Electromigration Checking Using a Stochastic Effective Current Model

Adam Issa
ECE Dept., University of Toronto,
Toronto, ON, Canada
adam.issa@mail.utoronto.ca

Valeriy Sukharev
Mentor, A Siemens Business,
Fremont, CA, USA
valeriy_sukharev@mentor.com

Farid N. Najm
ECE Dept., University of Toronto,
Toronto, ON, Canada
f.najm@utoronto.ca

ABSTRACT

Electromigration (EM) degradation evolves slowly towards failure, over a period of years. This is why EM checking methods use effective current models to represent the underlying circuit workload, which are typically constant (DC) currents over time. However, ignoring all input current variations around the mean can be risky, because low-frequency input variations can have a significant impact on EM, resulting in shorter than expected lifetimes. With the use of dark silicon and multimodal chip operation, such low-frequency changes in workload are becoming increasingly common in modern designs. Ignoring these variations can lead to false positives and must be avoided. We tackle this by developing a stochastic effective current model for the input current waveforms that is easy for users to specify and which allows stochastic estimation of the impact of input variability on the lifetime. User-provided guidance on the expected durations of various modes of operation is used to provide input current variances, which are then propagated to provide variances around the stress waveforms in the metal network, which gives a more realistic estimate of the EM lifetime. Variance propagation can be expensive for large systems, but a novel simulation-like framework will be presented that allows efficient variance propagation for large interconnect trees. This has revealed that the variance can be highly significant. Even when the standard deviation of the inputs is small, at around 20–30% of the mean, we see a 30–40% drop in the lifetimes.

KEYWORDS

Electromigration, power grid, stress, void, voltage-drop, variance.

ACM Reference Format:
Adam Issa, Valeriy Sukharev, and Farid N. Najm. 2020. Electromigration Checking Using a Stochastic Effective Current Model. In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD '20)*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3400302.3415635

ACKNOWLEDGMENTS

This work was supported in part by the Semiconductor Research Corporation (SRC) and by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

1 INTRODUCTION

In a metal line carrying significant current density, as often happens in integrated circuits (IC), the free electrons push and move the metal atoms in the direction of the electron wind, i.e., towards the anode end of the line; hence the name *electromigration* (EM) for this effect. The resulting atomic flow increases compressive stress at the anode and tensile stress at the cathode. If the levels of stress become high enough, a void may be created due to high tensile stress near the cathode, or a hillock (extrusion of metal through cracks in the dielectric) may form due to high compressive stress near the anode, which can either way result in circuit failure. With the confinement of metal lines in today's metal technology, voids are much more likely than hillocks and so one is often more concerned with the buildup of tensile stress. A void is created once the stress exceeds a certain level of stress, called the *critical stress*. The effect is slow and cumulative, and may take years to manifest as a chip failure.

While signal and clock lines do suffer from EM degradation, these lines carry bidirectional current and so have longer lifetimes due to healing. In contrast, metal lines in the power distribution network (PDN) carry mostly unidirectional current with no benefit of healing and so are more susceptible to EM failure. Thus our work is focused on EM in the PDN. The PDN consists of the power grid and the ground grid. Modern grids span multiple layers (often all the layers) of metallization and they consist of meshes of power and ground lines. Without loss of generality, we will focus on the power grid. Due to EM, metal atoms can travel between different connected branches on the same layer. However, they cannot travel through a via to other metal layers above and below, because of the metal liner under every via which acts as a barrier to atomic movement, but allows electron movement. As a result, EM-induced metal transport within a layer remains within that layer, so that the overall analysis problem is decomposed into sub-problems on different layers, corresponding to metal structures that are physically disconnected within any given layer. The vast majority of these planar structures turn out to be trees, i.e., they have no cycles. So, it is typical in the field to simply use the term *interconnect trees* to refer to these metal islands. In an interconnect tree, the nodes where two or more lines meet, or where a line ends, are called *junctions*. Junctions are where voids are most likely to nucleate, and so they are "nodes of interest." Because of the mesh structure of modern power grids, even though the first void may increase the resistance in the grid, it may not lead to an actual failure. A void causes a failure only if it results in a voltage-drop violation, and that may not happen until several voids have nucleated.

As a result of continued scaling of IC technology, EM has become a major reliability concern in modern design [11]. Today, it

ICCAD '20, November 2–5, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-8026-3/20/11…$15.00
https://doi.org/10.1145/3400302.3415635

is becoming harder to sign off on chip designs using state-of-the-art EM checking tools, as there is very little margin left between the predicted EM stress and that allowed by EM design rules [10]. This loss of safety margin can be traced back to the inaccurate and oversimplified nature of EM models used by existing tools. Standard practice in the industry is to break up a grid into isolated metal branches, then assess the reliability of each branch *separately* using Black's model [1]. This approach is inaccurate, for a number of reasons. For one thing, being an empirical model, the fitting parameters obtained for Black's model under accelerated testing conditions are not valid at actual operating conditions, and this leads to significant errors in lifetime extrapolation [3, 8]. Modern physics-based EM models do not have this problem. In addition, because they handle every line separately, in isolation, methods based on Black's model ignore the material flow between branches. In today's mesh structured power grids, atomic flux can flow freely between the branches of an interconnect tree and this has a big impact on the stress distribution and void nucleation.

1.1 Stress-based EM Simulation

Physics-based models for EM like Korhonen's model [5] have been found to be quite efficient as they are 1-dimensional (1D) and lend themselves to both analytical and numerical solutions. Once extended with boundary conditions, Korhonen's model becomes applicable to multi-line structures like interconnect trees, and this has been done in the recent past by several groups, including the EKM model (stands for extended Korhonen's model) by Chatterjee et al. [2]. EKM applies to whole interconnect trees, which overcomes the aforementioned inaccuracy of branch-by-branch analysis. Korhonen's equation is discretized in space and time to provide, for every interconnect tree, a linear time-invariant (LTI) system of the form

$$\dot{x} = Ax + \hat{B}\hat{u}, \tag{1}$$

where x is a vector of stress values at discretized nodes in the interconnect, A and $\hat{B}$ are the system and input matrices respectively and $\hat{u}$ is the input vector consisting of the terminal currents of the interconnect tree. Alternatively, as recently done in [4], the LTI system can be expressed in terms of the grid terminal currents, and this will be more useful for our needs, so we will be using the transformed system

$$\dot{x} = Ax + Bu, \tag{2}$$

where B is a matrix that includes dependence on the conductance matrix of the grid and u is a vector of the grid terminal currents. By numerically simulating these LTI systems, one for each interconnect tree, one can track the stress levels in the grid over time, and thus determine the times and locations of the sequence of void nucleations. Effectively, this leads to a stress-based *EM simulation* for on-chip metal structures. During the simulation, the voltages in the power grid are tracked until failure is declared when the voltage drop at any of the grid's port nodes exceeds some user-specified threshold. This provides a Time-To-Failure (TTF) estimate. However, the system and input matrices are functions of the branch diffusivities, which are random variables. Therefore, the randomness of branch diffusivities is taken care of by running a Monte Carlo loop that ultimately provides the mean time-to-failure (MTF) of the grid.

