLEN2SAE}$ can be expressed as (23) where t_{i} is the delay of the i^{th} stage of WLEN to SAE path.

$$T_{\mathrm{WLEN2SAE}} = t_1 + t_2 + \ldots + t_{\mathrm{N}} \tag{23}$$

Thus, the PDF of $T_{WLEN2SAE}$, $f_{TWLEN2SAE}(t)$ is obtained by (24)

$$f_{TWLEN2SAE}(t) = f_{t1}(t) * f_{t2}(t) * \ldots * f_{tN}(t), \tag{24}$$

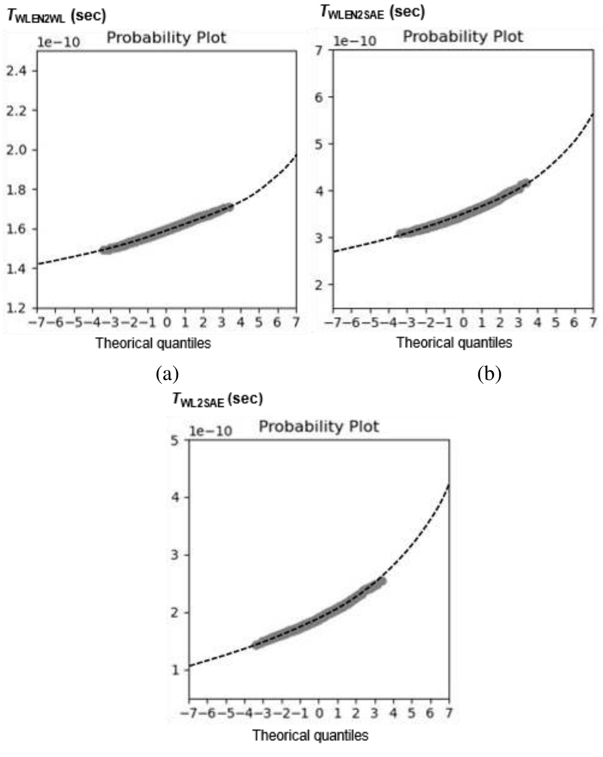
where * is the convolution operation. In the similar manner, the PDF of T_{WLEN2WL}, $f_{TWLEN2WL}(t)$, can also be determined.

With $f_{TWLEN2SAE}(t)$ and $f_{TWLEN2WL}(t)$, $f_{\mathrm{TWL2SAE}}(\mathrm{t})$ is determined through the convolution based on (1) as (25).

$$f_{\mathrm{TWL2SAE}}(\mathrm{t}) = f_{\mathrm{TWLEN2SAE}}(t) * f_{\mathrm{TWLEN2WL}}(-t) \tag{25}$$

Fig. 13 shows the Q-Q plot for CDF of T_{WLEN2E}, $T_{WLEN2SAE}$, and T_{WL2SAE} obtained from (23)–(25) with MC simulation results. To include the effects of the parasitic resistance and capacitance, the post layout simulations are performed. It is observed that the proposed method can accurately characterize $f_{\mathrm{TWL2SAE}}(\mathrm{t})$ up to a high sigma region.

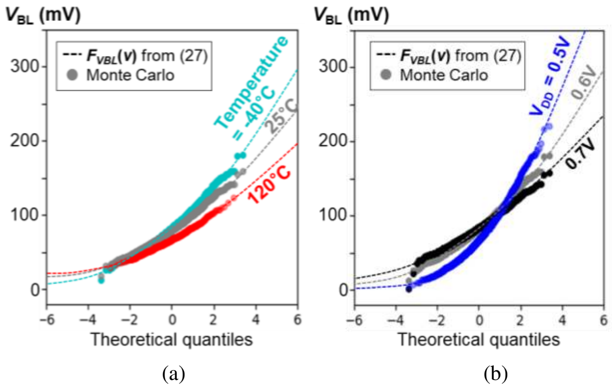
Because the integration and convolution are performed numerically, the resolutions for the operations should be sufficiently small to achieve a high accuracy. The effects of resolutions on the numerical calculation accuracy are discussed in Section IV.



(a) (b)

(c)

FIGURE 13. Q-Q plots for (a) $T_{WLEN2WL}$, (b) $T_{WLEN2SAE}$, and (c) T_{WL2SAE}.



(a) (b)

FIGURE 14. Q-Q plots for $F_{\mathrm{VBL}}(\mathrm{v})$ in (26) for (a) different temperature conditions when VDD = 0.6V and (b) different V_{DD} conditions for temperature is -40°C.

