Power Grid Voltage Integrity Verification

Maha Nizam
Department of ECE
University of Toronto
Toronto, Ontario, Canada
maha@eecg.utoronto.ca

Farid N. Najm
Department of ECE
University of Toronto
Toronto, Ontario, Canada
f.najm@utoronto.ca

Anirudh Devgan
Magma Design Automation
Austin, Texas
USA
devgan@magma-da.com

ABSTRACT

Full-chip verification requires one to check if the power grid is *safe*, i.e., if the voltage drop on the grid does not exceed a certain threshold. The traditional simulation-based solution to this problem is computationally expensive, because of the large variety of possible circuit behaviors that would need to be simulated; it also has the disadvantage that it requires full knowledge of the details of the circuit attached to the grid, thereby precluding early verification of the grid. We propose a power grid verification technique that can be applied before the complete circuit has been designed and without exact knowledge of the circuit currents. We use *current constraints*, which are upper bound constraints on the currents that can be drawn from the grid, as a way to capture the uncertainty about the circuit details and activity. Based on this, we propose two solution approaches. One approach gives an upper-bound on the worst-case voltage drop at every node of the grid. Another, less expensive approach, applies a *sufficient condition* (thus, this becomes a conservative approach) to check if the drop on the grid exceeds a given voltage threshold.

Categories and Subject Descriptors:
B.7 [**Integrated Circuits**]: Design Aids

General Terms: Design, Algorithms, Theory

Keywords: Power Grid, Voltage Drop, Voltage Integrity

1. INTRODUCTION

As the supply and threshold voltages are decreasing with technology scaling, full-chip verification is becoming more and more challenging. In particular, checking the integrity of the voltage on the power grid is becoming crucial. With lower supply voltages, smaller voltage drops become more significant and can cause longer circuit delays and lead to soft errors. Thus, supply voltage integrity verification is an integral part of both timing verification and noise immunity checking.

Voltage drop on the grid is mainly due to IR drop and Ldi/dt drop. IR drop is due to the resistance of the metal lines of the power network. The Ldi/dt drop is due to the self and mutual inductances of the power lines. In practice, the effect of inductance in the power grid is only significant at frequencies above a GHz or so, and it can therefore be ignored in many circuits. In this paper, we will use an RC model of the power grid, ignoring the inductance effects and focusing on IR drop.

Power grid verification via traditional circuit simulation requires full knowledge of the current waveform drawn by every circuit block attached to the grid. These current waveforms would be used to simulate the grid and get the voltage drop at every grid node. This drop is then compared to a maximum allowable voltage drop threshold to check if any node on the grid is unsafe (the voltage drop at this node is higher than the threshold). To verify the grid in this way requires a comprehensive set of current waveforms to make sure all the regions on the grid are simulated, and that the obtained voltage drop is a worst-case drop. Also, this method requires full knowledge of the circuit, which is a concern if one would like to verify the grid early in the design flow, before all the circuit details are available.

To overcome these problems, we will use the concept of *current constraints* [1] to capture the uncertainty about the circuit details and circuit behavior. These current constraints are a set of upper-bounds on the currents that would be drawn by the circuit. They can be obtained from simulations of the circuit or from knowledge of the overall power dissipation of the circuit blocks. Using these upper-bound constraints, we will show how to check if the maximum voltage drop on the power grid, under all possible transient current waveforms that satisfy the constraints, exceeds a certain voltage threshold.

A similar verification problem was previously formulated and solved in [1], but under the assumption that the currents on the grid are DC, which can under-estimate the worst-case drop. In this paper, we present a general solution to the problem, in which the currents are allowed to be arbitrary transient waveforms. We will present two solution approaches. First, we present an algorithm that computes an upper bound on the worst-case drop at every node of the grid. We give empirical data that shows that the upper bound can be tight, and that therefore the pessimism in the first approach is not too significant. In the second approach, we will present a method to check if the grid is safe with respect to a given voltage threshold, without attempting to find the exact maximum drop on the grid. This method applies a *sufficient condition* to perform the check, thereby leading to a *conservative approach* by which some grids which are declared unsafe may actually be safe, but no unsafe grid would be declared safe. The method does provide an indication of the nodes that did not pass the safety check, which allows one to focus on specific regions of the grid to perform more detailed analysis.

The rest of the paper is organized as follows. In the next section, a standard RC model of the power grid is adopted and the system equations explained, followed by a brief section 3 that describes a time discretization of the system equations. Current constraints are introduced in section 4 and our first solution approach is then given in section 5, whereby the worst-case voltage drop at every node is sought. Section 6 describes our second solution approach, which efficiently provides a safety check of the grid, and is followed by our conclusion.

2. THE POWER GRID RC MODEL

We consider an RC model of the power grid, where each branch is represented by a resistor and where there exists a capacitor from every node to ground. In addition, some grid nodes have ideal current sources (to ground) representing the currents drawn by the circuits tied to the grid at those nodes, and some grid nodes

ISLPED'05, August 8–10, 2005, San Diego, California, USA

have ideal voltage sources (to ground) representing the connections to the external voltage supply.