1.2 The Effective Current Problem

Electromigration damage is slow and cumulative over time, and damage may occur after a few years, or never, depending on the level of current density, the geometry of the lines, the temperature and the metal technology. In addition, for individual metal lines in test structures, many studies have been done over the last 50 years to assess the impact of the shape of the current waveform over time, be it DC, alternating, unidirectional pulsed, or other.

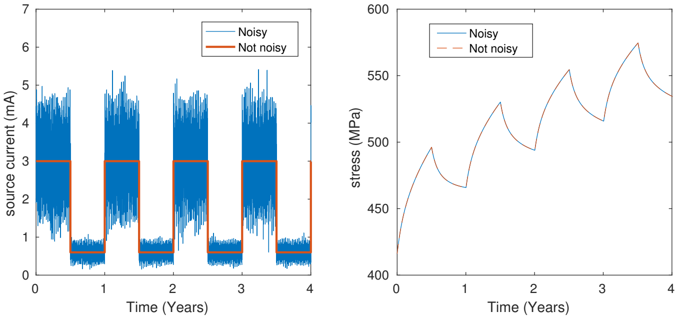
In using methods like EKM to predict the damage to chip power grids, an immediate question arises: what current stimulus should be applied at the current sources that load the grid? Specifically, what current waveforms should be applied? There are obviously an infinity of current waveforms that can be applied. One is interested in current stimulus that is representative of what the chip will generate during "typical" chip operation, but this is not helpful because there is no well-defined notion of typical operation. Ideally, one would like to run a simulation of the chip, under typical "workload," extract the source currents from that, then apply them to the grid as part of EM simulation. This is problematic for many reasons. It is very hard, almost impossible, in today's design flows to simulate chip workload representing several years worth of chip operation, and to extract the power tap currents for the whole design. Large chips can be simulated using high level architectural or functional models, but these models rarely have good supply current models, and the simulations would typically cover only a modest number of clock cycles. Simulating years worth of chip activity is practically impossible.

In the past, and specifically in the testing of metal structures composed of one or a few lines, much has been done to identify a suitable DC current level that represents a good *effective current* value which, when applied to the test structure produces the same mean time-to-failure (MTF) for that test structure as the MTF under the time-varying current. A very common method has been to use the DC average current value in a metal line, resulting from a time-average over a very long time period, as the effective DC current.

Considering the LTI system (2), it is clear that the effective current should be related to a convolution in some way. Taking a simple case for demonstration, for an LTI system with a single input and single output, and assuming zero initial stress, the stress at some point along the tree would be given by $\sigma(t) = \int_0^t h(\tau)i(t-\tau)d\tau$, where $h(\cdot)$ is the impulse response function and $i(\cdot)$ is the branch current waveform. Suppose the current was to be fixed over all previous time at some value i_{eff} and we require the resulting $\sigma(t)$ to come out to have the same value as in the above integral, then we would enforce $\int_0^t h(\tau)i_{\text{eff}}d\tau = \sigma(t) = \int_0^t h(\tau)i(t-\tau)d\tau$. But this would mean that the value of i_{eff} must depend on t, in fact

$$i_{\text{eff}}(t) = \frac{1}{\int_0^t h(\tau)d\tau} \int_0^t h(\tau)i(t-\tau)d\tau. \tag{3}$$

So the effective current is not truly a fixed DC average but it depends on the time for which the stress is to be computed. It is not a simple time average, but a weighted average based on the values of the impulse response function over all previous time. Because the impulse response dies down over time, the recent past of the current waveform matters much more than its distant past. So the long-term plain average of the current waveform is not really representative. Furthermore, evaluating the above expression for $i_{\text{eff}}(t)$ basically requires a simulation of the system itself for the specified current waveforms, so there is no merit or value in using this approach.



(a)

(b)

Figure 1: A low-frequency pulsed-DC input current (a) with a superimposed high-frequency component, and resulting stress response (b).

For a better effective current concept, we need a model or metric (but not a single number) that captures the aggregate characteristics of a current waveform, including its variability around its average. Because the dynamic system (2) for tracking the EM-induced stress is very slow, with time-constants on the order of days or weeks, the system effectively filters out all high-frequency input current variations, as seen in Fig. 1. Given this, one can see why the DC average *might* be useful to represent the aggregate impact of the inputs: it is the ultimate low-pass filter, and ignores everything but the DC component. However, because it masks *all* variations of the waveform around the average, the DC average also masks the effect of low-frequency variations, variations on the order of weeks or months. As shown in Fig. 1, the stress curve responds significantly to slow (low-frequency) input variations, such as a pulsed DC input, but is not sensitive to the high-frequency noise around the pulses. Ignoring significant stress variations leads to MTF estimates that are too optimistic. More realistic EM checking should take into account variations in the stress response arising from low-frequency variations in the applied currents.

Typically, low-frequency variations in the power supply currents are the result of major changes in functionality, e.g., due to changes in the *modes of operation* of the chip or, more precisely, of the large design blocks in the chip. Typical modes may include high performance, long battery life, standby, power down, etc. Without loss of generality, we will assume that the supply current of every design block k is represented by a single current source $i_k(t)$ that is attached to the power grid. In order to account for the impact of these low-frequency variations on the EM degradation of the power grid, we have pursued two objectives: 1) provide a low-burden mechanism for users to provide their knowledge of the high-level operation of the chip; 2) translate these specifications into variances of the stresses in the metal network, which immediately translate into changes (improvements) in the lifetime estimates.

