Verification and Codesign of the Package and Die Power Delivery System Using Wavelets

Imad A. Ferzli, Eli Chiprout, *Member, IEEE,* and Farid N. Najm, *Fellow, IEEE*

***Abstract*—As part of the design of large integrated circuits, one must verify that the power delivery network provides supply and ground voltages to the circuit that are within specified ranges. We introduce the concept of time–frequency description of circuit currents using wavelets, and use that to set up an optimization framework that finds the worst-case supply/ground voltage fluctuations. This framework allows for the quick determination of the impact of either the package or the die on the worst-case behavior, which enables their codesign. This approach has been applied to an industrial microprocessor design, resulting in realistic and nonobvious worst-case waveforms.**

***Index Terms*—Integrated circuit packaging, power grid, verification, voltage drop, wavelet transform.**

I. Introduction

THE INTENSE drive toward lower power designs has highlighted the need for robust design of a chip's power delivery network (PDN). The PDN, starting at the voltage regulation module (VRM), through the motherboard, package, and finally the on-die power grid, must supply a reliable source of power that is fairly free from fluctuations over time. A large drop in supply voltage may lead to timing violations or logic failure.

In order to ensure a stable power supply, designers are interested in knowing the lowest possible voltage, $V_{\min}$, which may be produced by the PDN and supplied to the transistors. $V_{\min}$ is a function of two major elements: the PDN system response, and the current draw or excitation produced by the die. To determine the PDN response, designers typically model the PDN using either *RLC* elements or electromagnetic models. While modeling the PDN system is generally accurate, there remains a significant source of error in modeling the worst-case current draw of the die: The number of state transitions is astronomical and searching the current space for the worst-case is a daunting task. Several attempts have been made to address this problem, including the computation of bounds on possible currents [1] or of current statistics [2]. A more formal approach has been proposed [3] that constrains the problem with known design bounds, rather than simulate or search through all possible scenarios.

In this paper, we introduce the concept of time–frequency descriptions of die currents using wavelets. Wavelet analysis has been used in the computer-aided design community for nonlinear circuit simulation [4] and applied in the analysis of early design die current profiles to compute current statistics [5]. In this paper, we show that wavelets are a natural way to comprehensively characterize die behavior in the time–frequency plane. We also show that this description may be used to extract new relevant bounds that will serve in determining the worst-case current draw.

The wavelet framework will allow us to find the worst-case current draw for a given PDN on a systematic and general basis. In the simplest cases, it yields solutions that are well-known such as the reverse pulse technique (RPT) [6]. In more complex cases, it opens up the possibility of obtaining nonobvious worst-case die current waveforms that mimic complex circuit behaviors.

The wavelet framework combines both the time and frequency domains. Since finding worst-case voltage drop on the PDN has both time and frequency dimensions, this framework will give us the best of both worlds. Purely time-domain techniques, including RPT, only deal with the simplest system descriptions, whereas purely frequency-domain methods lack accuracy, especially with state-of-the-art, multiresonant PDN systems [7].

Our approach provides a broad design-assist value. Using our approach, one can isolate particular frequency bands depending on the PDN hierarchy of interest and find an accurate worst-case scenario in the time-domain. Designers will also be able to make systematic early predictions about the trends in voltage drop, e.g., expected *IR* versus di/dt-drop, and to quickly iterate between worst-case waveforms with every change of package or die design, without writing new test cases for simulation or measurement, enabling true die-package codesign. Compared with [3], which relies on stepping through time or computing upper bounds on maximum voltage drop, our technique requires no time iterations and results in accurate strict worst-case scenarios. It is also more versatile, handling indiscriminately *RC* and *RLC* PDN models, and unlike [3], yields actual worst-case waveforms which enable *diagnostics*. Our approach has been applied in early design of an industrial microprocessor

Manuscript received August 14, 2008; revised May 28, 2009. Current version published December 18, 2009. This research was supported by Intel Corporation. This paper was recommended by Associate Editor V. Narayanan.

I. A. Ferzli and F. N. Najm are with the Department of Electrical and Computer Engineering, University of Toronto, ON M5S, Canada (e-mail: iferzli@cppib.ca; f.najm@utoronto.ca).

E. Chiprout is with the Strategic Computer-Aided Design Laboratories, Intel Corporation, Hillsboro, OR 97124 USA (e-mail: eli.chiprout@intel.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2009.2034563

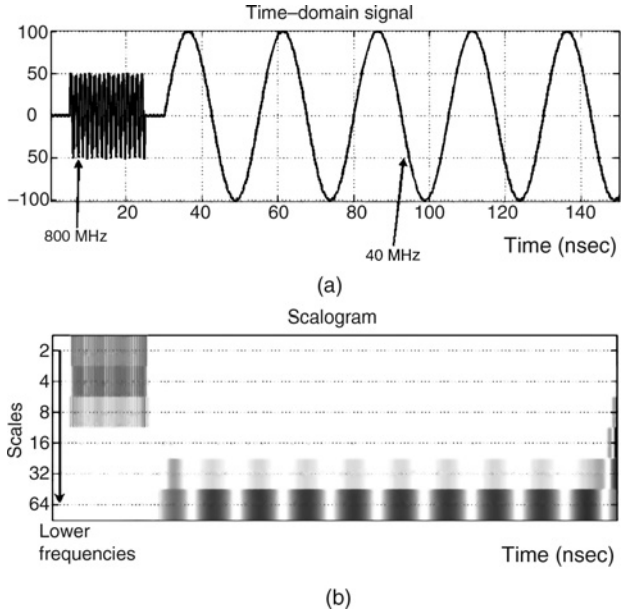

Fig. 1. Time–frequency description of a signal via a scalogram.

and has revealed realistic and nonobvious worst-case scenarios.

This paper is organized as follows: Section II introduces time–frequency analysis and paves the way for the continuous wavelet transform (CWT), discussed in Section III. The discrete wavelet transform (DWT) and its multiresolution analysis (MRA) framework are the subject of Section IV. The construction of the worst-case current stimulus is detailed in Section V and an optimization problem that discovers the worst-case stimulus and imposes appropriate constraints is formulated in Section VI. Experimental results are showcased in Section VII. We conclude in Section VIII.

II. Time–Frequency Analysis

Most readers are familiar with a time-domain description of a waveform, and a frequency-domain transform of it, usually obtained via a Fourier transform. However, there is another class of transforms which produce results between these two, called time–frequency transforms, of which wavelets are a major subset [8]. These transforms are useful if one wants to analyze the change of "frequency" behavior over time. The word *frequency* is in quotes because, by definition, a frequency lasts over an infinite time period. The usefulness of wavelets, however, lies in their ability to capture the "frequency properties" of time-limited bursts.

We begin with a simple example to give the reader an intuition of the time–frequency description. Fig. 1 (bottom) shows a time–frequency representation, known as a *scalogram*, of the time-domain current at the top. The signal is made of two sinusoidal components: a short, 800 MHz burst and a 40 MHz signal, starting at 25 ns. The x-axis of the scalogram is time (same as the signal), and the y-axis the so-called *scale*, which for now, it suffices to say, is akin to the inverse of frequency. Each entry in the scalogram represents the value of the CWT (Section III) over particular time windows at a particular scale. The scalogram shows how this representation follows the frequency contents of the signal in time: dark colorations, corresponding to large transform values, match the occurrence in time of the short 800 MHz burst at the low-to-middle scales, then the 40 MHz wave at the high-scales. Weak frequencies are too light to be visible on the scalogram, indicating that the scalogram is apt at highlighting the strength of a signal at certain frequencies over localized time windows. The scalogram extends insights into the waveform visually, and its values capture useful properties, such as a "wavelet frequency envelope" (Section VI-C).

Wavelet analysis essentially breaks down a signal into a weighted summation of wavelets. The CWT, discussed in Section III, is an infinite summation of wavelets and provides the starting point for understanding wavelet-based time–frequency analysis. The DWT, on the other hand, breaks the signal down into a finite number of wavelets, as we show in Section IV. The treatment provided in Sections III and IV is based on [8], and is tailored to our needs, leaving out all but the most relevant elements of wavelet analysis.