Let the power grid consist of $n+p$ nodes, where nodes 1,2,...,n have no voltage sources attached, and nodes $(n+1), (n+2), \ldots, (n+p)$ are the nodes where the p voltage sources are connected. Let c_k be the capacitance from every node k to ground. Let $i_k(t)$ be the current source connected to node k, where the direction of positive current is from the node to ground. We assume that $i_k(t) \geq 0$ and that $i_k(t)$ is defined for every node $k = 1, \ldots, n$, so that nodes with no current source attached have $i_k(t) = 0, \forall t$. Let $\mathbf{i}(t)$ be the vector of all $i_k(t)$ sources, $k = 1, \ldots, n$. Let $u_k(t)$ be the voltage at every node k, $k = 1, \ldots, n$, and let $\mathbf{u}(t)$ be the vector of all $u_k(t)$ signals, $k = 1, \ldots, n$.

Applying Kirchoff's Current Law (KCL) at every node, $k = 1, \ldots, n$, leads to:

$$\mathbf{G}\mathbf{u}(t) + \mathbf{C}\dot{\mathbf{u}}(t) = -\mathbf{i}(t) + \mathbf{G_0}\mathbf{V_{dd}} \tag{1}$$

where $\mathbf{G}$ and $\mathbf{G_0}$ are $n \times n$ conductance matrices resulting from the application of the traditional modified nodal analysis formulation [2] (simplified by the fact that all the voltage sources in this case are from a node to ground), $\mathbf{C}$ is an $n \times n$ diagonal matrix of node capacitances, and $\mathbf{V_{dd}}$ is a constant vector each entry of which is equal to V_{dd}.

If we set all $i_k(t) = 0, \forall t$, then obviously $u_k(t) = V_{dd}, \forall t$, so that the above system equation becomes:

$$\mathbf{G}\mathbf{V_{dd}} = \mathbf{G_0}\mathbf{V_{dd}} \tag{2}$$

By replacing $\mathbf{G_0}\mathbf{V_{dd}}$ by $\mathbf{G}\mathbf{V_{dd}}$ in (1), it can be re-written as:

$$\mathbf{G}[\mathbf{V_{dd}} - \mathbf{u}(t)] - \mathbf{C}\dot{\mathbf{u}}(t) = \mathbf{i}(t) \tag{3}$$

If we now define $v_k(t) = V_{dd} - u_k(t)$ to be the voltage drop at node k, and let $\mathbf{v}$(t) be the vector of voltage drops, then the system equation can be written as:

$$\mathbf{G}\mathbf{v}(t) + \mathbf{C}\dot{\mathbf{v}}(t) = \mathbf{i}(t) \tag{4}$$

This is a *revised* system equation which we can use to directly solve for the voltage drop values. Comparing (4) to (1), it is easy to see that the circuit described by this equation consists of the original power grid, but with all the voltage sources set to zero (short circuit) and all the current source directions reversed.

3. TIME DISCRETIZATION

We will be working with a discrete-time version of the system equation, which can obtained by considering that, for small $\Delta t > 0$, the derivative of a function $x(t)$ can be approximated by:

$$\dot{x}(t) \approx \frac{x(t) - x(t - \Delta t)}{\Delta t} \tag{5}$$

Applying this to $\dot{\mathbf{v}}(t)$ in (4), leads to:

$$\left(\mathbf{G} + \frac{\mathbf{C}}{\Delta t}\right)\mathbf{v}(t) = \frac{\mathbf{C}}{\Delta t}\mathbf{v}(t - \Delta t) + \mathbf{i}(t) \tag{6}$$

It will be useful to think of this equation as capturing the space of voltage vectors $\mathbf{v}(t)$ and $\mathbf{v}(t - \Delta t)$ that are allowed, given the current vector $\mathbf{i}(t)$.

Define a matrix $\mathbf{A}$ as:

$$\mathbf{A} = \left(\mathbf{G} + \frac{\mathbf{C}}{\Delta t}\right) \tag{7}$$

and let a_{ij} be the entry of $\mathbf{A}$ at row i and column j. As in [3], one can show that $\mathbf{A}$ is an M-matrix [4] because it satisfies the following:

1. $a_{ii} > 0 \ \forall i$
2. $a_{ij} \leq 0 \ \forall j \neq i$
3. $a_{ii} > \sum_{j \neq i} |a_{ij}|$, where strict inequality comes from the addition of the positive diagonal matrix $\mathbf{C}/\Delta t$ to the conductance matrix $\mathbf{G}$ and the fact that every node has non-zero parasitic capacitance.

Among other things, being an M-matrix means that the inverse of matrix $\mathbf{A}$ consists of only non-negative values.

As indicated above, we will assume throughout that $\mathbf{i}(t) \geq 0$, so that currents flow only from the grid into the circuitry tied to it. If we further assume that $\mathbf{v}(t - \Delta t) \geq 0$, then since $\mathbf{C}$ is a non-negative matrix, it is clear that (6) leads to $\mathbf{A}\mathbf{v}(t) \geq 0$. Since $\mathbf{A}^{-1}$ is also non-negative, then this leads to the fact that:

$$\mathbf{v}(t - \Delta t) \geq 0 \quad \Longrightarrow \quad \mathbf{v}(t) \geq 0 \tag{8}$$

By induction, if we assume that at some time in the past the voltage drops on the grid are non-negative (meaning that no grid node has a voltage higher than V_{dd}), then they will be non-negative for all time. This is a fairly mild requirement to make, because as we will see in section (6.2) it is possible to assume that at some time in the (distant) past, the grid voltages were all zero. Therefore, it will be implicitly assumed throughout this paper that:

$$\mathbf{v}(t) \geq 0, \forall t \tag{9}$$

so that voltage drop is never negative.