D. Derivation of bitcell failure rate

From the previous subsections, $f_{\mathrm{VOS}}(v)$, $F_{\mathrm{VBL}}(v_{\mathrm{bl}}|T_{\mathrm{WL2SAE}} = t)$, and $f_{\mathrm{TWL2SAE}}(t)$ can be obtained, that is, all the forms required to derive $P_{fail,bitcell}$ (and therefore Y_{R}) in (6) are available. The double integration in (6) for $P_{fail,bitcell}$ derivation can be interpreted as the two following steps; 1) the CDF of V_{BL} at T_{WL2SAE}, $F_{\mathrm{VBL}}(v)$, is derived first as (26), and 2) subsequently, the probability that the measured V_{BL} is smaller than V_{OS} is derived.

$$F_{VBL}(v) = \int_{T_{WL2SAE}} F_{VBL}(v \mid T_{WL2SAE} = t) f_{TWL2SAE}(t)dt \qquad (26)$$

Fig. 14(a) and (b) compare the CDF obtained by (26) with MC simulation results for the various conditions of temperatures and V$_{DD}$, respectively, implying that (26) can successfully characterize the V_{BL} distribution. Finally, $P_{fail,bitcell}$ can be derived from (27) that incorporates (26) and (6).

$$P_{fail,bitcell} = \int_{V_{OS}} F_{VBL}(v) f_{VOS}(v)dv \qquad (27)$$

According to Fig. 14(a), as the temperature is lowered, the worst-case V_{BL} at T_{WL2SAE} – V_{BL} at the left tail of $F_{VBL}(v)$ – is decreased. This is because, in low V$_{DD}$, the worst-case drain current is decreased in lower temperature, leading to decrease in the worst-case bitcell current. Thus, -40°C is the worst temperature corner for the sensing yield in the temperature range of -40°C~120°C.

Fig. 14(b) shows the effect of V$_{DD}$ on V_{BL} at T_{WL2SAE}. When V$_{DD}$ is decreased, there are two factors which oppositely affect V_{BL} at T_{WL2SAE}; 1) T_{WL2SAE} is increased due to larger gate delays to make V_{BL} at T_{WL2SAE} larger, and 2) the bitcell current is decreased to make V_{BL} at T_{WL2SAE} smaller. It should be noted that, in low V$_{DD}$, the variation of T_{WL2SAE} and the bitcell current becomes larger. Thus, V_{BL} at T_{WL2SAE} in the right tail of F_{VBL} becomes larger as V$_{DD}$ lowers. However, due to the enlarged variation in low V$_{DD}$, the bitcell current and T_{WL2SAE} can be exceedingly small. Thus, V_{BL} at T_{WL2SAE} in the left tail of F_{VBL}, which is critical for the sensing yield, can become very small in low V$_{DD}$. Thus, the sensing yield becomes degraded as V$_{DD}$ lowers. Therefore, the sensing yield of the SRAM should be evaluated at the low corner of V$_{DD}$.

IV. EXPERIMENT RESULTS

In this section, $P_{fail,bitcell}$ is estimated using the proposed method discussed in the previous section and compared with the other previous yield estimation methods in terms of the accuracy and efficiency. HSPICE post layout simulations are performed using ASAP 7 nm finFET technology [15], with V$_{DD}$ = 0.6V. The temperature is set to -40°C which is the worst temperature corner for sensing yield at V$_{DD}$ = 0.6V.

To model the variation in a transistor, V_{th} is randomly generated to follow Gaussian distribution whose standard deviation is σ_{Vth} given in (28) [21],

$$\sigma_{Vth} = \frac{A_{\Delta VT}/\sqrt{2}}{\sqrt{L_g W_g}} = \frac{A_{\Delta VT}/\sqrt{2}}{\sqrt{L_g N_{fin}(2H_{fin} + T_{fin})}} \qquad (28)$$

where $A_{\Delta VT}$ is Pelgrom constant which is determined based on the silicon measurement results of 7nm finFET in [22], and L_g, W_g, N_{fin}, H_{fin}, and T$_{fin}$ are gate length, gate width, the number of fin, fin height and fin thickness in a finFET, respectively.

A. Comparison

To verify the accuracy and the effectiveness of the proposed method, the derived results of Y_R are compared with those of the other yield estimation methods—QMC, minimized norm IS MNIS [5], scaled-sigma sampling SSS [23], and subset simulation SUS [24]—in terms of the accuracy and the simulation time. The failure rate obtained by the BMC simulation is used as the reference or the "golden" failure rate that is used for evaluating the accuracy of the other approaches.

For QMC, V_{BL} at T_{WL2SAE} is assumed to follow Gaussian distribution. For SSS, the four scaling factors {1.5, 2, 2.5, 3} are used for linear regression. For SUS, the objective failure rate is used as 0.1.

Because the $P_{fail,bitcell}$ estimated by all the methods vary according to the selection of the samples (that is, stochastic), instead of merely deriving a single value of $P_{fail,bitcell}$ through a one-time simulation, multiple runs of estimations are repeated—in this work, 100 times—to obtain the population of the $P_{fail,bitcell}$ values for each method. Consequently, the mean and standard deviation of $P_{fail,bitcell}$ can be obtained for each method.