For the purpose of finding the worst-case current draw, our approach is to use wavelet analysis to construct or synthesize a hypothetical worst-case current draw that results in maximum voltage drop given a number of time-domain and frequency-domain constraints that are known or can be derived in early design. The basic idea is to find a suitable set of wavelets to describe a current waveform (Section V) and formulate an optimization problem (Section VI) in the weights of these wavelets, whose solution results simultaneously in the shape of the worst-case current waveform and the value of the maximum voltage drop.

III. Continuous Wavelet Transform

The CWT is the starting point for understanding time–frequency analysis using wavelets. In its basic form, the CWT, denoted by $T(a, b)$, captures the correlation between a signal $x(t)$ and a time function $\psi(t)$, known as a *wavelet*, whose time-axis is *dilated* by a factor a, known as the *scale* parameter, and which is *translated* along the time-axis by b, called the *translation*

$$T(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} x(t)\psi\left(\frac{t-b}{a}\right) dt. \tag{1}$$

We denote the wavelet at scale a and translation b by $\psi_{a,b}(t)$

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\,\psi\left(\frac{t-b}{a}\right) \tag{2}$$

where the factor $1/\sqrt{a}$ normalizes the energy of all the wavelets

$$\int_{-\infty}^{+\infty} |\psi_{a,b}(t)|^2 dt = E_\psi \triangleq \int_{-\infty}^{+\infty} |\psi(t)|^2 dt. \tag{3}$$

There are three *admissibility conditions* which a real-valued time function $\psi(t)$ must satisfy in order to be a proper wavelet.

1) $E_\psi < \infty$: the wavelet is *localized in time*.
2) $\int_{-\infty}^{+\infty} \psi(t)dt = 0$: the wavelet has a zero long-term average.

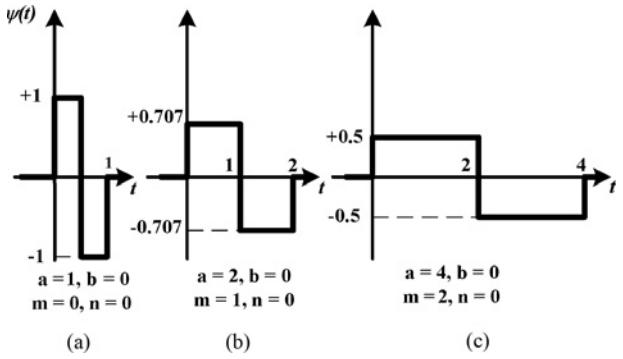

Fig. 2. Haar wavelets with various scales and translations.

3) $\int_{-\infty}^{+\infty} |\psi(f)|^2/|f|df < \infty$, where $\psi(f)$ is the Fourier transform of $\psi(t)$: the wavelet is *localized in frequency*.

The *Haar wavelet* shown in Fig. 2(a) and used throughout consists of one section of a square wave

$$\psi(t) = \begin{cases} +1, & 0 < t < 1/2 \\ -1, & 1/2 < t < 1 \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

Wavelets act like bandpass filters [8] with a given bandwidth depending on the scale a, and a *center frequency* f_c within this bandwidth where the Fourier spectrum is maximum. The center frequency for the Haar wavelet can be shown to occur at

$$f_c(a) \approx 2.33/(\pi a). \tag{5}$$

We now give a brief intuitive interpretation of the wavelet transform. One could think of the scale parameter a as an inverse frequency, and the translation parameter b as a simple time shift. If the scale parameter a is *increased*, the time span of the wavelet would *increase* while its center frequency would *decrease* and its bandwidth *shrink*, in accordance with $\mathcal{F}\{\psi(t/a)\} = |a|\psi(af)$. The transform value $T(a, b)$ can be qualitatively thought of as a measure of the match between the frequency contents of $x(t)$ *in the vicinity of* $t = b$ and the wavelet's bandwidth at scale a: the stronger the match, the larger $|T(a, b)|$. Hence, the name *time–frequency analysis*. While the Fourier transform reveals the frequency contents of a signal regardless of time, the wavelet transform matches the frequency contents of a signal around time $t = b$, with the frequency spectrum of the wavelet at scale a, $\forall(a, b) \in \mathbf{R}^2$, so that the wavelet transform acts like a *filter* (more precisely, a continuum of filters for all scales). The ability of the wavelet transform to capture time-localized frequency information has made it a powerful tool in signal processing. The reader may refer to [8] for a detailed quantitative treatment of this subject.

IV. Discrete Wavelet Transform

The DWT allows the breakdown of a signal into a finite number of wavelets. The starting idea is that the transform domain (a, b) has a lot of redundancy [8], and it is sufficient to select and use only certain discrete values of a and b. By far the most common choice is to use

$$a = 2^m \qquad \textit{and} \qquad b = 2^m n \tag{6}$$

where m and n are integers. This scheme results in a so-called *dyadic grid* in the transform domain. The notation is also changed, for clarity, so that the wavelet $\psi_{m,n}(t)$ now denotes

$$\psi_{m,n}(t) = \frac{1}{\sqrt{2^m}}\,\psi\left(\frac{t - 2^m n}{2^m}\right) = A_m \psi\left(2^{-m}t - n\right) \tag{7}$$

where we call $A_m = 2^{-m/2}$ the *amplitude* of the wavelet at scale m. Notice that $\psi_{0,0}(t) = \psi(t)$. Fig. 2 shows the Haar wavelet with $m = 0, 1, 2$, and $n = 0$. The transform integral becomes

$$T_{m,n} = \int_{-\infty}^{+\infty} x(t)\psi_{m,n}(t)\mathrm{d}t \tag{8}$$

referred to as the DWT. In the DWT literature it is common to refer to m and n as the scale and translation parameters, respectively, even though, strictly speaking, the true scale is $a = 2^m$ and the true translation is $b = 2^m n$.

Dyadic grid wavelets have the key property of forming an *orthonormal basis*, which enables reconstructing the signal $x(t)$

$$x(t) = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} T_{m,n}\psi_{m,n}(t). \tag{9}$$

The above is a *synthesis* of a signal in terms of an underlying basis of wavelets. It forms the basis of using wavelet decomposition to construct worst-case currents, as we show in Section V.

In the decomposition (9) both translations (n) and scales (m) run between $-\infty$ and $+\infty$. First, we limit the translations by observing that they only need to cover the duration of the signal. If N is an integer such that the time interval $[0, 2^N]$ covers the duration of $x(t)$, then no more than 2^{N-m} translations are necessary at scale m, which can be taken, without loss of generality, to be $n = 0, \ldots, 2^{N-m} - 1$, so that (9) becomes

$$x(t) = \sum_{m=-\infty}^{+\infty} \sum_{n=0}^{2^{N-m}-1} T_{m,n}\psi_{m,n}(t). \tag{10}$$

A. Multiresolution Analysis

MRA allows us to write (10) in terms of a finite number of scales m. MRA relies on companion functions to wavelets, called *scaling functions*, denoted $\phi(t)$, dilated and translated in the same way as wavelets

$$\phi_{m,n}(t) = \frac{1}{\sqrt{2^m}}\,\phi\left(\frac{t - 2^m n}{2^m}\right) = 2^{-m/2}\phi\left(2^{-m}t - n\right) \tag{11}$$

so that $\phi_{0,0}(t) = \phi(t)$. However, a scaling function is different from a wavelet in that its dc component is nonzero. In fact, $\int_{-\infty}^{+\infty} \phi(t)dt = 1$. The Haar scaling function is a pulse over $[0, 1]$: $\phi(t) = 1$, for $0 < t < 1$, and $\phi(t) = 0$, otherwise.

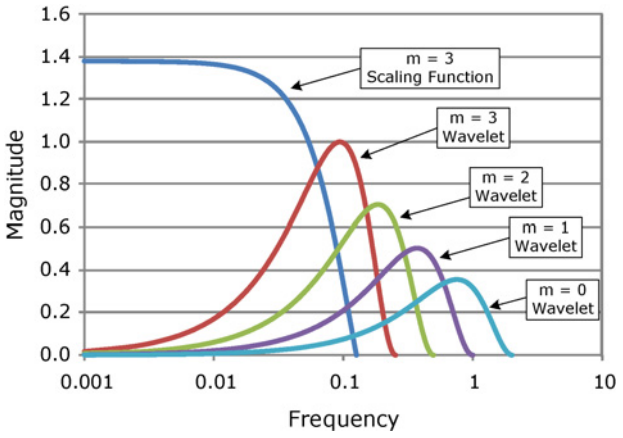

Fig. 3. MRA using the Haar system.

