A Deconvolution Error Avoidance Technique in Richardson-Lucy Method

Hiroyuki Yamauchi¹ and Worawit Somha²
¹Fukuoka Institute of Technology, Wajiro-Higashi, Higashi-ku, Fukuoka, Japan
²King Mongkut’s Institute of Technology, Ladkrabang, Bangkok, Thailand
E-mail: yamauchi@fit.ac.jp

Abstract. A deconvolution error avoidance technique in Richardson-Lucy deconvolution (RL-deconv) is proposed, which is used for inversely analysing the SRAM margin variations caused by the Random Telegraph Noise (RTN). The proposed technique reduces the phase difference between the deconvoluted RTN distribution and feedback-gain in the maximum likelihood (MLE) gradient iteration cycles. This avoids an unwanted positive feedback, resulting in a significant decrease in probability of ringing occurrence. The effects of the proposed technique on the deconvolution process are demonstrated. A quicker convergence benefit of the RL-deconv algorithm is also observed. It has been demonstrated that the proposed technique reduces its relative deconvolution errors by 100 times compared with the conventional RL-deconv. This provides an increase in accuracy of the fail-bit-count prediction by over 2-orders of magnitude while accelerating its convergence speed by 33 times of the conventional one.

Keywords: Blind deconvolution, Random telegraph noise, Richardson-Lucy deconvolution

1. Introduction
Estimating the fail probability for the static random access memory (SRAM) is expected to become a significant challenge because the time-dependent SRAM failures caused by the operating margin variations cannot be considered any more by only the ordinary Gaussian based analyses [1-6].

Figures 1(a) and 1(b) explain the background behind the above change. Considering that the tail length of \( g \) is expected to become no longer ignored, \( h \) has to be estimated by the convolution of \( f \) with \( g \), i.e., \( h=f \ast g \). In addition, once the tail length of \( g \) becomes longer than that for \( f \), the tail distribution of \( h \) is governed by \( g \) not \( f \), as shown in Figs. 1(a) and 1(b). Where, \( f \) and \( g \) denote the distributions of the Random Dopant Fluctuation (RDF) and the Random Telegraph Noise (RTN) caused SRAM operating margin variations, respectively. The symbol of \( \ast \) represents for the convolution.

Since the shape of the distribution of \( g \) follows a lognormal distribution, the tail of \( h \) does not follow the Gaussian distribution any more. Thus, the ordinary Gaussian based analyses [1-6] cannot be used for estimating the fail probability. For specific understanding of the required analysis example, the following scenarios are recounted by using Fig. 2(a): (1) A certain \( h \) within the product target specification (SP\(_{\text{prod}}\)) is predefined. The \( f \) is truncated at a certain point (TP) based on the screening condition. As a result, the \( f \) is converted to \( f_{\text{TP}} \). (2) The distribution \( g \) is unknown. However, there is a
case where $g$ has to be estimated to set a target value for the RTN $g$ reduction from a device engineering development point of view.

Figure 2(b) recounts another scenario: (1) $SP_{prod}$ and $g$ are predefined. (2) The truncated $f_{TP}$ is unknown but should be estimated because $h$ has to be set within the target specification $SP_{prod}$ [6], as shown in Fig. 2(b). The distributions of $g$ and $f_{TP}$ can be calculated in theory by the following deconvolution: $g = h \otimes^{-1} f_{TP}$ and $f_{TP} = h \otimes^{-1} g$, respectively. The symbol of $\otimes^{-1}$ represents the deconvolution. However, the deconvolution of the $g$ or $f_{TP}$ is sort of ill-posed inverse problem. Thus, a troublesome operation has to be expected unlike the forward problem such as the convolution: $h = f_{TP} \otimes g$ (Fig. 1(a)) [5-7].

Figure 1. (a) Concept of convolution ($h = f \otimes g$) and (b) deconvolution ($g = h \otimes^{-1} f$). Forward and inverse problems have to be solved.

Figure 2. Potential scenarios for application of deconvolution
(a) Based on product and screening specifications, process target for RTN reduction is determined.
(b) Based on product and RTN specifications, screening condition TP is determined.