The convergence conditions of the evaluated methods are set, such that the figure of merit $\rho(P_{fail,bitcell})$ defined as (29) is equal to 0.0865, implying 95% accuracy and 95% confidence level.

$$\rho(P_{fail,bitcell}) = \sqrt{VAR(P_{fail,bitcell})} / P_{fail,bitcell} \qquad (29)$$

Fig. 15(a) and (b) show the $P_{fail,bitcell}$ derived through various estimation methods and their $\rho(P_{fail,bitcell})$, respectively, versus the number of SPICE simulations N_{SPICE}. On purpose, the memory instance is designed to have a considerably larger $P_{fail,bitcell}$ than the 6 sigma yield (~3 sigma yield), such that the golden rate can be obtained through a practically acceptable number of SPICE simulations. Under the given condition, the golden P$_{fail,bitcell}$ = 0.00127 is determined by BMC that converges at N_{SPICE} = 1.5 × 10^{5}.

MNIS results are not shown in Fig. 15(a) and (b) because the $P_{fail,bitcell}$ estimated by MNIS exceedingly deviates from the golden $P_{fail,bitcell}$. This is because hundreds of transistors affect $P_{fail,bitcell}$ in a memory instance (that is, high dimensional variation space), and MNIS fails to determine the accurate optimal shifted vector, even within an exceedingly large number of SPICE simulations (~10^{5}). Non-optimal shifted vector results for an exceedingly small $P_{fail,bitcell}$. Under the given condition with the golden $P_{fail,bitcell}$ of 0.00127, $P_{fail,bitcell}$ estimated by MNIS is smaller than 10^{-50}, implying that MNIS is inappropriate.

Although QMC prediction converges with the 300-times smaller N_{SPICE} compared with BMC (~5k), $P_{fail,bitcell}$ estimated through QMC is ~0.4 that is more than thrice the golden value. This poor accuracy of QMC is attributed to

TABLE I
COMPARISON OF BITCELL FAILURE RATE ESTIMATION

Method	$P_{\text{fail,bitcell}}$	Relative Error (%)	N_{SPICE} for converge
BMC	0.00127	0 (Golden)	1.5×10^5
QMC	0.00391	208	5.0×10^3
MNIS	$<10^{-50}$	Large	-
SSS	0.00155	12.3	7.5×10^4
SUS	0.00141	11.1	5.0×10^4
Proposed (step = 0.5)	0.00132	4.7	1.25×10^3
Proposed (step = 0.2)	0.00131	3.15	8.0×10^3

the fact that QMC assumes that (V_{BL}–V_{OS}) follows the Gaussian distribution that is incorrect.

Although SSS shows a better accuracy and estimates $P_{\text{fail,bitcell}}$ = 0.00151 and has a relatively low error of ~12.3%, SSS requires a large N_{SAMPLE}, showing only a limited improvement in the convergence condition compared with BMC. This is because SSS requires multiple cases of SPICE simulations for different scale factors and has an increased uncertainty in the regression procedure with the approximated model. SUS converges with N_{SPICE} = 50k, implying that SUS has a three times better efficiency compared with BMC, with a fine accuracy ($P_{\text{fail,bitcell}}$ = 0.00141). However, SUS still requires 50k runs of SPICE simulations.

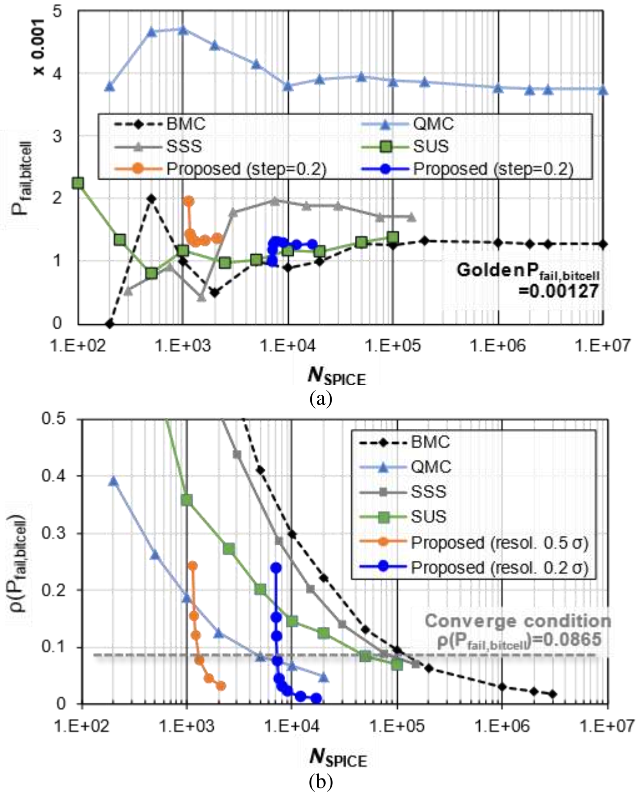
In the higher yield condition such as 6 sigma, where $N_{\text{SAMPLE}} > 10^{11}$ is required for the convergence with BMC, N_{SAMPLE} in the order of 10^5–10^6, implying that SUS and SSS are considerably more efficient than BMC. However, it still consumes a considerable amount of time even if only relevant transistors (over 100 dimensional, however) are involved for simulations. For instance, in the test environment used in this study that is 32-core Intel Xeon 2.30 GHz CPUs, 50k runs of post-layout SPICE simulations for one SRAM read operation takes ~10 h. Because the multiple number of the yield estimations should be repeated during the circuit optimization procedure, this runtime is critical and must be reduced. Table I summarizes the compared results for the different $P_{\text{fail,bitcell}}$ estimation methods.