It can be shown that scaling functions and wavelets at scale m are linear combinations of shifted scaling functions at scale $m-1$. In the Haar system, the expressions are

$$\psi_{m+1,n}(t) = \frac{1}{\sqrt{2}}[\phi_{m,2n}(t) - \phi_{m,2n+1}(t)] \tag{12}$$

$$\phi_{m+1,n}(t) = \frac{1}{\sqrt{2}}[\phi_{m,2n}(t) + \phi_{m,2n+1}(t)]. \tag{13}$$

These relationships will come in handy in Section V-B.

The utility of scaling functions becomes clear when we consider their frequency interpretation. Fig. 3 shows the frequency spectra $|\phi_{3,0}(f)|$ (scaling function) and $|\psi_{0,0}(f)|$, $|\psi_{1,0}(f)|$, $|\psi_{2,0}(f)|$, and $|\psi_{3,0}(f)|$ (wavelets), on a log-frequency scale, where for clarity, only the largest (first) "hump" is shown. The figure shows that, wavelets act as bandpass filters and the scaling function acts as a low-pass. In this example, all signal content up to about 0.01 is captured by $\phi_{3,n}(t)$, while higher frequency content is captured by $\psi_{3,n}(t)$, $\psi_{2,n}(t)$, etc. That is, in MRA, a collection of wavelets working together cover an overall pass-band with the scaling function covering the frequencies below this pass-band.

We need to limit the number of scales m needed in the decomposition (10) from both above and below, and the frequency spectrum of wavelets guides both choices. The key idea is that each wavelet is able to *resolve* a pass-band in which it is the "dominant" wavelet, as shown in Fig. 3. Since in practice, one is interested in signal content only up to a certain frequency $f_{\max}$, we set up the parameters of the expansion so that the smallest (fastest) wavelet has a pass-band that covers $f_{\max}$. This can be easily done, by using (5), to choose the slowest wavelet with center frequency greater than $f_{\max}$ as the fastest wavelet in the decomposition. This sets the smallest required scale $m_{\min}$.

While wavelets resolve frequencies in their pass-bands, the same cannot be said about scaling functions in the low-pass-band, in the sense that there is no resolution among the low-frequencies. In Fig. 3, there is no information on the relative magnitude of different frequencies below 0.01, as there is between frequencies of 0.01 and 1. This observation sets the guideline for choosing an upper limit $m_{\max}$ on the decomposition: one should choose the maximum scale $m_{\max}$ so that wavelets of scales $m_{\min}, \ldots, m_{\max}$ cover the frequency

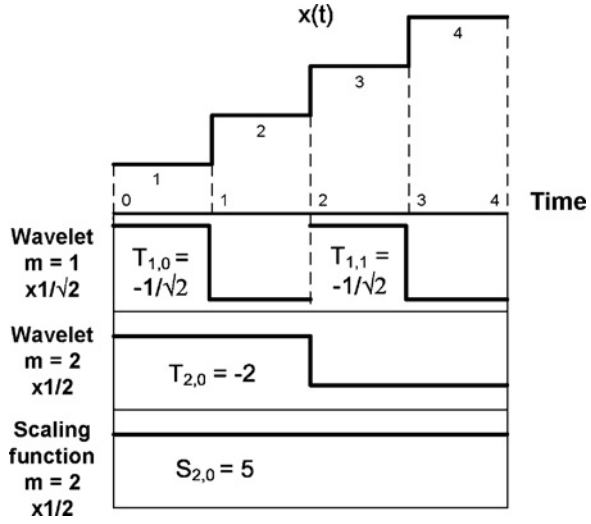

Fig. 4. Example MRA decomposition.

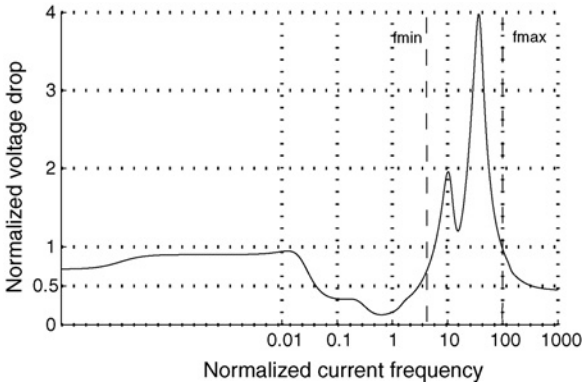

Fig. 5. Impedance plot in an early design microprocessor.

band $(f_{\min}, f_{\max})$, where one is interested in resolving signal content, and then use a scaling function for all lower frequencies. Without loss of generality, and conforming with the wavelet literature, we can number the scales $m_{\min}, \ldots, m_{\max}$ as $1, \ldots, m_0$, so that the full decomposition of the signal becomes

$$x(t) = \sum_{n=0}^{2^{N-m_0}-1} S_{m_0,n}\phi_{m_0,n}(t) + \sum_{m=1}^{m_0}\sum_{n=0}^{2^{N-m}-1} T_{m,n}\psi_{m,n}(t). \tag{14}$$

We emphasize that the decomposition is set up so that $\psi_{1,n}(t)$ and $\psi_{m_0,n}(t)$ are respectively the wavelets with the shortest and longest duration in the analysis.

The coefficients $S_{m_0,n}$ are referred to as the *approximation coefficients* and $T_{m,n}$ are the *detail coefficients*. The fast wavelet transform computes the approximation and detail coefficients, and we only show the skeleton of the algorithm for the Haar case, in Algorithm 1. It operates mostly by computing "coefficients from coefficients," in lines 3 and 4, which is the key to its efficiency. The starting point for the algorithm is the sequence $S_{0,n}$ of approximation coefficients at scale 0, which are the areas under the signal over consecutive

Algorithm 1: The FWT algorithm for the Haar case.

Require: $S_{0,0}, S_{0,1}, \ldots, S_{0,2^N-1}$
1: **for** $(m = 1, \ldots, m_0)$ **do**
2: **for** $(n = 0, 1, \ldots, 2^{N-m} - 1)$ **do**
3: $S_{m,n} = \frac{1}{\sqrt{2}}\left(S_{m-1,2n} + S_{m-1,2n+1}\right)$
4: $T_{m,n} = \frac{1}{\sqrt{2}}\left(S_{m-1,2n} - S_{m-1,2n+1}\right)$

unit-width time windows

$$S_{0,n} = \int_{-\infty}^{+\infty} x(t)\phi(t-n)dt = \int_{n}^{n+1} x(t)dt,\ n = 0, 1, \ldots, 2^N - 1. \quad (15)$$

We illustrate the MRA decomposition with a simple example, adapted from [8], and shown in Fig. 4. Consider the signal $x(t) = 1$, $0 \le t < 1$, $x(t) = 2$, $1 \le t < 2$, $x(t) = 3$, $2 \le t < 3$, and $x(t) = 4$, $3 \le t < 4$. Algorithm 1 yields the details and approximations at scales 1 and 2. For example, $T_{1,0}$ and $T_{1,1}$ are each $-1/\sqrt{2}$. $T_{2,0}$ and $S_{2,0}$ are respectively -2 and 5. To see how this decomposition reconstructs $x(t)$, consider for instance the first time step: noting from (12) and (13) that the wavelet and scaling function amplitudes are $1/\sqrt{2}$ for $m = 1$, and $1/2$ for $m = 2$, the contribution of $T_{1,0}$ is $-1/\sqrt{2} \times 1/\sqrt{2} = -1/2$, that of $T_{2,0}$ is $-2 \times 1/2 = -1$, and that of $S_{3,0}$ is $5 \times 1/2 = 5/2$. The total contribution of details and approximations at the first time step therefore adds up to 1, as expected.

V. Worst-Case Current Stimuli

Assume the PDN is a linear time-invariant *RLC* circuit, with v nodes and q current sources. We define a *PDN stimulus* as a collection of current waveforms $\{i_1(t), \ldots i_q(t)\}$, that simultaneously load the PDN, such that $i_j(t)$ attaches to the jth current source. In the following, we will often mention the case when the PDN has a single stimulus $i_j(t)$ on the jth current source and 0 on all other current sources.