Figure 3. Comparison of deconvolution $g(x)$ of $h(x)$ with $f(x)$, i.e., $g(x) = h(x) \otimes^{-1} f(x)$ in case of (a) tail length of $f(x) > g(x)$ and (b) $f(x) < g(x)$ corresponding to the case for future beyond 10nm.
The most ordinary SRAM statistical analyses have no choice but to rely on the Gaussian model for simplicity as follows:

\[ f = N(\mu, \sigma), \quad g = N(\mu_e, \sigma_e), \quad h = N(\mu_h, \sigma_h) \]  

(1)

where \( N \) shows Normal (i.e., Gaussian) distribution function and \( \mu \) and \( \sigma \) are mean and deviation, respectively.

The convolution result \( h = N(\mu_h, \sigma_h) \) is simply given by

\[ \mu_h = \mu + \mu_e, \quad \sigma_h^2 = \sigma^2 + \sigma_e^2 \]  

(2)

The deconvolution \( g = \varphi(\mu_e, \sigma_e) = h \mathcal{F}^{-1} f \) is also simply given by

\[ \mu_e = \mu_h - \mu, \quad \sigma_e^2 = \sigma_h^2 - \sigma^2 \]  

(3)

Thus, conventionally, it was straightforward to estimate both of \( h = f \mathcal{G} \) and \( g = h \mathcal{F}^{-1} f \).

On the other hand, if the tail length of lognormal distribution \( g \) no longer accounts for just a fraction but a large percentage of the overall \( h \) (see Fig. 3(b)), the non-Gaussian inverse problem needs to be solved by a complex numerical calculation. However, it must not be easy to use in the manufacturing field. Thus, we have to find the best practical way to solve the non-Gaussian deconvolution process.

These two main reasons mentioned above lead to a significant pressure to develop the user-friendly methods for solving the non-Gaussian inverse problem so that the user can easily figure out the unknown factors [5-7], although the SRAM designers are unfamiliar with such kind of methodology.

Since the Richardson-Lucy deconvolution (RL-deconv) algorithm [7-9] is most widely used technique for image processing of all the MATLAB® built-in deconvolution functions, we did the applicability evaluation for the SRAM margin analyses before [10-12].

In those papers, the authors demonstrated huge errors in the exponentially decreasing tails of the deconvolution of \( g \). The papers concluded that the built-in functions cannot be used because these errors were found in all of the deconvolution tools provided by off-the-shelf MATLAB® [10-12].

To the best of our knowledge, there have been very little qualified published solutions to the ringing error problems with the deconvolution for the real SRAM margin analyses [10-12].

The almost relevant ideas for the ringing reduction described in the publications rely on the human optical illusion trick for image recognition. Thus, the required ringing suppression level is unfortunately quite low, i.e., that is 10-orders of magnitude smaller dynamic range than our cases.

The authors previously proposed the alternative algorithms to avoid the ringing [10-13] but it causes an intolerable additional iteration cycles as a side effect of the required fine segmentations [6].

In that sense, we still have to rely on the maximum likelihood (MLE) gradient base algorithm that provides a much faster convergence characteristic than that for previously proposed one.

Thus, in order to solve the ringing issues by all possible means, we experimentally tried to figure out what is behind the ringing in the RL iterations and what is the most sensitive procedure to ringing noise amplification.

In this paper, we demonstrate the ringing elimination for the first time based on the proposed algorithm to control the key procedure and parameters for the RL deconvolution.

The rest of the paper is organized as follows. Review for MATLAB® built-in RL-deconv algorithm in terms of what is behind the ringing in the RL iterations and what is the most sensitive procedure to ringing noise amplification in Section 2. The proposed ringing elimination technique is proposed in Section 3. The advantages over the conventional MATLAB® built-in RL-deconv-functions are demonstrated in section 4, followed by conclusion in section 5.

2. Issues of Conventional Algorithm
2.1. Richardson-Lucy deconvolution algorithm

The Richardson-Lucy (RL) deconvolution uses the maximum likelihood (MLE) gradient iterations with the multiple convolution \( \otimes \) processes, as shown in the expressions of (4) – (7).

