A Static Pattern-Independent Technique for Power Grid Voltage Integrity Verification*

Dionysios Kouroussis
Department of ECE
University of Toronto
Toronto, Ontario, Canada
diony@eecg.utoronto.ca

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

ABSTRACT

Design verification must include the power grid. Checking that the voltage on the power grid does not drop by more than some critical threshold is a very difficult problem, for at least two reasons: *i)* the obviously large size of the power grids for modern high-performance chips, and *ii)* the difficulty of setting up the right simulation conditions for the power grid that provide some measure of a *realistic* worst case voltage drop. The huge number of possible circuit operational modes or workloads makes it impossible to do exhaustive analysis. We propose a static technique for power grid verification, where *static* is in the sense of *static* timing analysis, meaning that it does not depend on, nor require, user-specified stimulus to drive a simulation. The verification is posed as an optimization problem under user-supplied *current constraints*. We propose that current constraints are the right kind of abstraction to use in order to develop a practical methodology for power grid verification. We present our verification approach, and report on the results of applying it to a number of test-case power grids.

Categories and Subject Descriptors

B.7 [**Integrated Circuits**]: Design Aids

General Terms

Design, Algorithms, Verification

1. INTRODUCTION

With the advent of low-voltage technologies, the timing of modern integrated circuits is increasingly sensitive to supply voltage fluctuations. If the supply voltage drops by too much, the increased circuit delay may lead to soft errors. Thus, power grid analysis and verification has become central to high-performance chip design [1, 2, 3]. In high-performance chips, it is typical for the on-chip power distribution network to start out as a uniform grid structure which then becomes gradually non-uniform as it is transformed throughout the design process as dictated by the need for routing resources. For brevity, we will refer to the on-chip power supply network as *the power grid* or, simply, *the grid*.

*This research was supported by Micronet, with funding from ATI Technologies Inc. and from Altera Corp.

DAC 2003, June 2-6, 2003, Anaheim, California, USA.

There are many sources of voltage fluctuation within on-chip power grids, such as IR-drop, Ldi/dt drop, and resonance between the grid and the package. Often, especially for simulation of the power grid in the chip's core, at frequencies below a GHz or so, inductance is neglected and one is focused on only the IR-drop given an RC structure of the grid. Most simulation techniques for power grid analysis model the power grid as an RC network and use some form of circuit simulation to complete the simulation. When using any type of simulation technique, the user is required to provide information on current sources that represent typical currents drawn by the circuit off the grid. The current waveforms of these current sources may be determined by a prior simulation of the circuit under given input vector stimulus. One would then use these current waveforms to simulate the power grid in order to check the grid voltages. Specifically, one would like to check that the worst case voltage drop on the grid does not exceed some threshold. Unfortunately, given the very large number of possible circuit behaviors, one needs to simulate the circuit (for the currents) and the grid (for the voltage drops) for a large number of clock cycles or vector sequences, which is impractical. There is also a methodology problem in that one would like to verify the grid early on, before detailed information on the circuit is available.

We are interested in the possibility of determining whether the voltage fluctuations on the grid exceed user-supplied thresholds without complete knowledge of the circuit currents. Instead, we will assume that only *incomplete* information of the circuit currents is available, the type of information which may be easy to get early on in the design process. We will refer to a grid that meets all the voltage drop requirements under all possible currents that satisfy the incomplete current specification given by the user as being a *robust grid*. Given a power grid, and a set of incomplete current specifications, we are interested in checking if the grid is robust.

Exactly what form the incomplete current specification should take is a question that is best answered based on practical design methodologies. In the following, after pointing out a key monotonicity property of the power grid, we will describe one type of incomplete specification referred to as *current constraints*. These will essentially be upper bound constraints on the currents. We will then describe a technique for checking robustness under given current constraints, based on linear programming. This approach has been implemented and tested on a number of test cases. The results will be presented, with conclusions.

2. MONOTONICITY

We consider an RC model of the power grid, where each branch of the grid is represented by a resistor and where there exists a capacitor from every grid node to ground. In addition, some grid nodes have ideal current sources (to ground) representing the current drawn by the circuit tied to the grid at that point, and some grid nodes have ideal voltage sources (to ground) representing the connections to the external voltage supply. Given this model, we will point out a key *monotonicity* property of the power grid,

which will be useful in the following. Basically, this says that if we increase any of the currents driving the grid, at any point in time, then the overall voltage waveform at any point on the grid will either decrease or stay the same, but will not rise.