2 STOCHASTIC EFFECTIVE CURRENT MODEL

As might be expected, a good way of capturing uncertainty and variations is to use stochastic analysis. Thus, our mechanism for capturing user knowledge is what we call the *stochastic effective current model.* Formally, we model every block current source $i_k(t)$ as a stationary (wide-sense[1] stationary (WSS)) *stochastic process* with fixed mean μ_k and fixed variance σ_k^2. Furthermore, we assume that every block k has D_k modes, and the current in each mode j, where $j = 1, 2, \ldots, D_k$ is modelled as another stationary stochastic process $i_{kj}(t)$, with fixed mean μ_{kj}. In order to fully determine these various processes, we expect some user input, specifically, *for each block*:

(1) *The average power dissipation in each mode.* Knowing the supply voltage, this gives us the mean value μ_{kj} of the supply current in every mode of operation. *Optionally*, we can also make use of the variance σ_{kj}^2 of the supply current in each mode. Even approximate knowledge of the ratio σ_{kj}/μ would be useful, if available.

(2) *The average time spent in each mode.* We refer to this as the expected occupancy time or the sojourn time in each mode, to be denoted by τ_{kj}. Let $\tau_k = \sum_{j=1}^{D_k} \tau_{kj}$.

With such user specifications, the modes of operation easily translate to an *embedded Markov Chain* [6], which is a discrete-time Markov Chain describing the transitions of a continuous-time process, in this case the current source, between its states or modes of operation. Once in a given mode, we assume that a block can switch *equiprobably* to any other mode of operation. This is only a simplifying assumption, because we expect that it may not be easy for users to provide the actual probabilities. The method can certainly incorporate unequal transition probabilities, if available.

A few results follow from the above (details are omitted for brevity, but available in [4]). First, the probability that a block k is in any given mode j is fixed over time. We call these the *modal probabilities* and denote them by p_{kj}, and they can be shown to be

$$p_{kj} = \frac{\tau_{kj}}{\tau_k}. \tag{4}$$

From this, we can find the mean μ_k of the overall stochastic process $i_k(t)$ for the block, as

$$\mu_k = \sum_{j=1}^{D_k} \mu_{kj} p_{kj}, \tag{5}$$

and the variance of $i_k(t)$ can be shown to be

$$\sigma_k^2 = \sum_{j=1}^{D_k} p_{kj}(\sigma_{kj}^2 + \mu_{kj}^2) - \mu_k^2, \tag{6}$$

where σ_{kj}^2 is the variance of the current source $i_k(t)$ in its mode j of operation, if provided by the user. If not, it can be safely set to

[1] Throughout this work, we will use "stationary" process to refer to a *wide-sense stationary* stochastic process, i.e., a process with a fixed mean for which the mean of the product $x(t_1)x(t_2)$ depends only on $t_1 - t_2$, for any t_1, t_2. A stationary process also has a fixed variance over time.

zero so that

$$\sigma_k^2 = \sum_{j=1}^{D_k} p_{kj}\mu_{kj}^2 - \mu_k^2. \tag{7}$$

Another important characteristic of the process $i_k(t)$ is its *autocorrelation coefficient*, which depends on the notion of covariance. We will use $E[X]$ to refer to the *expected value*, i.e., the *mean*, of the random variable (RV) X. The *autocovariance function* of a scalar stochastic process $x(t)$ is

$$C_x(t_1, t_2) = E\left[x(t_1)x(t_2)\right] - E[x(t_1)]E[x(t_2)], \tag{8}$$

and its *autocorrelation coefficient* is defined as

$$\rho(t_1, t_2) = \frac{C_x(t_1, t_2)}{\sqrt{C_x(t_1, t_1)C_x(t_2, t_2)}}. \tag{9}$$

For any stationary process, $\rho(t_1, t_2) = C_x(t_1, t_2)/\sigma_x^2$ is an *even function* of only $t = t_1 - t_2$, i.e., $\rho(t) = \rho(-t)$. For our work, let $\rho_{kj}(t)$ be the autocorrelation coefficient of the supply current $i_{kj}(t)$, which is the current in block k when in mode j. If these are available, then the autocorrelation coefficient of the overall block current $i_k(t)$, to be denoted $\rho_k(t)$, can be shown (details in [4]) to be

$$\rho_k(t) = \sum_{j=1}^{D_k} \rho_{kj}(t) e^{-|t|/\tau_{kj}} p_{kj}. \tag{10}$$

If the user is unable to provide a variance for the current in every mode, then the $\rho_{kj}(t)$ are also not available and one can safely set them to 1, so that

$$\rho_k(t) = \sum_{j=1}^{D_k} e^{-|t|/\tau_{kj}} p_{kj}. \tag{11}$$

As a final refinement of the model, we have identified an effective time-constant τ_k^* such that the area under $e^{-|t|/\tau_k^*}$ is equal to the area under $\rho_k(t)$ from (11). It turns out that $\tau_k^* = \sum_{j=1}^{D_k} \tau_{kj} p_{kj}$, leading to the approximation

$$\tilde{\rho}_k(t) = e^{-|t|/\tau_k^*}, \tag{12}$$

which turns out to be quite accurate, as we will see below. It will also be of great utility for the general case approach using the shaping filters, as we'll see later in this work.

Table 1 shows four examples of the current source specifications for a block k with three modes. The user-provided settings are the modal mean current values μ_{k1}, μ_{k2} and μ_{k3}, and mean occupancy times τ_{k1}, τ_{k2} and τ_{k3}. Also shown are the computed modal probabilities p_{kj} and the resulting ratios σ_k/μ_k for the overall block current. Notice that widely varying block modal current means μ_{kj} lead to larger σ_k/μ_k values. It would've been a heavy burden on users if we had required them to specify both the mean μ_k and variance σ_k^2 of the overall block current, factoring in all the possible modes. Instead, a key benefit of this *stochastic effective current model*, based on very reasonable assumptions, is that it provides a way to compute these key properties based on user specifications that are easy to provide. In what follows, we will determine the effect of the stochastic effective current model on the stress simulation flow. We will demonstrate how the stochastic properties of the system inputs, which were computed from user specifications, translate to the system outputs.

Table 1: Sample user specifications for a block k with three modes, and the resulting σ_k/μ_k for the overall block current.