4. CURRENT CONSTRAINTS

As in [1], we define two types of constraints, *local constraints* and *global constraints*. Local constraints are upper bounds on individual current sources, which may represent the current drawn by individual cells, or larger blocks. They can be fixed values (DC) or time-waveforms (transient). They can be expressed as:

$$\mathbf{0} \leq \mathbf{i}(t) \leq \mathbf{I_L}, \forall t \geq 0 \quad \text{or} \quad \mathbf{0} \leq \mathbf{i}(t) \leq \mathbf{i_L}(t), \forall t \geq 0 \tag{10}$$

where $\mathbf{I_L}$ is a vector of fixed current values and $\mathbf{i_L}(t)$ is a vector of current waveforms. These constraints are defined for *every* node on the grid; nodes that are not connected to a current source have $I_L = 0$ or $i_L(t) = 0, \forall t \geq 0$. Local constraints, by themselves, are obviously insufficient for verification, because the chip components do not simultaneously draw their maximum currents. This is where global constraints come in.

Global constraints are upper bounds on the sum totals of current drawn by groups of current sources, and they may also be either DC or transient. If there is a total of k global constraints, they may be expressed in matrix form as:

$$\mathbf{0} \leq \mathbf{S}\mathbf{i}(t) \leq \mathbf{I_G}, \forall t \geq 0 \quad \text{or} \quad \mathbf{0} \leq \mathbf{S}\mathbf{i}(t) \leq \mathbf{i_G}(t), \forall t \geq 0 \tag{11}$$

where $\mathbf{S}$ is a $k \times n$ matrix of 0s and 1s that define the groups of current sources included in every constraint, $\mathbf{I_G}$ is a vector of fixed current values, and $\mathbf{i_G}(t)$ is a vector of current waveforms.

Local and global constraints can be combined as follows:

$$\begin{aligned} \mathbf{U}\mathbf{i}(t) \leq \mathbf{I_m}, \forall t \geq 0 \qquad & \text{or} \qquad \mathbf{U}\mathbf{i}(t) \leq \mathbf{i_m}(t), \forall t \geq 0 \\ \text{with} \quad \mathbf{i}(t) \geq \mathbf{0}, \quad & \forall t \geq 0 \end{aligned} \tag{12}$$

where $\mathbf{U}$ is an $(n + k) \times n$ matrix, whose first n rows form an indentity matrix, and whose last k rows are the $\mathbf{S}$ matrix. Notice that every entry of the matrix $\mathbf{U}$ is non-negative.

Current constraints offer a useful abstraction for capturing the uncertainty of circuit behavior, given the large number of logic vectors that can propagate through the circuit. They also offer a way to capture uncertainty about the circuit details early in the design flow. For circuits that are completely determined, local constraints may be obtained from exhaustive simulation of individual cells, or from a simple count of the transistors in that cell that are tied to the supply. Global constraints are chosen based on some knowledge of the peak power dissipation of a block or a chip, if that information is available from previous design knowledge or from test-case simulations. When little or no information is available, such as early in the design process, the global constraints may simply be specified as possible values to drive a "what if" type of analysis, or may be based on an estimate of the target block area coupled with knowledge of the typical power density of that technology (power dissipation per unit area). Being a means of capturing uncertainty in cases of limited information about the circuit and/or its activity, we envision that in practice

Table 1: Test-case power grids and results of the upper-bound algorithm

Power Grid				Constraints(mA)					Drop stats (% of Vdd)				Execution Cost	
Name	nodes	C4s	Sources	Local	Global				Min	Max	Mean	Std	#iterations	Time
G1	90	10	10	20	70	-	-	-	1.4	19	5.8	3.85	24	55.79 s.
G2	450	34	15	20	70	-	-	-	0.02	34.1	6.5	5.6	25	16.8 min.
G3	503	26	30	3	20	-	-	-	0.1	10.36	2.66	2.32	13	24.2 min.
G4	476	53	30	3	10	15	-	-	0.0	3.18	0.73	0.64	6	8.5 min.
G5	450	79	30	3	10	15	15	-	0.0	2.78	0.445	0.445	2	2.23 min.
G6	477	26	30	3	20	-	-	-	0.0	5.6	1.63	1	13	22 min.
G7	450	26	30	2	20	10	-	-	0.0	4.61	1.23	0.79	11	14.2 min.
G8	424	26	30	2	20	10	10	-	0.0	4.15	1.16	0.73	6	6 min.
G9	973	51	50	10	200	-	-	-	0.11	22.36	5.74	3.5	15	1.25 h.
G10	922	102	50	10	100	50	-	-	0.0	19.27	2.18	2.36	7	54 min.
G11	870	154	50	20	100	50	50	-	0.0	9.72	1.36	1.445	6	33 min.
G12	922	51	50	10	200	-	-	-	0.26	21.54	5.45	3.24	15	1.8 h.
G13	871	51	60	10	200	200	-	-	0.127	33.96	6.36	5	13	1.4 h.
G14	819	51	60	10	100	150	50	-	0.16	34.36	7.27	5.82	11	52 min.
G15	768	51	50	20	100	100	150	150	0.0	56.4	12.18	9.82	14	55 min.
G16	666	51	50	20	100	100	150	150	0.3	46	10.9	7.9	13	50 min.
G17	1924	101	100	10	200	-	-	-	0.27	24.8	4.81	3.27	10	9.4 h.
G18	1822	203	100	20	150	250	-	-	0.0	25.54	4.54	4.18	6	4.8 h.
G19	1721	304	100	20	250	200	150	-	0.0	16.7	2.9	2.87	6	3 h.
G20	1823	101	100	20	500	-	-	-	0.145	36	9.8	6.45	8	6 h.
G21	1721	101	100	30	300	200	-	-	0.1	68.5	17.54	11.6	9	6 h.
G22	1620	101	100	20	500	300	300	-	5	58	12.9	7.7	9	4.8 h.
G23	2420	151	100	50	700	-	-	-	0.0	87	15.9	16.9	17	11.4 h.