Our overarching aim is to use MRA to construct a *synthetic* worst-case PDN stimulus that maximizes voltage drop at a PDN node of interest, and to compute the resulting maximum drop. We call the stimulus "synthetic" because we do not observe it in actual traces or simulations, but rather, we construct it to yield the worst-case voltage drop which satisfies specified constraints.

More specifically, consider an *arbitrary* time point t_0. Our objective is to construct, for every node z of interest on the PDN, a worst-case stimulus which maximizes voltage drop on that node at $t = t_0$. It will become clear, in our formulation below, that t_0 is indistinguishable from any other time point. Therefore, *maximizing voltage drop at $t = t_0$ is tantamount to maximizing voltage drop at any arbitrary time point during circuit operation*. This point is key: although our problem is that of maximizing voltage drop, a transient quantity during circuit operation, we approach it by a single-time-point optimization.

We work backward from t_0 and construct the waveforms $\{i_1(t), \ldots i_q(t)\}$ that make up the PDN stimulus *indirectly*, computing for each a set of details ($T_{m,n}$) and approximations ($S_{m,n}$). As per (14), a waveform is a well-defined linear expression of its details and approximations. The situation is the reverse of the example in Fig. 4: we need to construct the waveform $x(t)$, by first setting the waveform duration to 4, then computing suitable values for $T_{1,0}$, $T_{1,1}$, $T_{2,0}$, and $S_{2,0}$ (under constraints), to build a piecewise constant waveform $(1, 2, 3, 4)$.

The computed waveform will be the solution to a voltage-drop-maximizing optimization problem, and will therefore be a guaranteed worst-case. Before formulating the optimization problem, we need to determine, for every waveform in the PDN stimulus: 1) the time span of its shortest wavelet; 2) its duration; and 3) its composition, in terms of wavelets and scaling functions at every scale. The remainder of this section addresses these issues, and we formulate the optimization problem in Section VI. In Sections V-A, V-B, and V-C we focus on some waveform $i_j(t)$ that is part of the worst-case stimulus for some PDN node z. We will refer to the pair (j, z) as the "input/output pair."

A. Time Span of the Shortest Wavelet

It is typical, in today's design processes, to characterize the PDN in terms of its impedance plot, i.e., a profile of voltage drop versus current excitation frequency. For one thing, this plot gives insight on the highest current frequency at which the PDN exhibits significant voltage drop. Fig. 5 shows the impedance plot of a high-level PDN in an early design high-performance microprocessor. Guided by such a plot, and based on PDN design expertise, designers are able to specify a maximum current frequency, $f_{\max}$, for which they are interested in verifying the PDN.

As we did in Section IV-A, we choose the shortest wavelet so as to have a center frequency equal to $f_{\max}$, and set its scale to $m = 1$. Noting that the scale a of a wavelet coincides with its time span, we compute the time span $a_{\min}$ of the shortest wavelet from (5)

$$a_{\min} = 2.33/(\pi f_{\max}). \quad (16)$$

Let u be the time unit over which the synthetic waveform is constant, which represents the time unit for our analysis. We have

$$u = a_{\min}/2 \quad (17)$$

see, e.g., Fig. 4. In that figure, if $f_{\max} = 300$ MHz, then $a_{\min}$ should be set to 2.47 ns, and u becomes 1.235 ns. Therefore, one time unit corresponds to 1.235 ns. Since scale $m = 1$ has a time span of $a_{\min} = 2u$, scales $m = 2, 3$, etc., therefore represent wavelets and/or scaling functions spanning $4u$, $8u$, and so on.

B. Waveform Duration

Guided by impedance plots, designers can also specify a lower-bound $f_{\min}$ on the frequency of interest, and this guides the choice of the largest scale m_0 needed in the waveform decomposition: we set the time span $a_{\max}$ of the longest wavelet such that its center frequency is at most

equal to $f_{\min}$. In addition, given the dyadic structure of the MRA decomposition, $a_{\max}/a_{\min}$ must be a power of 2. These requirements translate to

$$a_{\max} = 2^{\lceil \log_2(f_{\max}/f_{\min}) \rceil} a_{\min} \tag{18}$$

where $\lceil \cdot \rceil$ is the ceiling function. We deduce the largest scale m_0

$$m_0 = \left[\log_2(a_{\max}/a_{min}) + 1\right] = \left[\log_2(2 f_{\max}/f_{min}).\right] \tag{19}$$

Going back to the example of Fig. 4, and assuming that $f_{\max} = 300\,\text{MHz}$ and $f_{\min} = 200\,\text{MHz}$, then $a_{\min} = 2.47$ ns, and $a_{\max} = 2a_{\min} = 4.94$ ns. The largest scale is $m_0 = 2$.

We begin by describing qualitatively the process of computing the required duration d_j of the waveform $i_j(t)$. Recall that we are interested in the worst-case stimulus for node z. Let $h_{m,0,j,z}(t)$ be the voltage drop waveform on node z when the PDN has a single stimulus of $\psi_{m,0}(t)$ on current source j. Our basic idea is to inspect, for every scale $m = 1, 2, \ldots, m_0$, the time $d_{m,j}$ at which $h_{m,0,j,z}(t)$ dies down (to some negligible value, ϵ) and select $d_j = \max(d_{m,j})$, $m = 1, \ldots, m_0$. We do require that $d_{m,j}$ be an integer multiple of $2^m u$, which ensures that scale m contains an integer number of wavelets.

To see the reasoning behind this, we appeal once again to the example in Fig. 4, picturing the figure as a decomposition of some waveform $i_j(t)$. The inclusion of two wavelets at scale 1, as shown on the figure, assumes that the PDN response on node z, to a wavelet at scale 1 attached to current source j, takes longer than $2u$ but dies down in $4u$ ($d_{1,j} = 4u$), while the presence of a single wavelet at scale 2 implies that the response of the PDN to a wavelet at that scale dies down within $4u$ ($d_{2,j} = 4u$).

We now discuss how to compute $h_{m,0,j,z}(t)$, for all input/output pairs (j, z). A brute-force approach would be to simulate the PDN network qm_0 times, such that every simulation includes the PDN with a single stimulus consisting of a wavelet at a given scale (m_0 scales in total), attached to some current source (q sources in total), with voltage measured at every node. This job, however, can be done far more efficiently, requiring only q *step response* simulations of the PDN, followed by efficient linear operations.

Let $U(t)$ be the unit step function, defined as $U(t) = 0$, for $t < 0$, and $U(t) = 1$, for $t > 0$. Notice that

$$\phi_{0,0}(t) = U(t) - U(t - a_{\min}/2) = U(t) - U(t - u) \tag{20}$$

where u is the time unit. Let $y_{j,z}(t)$ and $s_{m,n,j,z}(t)$ be the voltage drop responses on node z of the PDN to a single stimulus on current source j consisting respectively of $U(t)$ and $\phi_{m,n}(t)$. Because the PDN network is linear, time-invariant, we can write from (20)

$$s_{0,0,j,z}(t) = y_{j,z}(t) - y_{j,z}(t - u). \tag{21}$$

Using (11), in which the time unit u is implicit, and the time-invariance of the PDN, we have

$$s_{m,n,j,z}(t) = s_{m,0,j,z}(t - n2^m u). \tag{22}$$

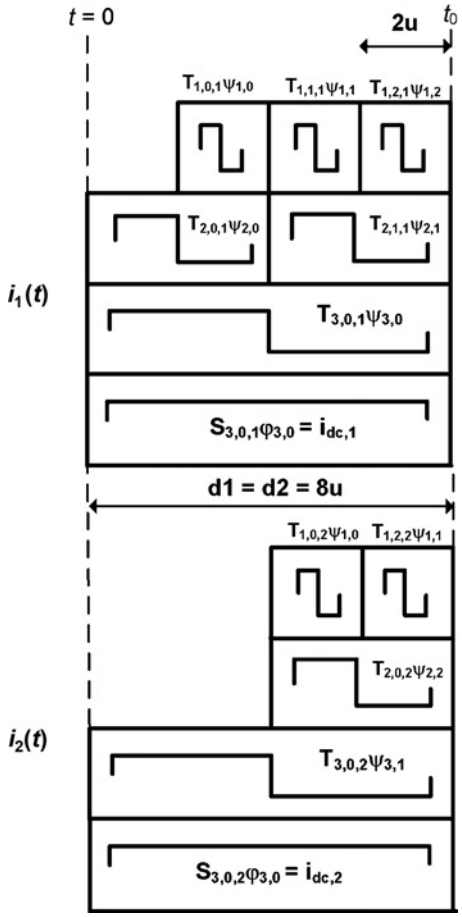

Fig. 6. Example composition for a synthetic stimulus $\{i_1(t), i_2(t)\}$.

