Incremental Power Grid Verification[*]

Abhishek
ECE Department
University of Toronto
Toronto, Ontario, Canada
abhishek@eecg.utoronto.ca

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

ABSTRACT

Verification of the on-die power grid is a key step in the design of complex high-performance integrated circuits. For the very large grids in modern designs, *incremental verification* is highly desirable, because it allows one to skip the verification of a certain section of the grid (internal nodes) and instead, verify only the rest of the grid (external nodes). We propose an efficient approach for incremental verification in the context of *vectorless* constraints-based grid verification, under *dynamic* conditions. The traditional difficulty is that the dynamic case requires iterative analysis of both the internal and external sections. This has been previously overcome for *simulation* purposes, but we provide the first solution for *verification*, through two key contributions: 1) a bound on the internal nodes' voltages is developed that eliminates the need for iterative analysis, and 2) a multi-port Norton approach is used to construct a reduced macromodel for the internal section. As a result, we demonstrate significant reductions in runtime, with speed-ups in the range of 3-8x, with negligible impact on accuracy.

Categories and Subject Descriptors

B.7.2 [**Integrated Circuits**]: Design Aids

General Terms

Performance, Algorithms, Verification

Keywords

Power Grid, voltage drop

1. INTRODUCTION

As supply voltages have decreased with technology scaling, the performance and reliability of modern integrated circuits (IC) have become increasingly susceptible to supply voltage fluctuations. Reduced voltage levels degrade the circuit timing performance and can lead to soft errors. Therefore, voltage integrity verification has become crucial for reliable high-speed chip design.

Most grid verification techniques use some form of circuit simulation to simulate the grid. Simulation-based approaches require complete knowledge of current waveforms drawn by the underlying logic circuitry, which are used to simulate the grid and determine the grid node voltage drops. However, verifying the grid in this way is prohibitively expensive, because the number of current traces required to cover all possible circuit behaviors is extremely large. Another disadvantage is that a simulation-based flow does not allow for early grid verification (when changes to the grid can most easily be incorporated) because no current traces may be available at that time.

To overcome these issues, a *vectorless* verification approach based on partial current specification in the form of *current constraints* was proposed in [1], and further developed in subsequent work over the last decade. Grid verification is reduced to a problem of finding the worst-case voltage drop over all possible currents that satisfy certain current constraints. In [2], the authors used an RC model of the power grid and gave an upper bound on the worst-case voltage drop using an iterative approach. A closed form expression for the upper bound was later proposed in [3], which involved solving a *linear program* (LP) for every grid node. An efficient way to reduce the size of the LPs based on the sparse approximate inverse technique was proposed in [4].

This previous work is useful for verifying the entire power grid but becomes an overkill when verification of only a part of the grid is required, a scenario that we refer to as *incremental verification.* In large modern grids, incremental verification has become desirable because the grids can be so large that verification on traditional workstations becomes impossible, and a divide-and-conquer approach becomes a necessity. Alternatively, incremental verification is desirable when design changes are made to a local region of a previously-verified grid, and the local impact of these changes needs to be verified. There are also various other cases, such as in case of IP reuse, where a portion of the grid may not need to be verified. In [5], a technique is given for incremental verification but only for the case of a resistive grid, under the influence of DC currents. In this paper, we extend incremental verification to the case of transient currents, i.e., a *dynamic verification* context, for the case of RC grids.

In our formulation, the user identifies a part of the grid that does not need to be verified, which we refer to as the *subgrid.* Verification is required only for grid nodes that are outside the subgrid, referred to as *external nodes* (nodes inside the subgrid are referred to as either *internal nodes* or *port nodes*, as we will see). A subgrid can be an arbitrary section of the grid, but must be a connected graph. Strictly speaking, the solution in the dynamic case requires an iterative relaxation-based analysis of both the internal and external grid sections. This difficulty was overcome in [6] for the purpose of circuit simulation (not vectorless verification), through the use of a multi-port Norton theorem. In this work, we provide the first solution in the dynamic case for the purpose of incremental vectorless verification, based on two contributions: 1) upper bounds on the voltage drops at internal nodes are efficiently computed, and used in lieu of the worst-case drops, and 2) a macromodel is constructed for the subgrid based on the movement of internal current sources by adapting the multi-port Norton theorem proposed in [6] to our verification framework, followed by reduction of a passive RC circuit by combining the moment-matching based approach as described in [7] and [8] with the nodal elimination based approach of [9].

The remainder of the paper is organized as follows. In section 2, we present the power grid model and the constraints-based approach to vectorless verification. Section 3 describes our pro-

[*]This work was supported in part by Advanced Micro Devices (AMD) and the Natural Sciences and Engineering Research Council (NSERC) of Canada.

DAC 2012 , June 3-7, 2012, San Francisco, California, USA
Copyright 2012 ACM 978-1-4503-1199-1/12/06 ...$10.00.

