An Error Reduction Technique in Richardson-Lucy Deconvolution 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. An error reduction technique for Richardson-Lucy deconvolution (RL-deconv) is proposed. The deconvolution is indispensable technique for inversely analysing the SRAM fail-bit probability variations caused by the Random Telegraph Noise (RTN). The proposed technique reduces the phase difference between the two distributions of the deconvoluted RTN and the feedback-gain in the maximum likelihood (MLE) gradient iteration cycles. This avoids an unwanted positive feedback, resulting in a significant decrease in probability of undesired ringing occurrence. A quicker convergence benefit of the RL-deconv algorithm while avoiding the ringing is achieved. 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 33times of the conventional one.

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

1. Introduction
Ordinary brief method for estimating the bit fail probability for the static random access memory (SRAM) is expected to be no longer applicable because the simple Gaussian based analyses method [1-6] cannot be used any more. The main reasons for this change are: (1) the tail distribution doesn’t obey Gaussian any more. It does follow log-normal distribution and (2) the tail length of the distributions of the SRAM operating margin will be no longer negligible small fraction.

Figures 1(a) and 1(b) compare the tail length of the log-normal distribution g. Once the tail length of g becomes longer than that for f, the shape of the tail distribution of h is governed by g not f, as shown in Fig. 1(b). Where, f and g denote the distributions of the Random Dopant Fluctuation (RDF) and the Random Telegraph Noise (RTN), respectively [1-3]. The symbols of ⊗⁻¹ and ⊗ represent for the operators for deconvolution and convolution, respectively.

Considering that the tail length of the distribution g is expected to become no longer ignored, the tail distribution of h will have to be estimated precisely by the convolution (⊗) of f with g, i.e., h=f⊗g.

Figures 2(a) and 2(b) compare the relationships among f, g, and h between the convolution and the deconvolution, respectively. Since the shape of the distribution of g follows a log-normal distribution, the tail distribution of h cannot be approximated by the Gaussian distribution any more. Thus, the ordinary Gaussian based analyses [1-6] cannot be used for estimating the tail of the h distribution.
Figure 1. 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.

Figure 2. (a) Concept of convolution ($h = f \otimes g$) and (b) deconvolution ($g = h \otimes^{-1} f$). Forward and inverse problems can be solved by convolution and deconvolution, respectively.

The most ordinary SRAM statistical analyses method have no choice but to rely on the Gaussian model for simplicity as follows:

$$f = N(\mu_f, \sigma_f), \quad g = N(\mu_g, \sigma_g), \quad h = N(\mu_h, \sigma_h)$$  \hspace{1cm} (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

$$\mu_h = \mu_f + \mu_g, \quad \sigma_h^2 = \sigma_f^2 + \sigma_g^2$$  \hspace{1cm} (2)

The deconvolution $g = \phi(\mu_g, \sigma_g) = h \otimes^{-1} f$ is also simply given by

$$\mu_g = \mu_h - \mu_f, \quad \sigma_g^2 = \sigma_h^2 - \sigma_f^2$$  \hspace{1cm} (3)

Thus, it was straightforward to estimate both of convolution ($h = f \otimes g$) and deconvolution ($g = h \otimes^{-1} f$) under the assumption that both of the distributions $f$ and $g$ obey Gaussian.

On the other hand, if the tail length of the log-normal distribution $g$ no longer accounts for just a fraction but a large percentage of the overall $h$ (see Fig. 1(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.

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.

Thus, in this paper, we proposed and discussed the practical way to solve the non-Gaussian deconvolution process based on the Richardson-Lucy deconvolution (RL-deconv) algorithm [7-9]. This is because the RL-deconv algorithm 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] and demonstrated huge errors in the exponentially decreasing tails of the deconvoluted $g$. The papers concluded that huge errors are observed in all of the deconvolution tools provided by off-the-shelf MATLAB® [10-12] and its error level is not acceptable for the accuracy request point of view of the target application.
To the best of our knowledge, there have been very little qualified published solutions to the ringing error problems with the deconvolution for the SRAM fail-bit probability analyses [10-12]. The almost relevant ideas described in the publications rely on the human optical illusion trick for the image recognition. Thus, the required ringing suppression level is unfortunately quite low, \( i.e., \) that is 10-orders of magnitude smaller dynamic range than that for the SRAM fail-bit probability.