DC constraints ($\mathbf{I_L}$ and $\mathbf{I_G}$) would be easier to specify than transient constraints ($\mathbf{i_L}(t)$ and $\mathbf{i_G}(t)$). Thus, in this paper we will perform the analysis under DC current constraints, although the work can be very easily extended to handle transient constraints. In summary, the current constraints can be expressed as:

$$0 \leq \mathbf{U}\mathbf{i}(t) \leq \mathbf{I_m}, \forall t \geq 0 \quad \text{with} \quad \mathbf{i}(t) \geq \mathbf{0}, \forall t \geq 0 \tag{13}$$

The currents are transient but their upper bounds are DC.

5. WORST-CASE VOLTAGE DROP

We now present our first solution approach. Given a set of current constraints we will show how to find the worst-case voltage drop at every node. In addition to checking if the grid is safe, this method gives information on the extent of the problem when the grid is unsafe. In sub-section 5.1, we will first see how one can set up a numerical optimization approach for solving this problem exactly, which will be seen to be too expensive. In the following sub-section, we then develop an efficient technique for finding an *upper bound* on the worst-case voltage drop on every node. We give empirical data to show that these bounds are tight and, therefore, useful in practice.

A starting point for the whole analysis is to assume that the current source currents are zero for all $t \leq 0$, so that the grid is safe and has zero voltage drop at time 0, and to then examine how the system evolves over time for $t \geq 0$. Given this, it is easy to see that the worst-case voltage at any node, over all possible currents that satisfy the constraints, is a non-decreasing function of time over $t \geq 0$ and, because it is bounded, must therefore converge to some steady state value. We are, naturally, interested in the steady state condition, where the solution becomes independent of the initial condition ($\mathbf{i}(t) = 0, \forall t \leq 0$), because that represents the true general solution for the grid. However, it will also be instructive to see how we approach the steady state.

5.1 Exact worst-case

Consider a sequence of (fixed value) time steps Δt, taken starting at time 0, and let $t = k\Delta t$, where $k > 0$ is an integer. Recall the discrete-time system equation (6), which can be written as:

$$\left(\mathbf{G} + \frac{\mathbf{C}}{\Delta t}\right)\mathbf{v}(t) - \frac{\mathbf{C}}{\Delta t}\mathbf{v}(t - \Delta t) = \mathbf{i}(t) \tag{14}$$

We can write the same equation at the previous time step as:

$$\left(\mathbf{G} + \frac{\mathbf{C}}{\Delta t}\right)\mathbf{v}(t - \Delta t) - \frac{\mathbf{C}}{\Delta t}\mathbf{v}(t - 2\Delta t) = \mathbf{i}(t - \Delta t) \tag{15}$$

This is repeated until we get to time $t - k\Delta t = 0$, where we get:

$$\left(\mathbf{G} + \frac{\mathbf{C}}{\Delta t}\right)\mathbf{v}(t - (k-1)\Delta t) = \mathbf{i}(t - (k-1)\Delta t) \tag{16}$$

because $\mathbf{v}(t - k\Delta t) = \mathbf{v}(0) = 0$. Notice that every equation of the above sequence expresses the current at every time point as a linear combination of the voltage drop vectors at the current and the previous time steps. If we are interested in finding the maximum voltage drop at node i at time t, we would have to perform the following optimization:

Maximize: $v_i(t)$
Subject to: for $q = 0, \ldots, (k-1)$:

$$0 \leq \mathbf{U}\left(\mathbf{G} + \tfrac{\mathbf{C}}{\Delta t}\right)\mathbf{v}(t - q\Delta t) - \mathbf{U}\tfrac{\mathbf{C}}{\Delta t}\mathbf{v}(t - (q+1)\Delta t) \leq \mathbf{I_m}$$

whereby we have enforced the current constraints (13) at every time point. The number of constraints is thus multiplied by the number of time steps, leading to a potentially very large problem, which can be prohibitively expensive in practice. For small grids, we can construct and solve the optimization problem, as we will show below, starting with $k = 1$, and gradually increasing k until we get the same result for two consecutive values of k, within some error tolerance ϵ. That steady-state result would be the worst-case voltage drop at every node in the general case, i.e., in the absence of a specific initial condition.

5.2 Upper bounds

We define a voltage vector $\mathbf{v}(t)$ to be *feasible* if there exist current source waveforms (with $\mathbf{i}(t) = 0, \forall t \leq 0$) that satisfy the current constraints and which can cause the grid to realize the voltage values in $\mathbf{v}(t)$ at time t (starting as always with $\mathbf{v}(0) = 0$). We define $\mathcal{V}_f(t)$ to be the set of all feasible voltage vectors at time t. We also define a voltage vector $\mathbf{v_{opt}}(t)$ as the solution of the following optimization problem:

$$\forall i : v_{opt,i}(t) = \max_{\mathbf{v}(t) \in \mathcal{V}_f(t)} v_i(t) \tag{17}$$

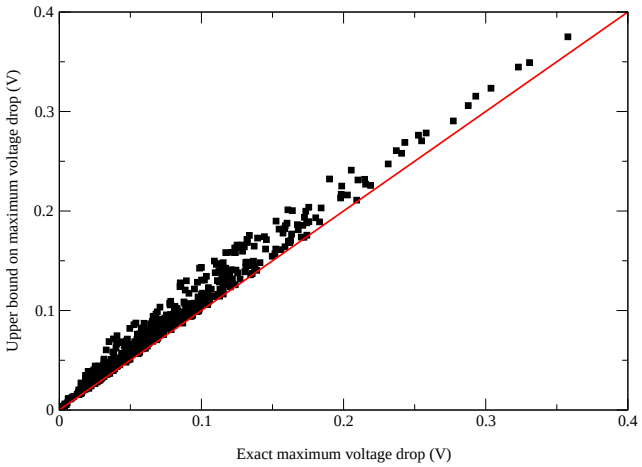

Figure 1: Voltage drop comparisons for all nodes in grids G1 and G2.