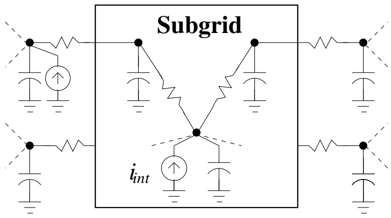


Figure 1: Original Grid

posed incremental verification approach. Implementation details are given in section 4, followed by experimental results in section 5. Finally, the paper is concluded in section 6.

2. BACKGROUND

2.1 The Power Grid Model

Consider an *RC* model of the grid where each branch is represented by a resistor and where there exists a capacitor from every node to ground. Some nodes have ideal current sources (to ground) to represent the current drawn by underlying circuitry, and some have ideal voltage sources to represent the connections to external power supply. Let the power grid consist of $n + p$ nodes, where nodes $1, 2, \ldots, n$ have no voltage sources attached, and the remaining nodes are nodes where the p voltage sources are attached. Let $i(t)$ be the element-wise non-negative vector of all current sources connected to the grid. We assume that $\forall k = 1, \ldots, n$, $i_k(t)$ is well-defined, so that nodes with no current source attached have $i_k(t) = 0$. Let $i(t)$ be the vector of all current sources $i_k(t)$ and $u(t)$ be the vector of nodal voltages. Applying Modified Nodal Analysis (MNA) to the grid leads to:

$$\mathbf{G}u(t) \;+\; \mathbf{C}\dot{u}(t) = -i(t) + \mathbf{G}_0 V_{dd} \tag{1}$$

where $\mathbf{G}$ and $\mathbf{G}_0$ are $n \times n$ conductance matrices, $\mathbf{C}$ is a $n \times n$ diagonal matrix of node capacitances, and V_{dd} is a constant vector each entry of which is equal to the supply voltage value. The matrix $\mathbf{G}$ is known to be diagonally-dominant, symmetric positive-definite, and an $\mathcal{M}$-matrix (so that $\mathbf{G}^{-1} \geq 0$). Let $v(t) = V_{dd} - u(t)$ be the vector of *v*oltage drops. The *RC* model for the power grid can then be written as [2]:

$$\mathbf{G}v(t) + \mathbf{C}\dot{v}(t) = i(t) \tag{2}$$

Note that this equation can be obtained directly by writing the MNA system for a modified network in which all voltage sources are shorted (set to 0) and all current sources are reversed. In the rest of this paper, it will be assumed that we are working with this *modified* topology so that, for example, certain power grid nodes may be connected to ground by a resistor.

In our proposed framework, the nodes inside the subgrid that are connected to external nodes are called port nodes while all the remaining subgrid nodes are referred to as internal nodes. Let n_{ext}, n_{prt}, and n_{int} be the number of external nodes, port nodes, and internal nodes respectively such that $n_{ext} + n_{prt} + n_{int} = n$. Because external nodes connect only to port nodes, the grid equation can now be written as:

$$\begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} & 0 \\ \mathbf{G}_{12}^T & \mathbf{G}_{22} & \mathbf{G}_{23} \\ 0 & \mathbf{G}_{23}^T & \mathbf{G}_{33} \end{bmatrix} \begin{bmatrix} v_{ext}(t) \\ v_{prt}(t) \\ v_{int}(t) \end{bmatrix} + \begin{bmatrix} \mathbf{C}_{ext} & 0 & 0 \\ 0 & \mathbf{C}_{prt} & 0 \\ 0 & 0 & \mathbf{C}_{int} \end{bmatrix} \begin{bmatrix} \dot{v}_{ext}(t) \\ \dot{v}_{prt}(t) \\ \dot{v}_{int}(t) \end{bmatrix} = \begin{bmatrix} i_{ext}(t) \\ i_{prt}(t) \\ i_{int}(t) \end{bmatrix} \tag{3}$$

where v_{ext} and i_{ext} are sub-vectors corresponding to voltage drops and current sources at external nodes, v_{prt} and i_{prt} correspond to voltage drops and current sources at port nodes, and v_{int} and i_{int} correspond to voltage drops and current sources at internal nodes. The matrices $\mathbf{G}$ and $\mathbf{C}$ are partitioned into sub-matrices of appropriate dimensions.

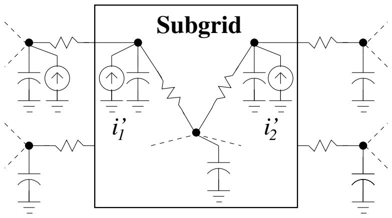


Figure 2: Modified Grid

Using a finite difference approximation as in [2], the system (3) can be written as:

$$\mathbf{A}v(t) = \frac{\mathbf{C}}{\Delta t} v(t - \Delta t) + i(t) \tag{4}$$

where $\mathbf{A} = \left(\mathbf{G} + \frac{\mathbf{C}}{\Delta t}\right)$ is also a symmetric positive-definite $\mathcal{M}$-matrix, so that $\mathbf{A}^{-1} \geq 0$.