Let the power grid consist of $n+p$ nodes, where nodes $1, 2, \ldots, n$ 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 application of the traditional modified nodal analysis formulation [4] (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}. The matrix $\mathbf{G}$ has several useful properties. It is symmetric positive definite [5] and can be shown to be an M-matrix [6] which means, among other things, that its inverse consists of only non-negative values. Notice that, 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:

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

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 one can solve directly for the voltage drop values. Notice 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. In the following, we will mainly be concerned with this *modified power grid* and the revised system of equations (4). We can now express the *monotonicity* property of the grid (see [7] for a proof) as follows:

PROPOSITION 1. **(monotonicity)** *If $\mathbf{v}(t)$ is the voltage drop due to $\mathbf{i}(t)$ and $\mathbf{v}^*(t)$ is the voltage drop due to $\mathbf{i}^*(t)$, then the power grid has the following property:*

$$\text{if} \quad \mathbf{i}^*(t) \geq \mathbf{i}(t), \forall t \geq 0, \quad \text{then} \quad \mathbf{v}^*(t) \geq \mathbf{v}(t), \forall t \geq 0 \tag{5}$$

which we will express by saying that the grid is monotone.

A similar result was earlier proven [8] for the special case of an RC tree driven by a single voltage source. Based on the monotonicity property, we can now make a couple of statements that will be useful below. Let I_k be an upper bound on $i_k(t)$ over the time period of interest, say $0 \leq t \leq \infty$. Let $I_1, I_2, \ldots, I_n$ form a $n \times 1$ vector $\mathbf{I}$ and let $\mathbf{V}$ be the solution of the system when the DC currents $\mathbf{I}$ are applied as inputs, which may be found by solving the DC system:

$$\mathbf{G}\mathbf{V} = \mathbf{I} \tag{6}$$

Then, from the monotonicity property, it is clear that $\mathbf{i}(t) \leq \mathbf{I}, \forall t \geq 0$ leads to $\mathbf{v}(t) \leq \mathbf{V}, \forall t \geq 0$. Finally, another related result is that, considering the DC system (6), if $\mathbf{I}^* \geq \mathbf{I}$, then $\mathbf{V}^* \geq \mathbf{V}$.

3. PEAK CURRENT CONSTRAINTS

We will develop two related notions of an incomplete current specification, which we will refer to as *current constraints*: *local constraints*, and *global constraints*.

3.1 Local Constraints

A local constraint relates to a single current source. For instance, one may specify that current $i_k(t)$ never exceeds a certain fixed level $I_{L,k}$, i.e., $i_k(t) \leq I_{L,k}, \forall t \geq 0$. This upper bound may be simply known from prior simulation, if the cell or block is already available, or it may be a best-guess based on the area of the cell or block and on perhaps the *power density* of the design (total power divided by total area). If further information is available on the circuit behavior over time, then the user may be able to specify an upper bound *waveform*, as a time function, so that $i_k(t) \leq i_{L,k}(t), \forall t \geq 0$. We will assume that every current source tied to the power grid has an upper bound associated with it, be it a fixed bound or a waveform bound. If a grid node does not have a current source attached to it, i.e., $i_k(t) = 0, \forall t \geq 0$, then we specify a fixed zero-current upper bound for that node, $I_{L,k} = 0$. This convention will be useful later on. In this way, we have a local constraint associated with every node of the power grid. We express these constraints in vector form as:

$$\mathbf{0} \leq \mathbf{i}(t) \leq \mathbf{I_L}, \forall t \geq 0 \quad \text{or} \quad \mathbf{0} \leq \mathbf{i}(t) \leq \mathbf{i_L(t)}, \forall t \geq 0 \tag{7}$$

Notice that, if *only* local constraints are provided, then checking robustness is trivial, due to the monotonicity of the power grid: simply set each current source to its maximum allowable value and simulate the grid. The resulting voltage drops are the maximum that can exist under these constraints. Of course, with only local constraints, the results can be very pessimistic because it is never the case that all chip components simultaneously draw their maximum current, hence the need for global constraints. Handling global constraints, however, is not as straightforward.

3.2 Global Constraints

It can also be useful to express constraints related to all current sources or to sub-groups of current sources. For instance, if the total power dissipation of the chip is known, even approximately, then one may say that the sum of all the current sources is no more than a certain upper bound. We refer to this type of constraint as a *global constraint*. In general, a global constraint corresponds to the case when the sum of the currents for a group of current sources is specified to have an upper bound. These constraints are useful to express the fact that certain groups of current sources (corresponding to certain functional blocks, or perhaps to the whole chip) draw no more than a certain total level of current, corresponding perhaps to the known total power dissipation for that block. The upper bound, corresponding to the jth global constraint, may be a fixed bound $I_{G,j}$, or a waveform bound $i_{G,j}(t)$. If m is the number of available global constraints, then we express all the global constraints in matrix form as:

$$\mathbf{U}\mathbf{i}(t) \leq \mathbf{I_G} \qquad \text{or} \qquad \mathbf{U}\mathbf{i}(t) \leq \mathbf{i_G}(t) \tag{8}$$

where $\mathbf{U}$ is a $m \times n$ matrix that contains only 0s and 1s.