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

In that sense, we still have to rely on the maximum likelihood (MLE) gradient base RL-deconv algorithm [7-9] because it provides a much faster convergence characteristic than that for the previously introduced segmentation methods. Thus, we have to solve the ringing issues in the RL-deconv algorithm by all possible means so that the benefits from the RL algorithm can be well exploited.

In this paper, we experimentally tried to figure out the root cause for the ringing and we demonstrated the ringing elimination for the first time based on the proposed algorithm to control the key procedure and parameters for the RL deconvolution.

For specific understanding of the application examples, the following deconvolution scenarios are recounted by using Fig. 3(a): (1) A certain \( h \) within the product target specification (\( \text{SP}_{\text{prod}} \)) is predefined. The \( f \) is truncated at a certain point (\( TP \)) based on the screening condition. As a result, the \( f \) is converted to the truncated \( f_{\text{TP}} \). (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 characteristics improvement point of view. Figure 3(b) recounts another scenario: (1) \( \text{SP}_{\text{prod}} \) and \( g \) are predefined. (2) The truncated \( f_{\text{TP}} \) is unknown but should be estimated because \( h \) has to be set within the target specification \( \text{SP}_{\text{prod}} \) [6], as shown in Fig. 3(b). The distributions of \( g \) and \( f_{\text{TP}} \) can be calculated in theory by the following deconvolution: 
\[ g = h \otimes^{-1} f_{\text{TP}} \] and 
\[ f_{\text{TP}} = h \otimes^{-1} g, \] respectively. However, the deconvolution of the \( g \) or \( f_{\text{TP}} \) is sort of ill-posed inverse problem. Thus, a troublesome operation has to be expected unlike the convolution: 
\[ h = f_{\text{TP}} \otimes g \] (Fig. 2(a)) [5-7].

The rest of the paper is organized as follows. Review for the MATLAB\textsuperscript{®} 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 the ringing noise amplification in Section 2. The proposed ringing elimination technique is proposed in Section 3. The advantages over the conventional MATLAB\textsuperscript{®} built-in RL-deconv functions are discussed based on the simulation results 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 algorithm uses the maximum likelihood (MLE) gradient iterations with the multiple convolution \((\otimes)\) processes, as shown in the expressions of (4) – (8). The super script \((t)\) for \(g, h, p\) and \(q\) denotes the number of iteration cycles.

\[
g^{(t+1)} = g^{(0)} \times \left( \frac{h}{f \otimes g^{(t)}} \right) \otimes f^\wedge \tag{4}
\]

\[
g^{(t+1)} = g^{(0)} \times q^{(0)} \tag{5}
\]

\[
q^{(0)} = p^{(0) \otimes f^\wedge} \tag{6}
\]

\[
p^{(0)} = h / ( f \otimes g^{(0)}) = h / h^{(0)} \tag{7}
\]

where, \(f^\wedge = \text{flipped } f\), \(h = g \otimes f\) and \(h^{(0)} = f \otimes g^{(0)}\)\(\tag{8}\)

2.2. Issues of RL-deconv algorithm
Figure 4 shows the simulated waveforms for \(p^{(0)}, q^{(0)}, g^{(0)}, h^{(0)}\) expressed by the above (4) – (8). Because the convolution process plays a role of low-pass filtering, a high-frequency noise on the waveforms are eliminated. However, a low-frequency noise amplification on the \(g^{(0)}\) (ringing of \(g^{(0)}\)), i.e., the ringing noise issue still remains, as shown in Fig. 4.

The probable root cause of the noise amplification is conceptually illustrated in Fig. 5.
In this paper, this noise amplification is referred to as “ringing error”.

![Figure 4](image-url)

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

![Figure 5](image-url)

**Figure 5.** Comparison of the cases of (a) \(\varphi = 0\) (no ringing happen) and (b) \(\varphi \neq 0\) (ringing can happen)
The mechanism of the ringing in the iteration cycles can be explained as follows:
(i) If the phase (x-position) difference \( \phi \) between \( g^0 \) and \( q^0 \) becomes larger than a certain value (see Fig. 5(b) compared with Fig. 5(a)), the successive deconvoluted value \( g^{(t+1)} \) is erroneously amplified. This is caused by the unwanted overlap by \( \phi \) between the \( g^0 \) and \( q^0 \), as shown in Fig. 5(b). Where, the superscript of \( (t) \) denotes the numbers of iteration cycles.
(ii) Mechanism of (i) is caused by a positive feedback to \( g^{(t+1)} \) in the next cycle. Then the noise amplification behavior can be seen on the curves of \( g^{(t+1)} \) in the iteration cycles.