Mode 1	Mode 2	Mode 3	σ_k/μ_k
μ_{k1} = 101 mA τ_{k1} = 212 ms p_{k1} = 33%	μ_{k2} = 102 mA τ_{k2} = 210 ms p_{k2} = 33%	μ_{k3} = 97 mA τ_{k3} = 220 ms p_{k3} = 34%	2 %
μ_{k1} = 130 mA τ_{k1} = 135 ms p_{k1} = 25%	μ_{k2} = 102 mA τ_{k2} = 172 ms p_{k2} = 33%	μ_{k3} = 80 mA τ_{k3} = 220 ms p_{k3} = 42%	20 %
μ_{k1} = 143 mA τ_{k1} = 121 ms p_{k1} = 23%	μ_{k2} = 98 mA τ_{k2} = 177 ms p_{k2} = 34%	μ_{k3} = 78 mA τ_{k3} = 220 ms p_{k3} = 43%	25 %
μ_{k1} = 162 mA τ_{k1} = 101 ms p_{k1} = 21%	μ_{k2} = 96 mA τ_{k2} = 170 ms p_{k2} = 34%	μ_{k3} = 74 mA τ_{k3} = 220 ms p_{k3} = 45%	33 %

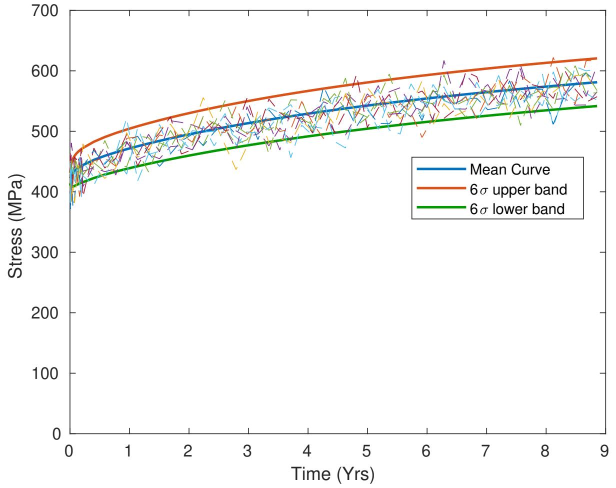


Figure 2: Stress response to Markov Chain samples with variance band capturing variations.

3 STOCHASTIC STRESS ANALYSIS

As mentioned earlier, the stress in metal lines or interconnect trees can be described by means of a linear time-invariant (LTI) system, usually expressed in the standard differential form

$$\dot{s}(t) = As(t) + Bu(t), \qquad \text{with } s(0) = s_0, \tag{13}$$

where $s(t)$ is the $n \times 1$ state vector, $u(t)$ is the $m \times 1$ input vector, $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$. The problem of assessing the impact of the input current variances arising from the Markov model on the variance of the stress around its mean as it evolves over time is a *generic* linear system problem: given the system input means and variances, find the state (or output) means and variances. The overall goal will be to combine the mean solution with the variance solution in order to generate a band around the mean curve that captures both the trend and the variation of the stress. To illustrate this point, consider Fig. 2 which shows the result of applying randomly generated input waveforms, as inputs to a system for an interconnect tree with 1904 nodes, which is extracted from the ibmpg2 industrial benchmark grid. The figure shows several time-varying stress responses surrounding the mean curve and mostly contained within the 6σ variance band, at one of the system outputs. By computing the variance directly, as we will see below, we are able to generate such variance bands without running any waveform sampling (which becomes very expensive for large trees). It is clear from the figure that the higher stress at the top-end of the 6σ band can cause void nucleation much earlier than one would get if working only with the mean curve. Strictly speaking, and in order to use a voltage-drop based failure criterion as in recent work [2], one should also assess the impact of the current variance on the voltage variance in the grid, because it might impact the failure time. However, we have found that the impact is minimal (about 0.1% of V_{dd}), so that it is sufficient to update only the mean of the voltage drop after void nucleation, which is done by simply solving the DC system $Gv = u$, where G is the grid conductance matrix and u is the current mean.

We assume that the stress system (13) is *stable*, i.e., all the eigenvalues of A have negative real parts, and that the input $u(t)$ is bounded, so that $s(t)$ is also bounded. The system description can also be given in integral form, as the solution of the above ordinary differential equation (ODE) system (13), which is

$$s(t) = e^{At} s_0 + \int_0^t e^{A(t-\tau)} Bu(\tau) d\tau. \tag{14}$$

For our work, the inputs $u_i(t)$ are stationary stochastic processes and the initial state s_0 is a vector of random variables (RVs), and we would like to compute the mean and variance over time at the system outputs. We will denote the variance of an RV X by σ_X^2, where $\sigma_X \geq 0$. In our context, the input processes are currents, and the system outputs are stress values at discretized points in the power grid. For practical reasons, we will assume that the $u_i(t)$ are mutually uncorrelated, and are uncorrelated from s_0. The $u_i(t)$ processes have fixed means μ_{u_i}, captured in the vector μ_u, as well as fixed variances $\sigma_{u_i}^2$. We will also assume that the components of the initial state s_0 are mutually uncorrelated, with variances captured in the vector $\sigma_{s_0}^2$.

An RV X can always be written as the sum of its mean μ_X and the RV $\tilde{X} = (X - \mu_X)$, which is obviously a zero-mean RV. It is often useful to simplify the presentation by working with zero-mean RVs, from which one can easily recover $X = \mu_X + \tilde{X}$. For a stochastic process $X(t)$, we can also set $\tilde{X}(t) = (X(t) - \mu_X(t))$ and work with the zero-mean stochastic process $\tilde{X}(t)$, from which one can easily recover $X(t) = \mu_X(t) + \tilde{X}(t)$. For an LTI system with WSS input processes $u_i(t) = \mu_{u_i} + \tilde{u}_i(t)$, the system response can be found by superposition, in two phases. In the first phase, we consider the system with the deterministic inputs $\mu_{u_i} = E[u(t)]$, starting from the deterministic initial state $\mu_s(0) = E[s_0]$. In the second phase, we consider the system inputs to be zero-mean RVs and proceed to compute the variances of the system outputs. The solution of the system in the first phase is in fact the output means, as can be seen by taking the expectation $E[\cdot]$ of both sides of (14) and using linearity. In this case, it is easy to compute the output $\mu_s(t)$ by performing a numerical simulation of the system with $E[s_0]$ as the initial state and the constants μ_{u_i} as the input stimuli. The mean simulation in this sense has been extensively studied in [2]. In this work, our focus is on the second phase, i.e., computing the state variances, which is much more involved as we will show in the next section.