2.2 Current Constraints

We perform verification using a vectorless approach in which the information about the currents drawn by underlying circuitry is not known. Instead, current constraints [1] are used, which capture the uncertainty about circuit behaviors and the fact that one is uncertain about the circuit currents early in the design flow. As in previous work, we use two types of constraints: *local constraints* and *global constraints*.

Local constraints are upper bounds on the individual current sources, where one specifies that the current $i_k(t)$ never exceeds a certain fixed level $i_{L,k}$. We assume that every current source tied to the grid has an upper bound associated with it, so that if a node does not have a current source attached, the upper bound for that current is 0. We can express these constraints as:

$$0 \leq i(t) \leq i_L, \quad \forall t \geq 0 \tag{5}$$

If only local constraints are provided, the problem is much simplified but the results become overly pessimistic, because it can never be the case that all the components of the chip are simultaneously drawing maximum currents. Global constraints are upper bounds on the sums of currents for groups of current sources. They represent the peak total power dissipation of a group of circuit blocks. Assuming that we have a total of m global constraints, then we can express them in matrix form as:

$$0 \leq \mathbf{S}i(t) \leq i_G, \quad \forall t \geq 0 \tag{6}$$

where $\mathbf{S}$ is a $m \times n$ matrix that contains only 0s and 1s, which indicate which current sources are present in each global constraint. Together, local and global constraints define a *feasible space* of currents, denoted by $\mathcal{F}$, such that $i(t)$ lies inside the feasible space ($i(t) \in \mathcal{F}$) if and only if it satisfies (5) and (6).

2.3 Power Grid Verification

Given a power grid, we are interested in finding the worst-case voltage drops at all the nodes, under all possible (transient) current waveforms $i(t)$ that satisfy the current constraints. In [3], the authors provide an upper bound v_{ub} on the worst-case voltage drop vector, so that $v(t) \leq v_{ub}, \forall t$, and this bound is given by:

$$v_{ub} = \left(\mathbf{I} + \mathbf{G}^{-1}\frac{\mathbf{C}}{\Delta t}\right) V_a \tag{7}$$

where V_a is the worst-case voltage drop vector at $t = \Delta t$ in the special case when $i(t) = 0, \forall t \leq 0$. Since $v(0) = 0$ in this special case, it follows from (4) that:

$$v(\Delta t) = \mathbf{A}^{-1} i(\Delta t) \tag{8}$$

and V_a can be expressed as:

$$V_a = \underset{\forall i(\Delta t) \in \mathcal{F}}{\text{emax}} \mathbf{A}^{-1} i(\Delta t) \tag{9}$$

where emax is an operator that denotes element-wise maximization of its vector argument, under the given constraints. In other words, V_a is the result of the `for`-loop: `for`$(k = 1, \ldots, n)$\{maximize the k^{th} element of the vector $\mathbf{A}^{-1}i(\Delta t)$, over all $i(\Delta t) \in \mathcal{F}$\}. Maximizing each element becomes a linear program (LP). Note that, because the definition of local and global constraints does not depend on time, then $\mathcal{F}$ is the same at every time point. Therefore, V_a is independent of t and we can drop the Δt argument, and write:

$$V_a = \underset{\forall i \in \mathcal{F}}{\text{emax}}\, \mathbf{A}^{-1} i \tag{10}$$

where, for the purpose of this optimization, i can be viewed as simply a "dummy variable", a $n \times 1$ real vector with units of current.Thus, the problem of finding the worst-case voltage drop is reduced to performing element-wise maximization of $\mathbf{A}^{-1}i$ over all $i \in \mathcal{F}$, to find V_a, followed by a standard linear system solve, to find v_{ub}.

3. PROPOSED APPROACH

In our proposed incremental verification approach, we are interested in the worst-case voltage drops at only the external nodes. As our first contribution in this paper, we propose an efficient way to compute the bounds on the worst-case voltage drops, benefiting from the fact that voltage drops at internal nodes are not required.

3.1 Efficient Bounds Computation

To compute v_{ub}, we need to have an estimate of worst-case voltage drop at internal nodes. From (3), we have:

$$\mathbf{G}_{23}^T v_{prt}(t) + \mathbf{G}_{33} v_{int}(t) + \mathbf{C}_{int} \dot{v}_{int}(t) = i_{int}(t) \tag{11}$$

which, after time-discretization, leads to:

$$v_{int}(\Delta t) = \mathbf{A}_{int}^{-1} i_{int}(\Delta t) - \mathbf{A}_{int}^{-1} \mathbf{G}_{23}^T v_{prt}(\Delta t) \tag{12}$$

where $\mathbf{A}_{int} = \left(\mathbf{G}_{33} + \frac{\mathbf{C}_{int}}{\Delta t}\right)$ is a symmetric positive-definite $\mathcal{M}$-matrix, so that $\mathbf{A}_{int}^{-1} \geq 0$. Because $-\mathbf{G}_{23}^T$ and $\mathbf{A}_{int}^{-1}$ are non-negative matrices, then in the special case used above to define V_a, we can write:

$$\underset{\forall i \in \mathcal{F}}{\text{emax}}(v_{int}(\Delta t)) \leq \mathbf{A}_{int}^{-1} \underset{\forall i \in \mathcal{F}}{\text{emax}}(i_{int}(\Delta t)) + \mathbf{T}^T \underset{\forall i \in \mathcal{F}}{\text{emax}}(v_{prt}(\Delta t)) \tag{13}$$

where $\mathbf{T} = -\mathbf{G}_{23}\mathbf{A}_{int}^{-1}$ is a transformation matrix that will also be useful later. Equation (13) gives an upper-bound on the worst-case voltage drops at $t = \Delta t$ for all internal nodes, so that we can write:

$$V_a = \underset{\forall i \in \mathcal{F}}{\text{emax}}(v(\Delta t)) \leq \begin{bmatrix} \mathbf{I}_{ext} & 0 & 0 \\ 0 & \mathbf{I}_{prt} & 0 \\ 0 & \mathbf{T}^T & \mathbf{A}_{int}^{-1} \end{bmatrix} \underset{\forall i \in \mathcal{F}}{\text{emax}} \begin{bmatrix} v_{ext}(\Delta t) \\ v_{prt}(\Delta t) \\ i_{int}(\Delta t) \end{bmatrix}$$

where $\mathbf{I}_{ext}$ and $\mathbf{I}_{prt}$ are identity matrices of sizes n_{ext} and n_{prt}, respectively. From this, and because $\mathbf{G}^{-1} \geq 0$, we have from (7) that:

$$v_{ub} \leq \left(\mathbf{I} + \mathbf{G}^{-1}\frac{\mathbf{C}}{\Delta t}\right) \begin{bmatrix} \mathbf{I}_{ext} & 0 & 0 \\ 0 & \mathbf{I}_{prt} & 0 \\ 0 & \mathbf{T}^T & \mathbf{A}_{int}^{-1} \end{bmatrix} \underset{\forall i \in \mathcal{F}}{\text{emax}} \begin{bmatrix} v_{ext}(\Delta t) \\ v_{prt}(\Delta t) \\ i_{int}(\Delta t) \end{bmatrix} \tag{14}$$

Because $\text{emax}_{\forall i \in \mathcal{F}}(i_{int}(\Delta t)) = i_L$ (the vector of local constraint values), this gives a faster way to compute an upper bound on v_{ub} which involves solving LPs for external and port nodes only, followed by standard linear solve. Our second contribution is the macromodeling of the internals of the subgrid, as described in the next sub-section.

3.2 Power Grid Macromodeling

Because the internals of the subgrid do not need to be verified, further performance improvement can be obtained by reducing or eliminating much of the subgrid network. Two steps are involved in this: 1) moving the internal current sources to the port nodes, which benefits from multi-port Norton theorem from previous work [6], and 2) reducing the remaining parasitic *RC* network inside the subgrid using model order reduction.

3.2.1 Moving Internal Current Sources

Norton's theorem is a fundamental theorem in circuit theory that converts any linear two-terminal network into a simple parallel circuit consisting of an equivalent current source, and an equivalent internal impedance. The equivalent current source value is the current that will flow through a short circuit between the two terminals [10]. In HiPRIME [6], multi-port Norton equivalent circuits were used to move the current sources internal to a block, to the ports. Previous work benefited from the multi-port Norton theorem for simulation purposes. In our work, we adapt this theorem for use in verification, where the current sources are not known, but are instead subject to current constraints.

Norton Equivalent Current Sources.

Applying the multi-port Norton theorem entails removing the current sources internal to the subgrid and replacing them by new current sources attached to the port nodes. The values of the new port current sources are found by a familiar construction in which 1) the subgrid is disconnected from the rest of the grid, 2) each port node is connected to ground via a short circuit, and 3) the current flowing through these short circuit connections (due to the applied internal current sources) is evaluated.

Consider a subgrid with n_{int} internal nodes and which is connected to the external grid through n_{prt} port nodes. The grid equation of the subgrid when the port nodes are shorted to ground is given by:

$$\mathbf{G}_{33} {v'}_{int}(t) + \mathbf{C}_{int} \dot{v}'_{int}(t) = i_{int}(t) \tag{15}$$

where ${v'}_{int}$ is the $n_{int} \times 1$ voltage drop vector at internal nodes. Let us call an internal node that connects to a port node k, a neighbor of k. The current through a port node to ground ${i'}_k(t)$ is given by:

$${i'}_k(t) = \sum_{\text{neighbors } j \text{ of } k} g_{kj} {v'}_{int_j}(t) \tag{16}$$

where g_{kj} is the conductance through which port node k is connected to internal node j and ${v'}_{int_j}(t)$ is the voltage drop for internal node j. In (3), $\mathbf{G}_{23}$ is the $n_{prt} \times n_{int}$ matrix consisting of all the conductance links from port nodes to internal nodes. Therefore, using (16), the Norton current vector $i'(t)$ can be written as:

$$i'(t) = -\mathbf{G}_{23} {v'}_{int}(t) \tag{17}$$

Modified Grid.