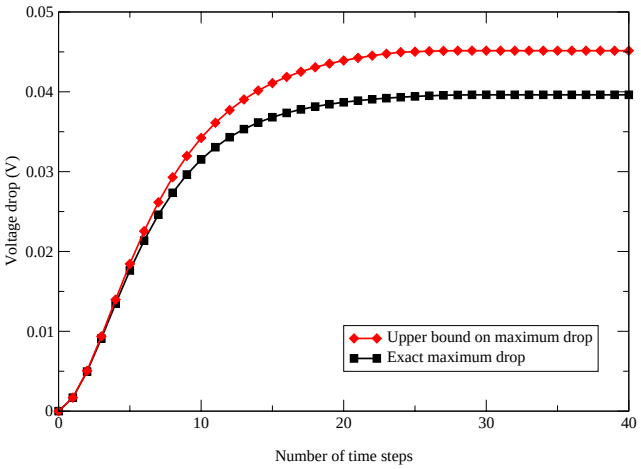

Figure 2: Voltage drop comparisons, over time, for a node in grid G1, with $\Delta t = 0.5$ps.

Notice that $v_{opt,i}(t)$ is nothing but the solution of the exact worst-case problem described in the preceding section. Notice also that $\mathbf{v_{opt}}(t)$ may not be a feasible voltage vector. Ideally, one would like to find $\mathbf{v_{opt}}(t)$ for every t, and especially for $t \to \infty$, but that can be very expensive, as we saw above. Instead, we will now propose an algorithm by which a vector $\mathbf{v_b}(t)$ can be efficiently computed such that $\mathbf{v_b}(t) \geq \mathbf{v_{opt}}(t), \forall t$.

From (6), the space of voltages allowed by the current constraints can be expressed as:

$$\mathbf{U}\frac{\mathbf{C}}{\Delta t}\mathbf{v}(t-\Delta t) \leq \mathbf{U}\left(\mathbf{G}+\frac{\mathbf{C}}{\Delta t}\right)\mathbf{v}(t) \leq \mathbf{U}\frac{\mathbf{C}}{\Delta t}\mathbf{v}(t-\Delta t)+\mathbf{I_m} \quad (18)$$

Suppose that we know that, at a certain time $t-\Delta t$, the voltage drop on the grid satisfies:

$$0 \leq \mathbf{v}(t-\Delta t) \leq \mathbf{v_b}(t-\Delta t) \quad (19)$$

Since $\mathbf{U}$ and $\mathbf{C}$ are non-negative matrices, then (18) and (19) lead to:

$$0 \leq \mathbf{U}\left(\mathbf{G}+\frac{\mathbf{C}}{\Delta t}\right)\mathbf{v}(t) \leq \mathbf{U}\frac{\mathbf{C}}{\Delta t}\mathbf{v_b}(t-\Delta t)+\mathbf{I_m} \quad (20)$$

Based on this, we can now present an algorithm to find the upper-bound on $\mathbf{v_{opt}}(t)$, as follows:

Step 1. $t = 0$, $\mathbf{v_b}(0) = 0$

Step 2. $t = t + \Delta t$

Step 3. For $i = 1, \ldots, n$:
Maximize: $v_{b,i}(t)$
Subject to:
$0 \leq \mathbf{U}\left(\mathbf{G}+\frac{\mathbf{C}}{\Delta t}\right)\mathbf{v_b}(t) \leq \mathbf{U}\frac{\mathbf{C}}{\Delta t}\mathbf{v_b}(t-\Delta t)+\mathbf{I_m}$

Step 4. If $\mathbf{v_b}(t) \approx \mathbf{v_b}(t-\Delta t)$, EXIT, else go to Step 2.

The algorithm is obviously much simpler than running the exact approach, because it involves an optimization across only one time step. The following claim proves the correctness of this algorithm.

CLAIM 1. *If $\mathbf{v_b}(t)$ is generated according to the above algorithm, then $\mathbf{v_b}(t) \geq \mathbf{v_{opt}}(t), \forall t \geq 0$.*

PROOF. Since the base case $\mathbf{v_b}(0) \geq \mathbf{v_{opt}}(0)$ is trivially satisfied (as an equality) at time 0, then the claim is true by induction if we can prove the following:

$$\mathbf{v_b}(t-\Delta t) \geq \mathbf{v_{opt}}(t-\Delta t) \quad \Rightarrow \quad \mathbf{v_b}(t) \geq \mathbf{v_{opt}}(t) \quad (21)$$

Define $\mathcal{V}_b(t)$ to be the set of all vectors $\mathbf{v}(t)$ that satisfy (20). Notice that $\mathcal{V}_b(t)$ is the solution space of the core optimization step in our algorithm, which gives $\mathbf{v_b}(t)$, and which may be summarized as:

$$\forall i : v_{b,i}(t) = \max_{\mathbf{v}(t)\in\mathcal{V}_b(t)} v_i(t) \quad (22)$$