Variance computation is computationally expensive for large scale system, such as one encounters in VLSI circuits. In fact, it is rarely attempted in EDA. Nevertheless, we will demonstrate a powerful new approach for performing this computation, based on numerical simulation, which allows us to handle moderate size problems. The rest of this paper is concerned with the computation of the variance curves of the system states where the analysis is now *based on zero-mean input processes and a zero-mean initial state.*

4 VARIANCE COMPUTATION

For any vector RV x with mean μ_x, we will use $\mathcal{V}_x$ to denote the covariance matrix $\mathcal{V}_x = E\left[(x - \mu_x)(x - \mu_x)^T\right]$, which is a symmetric matrix whose diagonal terms are the variances of the components x_i. Let a and b be two vector RVs and let $c = a + b$. If a_i and b_j are uncorrelated for every i and j, then it's easy to see that $\mathcal{V}_c = \mathcal{V}_a + \mathcal{V}_b$, due to $E\left[(a - \mu_a)(b - \mu_b)^T\right] = E\left[(b - \mu_b)(a - \mu_a)^T\right] = 0$. For the system (13), if we denote the first term of the right-hand-side (RHS) of the solution (14) by $a(t) = e^{At} s_0$ and the second (integral) term of the RHS by $b(t)$, then clearly $a(t)$ and $b(t)$ are uncorrelated, because they depend on the uncorrelated s_0 and $u(t)$, respectively, so that

$$\mathcal{V}_s(t) = \mathcal{V}_a(t) + \mathcal{V}_b(t), \tag{15}$$

where $\mathcal{V}_a(0) = \mathcal{V}_{s_0}$, which is the covariance matrix of the initial state s_0, and $\mathcal{V}_b(0) = 0$. We will tackle $\mathcal{V}_a(t)$ and $\mathcal{V}_b(t)$ separately.

4.1 Contribution of the Initial State ($\mathcal{V}_a(t)$)

The covariance matrix $\mathcal{V}_a(t)$ is the contribution due to the initial state and can be easily expressed, benefiting from $E[a(t)] = e^{At} E[s_0] = 0$ and $E[a^T(t)] = E[s_0^T] e^{A^T t} = 0$, as

$$\mathcal{V}_a(t) = E\left[a(t) a(t)^T\right] = E\left[e^{At} s_0 s_0^T e^{A^T t}\right], \tag{16}$$

so that

$$\mathcal{V}_a(t) = e^{At} \mathcal{V}_{s_0} e^{A^T t}. \tag{17}$$

In large scale problems, computation of the full matrix exponential e^{At} is prohibitively expensive because, while A is typically sparse, its exponential is often a full matrix. In fact, storing the full $\mathcal{V}_s(t)$ may itself be impractical. However, we don't actually need the full covariance matrix of the state; we only need its diagonal entries, i.e., we are interested in the scalars

$$\sigma_{s_i}^2(t) = e_i^T \mathcal{V}_s(t) e_i, \tag{18}$$

for certain $1 \leq i \leq n$ nodes of interest, where e_i is the $n \times 1$ vector with 1 in position i and 0 everywhere else. Thus, the contribution of the initial state to the state variance vector can be found by computing

$$\sigma_{s_i}^2(t) = e_i^T e^{At} \mathcal{V}_{s_0} e^{A^T t} e_i = (e^{A^T t} e_i)^T \mathcal{V}_{s_0} (e^{A^T t} e_i). \tag{19}$$

Computing the product of a matrix exponential by a vector is much more practical than computing the full matrix exponential, but it too is very expensive for large scale problems. Instead, notice that

$$\sigma_{s_i}^2(t) = p(t)^T \mathcal{V}_{s_0} p(t), \tag{20}$$

where $p(t) \triangleq e^{A^T t} e_i$ clearly satisfies the ODE system

$$\dot{p}(t) = A^T p(t), \quad \text{with } p(0) = e_i. \tag{21}$$

This system (21) can be *numerically simulated* over time and the resulting $p(t)$ used to compute $\sigma^2_{s_i}(t)$ at every time point, via (20).

Algorithm 1 Total variance computation under white noise input sources. Note, a_0, a_1 and b_{-1} are computed from the successive time steps in the VCBDF2 engine.

```
$t_0 = 0, p_0 = p_{-1} = e_i$
$\tilde{v} = 0, h_1 = h_{\min}$
$\sigma^2_{s_i}(t_0) = p_0^T \mathcal{V}_{s_0} p_0$
for ($k = 1, t_k < T, k = k + 1$) do
    $b = a_0 p_{k-1} + a_1 p_{k-2}$
    Solve $(h_k b_{-1} A - I)\, p_k = b$, for $p_k$ // VCBDF2

    $v = \tilde{v} + (h_k/2)\left(p_k^T Q p_k + p_{k-1}^T Q p_{k-1}\right)$ // TR
    Output $\sigma^2_{s_i}(t_k) = p_k^T \mathcal{V}_{s_0} p_k + v$
    $p_{k-2} = p_{k-1}, p_{k-1} = p_k, \tilde{v} = v$
    $h_{k+1} = \text{update}(h_k)$ // part of VCBDF2 engine
    $t_{k+1} = t_k + h_{k+1}$
end for
```

4.2 Contribution of the Inputs ($\mathcal{V}_b(t)$)

We now consider the solution due to only the inputs, i.e., with *zero initial state*, so that both $s(0) = 0$ and $\mathcal{V}_s(0) = 0$. To get to a general solution, we will benefit from the special case solution when the input currents to the LTI system are *white noise* processes, whose intensity[2] (or variance) is set equal to the computed Markov model variance (7). Although *white noise* inputs are not realizable in practice, they can serve as a useful idealized approximation. Indeed, we will see how white noise analysis allows us to handle the general case when the inputs are not white noise, by means of *shaping filters*.