3.3 Robustness

The local and global constraints can be combined into a single matrix inequality, as follows:

$$\begin{aligned} \mathbf{L}\mathbf{i}(t) \leq \mathbf{I_m} \qquad &\text{or} \qquad \mathbf{L}\mathbf{i}(t) \leq \mathbf{i_m}(t), \\ \text{with} \quad \mathbf{i}(t) \geq \mathbf{0}, \quad &\forall t \geq 0 \end{aligned} \tag{9}$$

where $\mathbf{L}$ is an $(n+m) \times n$ matrix of 0s and 1s, whose first n rows form an identity matrix (1s on the diagonal and 0s everywhere else) and whose remaining m rows correspond to the matrix $\mathbf{U}$, and where $\mathbf{I_m}$ and $\mathbf{i_m}(t)$ are $(n+m) \times 1$ vectors.

If we consider the case where one is verifying robustness under fixed (DC) currents, one is dealing with DC inputs $\mathbf{I}$, DC voltages $\mathbf{V}$, and the DC system $\mathbf{G}\mathbf{V} = \mathbf{I}$. The local constraints become $\mathbf{0} \leq \mathbf{I} \leq \mathbf{I_L}$, the global constraints are $\mathbf{U}\mathbf{I} \leq \mathbf{I_G}$, and their combination is:

$$\mathbf{L}\mathbf{I} \leq \mathbf{I_m}, \qquad \mathbf{I} \geq \mathbf{0} \tag{10}$$

Fixed current upper bounds will be referred to as *DC constraints*, otherwise, one is working with *transient constraints*.

Suppose that we are given, for each node k, the maximum allowable voltage drop at that node, $V_{m,k}$ (a voltage *threshold*) and let $\mathbf{V_m}$ be the vector of thresholds at all the nodes. We define the grid to be *robust* for a given set of constraints if and only if $\mathbf{V} \leq \mathbf{V_m}$ for *any* current vector $\mathbf{i}(t)$ that satisfies these constraints. In general, the voltage thresholds and the voltages themselves may be functions of time. If one is given a set of DC constraints, $\mathbf{I_m}$, then the following result is useful.

PROPOSITION 2. *The grid is robust for all transient currents whose peak values satisfy a given set of DC constraints if and only if it is robust for all DC currents that satisfy these constraints.*

PROOF. The forward direction is trivial: since a DC current is a special limiting case of a transient current, then robustness under a class of transient currents implies robustness under any DC currents that also belong to that class. The reverse direction is true because of the monotonicity property: assuming that the grid is robust under DC currents, then given a transient current assignment whose peaks satisfy the constraints, we can construct a DC current assignment by setting a DC current value equal to the peak value of each current source. This DC current assignment satisfies the constraints, therefore the grid must be robust under this assignment. Now, since for each current source the transient current is always below the corresponding DC current, then by (5), the grid is also robust for that transient current assignment. □

This result is useful because it provides that, *when the constraints are given as DC constraints*, then checking robustness under all DC currents that satisfy these constraints provides more than just robustness under DC currents - it provides that under a large class of transient currents (as given in proposition 2), the grid is also robust. Fixed upper-bound DC constraints are much easier to specify than transient upper-bound constraints. This is because in order to provide transient constraints, one requires much more design knowledge, including some notion of system or circuit timing. Thus, we anticipate that, in practice, power grid verification under fixed upper-bound constraints will be very useful, especially for verification early in the design process.

Considering proposition 2, one would like to prove a stronger statement. Specifically, it would be useful if one could prove that the grid is robust under *all* transient currents that satisfy the given DC constraints if and only if it is robust under some class of DC currents. This was not possible. Instead, we can make the following statement:

PROPOSITION 3. *A sufficient condition for the grid to be robust under all transient currents that satisfy a given set of DC constraints (local and global) is that it is robust under all DC currents that satisfy the local subset of the DC constraints.*

PROOF. Given a transient current assignment, we construct a DC current assignment by setting each current source DC value equal to the peak of the corresponding transient waveform. Since by assumption the grid is robust for these DC currents, then the voltage drop on the grid will only be reduced (due to monotonicity) if any of the transient currents are either *i)* reduced from their peak values at some time points, or *ii)* reduced from their peak value because of the imposition on them of the global constraints. □

If one is given a set of transient constraints, then if we denote by $\mathbf{I_m}$ a DC upper bound on the transient constraint waveforms, so that $\mathbf{i_m}(t) \leq \mathbf{I_m}, \forall t \geq 0$, then we say that these DC constraints *dominate* the given transient constraints. In this case, it is easy to prove the following:

PROPOSITION 4. *A sufficient condition for the grid to be robust under all transient currents that satisfy a given set of transient constraints is that the grid be robust under all DC currents that satisfy a set of local DC constraints that dominate the given local transient constraints.*

PROOF. The proof is similar to that of proposition 3 and is again based on the monotonicity of the grid. □

This condition is sufficient but not necessary. It means that, if given transient constraints, then a DC robustness check may be employed, but the results may be conservative and pessimistic. It remains an open question to develop a transient robustness check that works directly with given transient constraints.