To examine the simulation results of the proposed method, there are several aspects to be considered. In the proposed method, SPICE simulations are run for determining f_{VOS}, F_{VBL}, and f_{TWL2SAE}. Therefore, N_{SPICE} for the proposed method, $N_{\text{SPICE,prop}}$, is determined as (30),

$$N_{SPICE,prop} = N_{SPICE,OS} + N_{SPICE,VBL} + N_{SPICE,TWL2SAE} , \qquad (30)$$

where $N_{\text{SPICE,OS}}$, $N_{\text{SPICE,VBL}}$, and $N_{\text{SPICE,TWL2SAE}}$ are the number of SPICE simulations required for deriving f_{VOS}, F_{VBL}, and f_{TWL2SAE}, respectively.

It should be noted that $N_{\text{SPICE,prop}}$ in (30) does not adequately represent the efficiency of the proposed method. The reason is as follows. In the previous methods, the SPICE simulations should be run in the entire memory instance level. However, the proposed method runs SPICE simulations in the circuit composed of a considerably smaller number of transistors such as SA or a logic gate when f_{VOS} and f_{TWL2SAE} are determined. Thus, it is unfair to compare the efficiency of the proposed method with that of the previous methods using (30). To consider this runtime difference, the effective N_{SPICE}, $N_{\text{EFF,SPICE}}$, is defined as in (31) for the proposed method.



(a)

(b)

FIGURE 15. (a) $P_{\text{fail,bitcell}}$ and (b) $\rho(P_{\text{fail,bitcell}})$ versus N_{SPICE}

$$N_{EFF,SPICE} = \alpha N_{SPICE,OS} + N_{SPICE,VBL} + \beta N_{SPICE,TWL2SAE} \qquad (30)$$

In (31), α and β denote how much shorter the SPICE simulation time is for SA and logic gates forming the WL-to-SAE path, respectively, compared with the entire memory instance level SPICE simulation time required to derive F_{VBL}. Typically, α and β are much smaller than 0.1.

Another noticeable point of the proposed method is that F_{VBL} and f_{TWL2SAE} are not obtained by the simulations based on the random samples, but they are determined by the deterministic parameter sweep simulations. This implies that the accuracy of the proposed method is not affected by the appropriateness of the selection of the random samples, and there is no requirement for a large number of samples to reduce the uncertainty. Thus, $\rho(P_{\text{fail,bitcell}})$ is highly improved compared with the previous methods that mainly rely on the simulations based on the random samples. Only when f_{VOS} is determined, are the random samples employed.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111762, IEEE Access