Figure 6 shows the reason why the phase difference (x-position) between \( g^0 \) (dash-line) and \( q^0 \) (outer solid line) is made in the deconvolution processes expressed by equations (4)-(8).

The phase difference by \( \phi \) is developed through the two successive convolution (\( \otimes \)) processes in every iteration cycles: the 1st \( \otimes \) process in \( h^0 = f \otimes g^0 \), then the 2nd \( \otimes \) process in \( q^0 = (h/h^0) \otimes f^\Lambda \) (See the expressions of (4) – (8)). As a result, the x-position of \( q^0 \) is shifted by \( \phi \) from that for \( g^0 \), as shown in Fig. 6. As used herein, “x” in Fig. 6 corresponds to the number of \( \sigma \) for Gaussian. The two distributions \( g \) and \( f \) differently obey the Gamma \( \mathcal{G}(\alpha, \beta) \) and Gaussian \( \mathcal{N}(\sigma, \mu) \), respectively. To plot the two in the same x-y plot, the distribution parameters are normalized. For example, the \( g \) distribution is plotted normalized to the \( f \) distribution of \( \mathcal{N}(\sigma=1.0, \mu=0) \).

In this paper, in order to demonstrate the ringing elimination for the first time, we propose the algorithm for minimizing the phase error \( \phi \) with the control of the key procedure and parameters in the RL deconvolution in the following sections.

3. Proposed Ringing Elimination Methods
In this section, a method for elimination of the phase shift \( \phi \) of \( q^0 \) is explained.
The key is to use a certain filter \( \text{ALG} \) that is convoluted with \( p \) such that the position of \( q^0 \) (\( =p^0 \otimes\text{ALG} \)) gets closer to the curve of \( g^0 \). The \( \sigma \) value of the \( \text{ALG} \) distribution is the same as \( f \) but its mean value \( \mu \) is shifted by \( \theta \) (\( \mu=\theta \)), as shown in Fig. 7.

The operation of \( p^0 \otimes\text{ALG} \) is used for the phase shifting so that the phase of \( q^0 \) can be aligned with \( g^0 \) by designing the mean-shift value \( \theta \) of the \( \text{ALG} \) distribution, as shown in Fig. 7.
The proposed deconvolution process can be expressed by the following expression (9).

\[ g^{(t+1)} = g^{(t)} \times \left[ \left( \frac{h}{(f \otimes g)} \right) \otimes \text{ALG} \right] \]  

(9)

The difference from the conventional expression (4) is to introduce the distribution of $\text{ALG}$ for $f^\circ$ in (9). Thus, the key to successful elimination of the phase shift $\varphi$ is how to predict how much $\theta$ for $\text{ALG}$ is needed for the phase alignment of $q^{(t)}$ with $g^{(t)}$, as shown in Fig. 7.

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

In this subsection, how to determine the key parameters $\varphi$ and $\theta$ are described by using Figs. 8 and 9.

The amount of phase shift $\varphi$ 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 $\varphi$ depends on their values. For example, $\varphi$ can be expressed by $\varphi = k \times (\alpha_h - \alpha_f)$, as shown in Fig. 8.

   Where, $k$ is determined by the slope of the dotted line, as shown in Fig. 9(a).

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

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

![Figure 8](image)

**Figure 8.** (a) $\alpha_f$ and (b) $\alpha_h$ are average gradient of the distributions of $f$ and $h$.

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

![Figure 9](image)

**Figure 9.** (a) Relationships of $\varphi$ and gradient difference ($\alpha_h - \alpha_f$) between predefined $f$ and $h$.

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

Based on these relationships, $\theta$ can be determined for ringing elimination.
In order to discuss the tail length dependencies, three different tail lengths for RTN1, RTN2, and RTN3 distributions (see Fig. 10(a)) are assumed, which correspond to the scaling size of 40nm, 15nm, and 8nm in Fig. 10(b), respectively. Three sets of $\alpha$ and $\beta$ for $G(\alpha, \beta)$ for RTN1, RTN2, and RTN3 are assumed as shown in Fig. 10(a), respectively. As used herein, $g$ and $f$ obey Gamma $G(\alpha, \beta)$ and Gaussian $N(\sigma, \mu)$ distributions, respectively. The $G(\alpha, \beta)$ is plotted normalized to $f$ with $N(\sigma=1.0, \mu=0)$.