4. DC ROBUSTNESS

As was shown above, if we are given a set of DC constraints that are satisfied by the *peaks* of all possible transient currents, then checking robustness may be done for only DC currents and, therefore, may employ a DC model of the grid, meaning a resistive model. This is possible due to monotonicity of the grid and offers a major simplification of the problem.

4.1 A Sufficient Condition

Notice that a *sufficient* condition for the grid to be robust is:

$$\mathbf{I_L} \leq \mathbf{G V_m} \tag{11}$$

This is because, $\mathbf{I} \leq \mathbf{I_L}$ leads to $\mathbf{I} \leq \mathbf{G V_m}$ and since $\mathbf{G}^{-1}$ is non-negative, then $\mathbf{V} = \mathbf{G}^{-1}\mathbf{I} \leq \mathbf{V_m}$. This condition is very easy to check, requiring a single matrix vector multiplication. However, it is only a sufficient, not necessary, condition of robustness.

4.2 Voltage Formulation

By making use of the relationship $\mathbf{I} = \mathbf{GV}$, we can express the DC constraints in terms of voltages:

$$\mathbf{LGV} \leq \mathbf{I_m}, \quad \mathbf{V} \geq \mathbf{0} \tag{12}$$

Thus, the robustness checking problem can be expressed as:

Problem 1. Check if $\mathbf{V} \leq \mathbf{V_m}$ is satisfied for *all* voltages $\mathbf{V}$ that satisfy $\mathbf{LGV} \leq \mathbf{I_m}, \mathbf{V} \geq \mathbf{0}$.

Notice that the system equation $\mathbf{I} = \mathbf{GV}$ is implicitly satisfied by the first n rows of the matrix inequality $\mathbf{LGV} \leq \mathbf{I_m}$. This is because, as was expressed in relation to (7), that set of inequalities covers *all* the nodes, and any nodes with no current source attached are assigned a fixed zero-current upper bound constraint.

4.3 Normalization

We can normalize the variables and the constraints as follows. For $k = 1, 2, \ldots, n$, let $x_k = V_k / V_{m,k}$ form a dimensionless vector $\mathbf{x}$. If $\mathbf{g_1}, \mathbf{g_2}, \ldots, \mathbf{g_n}$ are the column vectors of $\mathbf{G}$ then:

$$\mathbf{GV} = \sum_{k=1}^{n} \mathbf{g_k} V_k = \sum_{k=1}^{n} \mathbf{g_k} x_k V_{m,k} \tag{13}$$

Dividing both sides of the first inequality in (12) by the current value $I_{nom} \stackrel{\triangle}{=} \max_{k=1,\ldots,n} I_k$, which is positive, we get:

$$\mathbf{L} \sum_{k=1}^{n} \mathbf{g_k} \frac{V_{m,k}}{I_{nom}} x_k \leq \frac{1}{I_{nom}} \mathbf{I_m} \tag{14}$$

We can transform the matrix $\mathbf{G}$ into a new matrix $\mathbf{G'}$ by multiplying every column k of $\mathbf{G}$ by $V_{m,k}/I_{nom}$ and call the product of $\mathbf{L}$ and $\mathbf{G'}$ a new matrix $\mathbf{M}$, so that:

$$\mathbf{Mx} = \mathbf{LG'x} = \mathbf{L} \sum_{k=1}^{n} \mathbf{g_k} \frac{V_{m,k}}{I_{nom}} x_k \tag{15}$$

Matrix $\mathbf{M}$ is dimensionless, as is vector $\mathbf{x}$. Furthermore, the right hand side vector $\mathbf{I_m}/I_{nom}$ is also dimensionless, call it a new $(n+m) \times 1$ vector $\mathbf{b}$, leading to the new dimensionless constraint set:

$$\mathbf{Mx} \leq \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0} \tag{16}$$

and a new normalized dimensionless formulation of the problem as follows:

Table 1: Voltage drop results. The % non-uniformity is the percentage of the original grid nodes that were removed so as to create a non-uniform grid.