4.2.1 *White Noise Analysis.*

So, assuming the input current sources are white noise processes, let us define the matrix Q as

$$Q \triangleq BKB^T, \tag{22}$$

where B is the input matrix of the LTI system (2), and K is the *intensity matrix* of the input vector u. Since the input processes are stationary and mutually uncorrelated, the matrix K is diagonal and its i^{th} diagonal entry is the intensity of the white noise process $u_i(t)$. It can be shown [9] that the covariance of the system states under white noise inputs and zero initial state is given by

$$\mathcal{V}_s(t) = \int_0^t e^{A\tau} Q e^{A^T \tau} d\tau. \tag{23}$$

Taking the derivative of both sides, we have

$$\dot{\mathcal{V}}_s(t) = e^{At} Q e^{A^T t}, \quad \text{with } \mathcal{V}_s(0) = 0, \tag{24}$$

and we're interested in the diagonal entries of $\dot{\mathcal{V}}_s(t)$, which are the derivatives of the variances,

$$\dot{\sigma}^2_{s_i}(t) = e_i^T \dot{\mathcal{V}}_s(t) e_i = (e^{A^T t} e_i)^T Q (e^{A^T t} e_i), \tag{25}$$

so that

$$\dot{\sigma}^2_{s_i}(t) = p(t)^T Q p(t), \quad \text{with } \sigma^2_{s_i}(0) = 0, \tag{26}$$

where $p(t) \triangleq e^{A^T t} e_i$ clearly satisfies

$$\dot{p}(t) = A^T p(t), \quad \text{with } p(0) = e_i. \tag{27}$$

The two ODE systems (26) and (27) are jointly numerically simulated using a hybrid Trapezoidal Rule (TR) and Variable Coefficient BDF2 (VCBDF2) method to give $\sigma^2_{s_i}(t)$. Notice that simulating the ODE system (27) also gives the contribution of the initial state by virtue of (20) and (21), so that, jointly with (26), the full variance solution is computed. The pseudocode for this simulation is shown in Algorithm 1. Fig. 3 shows an example of this variance simulation applied to one of the 120 junctions of a 1904 node system extracted from the ibmpg2 benchmark. The mean curve with a 6σ band shows the significance of the variance band for EM assessment.

Table 2: Details of the benchmarks used in this work.

Grid Name	Branches	Junctions	Trees	Current Sources
ibmpg1	10,853	11,562	709	5,387
ibmpg2	61,143	61,605	462	18,419
ibmpg3	401,412	409,601	8,189	100,527
ibmpg6	797,579	807,825	10,246	380,742

Table 3: CPU times and voiding times under white noise inputs.

System Size	Num. of	CPU Time for		1^{st} Nucleation	
(benchmark)	Junctions	Mean	Variance	Time	Drop
1904 (ibmpg2)	120	1.4 s	9.2 s	6.6 Yrs	42 %
3072 (ibmpg2)	193	1.5 s	23.7 s	14.0 Yrs	30 %
4400 (ibmpg3)	276	7.7 s	2.4 m	16.8 Yrs	16 %
5920 (ibmpg3)	371	8.1 s	3.5 m	0.8 Yrs	81 %
15440 (ibmpg3)	966	10.1 s	25.6 m	1.4 Yrs	75 %
15488 (ibmpg6)	969	9.6 s	19.7 m	10.4 Yrs	9 %

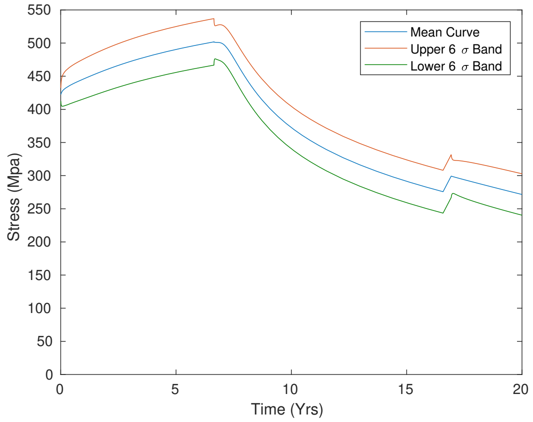


Figure 3: The variance band around its mean curve.

[2] The intensity of a white noise process is the magnitude of its power spectral density. It is common in the engineering literature to loosely refer to this as the "variance" of the process, but this is misleading because strictly-speaking the mathematical white noise process has infinite variance.

Some details of the various benchmark grids used in this work are available in Table 2. For the white noise case, Table 3 reports the times taken to simulate the mean and the variance profiles over a period of 20 years for various systems representing interconnect trees extracted from available industrial grid benchmarks. The variance profiles are computed for all junctions in each system. The table also reports the percentage reduction in the time to the first void nucleation in each tree, as a result of factoring in the variance. Note that for the system with 193 junctions, the variance simulation is significantly (221X) faster than the one carried out in [4], where the reported simulation time was 1.46 hours. The last row in the table corresponds to the largest single interconnect system found in the benchmarks. The variance band has been observed to catch many more void nucleation events per run compared to the mean simulation.

Table 4 shows the impact of accounting for input variance on the MTF estimates for three full-grid test cases. We report on the MTF reduction in different scenarios where we vary the inputs' σ/μ ratios keeping the same μ, and use the 6σ bands around the outputs. The Monte Carlo analysis (column 6 gives the number of samples) accounts for the variability in branch diffusivities and is performed in parallel on 11 cores of a 3GHz Linux machine with Xeon CPU and 128 GB of RAM. The impact on the MTF can reach up to 50% for a $\sigma/\mu = 33\%$ input variation. This shows the high risk incurred by neglecting variations in the current, and highlights the importance of using a more realistic approach for EM checking. The CPU time required is very good, ranging from 25 minutes for ibmpg1, up to 5–8 hours for a large grid (ibmpg3) with nearly 410K junctions.