The super script \( (t) \) for \( g, h, p, q \) denotes the number of iteration cycles

\[
g^{(t+1)} = g^{(t)} \times \left[ \frac{h (f \otimes g^{(0)}) \otimes f^\wedge}{h} \right] \\
g^{(t+1)} = g^{(0)} \times q^{(0)} \\
q^{(0)} = p^{(0)} \otimes f^\wedge \\
p^{(0)} = h \otimes h^{(0)} = h / h^{(0)}
\]

where, \( f^\wedge = \text{flipped } f \), \( h = g \otimes f \) and \( h^{(0)} = f \otimes g^{(0)} \)

2.2. Issues of RL-deconv algorithm

Because the convolution process plays a role of low-pass filtering, a high-frequency noise is suppressed. However, an amplified low-frequency noise (ringing of \( g^{(0)} \)) still remains, as shown in Fig. 4. The probable root causes of the noise amplification are shown in Fig. 5.

In this paper, this noise is referred to as “ringing error”.

**Figure 4.** Example of ringing of deconvoluted \( g^{(0)} \)

**Figure 5.** Comparison of the cases of (a) \( \phi = 0 \) (no ringing happen) and (b) \( \phi > 0 \) (ringing can happen)
The mechanism of the ringing in the iteration cycles can be explained as follows:

1. If the phase (x-position) error $\varphi$ between $g(t)$ and $q(t)$ becomes larger than a certain value (see Fig. 5(b) compared with 5(a)), the successive deconvoluted value $g^{(t+1)}$ is erroneously amplified. This is caused by the unwanted overlap by $\varphi$ between $g(t)$ and $q(t)$, as shown in Fig. 5(b). Where, the superscript of $(t)$ denotes the iteration numbers.

2. Mechanism of (1) causes a positive feedback to $g^{(t+1)}$ in the next cycle. Then larger ringing can be seen on the curves of $g^{(t+1)}$ as the number of iteration cycle increases.

Figure 6 shows the reason why and how much the phase shift by $\varphi$ is generated in the iteration cycles. The phase difference by $\varphi$ between $g^{(0)}$ (dash-line) and $q^{(0)}$ (outer solid line) occurs, as shown in Fig. 6. This is because the $\varphi$ is developed through the two successive convolution ($\otimes$) processes in every iteration cycles: 1st $\otimes$ in the process of $h(t) = f \otimes g(t)$, then 2nd $\otimes$ in the process of $q^{(0)} = (h/h^{(0)}) \otimes f^\circ$. As a result, the phase (x-position) of $q^{(0)}$ is shifted from that for $g^{(0)}$, as shown in Fig. 6.

3. Proposed Ringing Elimination Methods

In this section, a method for elimination of the phase error $\varphi$ of $q$ are explained. The key is to use a certain filter ALG that is convoluted with $p$ such that the position of $q = p \otimes \text{ALG}$ gets closer to the curve of $g^{(0)}$. The basic characteristic of the ALG distribution is the same as $f$ but its mean value is shifted by $\theta$.

The operation of $p \otimes \text{ALG}$ is used for the phase shifting so that the phase of $q$ can be aligned with $g^{(0)}$ by designing the mean-shift value $\theta$ of the ALG distribution, as shown in Fig. 7.

The proposed deconvolution process can be expressed by (8).

The difference from the conventional expression (4) is using ALG for $f^\circ$ in the proposed one.

\[
g^{(t+1)} = g^{(0)} \times [(h/(f \otimes g)) \otimes \text{ALG}] \quad (8)
\]
Thus, the key to successful elimination of the phase error $\phi$ is how to predict how much $\theta$ for ALG is needed for the phase alignment of $q$ with $g^0$, as shown in Fig. 7.

3.1. Design of key parameters $\phi$ and $\theta$

In this paper, how to determine the key parameters $\phi$ and $\theta$ are described by using Fig. 8 and Fig. 9.

The amount of phase error $\phi$ can be predicted based on predefined distributions of $f$ and $h$ as follows:

1. The average gradient values $\alpha_f$ and $\alpha_h$ for $f$ and $h$ are used because the value of $\phi$ depends on their values. For example, $\phi$ can be expressed by $\phi = k \times (\alpha_h - \alpha_f)$, as shown in Fig. 8.

2. The range of the value of $\theta$ where the phase error $\phi$ becomes low enough for ringing elimination is searched, as shown in Fig. 10. Then the relationship of $\phi$ and $\theta$ is drawn, as shown in Fig. 9(b).

As a result, the required mean shift $\theta$ of ALG for ringing elimination can be designed based on the average gradient of the distribution of given $f$ and $h$.

![Figure 8](image1)

(a) $\alpha_f$ and (b) $\alpha_h$ are average gradient of the distributions of $f$ and $h$.