Algorithm 2 leverages the PDN linearity to compute the $h_{m,0,j,z}(t)$.

Observe that lines 4 and 5 are a direct result of (12) and (13). Lines 1–6 need to be executed qv times, once for every (j, z) pair, to perform full characterization of the $h_{m,n,j,z}(t)$. However, these steps are efficient, and consist of waveform additions and shifting. The step responses $y_{j,z}(t)$ for every (j, z) pair can be obtained by way of q step function simulations of the PDN network, each simulation consisting of a single step function stimulus on one input, with the output voltage drop measured on all nodes.

C. Waveform Composition

We have seen that a waveform $i_j(t)$ consists of a number of wavelets at scales $1, \ldots, m_0$, as determined by the duration of the PDN response wavelets at every scale. In this section, we make further remarks on the composition of the waveform. We illustrate our remarks with the example in Fig. 6.

Let $n_{m,j}$ be the number of wavelets at scale m (in the terminology of Section V-B, $n_{m,j} = d_{m,j}/2^m u$). In Fig. 6, e.g., $n_{1,1} = 3$, $n_{2,1} = 2$, and $n_{3,2} = 1$. To every wavelet is associated an unknown detail coefficient, that will serve as a variable in the optimization problem in Section VI.

The reader may wonder why wavelets at the smallest scales do not cover the entire duration d_j, e.g., in Fig. 6, scale 1 for

$i_2(t)$ extends only $4u$ prior to t_0. While doing so would be correct, it would induce an unnecessary performance penalty, because the more wavelets the more optimization variables. For instance, if the PDN response to a single stimulus $i_2(t) = \psi_{1,0}(t)$ becomes insignificant within $4u$, then a wavelet at scale 1 attached to current source 2 and starting earlier than $t_0 - 4u$ does not have any meaningful bearing on the voltage drop at t_0.

The piecewise constant shape of the resulting waveform is a consequence of the Haar system, where wavelets and scaling functions are piecewise constant. We illustrate our ideas with the Haar system due to its simplicity, but our approach is the same with all other wavelets (which have scaling functions) [8].

Finally, we note that the scaling function's approximation coefficient in a waveform serves as its dc component, as can be seen in Figs. 4 and 6. Therefore, the overall composition of the waveform is a dc component and a number of shifted wavelets at scales $1, \ldots, m_0$, modulated by detail coefficients.

To summarize, consider the waveform $i_j(t)$, and denote its dc component by $i_{\mathrm{dc},j}$, its largest scale by m_0, the number of wavelets at scale m by $n_{m,j}$, and its detail coefficients by $T_{m,n,j}$

$$i_j(t) = i_{\mathrm{dc},j} + \sum_{m=1}^{m_0} \sum_{n=0}^{n_{m,j}} T_{m,n,j} \psi_{m,n}(t). \tag{23}$$

The decomposition of (23) is illustrated in Fig. 6.

VI. Optimization Problem

A. Objective Function

In this section, we formulate an optimization problem whose solution results simultaneously in the shape of the worst-case current waveform and the value of the maximum voltage drop at any point in time. Let $v(t)$ denote the voltage drop waveform on node z, due to an arbitrary PDN stimulus. Our objective is to maximize the voltage drop at $t = t_0$, or $v(t_0)$. Let $v_j(t)$ be the voltage drop waveform when the PDN has a single arbitrary stimulus $i_j(t)$. By linearity of the PDN, we write

$$v(t_0) = \sum_{j=1}^{q} v_j(t_0). \tag{24}$$

Thus, the objective function requires expressions for each $v_j(t_0)$.

Let $h_{m,n,j,z}(t)$ be the voltage drop waveform on node z, due to a single stimulus $\psi_{m,n}(t)$ attached to current source j. Following the waveform decomposition of Section V-C, and due to the time-invariance of the PDN system, we have

$$h_{m,n,j,z}(t_0) = h_{m,0,j,z}\left((n_{m,j} - n)2^m u\right). \tag{25}$$

From (25), we conclude that $h_{m,n,j,z}(t_0)$ is readily computed from the results of Algorithm 2.

Let k_j be the voltage drop on node z due to a dc stimulus of 1 A on current source j, i.e., the dc gain from current source j to node z. Combining (23) with (25), we write $v_j(t_0)$ as follows

$$v_j(t_0) = k_j i_{\mathrm{dc},j} + \sum_{m=1}^{m_0} \sum_{n=0}^{n_{m,j}} h_{m,0,j,z}\left((n_{m,j} - n)2^m u\right) T_{m,n,j}. \tag{26}$$

We combine (24) with (26) to write the objective function

$$v(t_0) = \sum_{j=1}^{q} k_j i_{\mathrm{dc},j} + \sum_{j=1}^{q} \sum_{m=1}^{m_0} \sum_{n=0}^{n_{m,j}} h_{m,0,j,z}\left((n_{m,j} - n)2^m u\right) T_{m,n,j}. \tag{27}$$

Therefore, the objective function is a linear combination of the $i_{dc,j}$ and $T_{m,n,j}$, our optimization variables.

Algorithm 2: Computation of $h_{m,0,j,z}(t)$ for a pair (j, z)

Require: u (suitable time unit), $y_{j,z}(t)$.

1: $s_{0,0,j,z}(t) = y_{j,z}(t) - y_{j,z}(t-u)$
2: $s_{0,1,j,z}(t) = s_{0,0,j,z}(t-u)$
3: **for** $m = 1, \ldots, m_0$ **do**
4: $\quad s_{m,0,j,z}(t) = \frac{1}{\sqrt{2}}\left[s_{m-1,0,j,z}(t) + s_{m-1,1,j,z}(t)\right]$
5: $\quad h_{m,0,j,z}(t) = \frac{1}{\sqrt{2}}\left[s_{m-1,0,j,z}(t) - s_{m-1,1,j,z}(t)\right]$
6: $\quad s_{m,1,j,z}(t) = s_{m,0,j,z}(t - 2^m u)$

B. Current/Power Constraints

A broad range of current/power bounds can be imposed on the PDN stimulus, given *specifications* known about the design at an early stage. Such bounds have been used in prior art to formulate a verification approach for power networks (see, e.g., [3]), and can be easily embedded into our wavelet-based framework. The simplest bounds are on the minimum and peak-currents, such as

$$0 \le i_j(t) \le i_{\max,j}, \qquad \text{for } 0 \le t \le t_0 \tag{28}$$

where the value of $i_{\max,j}$ is known by specification or simulation of the design block represented by $i_j(t)$. Another bound commonly available from design specification is a global power envelope $P_{\max}$ for a given chip. Denoting the supply voltage by V_{dd}, we can express the global power envelope as

$$\sum_{j=1}^{q} i_j(t) \le P_{\max}/V_{dd}, \qquad \text{for } 0 \le t \le t_0. \tag{29}$$

We can readily use (23) to integrate bounds such as (28) and (29) as linear *constraints* in terms of the optimization variables. This is best illustrated by appealing, once again, to the example in Fig. 6. For instance, a constraint $0 \le i_2(t) \le 1$ expands into

$$\begin{aligned}
0 \le\ & A_3 T_{3,0,2} + i_{\mathrm{dc},2} \le 1 \\
0 \le\ & A_1 T_{1,0,2} + A_2 T_{2,0,2} - A_3 T_{3,0,2} + i_{\mathrm{dc},2} \le 1 \\
0 \le\ & -A_1 T_{1,0,2} + A_2 T_{2,0,2} - A_3 T_{3,0,2} + i_{\mathrm{dc},2} \le 1 \\
0 \le\ & A_1 T_{1,1,2} - A_2 T_{2,0,2} - A_3 T_{3,0,2} + i_{\mathrm{dc},2} \le 1 \\
0 \le\ & -A_1 T_{1,1,2} - A_2 T_{2,0,2} - A_3 T_{3,0,2} + i_{\mathrm{dc},2} \le 1
\end{aligned}$$