4.2.2 The General Case and the Augmented System. In general, the inputs of our system (13) are not white noise, but we can construct another LTI system, called a *shaping filter* [9] [7], whose output has the same autocorrelation as the computed $\rho_{u_i}(t)$ and whose input is white noise. The augmented system, consisting of the concatenation of the new system (the shaping filter) with the original system then becomes the full system to be solved using the above white noise solution. We have already presented the modelling of the inputs $u(t)$ as stationary mutually uncorrelated Markov processes whose autocorrelation functions are sums of exponentials (11), arising from user specifications. We also showed how these autocorrelation functions can be approximated by a single exponential (12). This approximation facilitates the design of the filter (details in [4]), so that these inputs can each be generated by a scalar LTI system driven by a white noise input. Let m be the number of inputs to the LTI system (13). The resulting ODE structure for the set of input shaping filters is

$$\dot{u}(t) = Cu(t) + Dw(t), \qquad \text{with } u(0) = u_0, \tag{28}$$

where $w(t)$ is a $m \times 1$ input vector and C and D are non-singular diagonal $m \times m$ matrices. Thus, we are dealing with the $(n + m) \times (n + m)$ white-noise-driven *augmented system*

$$\begin{bmatrix} \dot{s} \\ \dot{u} \end{bmatrix} = \begin{bmatrix} A & B \\ 0 & C \end{bmatrix} \begin{bmatrix} s \\ u \end{bmatrix} + \begin{bmatrix} 0 \\ D \end{bmatrix} w \tag{29}$$

and we will define

$$x \triangleq \begin{bmatrix} s \\ u \end{bmatrix} \quad \tilde{A} \triangleq \begin{bmatrix} A & B \\ 0 & C \end{bmatrix} \quad \tilde{B} \triangleq \begin{bmatrix} 0 \\ D \end{bmatrix} \tag{30}$$

and

$$Q_w \triangleq DKD^T, \tag{31}$$

where K is the diagonal intensity matrix of $w(t)$. Therefore, we have the system

$$\dot{x}(t) = \tilde{A}x(t) + \tilde{B}w(t), \qquad \text{with } x(0) = \begin{bmatrix} 0 \\ u_0 \end{bmatrix}, \tag{32}$$

for which it will be useful to define the $(n + m) \times (n + m)$ matrix

$$\tilde{Q} \triangleq \tilde{B}K\tilde{B}^T = \begin{bmatrix} 0 & 0 \\ 0 & Q_w \end{bmatrix}. \tag{33}$$

Algorithm 2 Total variance computation under Markov inputs with shaping filters. Note, a_0, a_1 and b_{-1} are computed from the successive time steps in the VCBDF2 engine.

```
$t_0 = 0, p_0 = p_{-1} = e_i$
$q_0 = 0, \tilde{v} = 0, h_1 = h_{\min}$
$\sigma^2_{s_i}(t_0) = p_0^T \mathcal{V}_{s_0} p_0$
for ($k = 1, t_k < T, k = k + 1$) do
        $b = a_0 p_{k-1} + a_1 p_{k-2}$
        Solve $(h_k b_{-1} A - I) p_k = b$, for $p_k$ //VCBDF2

        $b = (I + (h_k/2)C^T)q_{k-1} + (h_k/2)B^T(p_k + p_{k-1})$
        $q_k = \left(I - (h_k/2)C^T\right)^{-1} b$    //Closed Form

        $v = \tilde{v} + (h_k/2)\left[q_k^T Q_w q_k + q_{k-1}^T Q_w q_{k-1}\right]$ //TR
        Output $\sigma^2_{s_i}(t_k) = p_k^T \mathcal{V}_{s_0} p_k + v$
        $q_{k-1} = q_k, \tilde{v} = v$
        $p_{k-2} = p_{k-1}, p_{k-1} = p_k$
        $h_{k+1} =$ update($h_k$) // part of VCBDF2 engine
        $t_{k+1} = t_k + h_{k+1}$
end for
```

As with the white noise case we saw earlier, the solution for the state covariance matrix $\mathcal{V}_x(t)$ for the full augmented system (with zero initial conditions) is given by

$$\mathcal{V}_x(t) = \int_0^t e^{\tilde{A}\tau} \tilde{Q} e^{\tilde{A}^T \tau} d\tau. \tag{34}$$

Taking the derivative of both sides, we have

$$\dot{\mathcal{V}}_x(t) = e^{\tilde{A}t} \tilde{Q} e^{\tilde{A}^T t}, \qquad \text{with } \mathcal{V}_x(0) = 0, \tag{35}$$

and we're interested in the diagonal of $\dot{\mathcal{V}}_x(t)$, which holds the derivatives of the variances. Let $\tilde{e}_i \in \mathbb{R}^{(n+m)}$ be zero everywhere except for its ith entry which is 1, with $i \leq n$, so that $\tilde{e}_i^T = \begin{bmatrix} e_i^T & 0 \end{bmatrix}$ and

$$\begin{aligned} \dot{\sigma}^2_{x_i}(t) = \tilde{e}_i^T \dot{\mathcal{V}}_x(t) \tilde{e}_i &= (e^{\tilde{A}^T t} \tilde{e}_i)^T \tilde{Q} (e^{\tilde{A}^T t} \tilde{e}_i), \\ \text{with } \sigma^2_{x_i}(0) &= 0, \end{aligned} \tag{36}$$

and note that $\dot{\sigma}^2_{s_i}(t) = \dot{\sigma}^2_{x_i}(t)$ due to $i \leq n$. Let $p(t) \in \mathbb{R}^n$ and $q(t) \in \mathbb{R}^m$ be such that

$$\begin{bmatrix} p \\ q \end{bmatrix} = e^{\tilde{A}^T t} \tilde{e}_i, \tag{37}$$

Table 4: Full grid variance simulation using white noise inputs, with 6σ bands.

Grid Benchmark	Input $\frac{\sigma}{\mu}$	CPU Time for Mean	CPU Time for Variance	CPU Time Total	Num. of Samples	MTF	% Drop in MTF
ibmpg1	1%	3.16 m	47.92 m	51.08 m	61	4.89 Yrs	30 %
	2%	3.55 m	38.49 m	42.04 m	57	4.14 Yrs	40.7%
	5%	54.6 s	23.32 m	24.23 m	54	2.55 Yrs	63.5 %
ibmpg2	20%	5.4 m	1.13 h	1.22 h	36	9.21 Yrs	23.12 %
	25%	4.2 m	1.07 h	1.14 h	39	8.06 Yrs	32.7 %
	33%	3.6 m	1.05 h	1.11 h	51	6.12 Yrs	49 %
ibmpg3	20%	31.40 m	4.00 h	4.52 h	8	6.46 Yrs	7.1 %
	33%	1.27 h	6.57 h	7.84 h	8	6.37 Yrs	8.5 %

from which it's clear that $\left[p(0)^T \;\; q(0)^T\right] = \tilde{e}_i^T$, so that $p(0) = e_i$ and $q(0) = 0$, and we can write

$$\dot{\sigma}^2_{s_i}(t) = \left[p^T \quad q^T\right] \begin{bmatrix} 0 & 0 \\ 0 & Q_w \end{bmatrix} \begin{bmatrix} p \\ q \end{bmatrix} = q^T Q_w q. \tag{38}$$