Consider any $\mathbf{v}(t) \in \mathcal{V}_f(t)$, then there exists another feasible voltage vector $\mathbf{v}(t-\Delta t)$ such that $\mathbf{v}(t)$ and $\mathbf{v}(t-\Delta t)$ satisfy (18). Since $0 \leq \mathbf{v}(t-\Delta t) \leq \mathbf{v_{opt}}(t-\Delta t)$, due to (17), and assuming $\mathbf{v_{opt}}(t-\Delta t) \leq \mathbf{v_b}(t-\Delta t)$, consistent with (21), then (19) is true, leading to (20), so that that $\mathbf{v}(t) \in \mathcal{V}_b(t)$. This means that $\mathcal{V}_f(t)$ is a subset of $\mathcal{V}_b(t)$, i.e.:

$$\mathcal{V}_f(t) \subset \mathcal{V}_b(t) \quad (23)$$

Given the definition of $\mathbf{v_{opt}}(t)$ in (17) and given (22), it then follows that $\mathbf{v_b}(t) \geq \mathbf{v_{opt}}(t)$, because maximizing over a superset of $\mathcal{V}_f(t)$ must give a result that is at least as large as maximizing over $\mathcal{V}_f(t)$. This completes the proof. □

When the algorithm terminates, we exit with the values obtained at the last iteration, and we take these values as the upper bounds on the worst-case voltage drop at every node.

5.3 Experimental Results

It is clear that the optimization step in the above algorithm is a *linear program* (LP). To solve this LP, we have used a Primal-Dual Interior Point Method algorithm [5], using the PCx package [6]. A number of tests were conducted on a set of randomly-generated test-bench power grids (as was done in [1]) and simulated on a 1 GHz SUN machine with 4 GB of memory. For every grid, the number of nodes and the number of voltage sources (C4) are specified up-front and an initial uniform (regular) grid is generated. Then, a certain percentage of nodes are eliminated from the grid in order to make it somewhat non-uniform.

Table 1 shows all the test cases. It also shows the number of sources that are connected to the grid (#C4s). These sources are randomly placed on the grid. For each test case, a single value is chosen as the local upper-bound constraint, as shown in the table, and a set of global constraints are specified. The simulation results, showing the number of iterations (time steps) before completion and the total CPU time required, are also shown in Table 1. The results indicate that the number of time steps by which convergence is reached is fairly small.

For each grid, the table also shows statistics of the worst-case voltage drop values found at all the nodes: minimum, maximum, mean, and standard deviation. Notice how increasing the number of voltage supplies (C4s) on the grid decreases the maximum and other stats of the voltage drop. Fig. 1 shows a correlation plot with the exact maximum voltage drop on one axis and the upper bound on the drop on the other. The figure shows closely aligned

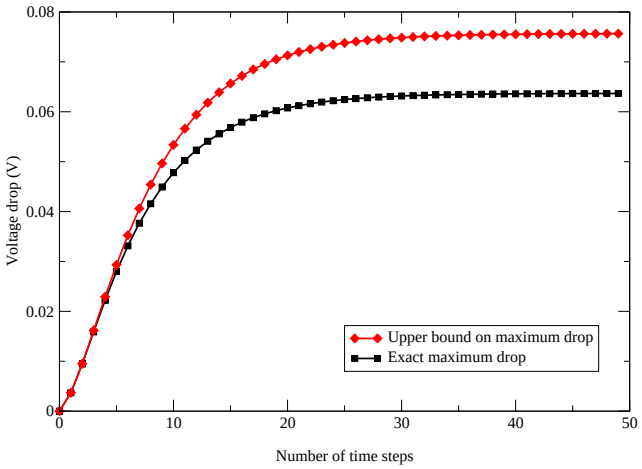

Figure 3: Voltage drop comparisons, over time, for a node in grid G2, with $\Delta t = 0.5$ps.

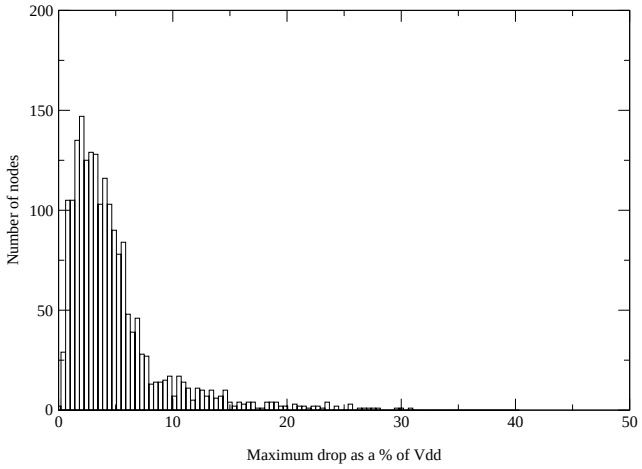

Figure 4: Voltage drop histogram for grid G17.

points, which means that the upper bound is fairly tight, for all nodes in grids G1 and G2. Figs. 2 and 3 show the voltage drop versus time for a node in grid G1 and another in grid G2, respectively. The exact solution for G1 took 7 minutes while the upper bound algorithm took only 55 seconds, and the exact solution for G2 took 18 hours, while the upper bound algorithm took only 17 minutes. This obviously limits the size of the grids for which we can offer comparisons to the exact solution, and the comparison results for these two small grids were typical of what we saw. Finally, Fig. 4 shows a histogram of the node voltage drops in grid G17, obtained with the upper bound algorithm.