Power Grid				Constraints (A)					% V_{dd} drop			
Name	# nodes	# C4s	% Non-Uniformity	Local	Global				Max.	Min.	Mean	Std. Dev.
N-529-C4-26-RD0	503	26	0	0.03	5	-	-	-	14.40	1.39	7.11	2.81
N-529-C4-53-RD0	476	53	0	0.03	7	2	-	-	9.15	0.22	3.02	1.89
N-529-C4-79-RD0	450	79	0	0.05	10	2	2	-	15.34	0.40	4.23	3.11
N-529-C4-26-RD26	477	26	5	0.03	5	-	-	-	15.08	1.22	7.75	3.29
N-529-C4-26-RD53	450	26	10	0.01	3	1	-	-	9.15	0.23	3.03	1.89
N-529-C4-26-RD79	424	26	15	0.015	3	1	1	-	8.44	0.34	4.47	2.01
N-1024-C4-51-RD0	973	51	0	0.03	15	-	-	-	19.10	0.76	7.48	3.38
N-1024-C4-102-RD0	922	102	0	0.035	17	2.5	-	-	24.33	0.36	5.29	4.29
N-1024-C4-154-RD0	870	154	0	0.035	17	8	8	-	8.22	0.28	2.57	1.43
N-1024-C4-51-RD51	922	51	5	0.03	15	-	-	-	17.78	0.64	7.00	3.16
N-1024-C4-51-RD102	871	51	10	0.03	15	2	-	-	30.54	0.73	8.12	5.12
N-1024-C4-51-RD154	819	51	15	0.03	15	2	2	-	32.78	0.28	9.05	6.09
N-1024-C4-51-RD205	768	51	20	0.02	10	2	2	2	19.33	0.01	5.85	4.42
N-1024-C4-51-RD307	666	51	30	0.02	6	1	1	1	18.37	0.20	5.32	3.28
N-2025-C4-101-RD0	1924	101	0	0.03	30	-	-	-	31.57	0.63	8.42	5.73
N-2025-C4-203-RD0	1822	203	0	0.03	30	6	-	-	13.14	0.36	3.48	2.03
N-2025-C4-304-RD0	1721	304	0	0.03	30	6	6	-	6.11	0.24	1.95	1.00
N-2025-C4-101-RD101	1823	101	5	0.03	30	-	-	-	22.34	0.80	7.32	3.45
N-2025-C4-101-RD203	1721	101	10	0.03	30	3	-	-	28.82	0.21	7.45	4.75
N-2025-C4-101-RD304	1620	101	15	0.03	30	3	3	-	24.36	0.65	8.49	4.76
N-3025-C4-151-RD454	2420	151	15	0.03	33	-	-	-	20.04	0.18	5.96	3.00

Problem 2. Check if $\mathbf{x} \leq \mathbf{1}$ is satisfied for *all* vectors $\mathbf{x}$ that satisfy $\mathbf{Mx} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}$.

Thus, the problem is now expressed as a check for the existence of a feasible point (a point that satisfies the constraints) that lies outside the unit cube.

4.4 Linear Programming

It is significant that the constraints (16) are *linear* and we propose to construct a linear program (LP) around them as a way to check robustness. We will refer to the space of voltages represented by these constraints as the *feasible space*. Note that this space, being bound by joint linear constraints, is convex, as in a standard LP. The space of voltages defined by $\mathbf{x} \leq \mathbf{1}$ is also convex. Thus, one way of checking robustness, is to take the nodes one at a time, and solve an LP every time in which the objective is to maximize the voltage at that node:

$$\begin{array}{ll} \text{Maximize} & x_k \\ \text{Subject to} & \mathbf{Mx} \leq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{array} \tag{17}$$

As one solves the LP, one can of course stop when x_k exceeds a 1 value, and declare a violation. If, instead, the maximum (normalized) voltage at that node is less than 1, then that node is "safe", and one can then switch to another node and start solving a new LP with the new objective function. One can keep doing this until a violation is found or all nodes have been proven safe, in which case the grid is declared robust. A useful by-product of this (for subsequent optimization) is the voltage slacks at each node (the smallest difference between V_k and $V_{m,k}$).

This process can be expensive, of course, especially if one has to go through all the nodes. One obvious way of speeding this up is to check violations at *all* the nodes as the LP tries to maximize one x_k. We can do this by checking whether $\mathbf{x} \leq \mathbf{1}$ at every step of the LP. But, more importantly, it is possible to achieve significant reduction in computational cost by not initiating a new LP for every node. Instead, using the simplex method, it is possible to simply update the objective function row in the *simplex tableau* [9] in order to "switch" the LP to a new objective function for a new node, while leaving the whole of the (large) constraint part of the tableau intact. In this way, we have found that upon switching to a new node, it typically takes the LP a small number of basic/non-basic exchanges to find the maximum for the new node. Recall that a basic/non-basic exchange is a sequence of row-operations by which a column in the simplex tableau [9] is converted so that all its entries are 0 except for one which is 1. This is the basic iterative step of the simplex method.