“x” of raw score in Fig. 10(a) corresponds to the $z$-number (i.e., number of $\sigma$ for Gaussian).

The operating voltage (VCC) is one of examples being used for “x”, as shown in Fig. 1.

![Figure 10](image1.png)

**Figure 10.** (a) Probability density function comparisons among RDF, RTN1, RTN2, and RTN3. (b) Trend of variation amplitude comparison between RDF $f(x)$ and RTN $g(x)$.

Figure 11 shows the RTN tail length dependencies of the required $\theta$. It is found that the allowable range of the required $\theta$ (i.e., margin of $\theta$) is wide enough to tolerate the error from $\Delta \theta = 1$ to 2, as shown in Fig. 11. In addition, it depends on the slope (i.e., tail length) of the RTN distribution.

![Figure 11](image2.png)

**Figure 11.** RTN dependencies of required mean shift value $\theta$.
(a) RTN1, (b) RTN2, and (c) RTN3. No ringing happens in the zone for $\theta_1$, $\theta_2$, and $\theta_3$. 
3.2. Demonstration of deconvolution with ALG

The deconvolution results $g^{(t)}$ at each iteration cycle number ($\#$) of 1, 2, and 10 are shown in Figs. 12(a), (b), and (c) (top to bottom), while changing the mean shift ($\theta_A, \theta_B, \theta_C$) for ALG (left to right), respectively.

It is exhibited that the ringing errors on $g^{(t)}$ are amplified for the cases of $\theta_A$ and $\theta_B$ as the iteration cycles number $\#$ is increased from 1 to 10, as shown in Figs 12(a) and (b), respectively.

On the other hand, these errors are fully suppressed in the case of $\theta_C$ set within allowable zone, as shown in Fig. 12(c). The $q$ and $p$ for the case of $\theta = \theta_C$ becomes flat at the $\# = 10$ and their value are almost 1 (i.e., achieved convergence) in the overall range of $x$, as shown in Fig. 12(c). It is found that an appropriate design of $\theta$ enables to eliminate the ringing behaviors.

Compared with the Fig. 12(c), the distributions for $q$ and $p$ in the cases of $\theta = \theta_A$ and $\theta_B$ are fluctuated, as shown in Figs 12(a) and (b). This is caused by the phase misalignment of the feedback gain $q^{(t)}$ with the deconvoluted $g^{(t)}$ as explained in the previous section.

![Figure 12. ALG mean shift ($\theta$) dependencies of the ringing behaviors (left to right).
left (a): $\theta_A = +2$, center (b): $\theta_B = -2.8$, right (c): $\theta_C = -1.2$
Iteration cycles dependencies of deconvolution $g^{(t)}$ distributions (up to down)
Iteration cycles $\#$ = 1 (top), 2 (middle), and 10 (bottom), respectively.](image-url)
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 $g^{(t)}$ 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 deconvolution errors and its impact on the SRAM fail-bit count (FBC) prediction accuracy are compared between this work and the conventional one in this section.

4.1. SRAM Fail-bit Count Estimation

Figure 14 shows the concept of how does the deconvolution errors of $g^{(t)}$ gives impact on the FBC prediction errors. $\text{FBC}_{\text{EXP}}(x_p)$ for the golden case (expected no $g^{(t)}$ error case) represents the FBC at $x_p$. FBC is given by the integration of the curve of the cumulative density function (cdf) of $h = f \otimes g$ from $+\infty$ to $x_p$ (see Fig. 14 (a)). This value corresponds to the fail bit probability. Thus, if the extracted $h^{(t)}$ line
is deviated from the golden line of the $h$ (see Fig. 14 (b)), the value of $FBC_{EXTR}(x_p)$ becomes different from that for the golden value of $FBC_{EXP}(x_p)$. This results in the FBC prediction errors. If the value of $FBC_{EXTR}(x_p)$ is larger or smaller than that for $FBC_{EXP}(x_p)$, this corresponds to the case for an over-estimation or under-estimation of FBC, respectively.

**Figure 14.** Concept of fail bit count (FBC) errors caused by deconvolution errors.
(a) $FBC_{EXP}$ at each x-point is given by integration of expected $h=f \otimes g$.
(b) If extracted $h^{(0)}$ is deviated from $h$, $FBC_{EXTR}$ (right figure) has to be expected.