6. GRID SAFETY CHECK

We will now present our second solution approach. Given the set of current constraints and a maximum voltage drop threshold, we will show how one can check if the grid is safe, i.e., if the voltage drop is below threshold at all the nodes, under all possible transient currents that satisfy the constraints, without having to find the worst case voltage drop at every node.

6.1 Conditional safety

Let $\mathbf{v_{max}}$ be an $(n \times 1)$ vector, where $v_{max,i}$ is the maximum allowable voltage drop at node i, for $i = 1, \ldots, n$, and suppose that we know that at a certain time $(t - \Delta t)$, the grid was safe, so that:

$$0 \leq \mathbf{v}(t - \Delta t) \leq \mathbf{v_{max}} \tag{24}$$

Since $\mathbf{U}$ and $\mathbf{C}$ are non-negative matrices, then (18) and (24) lead to:

$$0 \leq \mathbf{U}\left(\mathbf{G} + \frac{\mathbf{C}}{\Delta t}\right)\mathbf{v}(t) \leq \mathbf{U}\frac{\mathbf{C}}{\Delta t}\mathbf{v_{max}} + \mathbf{I_m} \tag{25}$$

Notice that the space of voltages $\mathbf{v}(t)$ captured by (25) is a superset of that captured by (18). We can now formulate the following *conditional safety* checking problem:

PROBLEM 1. *If the grid is safe at time $(t - \Delta t)$, check if it is safe at time t.*

To solve this problem, it is sufficient to try and maximize every component of $\mathbf{v}(t)$, subject to the constraints (25), as follows:

For $i = 1, \ldots, n$:
Maximize: $v_i(t)$
Subject to: $0 \leq \mathbf{U}\left(\mathbf{G} + \frac{\mathbf{C}}{\Delta t}\right)\mathbf{v}(t) \leq \mathbf{U}\frac{\mathbf{C}}{\Delta t}\mathbf{v_{max}} + \mathbf{I_m}$

If it turns out that the answer is YES, so that $\mathbf{v}(t) \leq \mathbf{v_{max}}$, then by induction we can say that if the grid is safe at some previous time t_0, it will be safe for all times $t \geq t_0$. If the answer is NO, then the result is inconclusive. Those nodes that violate the threshold, leading to the NO answer, are possibly unsafe, but one doesn't know for sure. Thus, the approach is conservative.

If we are seeking more accuracy, we can use an algorithm similar to the one used in the exact worst-case solution, but using only a few time steps, as follows. Suppose that we know that the grid was safe between time $(t - k\Delta t)$ and $(t - \Delta t)$, so that:

$$0 \leq \mathbf{v}(t - q\Delta t) \leq \mathbf{v_{max}} \quad \text{for} \quad q = 1, \ldots, k \tag{26}$$

We can formulate a new conditional safety checking problem as follows:

PROBLEM 2. *If the grid is safe at all times $(t - q\Delta t)$, for $q = 1, \ldots, k$ check if it is safe at time t.*

As in section 5, we can write the following inequality:

$$\left(\mathbf{G} + \frac{\mathbf{C}}{\Delta t}\right)\mathbf{v}(t-(q-1)\Delta t) - \frac{\mathbf{C}}{\Delta t}\mathbf{v}(t-q\Delta t) = \mathbf{i}(t-(q-1)\Delta t) \tag{27}$$

To find the maximum voltage drop at time t, we should maximize the drop at each node, given the two sets of constraints in (26) and (27), as follows:

For $i = 1, \ldots, n$:
Maximize: $v_i(t)$
Subject to: for $q = 1, \ldots, k$:
$0 \leq \mathbf{v}(t - q\Delta t) \leq \mathbf{v_{max}}$
$0 \leq \mathbf{U}\left(\mathbf{G} + \frac{\mathbf{C}}{\Delta t}\right)\mathbf{v}(t-(q-1)\Delta t) - \mathbf{U}\frac{\mathbf{C}}{\Delta t}\mathbf{v}(t-q\Delta t) \leq \mathbf{I_m}$

In the next sub-section, we will show that the grid can always be assumed to have been safe prior to some time in the distant past. This will provide us with a *base case* that can be combined with the inductive step in Problems 1 and 2, to check if the sufficient condition for the grid safety is met at all time.

6.2 Initial condition

In order to guarantee that the grid was safe *before some time* t_0 in the past, it is sufficient to consider that the circuit currents are zero for all $t \leq t_0$, and then become possibly non-zero after that. This is such a trivial condition that one may want to simply *assume* it and move on. However, in order to satisfy the critical reader, we will show that even if nothing is known about the currents for all previous times $(-\infty, t]$, we can identify a previous time prior to which the grid may be assumed to have been safe. To see this, consider that the grid is a stable linear system with bounded inputs (the source currents) and bounded outputs (the

Table 2: Verification results showing the number of nodes that violate the different voltage drop constraints.