5. EXPERIMENTAL RESULTS

This method has been implemented and tested on a number of test-case power grids, using a Dell machine with a 1.6GHz Intel Pentium-4 CPU and 2.0GB of RAM. Not having access to power grids from industrial designs, and because we need a large number of grids to test our approach under different conditions, we have opted to *generate* a number of grids ourselves. The grid generation process is automatic, and employs a random number generator, as well as user-specified technology and topology parameters. Starting with a square uniform grid of a given size, we proceed to randomly delete a user-specified percentage of nodes (which we call the *degree of non-uniformity*), thus rendering the grid structurally non-uniform. Typical geometric and physical grid characteristics (e.g. grid dimensions) as well as characteristics of the fabrication process (e.g. sheet resistance of a particular level of metallization) are given by the user, leading to an initial value of the conductance of every branch. When a node is deleted, the conductances of the remaining surrounding edges (branches) are increased by a random amount around a user-specified percentage of their initial values. The rationale behind this is to allow the non-uniform grid to be loaded with currents comparable to its uniform predecessor while exhibiting comparable IR-drops. The number of V_{dd} (C4) sites and leakage current sources are supplied by the user; the C4s are then distributed at random over the grid nodes.

Table 1 shows the characteristics of a number of test grids, along with the statistics of the worst case voltage drop for each, as a percentage of V_{dd}. In all the runs, the voltage thresholds at all the nodes were specified to be artificially large so as not to create any violations. In this way, the optimization technique is allowed to operate on all the nodes, yielding the full distribution of worst-case voltage drops at all the nodes. Some typical distributions

Table 2: Complexity Results.

Power Grid	Complexity Analysis							
	First Solve		Successive Solves (iterations & time)					
	Iter.	time(s)	Min.	Max.	Mean	Std. Dev.	time/iter.(ms)	Total time
N-529-C4-26-RD0	462	1.63	0	81	6.59	13.11	3.6	13.22s
N-529-C4-53-RD0	385	1.13	0	225	27.55	35.24	2.9	41.12s
N-529-C4-79-RD0	374	0.95	0	256	32.62	40.32	2.5	40.04s
N-529-C4-26-RD26	432	1.29	0	72	7.76	13.03	3.1	12.47s
N-529-C4-26-RD53	441	1.10	0	59	5.32	10.60	3.6	6.91s
N-529-C4-26-RD79	391	0.84	0	64	6.37	10.84	3.3	6.98s
N-1024-C4-51-RD0	935	15.38	0	226	20.67	33.78	16.4	345.07s
N-1024-C4-102-RD0	792	11.80	0	412	68.85	58.67	14.7	973.87s
N-1024-C4-154-RD0	739	9.63	0	399	118.61	67.80	13.1	1364.50s
N-1024-C4-51-RD51	871	13.08	0	212	17.73	31.68	19.2	319.28s
N-1024-C4-51-RD102	824	10.79	0	219	24.12	31.69	13.1	285.71s
N-1024-C4-51-RD154	776	9.01	0	206	20.46	29.73	11.6	188.73s
N-1024-C4-51-RD205	732	7.68	0	154	11.06	19.93	10.5	97.86s
N-1024-C4-51-RD307	722	7.50	0	216	25.85	31.63	10.4	215.16s
N-2025-C4-101-RD0	1842	144.03	0	579	82.33	70.50	64.1	2.88h
N-2025-C4-203-RD0	1574	90.78	0	720	207.87	112.25	57.3	6.06h
N-2025-C4-304-RD0	1325	68.05	0	813	364.64	106.82	51.6	8.97h
N-2025-C4-101-RD101	1721	98.79	0	505	72.82	68.68	57.5	2.16h
N-2025-C4-101-RD203	1631	83.61	0	471	92.75	67.87	51.3	2.3h
N-2025-C4-101-RD304	1574	71.33	0	475	91.46	66.59	45.3	1.89h
N-3025-C4-151-RD454	2241	250.98	0	753	211.70	92.02	102.5	14.7h

are shown in Figs. 1, 2, 3, 4, 5, and 6, and the statistics of the data from the histograms of all test cases are given in Table 1. With these histograms, a designer can quickly determine which (and how many) nodes are susceptible to large voltage drop, and can focus on them for further work. Notice how an increase in the number of C4 sites for the same grid (such as from Fig. 5 to Fig. 6) produces a reduction in the overall voltage drop. A visual display, such as the contour plot in Fig. 7, can also aid in highlighting layout regions that should be of concern.