**Figure 15.** SRAM memory density dependencies of fail bit count (FBC) errors.
Herein, the total number of SRAM bits of $10^6$ (1000 chips x 1Kbit) to $10^{15}$ (1000 chips x 1Tbit) are assumed. Thus, the only 1bit fail probability for 1000 chips of 1Kbit, 1Mbit, 1Gbit, and 1Tbit are $1/10^6$, $1/10^9$, $1/10^{12}$, and $1/10^{15}$, respectively.

Thus, the position of $x_p$ depends on the total memory bits. For example, the position of $x_p$ for $10^{15}$ is shifted to the right edge side of the tail compared with that for the case for $10^6$. This means that the accuracy in the tail region becomes more important as the number of SRAM bits (density) increase. Figure 15 shows the SRAM memory density dependencies of the FBC prediction errors.

As you can see in Fig. 15, the fluctuation of $h(t)$ caused by $g(t)$ ringing increases the over-estimated or under-estimated FBC prediction errors. On the other hand, this work well eliminates the ringing, resulting in much less over/under estimation for the case of RTN1, RTN2, and RTN3 distributions.

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 the polarity (+ and -) of the errors are ignored in the relative error plot.

![Figure 16](image)

**Figure 16.** 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.

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.
Figure 17. Comparisons of cdf error of $h^{(k)} = g^{(0)} \otimes f$ 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 $g^{(0)}$ 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 ($h^{(k)} = g^{(0)} \otimes f$) of the RTN $g^{(0)}$ with the RDF $f$ by over 2-orders of magnitude while accelerating its convergence speed by 7~30 times of the conventional one.

References

[1] K. Takeuchi, et al, “Comprehensive SRAM Design Methodology for RTN Reliability”, Digest of IEEE Symposium on VLSI Technology, (2011), pp. 130-131

[2] X. Wang, et al, “RTS amplitude distribution in 20nm SOI FinFETs subject to Statistical Variability”, SISPAD 2012, (2012) pp.296-299

[3] X. Wang, et al, “Simulation Study of Dominant Statistical Variability Sources in 32-nm High-k/Metal Gate CMOS”, IEEE Electron Device Letters, vol.33, no.5, pp. 643-645, 2012
[4] J. Franco, et al, “RTN and PBTI-induced Time-Dependent Variability of Replacement Metal-Gate High-k InGaAs FinFETs”, IEEE Electron Device Meeting, Dec. 2014

[5] Q. Shi, et al, “Application of Iterative Deconvolution for Wire Fault Location via Reflectometry”, Digest of IEEE Int. Symp. on Instrumentation & Measurement, Sensor Network and Automation (IMSNA), (2012), pp. 102-106

[6] W. Somha, H.Yamauchi, “Convolution/Deconvolution SRAM Analyses for Complex Gamma Mixtures RTN Distributions”, Digest of IEEE ICICDT-2013,(2013), pp. 33-36

[7] Fish D. A. et al, "Blind deconvolution by means of the Richardson–Lucy algorithm", Journal of the Optical Society of America A 12 (1): pp.58–65

[8] White R. L, "Image Restoration Using the Damped Richardson-Lucy Method" The Restoration of HST Images and Spectra II, Space Telescope Science Institute, 1994, pp.104-110

[9] F. Dell'Acqua1, "A Modified Damped Richardson-Lucy Algorithm to Improve the Estimation of Fiber Orientations in Spherical Deconvolution", Proc. Intl. Soc. Mag. Reson. Med. 16 (2008)

[10] H.Yamauchi, W. Somha, “Comparative Study on Deconvolution Function Dependencies of RTN/RDF Effect Estimation Errors in Analyzing Sub-nm-Scaled SRAM Margins", IEEE MWSCAS 2014, pp.230-233 (2014).

[11] H.Yamauchi, W. Somha, “Errors in Solving Inverse Problem for Reversing RTN Effects on VCCmin Shift in SRAM Reliability Screening Test Designs”, IEEE SOCC 2014, pp.318-323 (2014)

[12] H.Yamauchi, W. Somha, “Deconvolution Algorithm Dependencies of Estimation Errors of RTN Effects on Subnano-Scaled SRAM Margin Variation”, IFIP/IEEE VLSI-SoC 2014, pp.97-102 (2014)

[13] H.Yamauchi, W.Somha , “Ringing Error Prevention Techniques in Lucy-Richardson Deconvolution Process for SRAM Space-Time Margin Variation Effect Screening Designs”, 16th IEEE Latin-American Test Symposium (2015)