(c) Relationships of $\phi$ and gradient difference ($\alpha_h - \alpha_f$) between predefined $f$ and $h$. $\phi$ depends on ($\alpha_h - \alpha_f$). $k$ is determined by the slope of dotted line shown in Fig. 9(a).

![Figure 9](image2)

(a) Relationships of $\phi$ and gradient difference ($\alpha_h - \alpha_f$) between predefined $f$ and $h$.

(b) Relationships of $\phi$ and required mean shift value $\theta$ by ALG.

Based on this relationships, $\theta$ can be determined for ringing elimination.
Figure 10 shows the RTN dependencies of required mean shift value ($\theta$). It is found that the range of required $\theta$ to eliminate ringing is wide enough and can tolerate margins of error from $\Delta \theta = 1$ to 2 as shown in Fig. 10. In addition, it depends on the slope of the RTN.

![Figure 10](image1.png)

**Figure 10.** RTN dependencies of required mean shift value $\theta$.
(a) RTN1, (b) RTN2, and (c) RTN3. No ringing happens at $\theta_1$, $\theta_2$, and $\theta_3$.

Three-cases of RTN1, RTN2, and RTN3 are assumed, which correspond to the scaling size of 40nm, 15nm, and 8nm in Fig. 11(a), respectively. Three sets of $\alpha$ and $\beta$ for $G(\alpha, \beta)$ for RTN1, RTN2, and RTN3 are assumed as shown in Fig. 11(b), respectively.

As used herein, $g$ and $f$ obey Gamma $G(\alpha, \beta)$ and Gaussian $N(\sigma, \mu)$ distributions, respectively. To normalize the amplitude ratio of $g$ to $f$, $\sigma$ and $\mu$ for $N(\sigma, \mu)$ is assumed 1.0 and 0, respectively. 

“x” of raw score in Fig. 11(b) corresponds to the z-number (i.e., number of $\sigma$ for Gaussian).

The operating voltage (VCC) can be used for “x”, as shown in Fig. 3.
3.2. Demonstration of deconvolution with ALG

The deconvolution results $g^{(r)}$ at each iteration cycle number $#$ of 1, 2, and 10 are shown in Fig. 12, while changing the mean shift $(\theta_A, \theta_B, \theta_C)$ for ALG, respectively.

It is exhibited that the ringing errors occur for the cases of $\theta_A$ and $\theta_B$. However, these errors are eliminated in the case of $\theta_C$ set within allowable zone, as shown in Fig. 12(a), (b), and (c), respectively.

The $q$ and $p$ for the case of $\theta = \theta_C$ becomes flat and their value are almost 1 in the overall range of $x$.

On the other hand, the ringing behaviors on the curves of $q$ and $p$ are seen in the cases of $\theta = \theta_A$ and $\theta_B$. It is found that an appropriate design of $\theta$ enables to eliminate the ringing behaviors, as shown in Fig. 12(c).

![Figure 12](image)

Figure 12. ALG mean shift ($\theta$) dependencies of the ringing errors.

left (a): $\theta_A = +2$, center (b): $\theta_B = -2.8$, right (c): $\theta_C = -1.2$

Comparisons of deconvolution $g(t)$ at different iteration cycles of #1 (top), #2 (middle), and #10 (bottom).
Figure 13. Demonstrations of ringing elimination for (a) RTN1, (b) RTN2, and (c) RTN3. Mean shift of ALG for RTN1, RTN2, and RTN3 are $\theta_1 = -1.8$, $\theta_2 = -3.5$ and $\theta_3 = -7.0$, respectively.

Figure 13 demonstrates the $g^0$ deconvolution results for (a) RTN1, (b) RTN2, and (c) RTN3, respectively. The mean shifts of ALG for RTN1, RTN2, and RTN3 are set to $\theta_1 = -1.8$, $\theta_2 = -3.5$ and $\theta_3 = -7.0$, respectively. Those values of $\theta$ are within allowable zones, as shown in Fig. 10.

It is found that appropriate designs of $\theta$ depending on the different RTN distributions enable to eliminate the ringing behavior for (a) RTN1, (b) RTN2, and (c) RTN3, respectively.

4. Deconvolution Errors and Its Error Impact on SRAM Fail Bit Count Prediction Accuracy

The comparisons of the deconvolution errors and its impact on the SRAM fail-bit count (FBC) prediction accuracy between this work and the conventional one are discussed in this section.