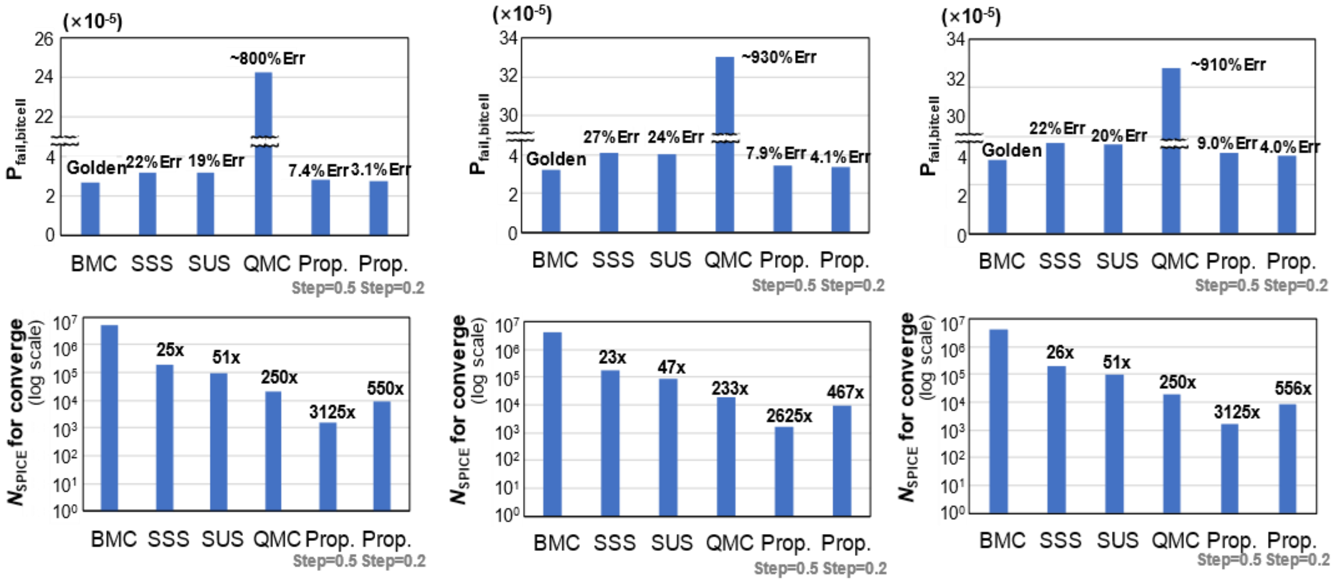


FIGURE 16. $P_{fail,bitcell}$ and $\rho(P_{fail,bitcell})$ of the six methods for three different circuit designs: (a) WL is increased by 50 mV, (b) additional delay buffers are used for SAE generation, and (c) SA size is increased for halving the value of V_{OS}.

However, this is not a high burden because μ_{OS} and σ_{OS} are converged in a relatively small number of random samples and SA consists of seven transistors that is considerably smaller than the entire memory instance.

A distinctive feature of the proposed method is that the accuracy is dependent on the step size used for numerical calculation because the integral is numerically calculated for deriving F_{VBL} or $f_{TWL2SAE}$ in (15) and (20), respectively. The integral in (15) is numerically evaluated as (32).

$$\iint_{Z(v|t)} f_{PG-PD}(z_{PG}, z_{PD}) dz_{PG} dz_{PD} = \sum_{Z(v|t)} f_{PG-PD}(z_{PG}, z_{PD}) \Delta z_{PG} \Delta z_{PD} \quad (32)$$

The step size, Δz_{PG} or Δz_{PD}, should be set sufficiently small to ensure accuracy; however, a smaller step increases the required number of SPICE runs. For example, if the range of $z_{PG}^2 + z_{PD}^2 \leq 8$ is covered, the number of SPICE runs required for (32) with the step size of 0.5 and 0.2 are 797 and 5025, respectively. This is also true for the integral calculation in the logic gate delay distribution in (20).

In Fig. 15 (a) and (b), $P_{fail,bitcell}$ and $\rho(P_{fail,bitcell})$ of the proposed method are shown, as derived on the basis of the step size of 0.5 and 0.2. The range of V_{th} is covered for the range of $z_{PG}^2 + z_{PD}^2 \leq 8$ and $z_{M1}^2 + z_{M2}^2 \leq 8$ when deriving F_{VBL} or $f_{TWL2SAE}$, respectively. For a fair comparison, $N_{EFF,SPICE}$ defined in (31) is used for the horizontal axis, while both α and β are set to 0.1 that is conservatively large. Compared with the previous methods, the proposed method converges speedily because the random samples are utilized only when f_{VOS} is determined, as explained.

When the step size is 0.5, the proposed method is converged within only N_{SAMPLE} = 1.25k that is 120 times smaller than BMC, and $P_{fail,bitcell}$ is estimated as 0.00132 that produces an error of 4.7%. When the step size is 0.2, the convergence condition is met N_{SAMPLE} = 8k, $P_{fail,bitcell}$ is estimated as 0.00131 that produces a 3.15% error. As expected, the error is decreased according to the reduced step size. In terms of the efficiency and accuracy, the proposed method affords the best results among the different yield estimation methods.

It is valuable to apply the methods and examine the results under higher yield conditions. To enhance the yield close to 4 sigma ($P_{fail,bitcell}$ ~3×10^{-5}), where the golden rate can still be derived through BMC within a few days or weeks, the SRAM circuit is revised in three respects; subsequently, the six methods are applied to estimate and compare $P_{fail,bitcell}$. The three circuit revisions are as follows: (1) higher WL voltage (+50 mV) is used, (2) additional delay buffers are used (12 additional inverters in the path with a load capacitor), (3) increasing the transistor width (fin number) in SA for halving V_{OS} variation.

Fig. 16(a), (b), and (c) compare the $P_{fail,bitcell}$ and $\rho(P_{fail,bitcell})$ of the six methods for the three different circuit revisions. Compared with the low yield condition shown in Fig. 15 and Table I, the efficiencies of the non-BMC methods are considerably improved compared with that of BMC, while the proposed method exhibits the best accuracy and efficiency. QMC is not applicable because it is highly inaccurate.