The computational cost of verification is given in Table 2. The cost is reported in terms of basic iterations (one iteration is one basic/non-basic exchange in the simplex tableau), and in terms of CPU time (per iteration and total). The column labeled "First Solve" gives the cost of solving for the worst-case voltage at the first node. "Successive Solves" refers to the cost of solving subsequent nodes. The first solve is the hardest because the optimizer is starting from an arbitrary feasible point. Successive nodes are easier to solve because they are chosen by traversing small steps on the grid, so that successive solutions require only a small number of iterations (using our technique for objective function switching), as can be seen from the table.

In spite of the efficiency of objective function switching, it emerges from these results that even though the time per node may be small (a mean of up to about 20 seconds for these examples), the time to analyze the whole grid is still large (up to several hours in some cases) and, in practice, one would like to be able to verify larger grids than these. When given a voltage threshold, of course, one does not need to analyze the whole grid, but it remains a fact that one may need to occasionally solve the whole grid. The reason for the computational complexity is that, even though the circuit $\mathbf{G}$ matrix is very sparse, the simplex tableau quickly fills up and soon enough one is working with a near-full tableau. In any case, this technique is the first of its kind to use a static approach based on current constraints, and it is *exact*, thus it can form an accurate benchmark to which other future techniques may be compared.

6. CONCLUSION

Voltage drop on the power grid is a key concern for design of modern integrated circuits. Analysis of the grid by simulation is complicated by the need for stimulus (vectors, patterns, waveforms) to drive a simulator. In practice, obtaining such stimulus is either hard or impossible, especially early in the design process and, in any case, the stimulus would need to be very long to check all interesting cases. We have proposed a pattern-independent *static* approach to power grid voltage verification. By running a sequence of linear programs, the grid is verified under a set of user-supplied *current constraints*. Current constraints represent incomplete information on the circuit behavior and as such are much easier to obtain in practice.

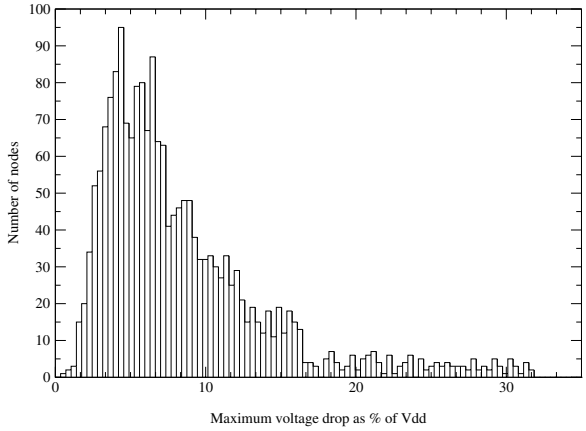

Figure 1: Node voltage drop histogram for grid N-2025-C4-101-RD0.

7. REFERENCES

[1] A. Dharchoudhury et. al. Design and analysis of power distribution networks in PowerPC^{TM} microprocessors. In *35th Design Automation Conference*, San Francisco, 1998.

[2] G. Steele et. al. Full-chip verification methods for DSM power distribution systems. In *35th Design Automation Conference*, San Francisco, 1998.

[3] M. Zhao et. al. Hierarchical analysis of power distribution networks. In *37th Design Automation Conference*, 2000.

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

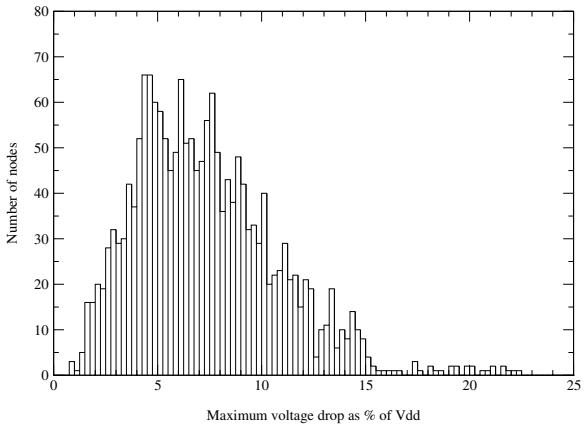

Figure 2: Node voltage drop histogram for grid N-2025-C4-101-RD101.

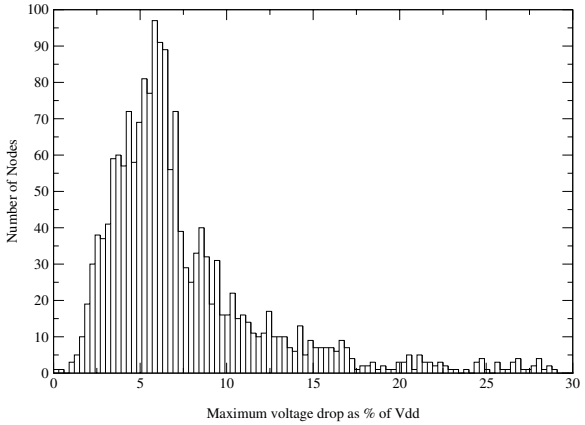

Figure 3: Node voltage drop histogram for grid N-2025-C4-101-RD203.