4.1. SRAM Fail-bit Count Estimation

Figure 14. Concept of fail bit count (FBC) errors caused by deconvolution errors.

(a) $\text{FBC}_{\text{exp}}$ at each x-point is given by integration of expected $h=fg$.

(b) If extracted $h^0$ is deviated from $h$, $\text{FBC}_{\text{extr}}$ (right figure) has to be expected.
Figure 14 shows the concept of the FBC errors caused by deconvolution errors of $g^{(t)}$. FBC_{EXP}(x_p) represents the integration of the expected $h = f \ast g$, i.e., the curve of the cumulative density function (cdf) of $h$ (see Fig. 14 (a)). Thus, if extracted $h^{(t)} = f \ast g^{(t)}$ is deviated from the expected $h$ (see Fig. 14 (b)), the FBC_{EXTR}(x_p) at $x_p$ is deviated from the FBC_{EXP}(x_p). If FBC_{EXTR}(x_p) is larger (smaller) than FBC_{EXP}(x_p), this means an over-estimation (under-estimation).

The attention point of raw score $x_p$ is determined by the total bits so that the probability density function (pdf) at $x_p$ can corresponds to the 1-bit fail probability.

Herein, total bit of $10^6$ (1000chips x 1Kbit) to $10^{15}$ (1000 chips x 1Tbit) are assumed. For example, 1bit fail probability of 1000chips of 1Kbit, 1Mbit, 1Gbit, and 1Tbit are $1/10^6$, $1/10^9$, $1/10^{12}$, and $1/10^{15}$ respectively.

Thus, the $x_p$ for the cases of total bit of $10^6$ is smaller than that for the case for $10^{15}$.

Figure 15 shows the SRAM memory density dependencies of the FBC prediction errors.

![Graph](image)

Figure 15. SRAM memory density dependencies of fail bit count (FBC) errors.

As you can see in Fig. 15, the fluctuation of $h^{(t)}$ caused by $g^{(t)}$ ringing increases over-estimated or under-estimated FBC prediction errors. This work eliminates the ringing, resulting in much less over/under estimation.

The relative deconvolution error of $g^{(t)}$ across raw score $x$ for (a) RTN1, (b) RTN2, and (c) RTN3 are compared between conventional and this work, as shown in Fig. 16. It is found that this work can reduce the relative deconvolution error of $g^{(t)}$ by 2-3 orders of magnitude compared with the conventional one. It is noticed that since the polarity (plus+ and minus-) of the errors are ignored in the relative error plot, the amplitude of $g^{(t)}$ ringing error looks smaller than that of Fig. 12.

Figure 17 shows the comparisons of cumulative density function cdf($x_p$) at $x_p$ where the pdf of golden $h(x)$ is $10^{-12}$ for RTN1, RTN2, and RTN3 between the conventional and this work. The cdf($x_p$) value corresponds to the predicted fail-bit counts for 1000-chips of 1Gbit SRAM.
It is found that this work can reduce both of the errors of FBC prediction and required iteration cycles for convergence, as shown in Fig. 17.

**Figure16.** Comparisons of relative deconvolution error of $g^{(t)}$ across raw score $x$ for (a) RTN1, (b) RTN2, and (c) RTN3 between conventional and this work.

**Figure17.** Comparisons of cdf error of $h^{(t)} = g^{(t)} \otimes f$ and its iteration cycles required for convergence for (a) RTN1, (b) RTN2, and (c) RTN3 between conventional and this work.
5. Summary and Discussions

The proposed ringing prevention technique successfully circumvents the ringing error thanks to reducing the phase difference between the feedback-gain \( q \) and deconvolution target distributions \( g(t) \) in RL-deconv iteration cycles. It is found that the proposed one can avoid any unwanted positive feedbacks, resulting in no error amplification. It has been shown that the proposed technique reduces its relative errors of the RTN deconvolution by \( 10^2 \sim 10^3 \) times compared with the conventional RL-deconv. This enables to increase accuracy of the fail-bit-count prediction based on the cdf of the convolution (\( h(t) = g(t) \ast f \)) of the RTN \( g(t) \) with the RDF \( f \) by over 2-orders of magnitude while accelerating its convergence speed by 7~30 times of the conventional one.

References