The grid resulting after the internal current sources of the subgrid have been removed and replaced by the new port current sources will be referred to as the *modified grid.* In this modified grid, the voltage drops at the nodes will be denoted by $\hat{v}(t)$, and the system equation becomes:

$$\mathbf{G} \begin{bmatrix} \hat{v}_{ext}(t) \\ \hat{v}_{prt}(t) \\ \hat{v}_{int}(t) \end{bmatrix} + \mathbf{C} \begin{bmatrix} \dot{\hat{v}}_{ext}(t) \\ \dot{\hat{v}}_{prt}(t) \\ \dot{\hat{v}}_{int}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{ext} & 0 & 0 \\ 0 & \mathbf{I}_{prt} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_{ext}(t) \\ i_{prt}(t) \\ i_{int}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{I}_{prt} \\ 0 \end{bmatrix} i'(t)$$

which can also be written as:

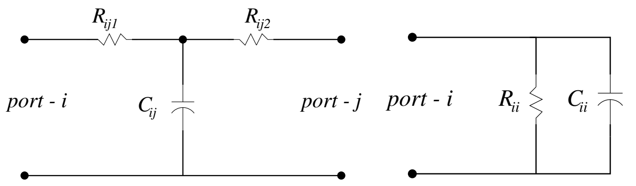
$$\mathbf{G}\hat{v}(t) + \mathbf{C}\dot{\hat{v}}(t) = \mathbf{M}i(t) + \mathbf{N}i'(t) \tag{18}$$

where $\mathbf{M}$ is a $n \times n$ matrix consisting of $\mathbf{I}_{ext}$ and $\mathbf{I}_{prt}$, and $\mathbf{N}$ is a $n \times n_{prt}$ matrix consisting of $\mathbf{I}_{prt}$. Time-discretizing (18) gives:

$$\mathbf{G}\hat{v}(t) + \frac{\mathbf{C}}{\Delta t}\big(\hat{v}(t) - \hat{v}(t - \Delta t)\big) = \mathbf{M}i(t) + \mathbf{N}i'(t) \tag{19}$$

We now return to the special case situation used to define V_a earlier. In that case, the voltage in the modified grid at time $t = \Delta t$ is given by:

$$\hat{v}(\Delta t) = \mathbf{A}^{-1}\left(\mathbf{M}i(\Delta t) + \mathbf{N}i'(\Delta t)\right) \tag{20}$$



(a) T-model (b) Parallel RC model

Figure 3: Macromodels for synthesis

Likewise, time discretizing (15) and evaluating at $t = \Delta t$, we get:

$$v'_{int}(\Delta t) = \mathbf{A}_{int}^{-1} i_{int}(\Delta t) \tag{21}$$

From (17) and (21),

$$\begin{aligned} i'(\Delta t) &= -\mathbf{G}_{23} v'_{int}(\Delta t) \\ &= -\mathbf{G}_{23}\mathbf{A}_{int}^{-1} i_{int}(\Delta t) \equiv \mathbf{T} i_{int}(\Delta t) \end{aligned} \tag{22}$$

which can also be written as:

$$i'(\Delta t) = \begin{bmatrix} 0 & 0 & \mathbf{T} \end{bmatrix} \begin{bmatrix} i_{ext}(\Delta t) \\ i_{prt}(\Delta t) \\ i_{int}(\Delta t) \end{bmatrix} = \mathbf{P} i(\Delta t) \tag{23}$$

where $\mathbf{P}$ is a $n_{prt} \times n$ matrix that contains $\mathbf{T}$. Using (23) in (20), we get:

$$\hat{v}(\Delta t) = \mathbf{A}^{-1}(\mathbf{M} + \mathbf{N}\mathbf{P}) i(\Delta t) = \mathbf{A}^{-1}\hat{\mathbf{M}} i(\Delta t) \tag{24}$$

where:

$$\hat{\mathbf{M}} = \begin{bmatrix} \mathbf{I}_{ext} & 0 & 0 \\ 0 & \mathbf{I}_{prt} & \mathbf{T} \\ 0 & 0 & 0 \end{bmatrix} \tag{25}$$

The above results will be used in the following to efficiently verify the external nodes in the modified grid. From Norton's theorem, the modified grid will exhibit the same voltage response at external and port nodes as the original grid. Therefore, $\forall i \in \mathcal{F}$:

$$\begin{aligned} v_{ext}(\Delta t) &= \hat{v}_{ext}(\Delta t) \\ v_{prt}(\Delta t) &= \hat{v}_{prt}(\Delta t) \end{aligned}$$

so that:

$$\underset{i \in \mathcal{F}}{\text{emax}}(v_{ext}(\Delta t)) = \underset{i \in \mathcal{F}}{\text{emax}}(\hat{v}_{ext}(\Delta t)) \tag{26}$$

$$\underset{i \in \mathcal{F}}{\text{emax}}(v_{prt}(\Delta t)) = \underset{i \in \mathcal{F}}{\text{emax}}(\hat{v}_{prt}(\Delta t)) \tag{27}$$

Therefore, any verification that we will do below on the external nodes in the modified grid will also verify the same nodes in the original grid.