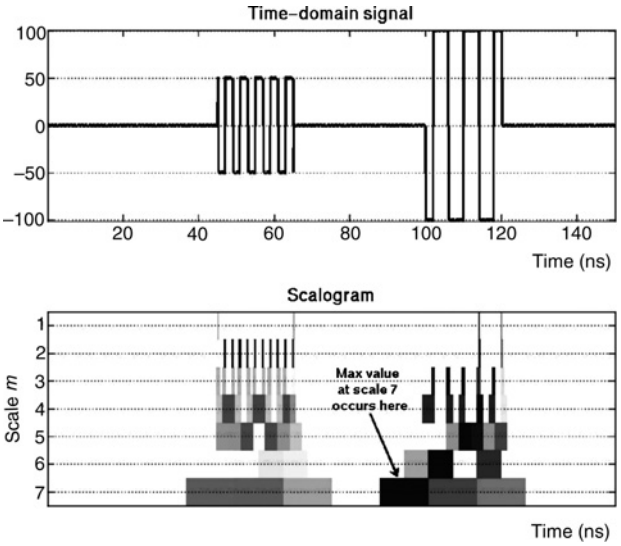

Fig. 7. DWT-scalogram establishes the maximum frequency footprint of a given test. Maximum values are darkest.

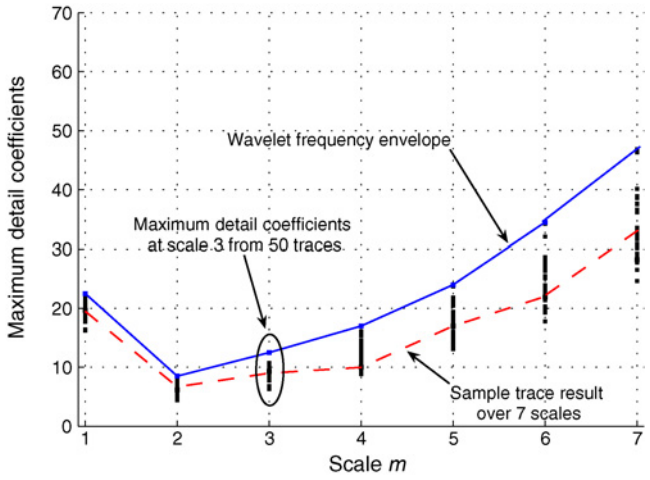

Fig. 8. Wavelet frequency envelope in an early stage microprocessor.

where $A_m = 2^{-m/2}$ is wavelet amplitude at scale m, as in (7).

A third type of constraints that designers can infer is in the form of *max delta*, i.e., a bound on the change in current between successive time units. It can be written in the form

$$-\delta \le i_j(t) - i_j(t-u) \le \delta, \qquad \text{for } u \le t \le t_0. \tag{30}$$

These constraints are not the only ones possible. Any linear constraints in the instantaneous values of $i_j(t)$ (e.g., waveform averages) can do: Since every $i_j(t)$ is a linear combination of some of the optimization variables $i_{\mathrm{dc},j}$ and $T_{m,n,j}$, any linear inequality in terms of the $i_j(t)$ is a linear constraint on the problem.

C. Wavelet Frequency Envelope

We mentioned in Section II that scalograms capture useful properties about the time-signal being analyzed. One such property is the *wavelet frequency envelope*, or the maximum transform value along every scale. When the signal analyzed is a current or switching trace, this value is a proxy for the maximum energy burst for a given frequency band (the wavelet's pass-band at each scale) generated by the current trace.

From the perspective of PDN design, we would want to analyze commonly used high-level benchmark power simulation traces (architectural, register transfer level or logic) in order to generate time–frequency constraints. These constraints would complement current and power bounds, which are normally specified *a priori* by designers.

The wavelet frequency envelope can be formed by applying Algorithm 1 on the available traces and taking the maximum observed values of the resulting detail coefficients, for every scale: We take these values to form a set of optimization constraints. This process is shown in Fig. 7, which illustrates a DWT-based seven-level scalogram (bottom panel) on a time-signal similar to a typical switching trace. Dark squares on the scalogram indicate large values of the wavelet transform at the corresponding scale and time window. A wavelet frequency envelope, obtained after analyzing 50 power traces of an early stage microprocessor design, is shown in Fig. 8 (scale $m = 1$ represents a wavelet spanning 10 clock cycles). Every individual trace results in a scalogram similar to Fig. 7, from which scale-wise maximum detail coefficient values are extracted (the dashed line on Fig. 8 is one such trace). The wavelet frequency envelope, shown in a solid line, is obtained from the maximum details at every scale, observed in any test.

Let $T_{m,\max,j}$ be the value of the wavelet frequency envelope for $i_j(t)$ at scale m. We write optimization constraints as

$$-T_{m,\max,j} \le T_{m,n,j} \le T_{m,\max,j}, \quad \forall n = 0, \ldots, n_{m,j} - 1. \tag{31}$$

Going back to Fig. 6 for an example, assume that, in absolute value, the details at scale 2 for $i_1(t)$ are less than $T_{2,\max,1} = 5$, and those at scale 3 for $i_2(t)$ less than $T_{3,\max,2} = 6$. This would imply the following three constraints: $-5 \le T_{2,0,1} \le 5$, $-5 \le T_{2,1,1} \le 5$, and $-6 \le T_{3,0,2} \le 6$. It is easy to see that constraints in the form of (31) are linear in the optimization variables $T_{m,n,j}$. Therefore, they are readily usable in our problem.

D. Problem Formulation and Solution

$$\max v(t_0) = \sum_{j=1}^{q} k_j i_{\mathrm{dc},j} + \sum_{j=1}^{q} \sum_{m=1}^{m_0} \sum_{n=0}^{n_{m,j}} h_{m,0,j,z}\left((n_{m,j} - n)2^m u\right) T_{m,n,j}$$

such that

$$\begin{aligned} &\text{peak current} && 0 \le i_j(t) \le i_{\max,j}, \quad 0 \le t \le t_0 \\ &\text{(peak power)} && \sum_{j=1}^{q} i_j(t) \le P_{\max}/V_{dd}, \quad 0 \le t \le t_0 \\ &\text{(max delta)} && -\delta \le i_j(t) - i_j(t-u) \le \delta, \quad u \le t \le t_0 \end{aligned}$$

$$\text{(current expression)} \qquad i_j(t) = i_{\mathrm{dc},j} + \sum_{m=1}^{m_0} \sum_{n=0}^{n_{m,j}} T_{m,n,j} \psi_{m,n}(t)$$

$$\text{(wavelet envelope)} \qquad -T_{m,\max,j} \le T_{m,n,j} \le T_{m,\max,j},$$
$$m = 1, \ldots, m_0, \ \ n = 0, \ldots, n_{m,j}-1.$$

The above problem is a linear program (LP) in the $T_{m,n,j}$ and $i_{\mathrm{dc},j}$. The solution of this problem leads to the worst-case voltage drop on node z of the PDN, whereas $\arg\max v(t_0)$ yields the optimization variables, which can directly be plugged into (23) to construct synthetic waveforms $i_j(t)$, $j = 1, \ldots, q$, leading to the worst-case synthetic stimulus for node z.

VII. Experimental Results

A. PDN Verification and Trend Prediction

We tested the proposed approach on several models of an early stage, high-performance microprocessor design, and in this section, we showcase how it provides design-assist value. As is standard practice among microprocessor design groups, we used PDN/die models consisting of *RLC* networks from the VRM to the die, with the die modeled as lumped current sources [7]. Such models are known to capture well the die-level voltage drop. The justification for a lumped die model lies in that, seen from the VRM, voltage drop across the die is, up to high-frequencies, predominantly global [9].