According to Fig. 16, SSS and SUS require a 10^5 order of N_{SPICE}, while the proposed method requires a 10^4 order of N_{SPICE} to meet the convergence condition. Because the target yield is typically higher than 4 sigma (~6 sigma), the required N_{SPICE} is larger. Although these N_{SPICE} values are considerably smaller than BMC, the runtime for N_{SPICE} = ~10^4–10^5 is still significant. Considering the fact that the $P_{fail,bitcell}$ estimation procedure should be repeated multiple times during the circuit optimization, it is inevitable to spend a significantly long time to derive $P_{fail,bitcell}$. In other words, over 10^5 times of SPICE runs for full memory

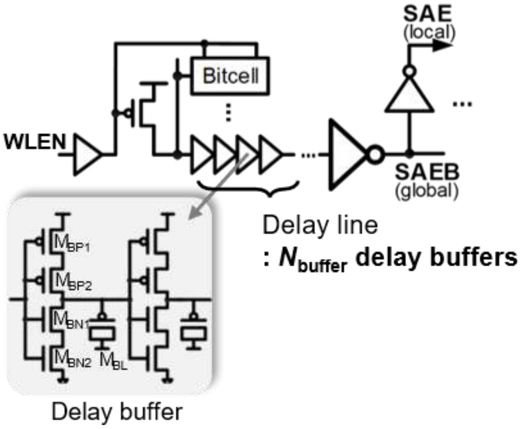


FIGURE 17. Delay buffer circuits for SAE generation.

instance circuit are required every time the circuit is revised in SSS or SUS. However, in the proposed method, additional SPICE runs need not be performed to estimate $P_{\text{fail,bitcell}}$ after the circuit is revised; conversely, SPICE runs are only required for a small part of the circuit instead of a full memory instance. This is another merit of the proposed method that is covered in detail in the following subsection.

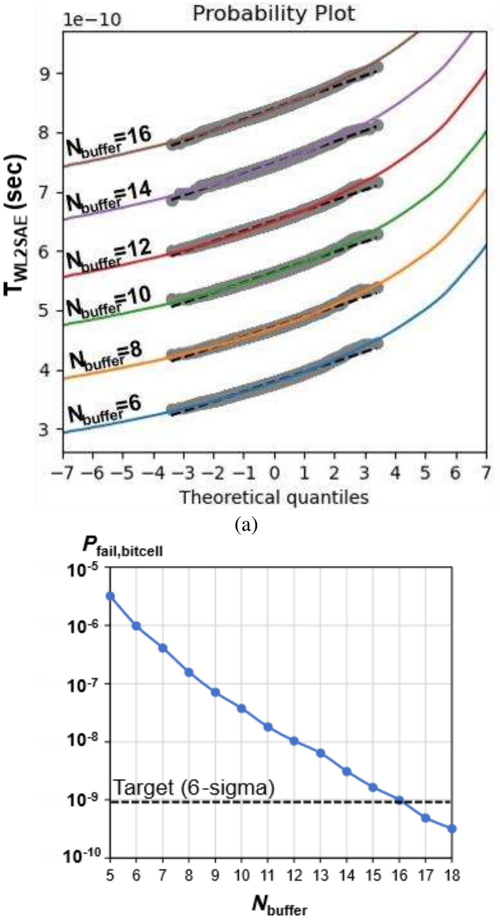
B. Application to circuit optimization

In addition to the reduction in the computational costs in the yield estimation, another significant advantage of the proposed method is the ability to accelerate the circuit design optimization procedure. One of the most challenging tasks in the SRAM circuit design is the optimization of the delay circuits for generating SAE signal. If SAE is triggered exceedingly early, Y_R is degraded. On the contrary, the delayed triggering of SAE increases the read access time and unnecessary power consumption. Thus, the SAE generation circuit should be carefully designed considering these two aspects; 1) the delay imposed on SAE should be minimized, provided that 2) the target Y_R (or $P_{\text{fail,bitcell}}$) is satisfied.

The SAE generation circuit is revisited in Fig. 17 that includes a delay line composed of multiple buffers. To obtain a sufficient delay, the delay buffer is designed with the stacked pFETs and nFETs ($M_{BP1,2}$ and $M_{BN1,2}$), and a load pFETs (M_{BL}) is used.

To optimize the SAE circuit design on the basis of the previous methods, the yield estimation procedures should be repeatedly invoked by adding or removing the delay buffers until the design goal is achieved. This procedure necessitates computationally heavy SPICE circuit simulations. However, the proposed yield estimation does not require additional SPICE simulations for this optimization process because the distribution of the T_{WL2SAE} is obtained based on (24), and (25) only necessitates convoluting the predetermined PDF of the basic logic gates, including the delay buffer shown in Fig. 17.

Fig. 18(a) shows the distribution of T_{WL2SAE} for different number of buffers obtained by (24) and (25) that are compared with the MC simulation results. As explained, SPICE runs are required only when the delay distributions of logic gates and delay buffer are derived. Adding or removing buffer in the delay line does not necessitate the additional SPICE simulations. Thus, the results shown in Fig. 18(a)—f_{TWL2SAE} for different N_{buffer} cases—can be obtained without additional SPICE simulations. Based on Fig. 18(a), $P_{\text{fail,bitcell}}$ can be obtained as Fig. 18(b) according to different N_{buffer}. Assuming the target read access yield is 6 sigma, corresponding to $P_{\text{fail,bitcell}} = 9.86 \times 10^{-10}$, N_{buffer} should be larger than 16. The step size of 0.2 is used and $N_{\text{SPICE,eff}}$ is set to 15000 to cause the $P_{\text{fail,bitcell}}$ to converge under the 6 sigma yield condition. In this manner, the proposed method exhibits a significantly improved efficiency, compared with the previous methods.