3.2.2 Subgrid Reduction

After moving the internal current sources, we are left with a subgrid consisting only of parasitic *RC* elements. Therefore, we can use a passive Model Order Reduction (MOR) technique to reduce the internals of the subgrid. The reduction approach that we have found applicable and beneficial for this work combines the two standard techniques of moment-matching and node elimination. Because this is a mix-and-match of various content from previous work, it is useful for us to describe some existing techniques, for clarity.

Calculation of Moments.

We use a nodal-formulation based method [8] to compute moments of the system transfer function. The passive subgrid is first isolated from the rest of the grid by removing (i.e., make into an open-circuit) the connections from all port nodes to external nodes. Therefore, the isolated subgrid can be represented in the s-domain by:

$$\left(\begin{bmatrix} \hat{\mathbf{G}}_{22} & \mathbf{G}_{23} \\ \mathbf{G}_{23}^T & \mathbf{G}_{33} \end{bmatrix} + s \begin{bmatrix} \mathbf{C}_{prt} & 0 \\ 0 & \mathbf{C}_{int} \end{bmatrix} \right) \begin{bmatrix} v_{prt}(s) \\ v_{int}(s) \end{bmatrix} = \begin{bmatrix} i_{prt}(s) + i'(s) \\ 0 \end{bmatrix} \tag{28}$$

where $\hat{\mathbf{G}}_{22}$ is an $n_{prt} \times n_{prt}$ conductance matrix that is derived from $\mathbf{G}_{22}$ by removing the connections from port nodes to external nodes represented by $\mathbf{G}_{12}^T$. The admittance looking into the ports of the subgrid, a matrix $\mathbf{Y}(s)$, can be approximated as [11]:

$$\mathbf{Y}(s) \approx \mathbf{M}_0 + \mathbf{M}_1 s \tag{29}$$

where $\mathbf{M}_0 = \hat{\mathbf{G}}_{22} - \mathbf{G}_{23}\mathbf{V}$ and $\mathbf{M}_1 = \mathbf{C}_{prt} + \mathbf{V}^T\mathbf{C}_{int}\mathbf{V}$ are the $n_{prt} \times n_{prt}$, zero and first-order moment matrices with $\mathbf{V} = \mathbf{G}_{33}^{-1}\mathbf{G}_{23}^T$. Because of the quadratic form of $\mathbf{M}_1$, it is clear that it is a non-negative matrix and this will be useful below.

Moment-Matching.

For circuits with a non-negative $\mathbf{M}_1$ matrix, a 2π-model between pairs of ports was constructed in [7] by matching the zero and first-order moments. The circuit between a pair of ports is synthesized using a T-model as shown in Fig. 3(a) and port-to-ground elements are modeled with a parallel *RC* model as shown in Fig. 3(b). The elements of the T-model are given by [7]:

$$\begin{aligned} R_{ij1} &= \frac{-\sqrt{m_1^{jj}}}{m_0^{ij}\left(\sqrt{m_1^{ii}} + \sqrt{m_1^{jj}}\right)} \\ R_{ij2} &= \frac{-\sqrt{m_1^{ii}}}{m_0^{ij}\left(\sqrt{m_1^{ii}} + \sqrt{m_1^{jj}}\right)} \\ C_{ij} &= \frac{\left(\sqrt{m_1^{ii}} + \sqrt{m_1^{jj}}\right)^2 m_1^{ij}}{\sqrt{m_1^{ii} m_1^{jj}}} \end{aligned} \tag{30}$$

where m_0^{ij} and m_1^{ij} are the $(i,j)^{th}$ elements of $\mathbf{M}_0$ and $\mathbf{M}_1$, respectively. To macromodel the port-to-ground connections for port i, the authors [7] provide:

$$R_{ii} = \left(m_0^{ii} + \sum_{\substack{j=1 \\ i \neq j}}^{n_{prt}} m_0^{ij} \right)^{-1}, \quad C_{ii} = m_1^{ii} - \sum_{\substack{j=1 \\ i \neq j}}^{n_{prt}} \frac{C_{ij} R_{ij2}^2}{(R_{ij1} + R_{ij2})^2} \tag{31}$$

Node Elimination.