In addition, we have

$$\begin{bmatrix} \dot{p} \\ \dot{q} \end{bmatrix} = \frac{d}{dt}\left(e^{\tilde{A}^T t}\tilde{e}_i\right) = \tilde{A}^T\left(e^{\tilde{A}^T t}\tilde{e}_i\right) = \begin{bmatrix} A^T & 0 \\ B^T & C^T \end{bmatrix} \begin{bmatrix} p \\ q \end{bmatrix}. \tag{39}$$

As a result, the variance can be found based on the following three ODE systems,

$$\dot{\sigma}^2_{x_i}(t) = q^T Q_w q, \qquad \text{with } \sigma^2_{x_i}(0) = 0, \tag{40}$$

$$\dot{p} = A^T p, \qquad \text{with } p(0) = e_i, \tag{41}$$

$$\dot{q} = C^T q + B^T p, \qquad \text{with } q(0) = 0. \tag{42}$$

These three systems can be jointly numerically simulated using VCBDF2 and TR, as previously described, to give $\sigma^2_{s_i}(t)$, as shown in Algorithm 2, which actually provides the full variance solution, including the contribution of the initial state. Note the simplification arising from the fact that C is a diagonal matrix, so that

$$\left(I - (h/2)C^T\right)^{-1} b = \left[\frac{b_1}{1-(h/2)c_{11}} \quad \cdots \quad \frac{b_n}{1-(h/2)c_{nn}}\right]^T, \tag{43}$$

which is needed to compute q. Table 5 reports the CPU times of simulating the mean and variance curves over 20 years for systems for trees extracted from ibmpg2. We can see up to 85% drop in the time to first voiding estimates for inputs with $\sigma/\mu = 33\%$. The CPU cost depends on the grid size, since the input matrix B in (42), which is a function of the conductance matrix of the grid, is explicitly used in the simulation. We used occupancy times of the order of 100μs to 10ms. For a simulation of the full ibmpg2 benchmark grid using shaping filters, we found that one sample can be obtained in a total of 5.8 CPU hours. The drop in sample TTF in this example was found to be as high as 69.5%. The difference in the CPU times between the white noise case v.s. the general case is due to the matrix B, whose computation and storage can be a significant overhead.

Table 5: Results for the general case using shaping filters, for some large interconnect trees from the ibmpg2 benchmark.

System Size	Num. of Junctions	CPU Time for Mean	CPU Time for Variance	1^{st} Nucleation Time	1^{st} Nucleation % Drop
928	59	1.50 s	27.37 s	6.80 Yrs	66 %
1808	114	1.47 s	1.47 m	3.06 Yrs	84.7 %
1904	120	1.44 s	1.61 m	4.34 Yrs	62.1 %
3072	193	1.56 s	4.64 m	3.71 Yrs	81.4 %

5 CONCLUSION

In this work, we have addressed the optimism that's implicit in the use of a fixed DC average as the effective current to be used for EM checking. As an alternative, we developed a model for the input current sources that is driven by user specifications that are easy to provide. We then presented a novel approach to variance computation, whose key feature is to express the variance computation as the output of an ODE problem that can be numerically simulated. We make use of shaping filters, a concept that is adapted from the study of communication systems. The MTF reduction as a result of taking the variance into account is quite significant, reaching ≈ 50% for input σ/μ from 2% to 33% in some cases. EM checking without the variance is simply too optimistic and can be misleading for design. Even though the grids in our test cases are of modest size, the performance is very promising and it is hoped, now that the significance of the variance has been demonstrated, that future work will provide further performance improvements and so allow the handling of very large grids.

REFERENCES

[1] J. R. Black. 1969. Electromigration-A brief survey and some recent results. *IEEE Trans. on Electronic devices* 16, 4 (Apr. 1969), 338–347.

[2] S. Chatterjee, V. Sukharev, and F. N. Najm. 2017. Power grid electromigration checking using physics-based models. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 37, 7 (2017), 1317–1330.

[3] M. Hauschildt, C. Hennesthal, G. Talut, O. Aubel, M. Gall, K. B. Yeap, and E. Zschech. 2013. Electromigration early failure void nucleation and growth phenomena in Cu and Cu(Mn) interconnects. In *IEEE International Reliability Physics Symposium (IRPS)*. 2C.1.1–2C.1.6. https://doi.org/10.1109/IRPS.2013.6531951

[4] A. Issa. 2019. *A Stochastic Approach to Electromigration Checking for Chip Power Grids.* MASc Thesis. University of Toronto.

[5] M. A. Korhonen, P. Børgesen, K. N. Tu, and C.-Y. Li. 1993. Stress evolution due to electromigration in confined metal lines. *Journal of Applied Physics* 73, 8 (1993), 3790–3799.

[6] A. Leon-Garcia. 2017. *Probability, statistics, and random processes for electrical engineering.* Pearson Education.

[7] F. L. Lewis, L. Xie, and D. Popa. 2008. *Optimal and Robust Estimation: With an Introduction to Stochastic Control Theory* (2nd ed.). Taylor & Francis Group.

[8] J. R. Lloyd. 2007. Black's law revisited-Nucleation and growth in electromigration failure. *Microelectronics Reliability* 47, 9–11 (2007), 1468 – 1472.

[9] P. S. Maybeck. 1979. *Stochastic Models, Estimation and Control (Vol. 1).* Academic Press, New York, NY.

[10] A. S. Oates. 2016. Interconnect reliability challenges for technology scaling: A circuit focus. In *IEEE International Interconnect Technology Conference / Advanced Metallization Conference (IITC/AMC).* San Jose, CA, 59–59. https://doi.org/10.1109/IITC-AMC.2016.7507680

[11] J. Warnock. 2011. Circuit design challenges at the 14nm technology node. In *ACM/IEEE 48th Design Automation Conference.* San Diego, CA, 464–467.