(a)

(b)

FIGURE 18. (a) Q-Q plots of T_{WL2SAE} for different N_{buffer} and (b) the resultant estimated $P_{fail,bitcell}$ according to N_{buffer}.

Instead of changing N_{buffer}, the delay buffer design itself (for example, the size of transistor) can be revised to adjust T_{WL2SAE} for circuit optimization. In addition, the SA size can be adjusted to reduce V_{OS}. In these circuit revisions, the proposed method requires additional SPICE runs. However, additional SPICE can be run at a small circuit level (for example, the delay buffer or SA), instead of at a full memory instance level. Consequently, $P_{\text{fail,bitcell}}$ can be derived considerably more efficiently compared with the other methods.

V. CONCLUSION

We propose a method that accurately and efficiently estimates the read access yield in high-density SRAM. In the proposed method, the SRAM is partitioned into three circuit parts—the control signal generation circuit, the bitcell array, and SA. These three circuit parts determine the three key parameters, T_{WL2SAE}, V_{BL}, and V_{OS}, respectively. The probability distributions of the three parameters are derived through different approaches, considering the respective characteristics. Because only V_{OS} is determined by the random samples, $P_{\mathrm{fail,bitcell}}$ estimated by the proposed method can converge within a much smaller runtime, with a higher accuracy. The proposed method can achieve 500–3000× improvement in the speed for 4 sigma yield over BMC, and 10–100× over the other state-of-art methods.

More importantly, in the circuit optimization procedure, the proposed method does not require additional SPICE runs or requires SPICE runs only for small circuit parts, instead of the entire memory instance. Thus, the proposed method can significantly reduce the computational cost of yield optimization in SRAM.

The proposed method is customized to derive the sensing yield in the SRAM. However, the write or read stability yield are also critical for the SRAM. As a future work, we would like to develop the write and read stability yield estimation method, which can have high accuracy and efficiency.

REFERENCES

[1] K. Agarwal and S. Nassif, "Statistical Analysis of SRAM Cell Stability," in *Proc. Design Autom. Conf.*, Jul. 2006, pp. 57-62.

[2] E. Grossar, M. Stucchi, K. Maex and W. Dehaene, "Read Stability and Write-ability Analysis of SRAM Cells for Nanometer Technologies," *IEEE J. Solid-State Circuits*, vol. 41, no. 11, pp. 2577-2588, Nov. 2006.

[3] C. Wann, R. Wong, D. J. Frank, R. Mann, S.-B. Ko, P. Croce, D. Lea, D. Hoyniak, Y.-M. Lee, and J. Toomey, "SRAM cell design for stability methodology," *IEEE VLSI-TSA International Symposium on VLSI Technology, (VLSI-TSA-Tech).*, pp. 21-22, 2005.

[4] M. Wang, C. Yan, X. Li, D. Zhou, and X. Zeng, "High-Dimensional and Multiple-Failure-Region Importance Sampling for SRAM Yield Analysis," *IEEE Trans. VLSI Syst.,* vol. 25, no. 3, pp. 806–819, 2017.

[5] L. Dolecek, M. Qazi, D. Shah and A. Chandrakasan, "Breaking the simulation barrier: SRAM evaluation through norm minimization", in *Proc. Int. Conf. Comput. Aided Design*, 2008, pp. 322-329.

[6] L. Pang, M. Yao and Y. Chai, "An Efficient SRAM yield Analysis Using Scaled-Sigma Adaptive Importance Sampling," in *Proc 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2020, pp. 97-102.

[7] R. Kanj, R. Joshi and S. Nassif, "Mixture importance sampling and its application to the analysis of SRAM designs in the presence of rare failure events,"in *Proc Design Autom. Conf.*, 2006, pp. 69-72.

[8] J. Yao, Z. Ye, and Y. Wang, "Importance boundary sampling for SRAM yield analysis with multiple failure regions," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 33, no. 3, pp. 384–396, Mar. 2014.

[9] F. Gong, H. Yu, Y. Shi, D. Kim, J. Ren, and L. He, "QuickYield: An efficient global-search based parametric yield estimation with performance constraints," in *Proc. Design Autom. Conf.*, 2010, pp. 392–397.

[10] A. Singhee and R. A. Rutenbar, "Statistical blockade: very fast statistical simulation and modeling of rare circuit events, and its application to memory design," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 28, no. 8, pp. 1176–1189, 2009.

[11] F. Gong, Y. Shi, H. Yu, and L. He, "Parametric yield estimation for SRAM cells: Concepts, algorithms and challenges," in *Proc Design Autom. Conf.*, 2010, pp. 1–12.