	Voltage Drop Threshold (% of Vdd)						Execution Time	
	V_{max} = 5 %		V_{max} = 10 %		V_{max} = 50 %			
Name	$k=1$	$k=3$	$k=1$	$k=3$	$k=1$	$k=3$	$k=1$	$k=3$
G1	50	29	40	22	0	0	2.01 s.	30.1 s.
G2	153	88	72	37	0	0	59.12 s.	10 min.
G3	146	62	81	41	0	0	71.38 s.	12 min.
G4	47	17	5	0	0	0	57.65 s.	11 min.
G5	6	4	54	25	0	0	51.17 s.	11 min.
G6	72	28	48	28	10	4	60.06 s.	12 min.
G7	57	29	35	14	0	0	56.25 s.	8.6 min.
G8	98	47	422	2	12	8	47.48 s.	5.6 min
G9	238	129	59	30	0	0	6.06 min.	73 min.
G10	199	112	99	48	5	0	5.92 min.	63 min.
G11	170	117	80	56	0	0	5.63 min.	43.4 min.
G12	301	209	193	120	1	1	4.31 min.	50 min.
G13	380	267	273	188	130	55	3.6 min.	33 min.
G14	368	266	272	181	82	28	3.44 min.	26.8 min.
G15	459	361	337	262	206	115	2.84 min.	21.7 min.
G16	441	358	303	210	108	66	2.96 min.	24.3 min.
G17	776	693	439	271	73	50	33.36 min.	11 h.
G18	486	316	324	252	0	0	29.9 min.	6 h.
G19	339	219	57	34	0	0	21.56 min.	3.4 h.
G20	1231	1093	728	561	25	15	24.18 min.	5.8 h.
G21	1293	1082	785	575	162	125	25.34 min.	4.3 h.
G22	1137	932	704	500	65	47	14.67 min.	2.38 h.
G23	1205	982	1113	970	281	217	51.72 min.	7.8 h.

node voltages). Using standard linear system theory, if the voltage drop at time t_0 is $\mathbf{v}(t_0)$, then the voltage drop at any time $t \geq t_0$ is given by:

$$\mathbf{v}(t) = e^{-\mathbf{C}^{-1}\mathbf{G}(t-t_0)}\mathbf{v}(t_0) + \mathbf{v_s}(t) \tag{28}$$

where $\mathbf{v_s}(t)$ is the response due to the input currents alone, over $[t_0, t]$, with zero initial conditions. A key point to remember is that, as t increases, the dependence of the voltage drop on the initial conditions decreases. Therefore, if one considers a $t_0 \ll t$, such that $\mathbf{v}(t)$ is insensitive to $\mathbf{v}(t_0)$, then one may assume that $\mathbf{v}(t_0) = 0$ without affecting the voltages at time t. As a matter of fact, we can assume that at all time $t' \leq t_0$, we have $\mathbf{v}(t') = 0$. Thus, the grid can be assumed safe at all times before t_0 without affecting the grid safety at time t. Therefore, for any grid, we can assume that it was safe at some number of time steps k in the past. With this, solving Problems 1 or 2, two approaches which were introduced to check conditional safety, thus turns out to be sufficient to check safety *unconditionally*.

6.3 Experimental Results

The optimization problems above were implemented using the same optimization package as before, for the same test cases in Table 1. For each grid, we check the safety under three different voltage drop thresholds ($\mathbf{v_{max}}$). Table 2 shows the number of nodes that violate every threshold and the number of nodes that are safe, using both a one-step only solution ($k = 1$) and a solution with 3 steps involved ($k = 3$). Notice how the number of unsafe nodes decreases with increasing threshold and how increasing the number of voltage sources on the grid decreases the number of unsafe nodes. Also notice how the solution including 3 time steps indicates a smaller number of unsafe nodes. Table 2 also shows the CPU time required.

Looking at the CPU time, we notice that a grid of just under 2,500 nodes takes just under an hour to be checked. Such a grid is of course extremely small compared to full-chip grids that contain millions of nodes. Nevertheless, our approach is important for at least two reasons: 1) it is the *first* approach to rigorously check the safety of the grid in a truly vectorless approach - remember that we are checking the grid over *all transient currents* that satisfy the constraints, 2) coupled with a hierarchical modeling approach, our method can be applied either to parts of the grid, or to the top-level main feeder network of the grid. The ability to test the main feeder network early in the design flow is a major advantage of our technique. We believe that these techniques can lead to practical methods for early vectorless grid verification.

7. CONCLUSION

Checking the safety of the power grid is an essential part of chip verification. To do this by circuit simulation requires gathering complete information about the circuit and doing expensive simulation. Instead, we propose a verification approach based on a set of current constraints (upper bounds on the current sources attached to the grid). We first gave a conservative algorithm to get an upper bound on the maximum voltage drop at every node of the grid, we then offered two simple safety check algorithms.

8. REFERENCES

[1] D. Kouroussis and F. N. Najm. A static pattern-independent technique for power grid voltage integrity verification. In *ACM/IEEE 40th Design Automation Conference (DAC-03)*, pages 99–104, Anaheim, CA, June 2-6 2003.

[2] L. T. Pillage, R. A. Rohrer, and C. Visweswaraiah. *Electronic Circuit and System Simulation Methods*. McGraw-Hill, New York, 1995.

[3] J. N. Kozhaya, S. R. Nassif, and F. N. Najm. A multigrid-like technique for power grid analysis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 21(10):1148–1160, October 2002.

[4] H. Kim, J. Xu, and L. Zikatanov. A multigrid method based on graph matching for convection-diffusion equations. *Numerical Linear Algebra with Applications*, 10(1-2):181–195, December 2002.

[5] R. Saigal. *Linear Programming: a Modern Integrated Analysis*. Kluwer Academic Publishers, 1995.

[6] J. Czyzyk, S. Mehrotra, M. Wagner, and S. J. Wright. *PCx User Guide (Version 1.1)*. Optimization Technology Center, Northwestern University, November 1997.