Note that the above T-model generates an extra node (a new internal node) for each pair of ports. If we have n_{prt} port nodes, generating a T-model for every pair of ports can be expensive because it will result in $n_{prt}(n_{prt} - 1)/2$ new nodes. To overcome this issue, we can eliminate many of the new internal nodes by using the nodal elimination based reduction approach proposed in [9]. For every new internal node, the nodal time constant [9] is given by:

$$\tau = \frac{C_{ij} R_{ij1} R_{ij2}}{R_{ij1} + R_{ij2}} \tag{32}$$

If $\tau < \tau_N$ where τ_N is a user specified nodal time constant value, the internal node can be eliminated by adding capacitors C_i and C_j to port nodes i and j and resistors R_{ij} between i and j. The capacitors and resistors are given by:

$$\begin{aligned} R_{ij} &= R_{ij1} + R_{ij2} \\ C_i = \frac{C_{ij} R_{ij2}}{R_{ij1} + R_{ij2}}&, \quad C_j = \frac{C_{ij} R_{ij1}}{R_{ij1} + R_{ij2}} \end{aligned} \tag{33}$$

Sparsification.

Once the system matrices of the reduced model are formed, it turns out that they contain large numbers of negligible (near zero) entries. As a final step in the reduction, therefore, we have found it useful to apply a final sparsification step, where entries whose absolute value is below a small value κ are automatically set to 0. We have found the error resulting from this to be insignificant and very much worth the effort.

Algorithm 1 INCR_VERIFY

```
Input: Partitioned power grid matrices in (3), $\tau_N$, $\kappa$, $\delta_1$ and $\delta_2$
Output: Upper bounds on worst-case voltage drops for external
    nodes
Construct subgrid matrices in (28)
($\mathbf{T}$, $\tilde{\mathbf{G}}_{sub}$, $\tilde{\mathbf{C}}_{sub}$) = MACRO(subgrid matrices, $\tau_N$, $\kappa$, $\delta_2$)
Construct $\tilde{\mathbf{M}}$, $\tilde{\mathbf{G}}$, $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{A}}$
for ($j = 1, \ldots, n_{ext} + n_{prt}$) do
    Compute $j^{th}$ row of $\tilde{\mathbf{A}}^{-1}$ using SPAI [4] with $\delta = \delta_1$
    Multiply that row by the columns of $\tilde{\mathbf{M}}$, get row vector $\mathbf{d}$
    Maximize: $\mathbf{d} \cdot i$, subject to: $i \in \mathcal{F}$
end for
Compute $\tilde{v}_{ub}$ using (37)
```

Final Reduced Model.

After applying all the reduction techniques discussed above, we end up with new conductance ($\tilde{\mathbf{G}}_{sub}$) and capacitance ($\tilde{\mathbf{C}}_{sub}$) matrices for the isolated subgrid, given by:

$$\tilde{\mathbf{G}}_{sub} = \begin{bmatrix} \tilde{\mathbf{G}}_{22} & \tilde{\mathbf{G}}_{23} \\ \tilde{\mathbf{G}}_{23}^T & \tilde{\mathbf{G}}_{33} \end{bmatrix}; \quad \tilde{\mathbf{C}}_{sub} = \begin{bmatrix} \tilde{\mathbf{C}}_{prt} & 0 \\ 0 & \tilde{\mathbf{C}}_{int} \end{bmatrix} \qquad (34)$$

where $\tilde{\mathbf{G}}_{22}$ and $\tilde{\mathbf{C}}_{prt}$ are the modified port-to-port conductance matrix and capacitance matrix respectively, $\tilde{\mathbf{G}}_{33}$ and $\tilde{\mathbf{C}}_{int}$ are $\tilde{n}_{int} \times \tilde{n}_{int}$ conductance and capacitance matrices for the $\tilde{n}_{int}$ new internal nodes, and $\tilde{\mathbf{G}}_{23}$ and $\tilde{\mathbf{G}}_{23}^T$ are matrices consisting of connections between port nodes and newly formed internal nodes.

3.2.3 Verification after Macromodeling

After macromodeling, the *reduced* grid matrices can be constructed by stitching together the reduced subgrid and external grid matrices using the connections from external nodes to port nodes. The reduced grid matrices are given by:

$$\tilde{\mathbf{G}} = \begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} & 0 \\ \mathbf{G}_{12}^T & \tilde{\mathbf{G}}_{22}' & \tilde{\mathbf{G}}_{23} \\ 0 & \tilde{\mathbf{G}}_{23}^T & \tilde{\mathbf{G}}_{33} \end{bmatrix}, \quad \tilde{\mathbf{C}} = \begin{bmatrix} \mathbf{C}_{ext} & 0 & 0 \\ 0 & \tilde{\mathbf{C}}_{prt} & 0 \\ 0 & 0 & \tilde{\mathbf{C}}_{int} \end{bmatrix} \qquad (35)$$