[12] R. Fragasse, R. Tantawy, B. Dupaix, T. Dean, D. Disabato, M. R. Belz, D. Smith, J. Mccue, and W. Khalil, "Analysis of SRAM Enhancements Through Sense Amplifier Capacitive Offset Correction and Replica Self-Timing," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 6, pp. 2037—2050, 2019.

[13] T. S. Doorn, J. A. Croon, E. J. W. ter Maten, and A. Di Bucchianico, "A yield centric statistical design method for optimization of the SRAM active column," in *Proc. ESSCIRC*, Athens, Greece, 2009, pp. 352–355.

[14] L. Dolecek, M. Qazi, D. Shah, and A. Chandrakasan, "Breaking the simulation barrier: SRAM evaluation through norm minimization," in *Proc. Int. Conf. Comput. Aided Des.*, San Jose, CA, USA, 2008, pp. 322–329.

[15] L. T. Clark, V. Vashishtha, L. Shifren, A. Gujja, S. Sinha, B. Cline, C. Ramamurthy, and G. Yeric, "ASAP7: A 7-nm finFET predictive process design kit," Microelectronics Journal, vol. 53, pp. 105-115, 2016.

[16] B. Wicht, T. Nirschl and D. Schmitt-Landsiedel, "Yield and speed optimization of a latch-type voltage sense amplifier," *IEEE J. Solid-State Circuits*, vol. 39, no. 7, pp. 1148-1158, July. 2004.

[17] L. Bagheriye, S. Toofan, R. Saeidi and F. Moradi, "Offset-Compensated High-Speed Sense Amplifier for STT-MRAMs," *IEEE Trans. VLSI Syst.*, vol. 26, no. 6, pp. 1051-1058, June. 2018.

[18] N. Damrongplasit, L. Zamudio, T. K. Liu and S. Balasubramanian, "Threshold Voltage and DIBL Variability Modeling Based on Forward and Reverse Measurements for SRAM and Analog MOSFETs," *IEEE Trans. Electron Devices*, vol. 62, no. 4, pp. 1119-1126, April. 2015.

[19] P. Jain and B. P. Das, "On-Chip Threshold Voltage Variability Detector Targeting Supply of Ring Oscillator for Characterizing Local Device Mismatch,"in *Proc 2019 IEEE 32nd International Conference on Microelectronic Test Structures (ICMTS)*, 2019, pp. 120-125.

[20] M. D. Giles, N. A. Radhakrishna, D. Becher, A. Kornfeld, K. Maurice, S. Mudanai, S. Natarajan, P. Newman, P. Packan, and T. Rakshit, "High sigma measurement of random threshold voltage variation in 14nm Logic FinFET technology," *in Proc 2015 Symposium on VLSI Technology (VLSI Technology)*, June. 2015, pp. T150-T151.

[21] K. J. Kuhn, M. D. Giles, D. Becher, P. Kolar, A. Kornfeld, R. Kotlyar, S. T. Ma, A. Maheshwari, and S. Mudanai, "Process Technology Variation," in *IEEE Trans. Electron Devices*, vol. 58, no. 8, pp. 2197-2208, Aug. 2011.

[22] D. Ha, C. Yang, J. Lee, S. Lee, S. Lee, K.-I. Seo, H. Oh, E. Hwang, S. Do, and S. Park, "Highly manufacturable 7nm FinFET technology featuring EUV lithography for low power and high performance applications," *2017 Symposium on VLSI Technology*, pp. T68-T69, 2017.

[23] S. Sun, X. Li, H. Liu, K. Luo, and B. Gu, "Fast Statistical Analysis of Rare Circuit Failure Events via Scaled-Sigma Sampling for High-Dimensional Variation Space," *IEEE Trans. on Comput. -Aided Design of Integr. Circuits Syst.,* vol. 34, no. 7, pp. 1096-1109, 2015.

[24] S. Sun and X. Li, "Fast statistical analysis of rare circuit failure events via subset simulation in high-dimensional variation space," in *Proc. Int. Conf. Comput. Aided Des.*, 2014, pp. 324-331.

GIDONG BAEK was born in Seoul, Korea in 1995. He received the B.S. degree in electronic engineering from Kwangwoon University, Seoul, Korea, in 2020. where he is currently pursuing the M.S. degree in electronic engineering. His research interests include SRAM peripheral circuit design, SRAM yield estimation, in memory computation and on-chip thermal sensor.

HANWOOL JEONG was born in Seoul, Korea, in 1987. He received the B.S in electrical and electronic engineering from Yonsei University, Seoul, Korea, in 2012. He received the Ph.D degree in electrical and electronic engineering from Yonsei University, Seoul, Korea, in 2017.

He was with Foundry Division of Samsung Electronics Company, Ltd., Hwaseong, Korea, from 2017 to 2019, where he was involved with the circuit design and verification of 4nm/5nm memory compiler. Since 2019, he has been a professor in Kwangwoon University, Seoul, Korea. His current research interests include memory circuit design, low-voltage/low power digital logic and neuromorphic & machine learning circuit design.