Fig. 9 illustrates a full-die current stimulus of duration $t_0 = 47$ ns, which captures worst-case voltage drop in a high-frequency band. We substantiated our voltage drop calculations by simulating the PDN with this load: The computed worst-case drop of 120 mV matched with simulation, and occurred at t_0, as expected. However, simulation shed additional light in showing the computed worst-case waveform gradually building up the magnitude of voltage oscillations until culminating at t_0. Our approach covers arbitrary frequency bands by generating waveforms of various durations: for a wide band of a few hundred kilohertz to a few hundred megahertz, we generated a stimulus lasting about 6 μs. We note that, in general, the resulting worst-case stimuli were characterized by fairly regular, low-frequency variations at the beginning, with higher-frequency components (wavelets) progressively kicking in, creating local fine patterns that intensify near the end point, t_0.

Our approach may be construed as a generalization of RPT [6]. While RPT works backward from t_0 to construct a worst-case current waveform simply as a sequence of full-swing step functions $I_{\max}$ to $I_{\min}$ and $I_{\min}$ to $I_{\max}$, our approach yields finer patterns in the current waveform, visible on Fig. 9. The reason is that it embeds sophisticated considerations in ways RPT cannot: minimum and maximum waveform frequency, *max delta*, wavelet frequency envelope, and global power constraints. Indeed, when we stripped our optimization formulation from these constraints, we observed that our worst-case waveforms matched with RPT-generated ones, but, as expected, these waveforms overestimate maximum voltage drop with respect to their constrained counterparts, and our experiments indicated 16–25% overestimation. Therefore, our technique has the clear benefit of offering less pessimistic

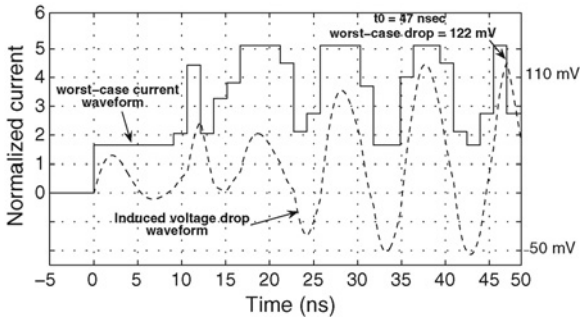

Fig. 9. Example worst-case current waveform and its induced voltage drop.

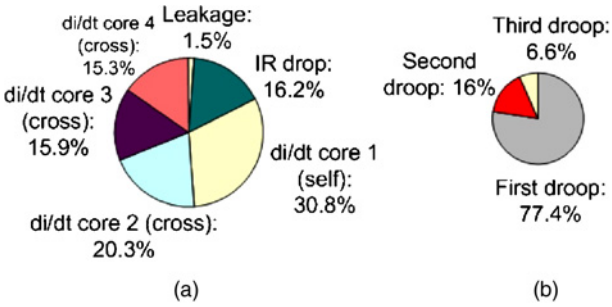

Fig. 10. Predicted voltage drop breakdown for early stage microprocessor models. (a) Predicted voltage drop breakdown on Core 1 in an early-stage four-core design. (b) Predicted breakdown of $\mathrm{d}i/\mathrm{d}t$ voltage drop.

predictions, and broader user-characterization of current waveforms than is possible with RPT. Furthermore, complex PDN models with several current sources, which are simply beyond RPT, can be naturally handled with our technique.

Our approach can help designers predict voltage drop *trends* in an early design stage. By that, we are referring to the ability to systematically quantify the relative magnitudes of different voltage drop components. To illustrate this idea, we applied our analysis on a four-core early stage design, and measured the contribution of leakage, IR-drop, and $\mathrm{d}i/\mathrm{d}t$ switching activity for each core, on the worst-case voltage drop on Core 1. For example, once the optimization problem of Section VI is solved, the impact on Core 1 of $\mathrm{d}i/\mathrm{d}t$ switching that occurs on Core j is computed as

$$\sum_{m=1}^{m_0} \sum_{n=0}^{n_{m,j}} h_{m,0,j,1} \left((n_{m,j} - n) 2^m u \right) T_{m,n,j} / \max v(t_0).$$

Fig. 10(a) shows sample results: IR-drop was found to contribute 18.2 % of the total drop (with a separate 1.5% share for leakage). And while it is expected that the largest individual contributor to $\mathrm{d}i/\mathrm{d}t$ drop on Core 1 is switching activity on the same core (30.8 %), the combined impact of the other cores exceeds that individual contribution (51.5%), thereby firmly establishing the need for an integrated cross-core analysis.

A key advantage of our framework is being frequency-aware. Modern PDNs have complex frequency responses with several resonant modes, each determined by a set of electrical and design parameters. For example, the resonant mode at the highest frequency, commonly referred to as "first droop" [7] is a strong function of package inductance and die capacitance.

Lower frequency resonance, i.e., "second droop," and "third droop" [7], depend on other parameters including motherboard design and bulk capacitors. Designers of different stages of the PDN will therefore be interested in gauging voltage drop across specific frequency ranges: e.g., package designers may care only about minimizing worst-case voltage drop in the first droop range. While current methods to gauge the extent of first, second, and third droop are tedious, measurement-heavy and limited [7], our method naturally incorporates theses considerations by specifying $f_{\max}$ and $f_{\min}$ and leveraging the wavelet filtering properties. Fig. 10(b) shows the breakdown of the three resonant modes in terms of their share of worst-case $\mathrm{d}i/\mathrm{d}t$ voltage drop. It is worth noting that each core required about 500 optimization variables to cover the frequency range of first, second, and third droop.

B. Package/Die Codesign

We applied our approach to the optimization of package inductance L_{pkg} for an early design microprocessor model. The idea was to vary L_{pkg} around the design point and compute the expected worst-case voltage drop with die constraints unchanged (currents, power, frequency). The alternative for designers today is to discover the L_{pkg} value which results in the least voltage drop by simulating the power delivery network for all traces and all candidate L_{pkg} values. Compared with this optimization-by-simulation, our approach has two key advantages: besides the obvious one that simulations are not guaranteed to uncover worst-case drop, simulating the PDN for several (and we used 50) multimillion-cycle traces is simply impracticable. In this respect, our approach is both faster and more accurate.

Fig. 11 shows a plot of the predicted maximum voltage drop versus L_{pkg}. The x-axis depicts the percentage variation from the current design point, represented by the 0 point. The figure shows a plateau around the current design point and reveals that designers have approximately a 30% headroom in their choice of L_{pkg} without incurring a voltage drop penalty. In fact, the figure predicts a slight improvement in the worst-case voltage drop if L_{pkg} were to be increased by about 25% from its planned value. This, somewhat surprising finding, stems from the fact that a change in L_{pkg} changes the PDN frequency response, and the interplay between the new frequency response and the wavelet envelope bounds (which are frequency constraints), results in a net decrease in the maximum voltage drop. Such feedback is of great value to designers: the possibility to change L_{pkg} has direct financial bearing on the chip design. To our knowledge, no other method or commercial tool provides such feedback today.

C. Application to On-Die Power Grids

The on-die component of the PDN, widely known as the *power grid*, is a mesh structure with several layers of metallization, connected "from above" to the package, and "from below" to circuit devices [9]. To test our approach on power grids, we carried out an experiment where we fixed the dimension of a grid (5 mm × 5 mm), total power budget (1 W), and die capacitance (200 nF), then progressively added

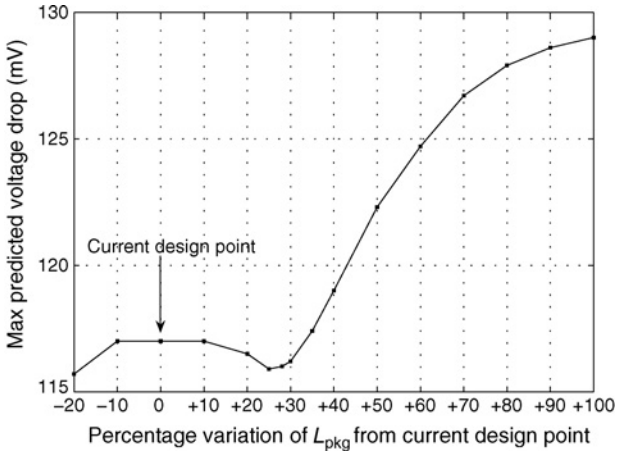

Fig. 11. Application to L_{pkg} optimization.