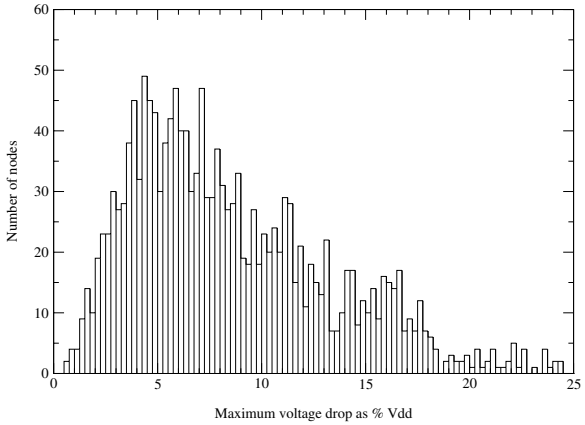

Figure 4: Node voltage drop histogram for grid N-2025-C4-101-RD304.

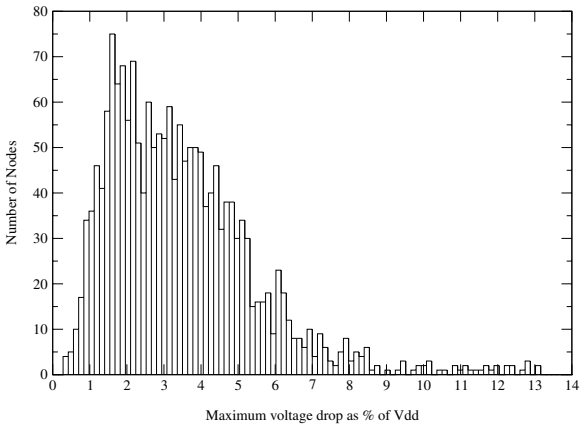

Figure 5: Node voltage drop histogram for grid N-2025-C4-203-RD0.

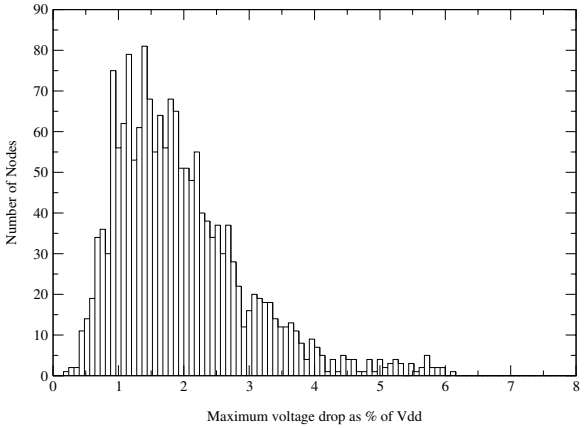

Figure 6: Node voltage drop histogram for grid N-2025-C4-304-RD0.

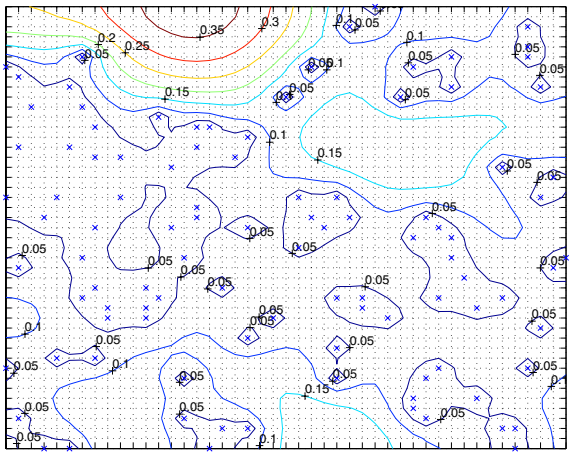

Figure 7: Contour plot of maximum voltage drop for a grid with 2025 nodes.

[5] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 1996.

[6] A. Berman and R. J. Plemmons. *Nonnegative Matrices in the Mathematical Science*. Academic Press, New York, 1996.

[7] H. Kriplani, F. N. Najm, and I. Hajj. Pattern independent maximum current estimation in power and ground buses of CMOS VLSI circuits: algorithms, signal correlations, and their resolution. *IEEE Transactions on Computer-Aided Design*, 14(8):998–1012, August 1995.

[8] Jorge Rubenstein, Paul Penfield, and Mark A. Horowitz. Signal delay in RC tree networks. *IEEE Trans. on Computer-Aided Design*, 2(3):202–211, July 1983.

[9] D. G. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley, Reading, MA, 2nd edition, 1984.