TABLE I
APPLICATION TO POWER GRID VERIFICATION

Grid Size (#Nodes)	Simulation Time (h)	Optimization Time (182 Nodes) (s)	Average #Variables	Max Vdrop (mV)
11 100	1.6	11	675	59
14 300	1.7	8	604	35
26 000	3.6	5	543	20
36 500	11.3	4	522	13

metallization layers and verified the grid with every new layer. The grid was modeled as an *RC* network [9] and the power budget nonuniformly distributed over 182 current loads. Grid verification consisted of finding the worst-case voltage drop in the 50 MHz—1 GHz range, over the 182 nodes with loads attached. Table I shows the results: Beginning with two metal layers M1 and M2 (row 1), we added M3 (row 2), then M4 (row 3), then M5 and M6 (row 4). Results were obtained on a server with two dual-core, 2.6 GHz processors.

We divided runtime into two components: 1) simulation of step-responses (Section V-B), carried out with HSpice; and 2) solution of the optimization problem (Section VI-D), done with PCx, a freely available LP package. It is clear that simulation is the overwhelming bottleneck. However, since the required simulations are generic step responses, simulators tailored for the power grid will drastically improve efficiency. Also, simulation is a precharacterization step: users need to do it once for a given PDN and can re-run the verification with different constraints or over different frequency bands without further simulation. More importantly, these results show that the cost of maximizing voltage drop, which essentially searches the feasibility space of currents in time, came down to the computation of step responses.

The second highlight is the efficiency of the optimization itself, due to the relatively small number of variables needed per optimization problem (column 4 reports the average number over the 182 problems solved). The key lies in the spatial locality of power grids [9]: excitations from some current source have little impact on the grid beyond a certain neighborhood around the source. Locality has also a frequency aspect: a high-frequency excitation propagates over a smaller

neighborhood than a low-frequency one. Our framework picks up spatial locality seamlessly, while computing wavelet responses from (25) and Algorithm 2: if the response on node z to a wavelet at some scale and source is negligible at t_0, the corresponding variable is dropped out of the zth optimization problem, leaving, for every node, only the set of relevant sources and scales (frequencies), as optimization variables. This also explains the trend of requiring fewer optimization variables with more grid layers: all other things equal, the greater the connectivity of the grid (more metal), the less it is prone to voltage drop, and the smaller the neighborhood of influence per source at a given frequency.

VIII. Conclusion

We introduced the concept of time–frequency description of circuit currents using wavelets, and formulated an optimization framework that solves for the worst-case supply voltage drop. Using this framework, we were able to integrate a broad range of constraints, and introduce new ones based on time–frequency die behavior. We applied this framework on an early stage design process, for package-die codesign, and on power grids, showing how it naturally fills designers' needs for sytematic predictions and characterizations of the power delivery network. The proposed approach was applied in early design of a four-core industrial microprocessor. The application of this approach revealed realistic and nonobvious worst-case scenarios, including the contribution to maximum voltage drop of the die, package, and motherboard stages, and enabled what–if analysis to discover optimal design points.

References

[1] H. Kriplani, F. N. Najm, and I. N. Hajj, "Pattern independent maximum current estimation in power and ground buses of CMOS VLSI circuits: Algorithms, signal correlations, and their resolution," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 14, no. 8, pp. 998–1012, Aug. 1995.

[2] S. Pant, D. Blaauw, V. Zolotov, S. Sundareswaran, and R. Panda, "A stochastic approach to power grid analysis," in *Proc. 41st Ann. Des. Automat. Conf.*, San Diego, CA, 2004, pp. 171–176.

[3] M. Nizam, F. N. Najm, and A. Devgan, "Power grid voltage integrity verification," in *Proc. IEEE Int. Symp. Low-Power Electron. Des.*, 2005, pp. 239–244.

[4] D. Zhou and W. Cai, "Fast wavelet collocation method for high-speed circuit simulation," *IEEE Trans. Circuits Syst. I Fund. Theory Appl.*, vol. 46, no. 8, pp. 920–930, Aug. 1999.

[5] R. Joseph, Z. Hu, and M. Martonosi, "Wavelet analysis for microprocessor design: Experiences with wavelet-based dI/dt characterization," in *Proc. Int. Symp. High-Performance Comput. Arch.*, Madrid, Spain, 2004, pp. 36–46.

[6] V. Drabkin, C. Houghton, I. Kantorovich, and M. Tsuk, "Aperiodic resonant excitation of microprocessor power distribution systems and the reverse pulse technique," in *Proc. IEEE Topical Meet. Elect. Performance Electron. Packag.*, 2002, pp. 175–178.

[7] T. Rahal-Arabi, G. Taylor, M. Ma, and C. Webb, "Design and validation of the Pentium R III and Pentium R IV processors power delivery," in *Proc. IEEE Symp. Very Large Scale Integr. Circuits*, 2002, pp. 220–223.

[8] P. S. Addison, *The Illustrated Wavelet Transform Handbook*, New York: Taylor & Francis, 2002.

[9] S. Pant and E. Chiprout, "Power grid physics and implications for CAD," in *Proc. Des. Autom. Conf.*, San Francisco, CA, 2006, pp. 199–204.

Imad A. Ferzli received the B.E. degree in computer and communications engineering from the American University of Beirut, Beirut, Lebanon, in 2001, and the M.A.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Toronto, Toronto, ON, Canada, in 2004 and 2009, respectively. His research focused on the verification and analysis of power grids and statistical methods in integrated circuits.

During the course of his graduate studies, he interned at Magma Design Automation, San Francisco, CA, and Intel Corporation, Hillsboro, OR. After his Ph.D., he pursued research interests in financial markets by starting his career as an associate in investment risk management at the Canada Pension Plan Investment Board and is currently working on quantitative analytics and risk modeling.

Farid N. Najm (S'85–M'89–SM'96–F'03) received the B.E. degree in electrical engineering from American University of Beirut, Beirut, Lebanon, in 1983, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Illinois at Urbana-Champaign (UIUC), Champaign, in 1986 and 1989, respectively.

From 1989 to 1992, he was a Member of the Technical Staff at Texas Instruments Incorporated, Dallas, TX. He then joined the Department of Electrical and Computer Engineering at UIUC as an Assistant Professor and became Associate Professor in 1997. In 1999, he joined the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada, where he is currently a Professor and Chair. His research interests include computer-aided design for very large scale integration, with an emphasis on circuit level issues related to power, timing, variability, and reliability.

Dr. Najm is is an Associate Editor for the IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. He has received an IEEE Transactions on Computer-Aided Design Best Paper Award, a National Science Foundation (NSF) Research Initiation Award, and an NSF Career Award. He was an Associate Editor of the IEEE Transactions on Very Large Scale Integration Systems from 1997 to 2002. He serves on the executive committee of the International Symposium on Low-Power Electronics and Design (ISLPED), and has served on the technical committees of various conferences, including the International Conference on Computer-Aided Design, Design Automation Conference, Custom Integrated Circuits Conference, International Symposium on Quality Electronic Design, and ISLPED.

Eli Chiprout (M'90) received the B.Eng. degree in electrical engineering from McGill University, Montreal, QC, Canada, in 1988, and the M.S. and Ph.D. degrees in electrical engineering from Carleton University, Ottawa, ON, Canada, in 1992 and 1994, respectively.

He is currently a Principal Engineer for research at Strategic Computer-Aided Design Laboratories, Intel Corporation, Hillsboro, OR, and is Research Lead and Manager for engineers working on post-silicon validation of microprocessors. From 1994 to 2000, he was a Research Staff Member at IBM's research division both in the T. J. Watson Research Center, Armonk, NY, and at the Austin Research Laboratory, Armonk, NY. He has published extensively in multiple conference proceedings and IEEE journals. His research interests include power delivery modeling and optimization, nonlinear macromodeling, variational models, and silicon correlation.

Dr. Chiprout has received the Best Paper Award at the IEEE Design Automation Conference in July 2006. He received the university medal for outstanding research work from Carleton University.