where $\tilde{\mathbf{G}}_{22}'$ is the updated port-to-port conductance matrix in which the connections from external nodes to port nodes have been added. Since we have used a realizable macromodeling approach, $\tilde{\mathbf{G}}$ is also a $\tilde{n} \times \tilde{n}$ symmetric positive-definite $\mathcal{M}$-matrix, where $\tilde{n} = n_{ext} + n_{prt} + \tilde{n}_{int}$. From (24), the voltage at time $t = \Delta t$ for the modified grid in the special case is given by:

$$\tilde{v}(\Delta t) = \tilde{\mathbf{A}}^{-1}\tilde{\mathbf{M}}i(\Delta t) \qquad (36)$$

where $\tilde{v}$ is a $\tilde{n}$-vector of voltage drops at external, port and new internal nodes, $\tilde{\mathbf{A}} = \left(\tilde{\mathbf{G}} + \frac{\tilde{\mathbf{C}}}{\Delta t}\right)$ is a $\mathcal{M}$-matrix and $\tilde{\mathbf{M}}$ is a $\tilde{n} \times n$ matrix with the first $n_{ext} + n_{prt}$ rows equal to $\hat{\mathbf{M}}$ defined in (25), and the remaining rows have all entries equal to 0. The worst-case voltage drop at external and port nodes can be found by element-wise maximization of $\tilde{v}_{ext}$ and $\tilde{v}_{prt}$. The upper bound on worst-case voltage drops can be computed by using (14) and is given by:

$$\tilde{v}_{ub} = \left(\mathbf{I} + \mathbf{G}^{-1}\frac{\mathbf{C}}{\Delta t}\right) \begin{bmatrix} \mathbf{I}_{ext} & 0 & 0 \\ 0 & \mathbf{I}_{prt} & 0 \\ 0 & \mathbf{T}^T & \mathbf{A}_{int}^{-1} \end{bmatrix} \underset{\forall i \in \mathcal{F}}{\text{emax}} \begin{bmatrix} \tilde{v}_{ext}(\Delta t) \\ \tilde{v}_{prt}(\Delta t) \\ i_{int}(\Delta t) \end{bmatrix} \qquad (37)$$

where $\tilde{v}_{ub}$ is the $\tilde{n}$-vector of upper bounds on worst-case voltage drops with the first n_{ext} entries corresponding to upper bounds for external nodes.

4. IMPLEMENTATION

The overall flow of the proposed incremental verification approach is given in Algorithm 1. We start with a user-specified power grid and subgrid, along with parameter values for τ_N, κ, and error tolerance values (δ_1, δ_2) for the sparse approximate inverse (SPAI [4]) engine. The grid matrices are appropriately partitioned and macromodeling of the subgrid is performed. As a result of macromodeling, the size of the original power grid gets reduced and the internal current sources are moved to port nodes. The inverse of the matrix $\tilde{\mathbf{A}}$ is then computed, and every row is multiplied by $\tilde{\mathbf{M}}$ to account for the effect of movement of internal current sources. Next, we maximize the voltage drop at external and port nodes, and compute the upper bounds on worst-case voltage drops by using the efficient bounds computation approach.

Algorithm 2 MACRO(subgrid matrices, τ_N, κ, δ_2)

```
Output: $\mathbf{T}$ in (13), $\tilde{\mathbf{G}}_{sub}$ and $\tilde{\mathbf{C}}_{sub}$ in (34)
Construct $\mathbf{A}_{int}$
for (every port node $k$) do
    Find neighbors of $k$ and $g_{kj}$ in (16)
    for (every neighbor $j$ of $k$) do
        Compute the $j^{th}$ row of $\mathbf{A}_{int}^{-1}$ using SPAI [4] with $\delta = \delta_2$
        Multiply the row entries by $g_{kj}$
        Add the row entries to the $k^{th}$ row of $\mathbf{T}$
    end for
end for
Compute $\mathbf{M}_0$ and $\mathbf{M}_1$
for (every pair of ports $i$, $j$) do
    Compute $R_{ij1}$, $R_{ij2}$ and $C_{ij}$ using (30)
end for
for (every port node $k$) do
    Compute $R_{kk}$ and $C_{kk}$ using (31)
end for
for (every new internal node created) do
    Compute $\tau$ using (32)
    if $\tau < \tau_N$ then
        Compute $R_{ij}$, $C_i$ and $C_j$ using (33)
        Eliminate the new internal node
    end if
end for
Drop insignificant connections with conductance less than $\kappa$
Construct $\tilde{\mathbf{G}}_{sub}$ and $\tilde{\mathbf{C}}_{sub}$
```

The macromodeling algorithm is presented in Algorithm 2. To avoid the cost of constructing the full matrix $\mathbf{A}_{int}^{-1}$, we generate one row of the transformation matrix $\mathbf{T} = -\mathbf{G}_{23}\mathbf{A}_{int}^{-1}$ at a time, and that is possible through the use of SPAI [4] which is inherently parallelizable and can compute a single column/row of the approximate inverse. For a port node, we first identify the neighbors (k), and the connections (g_{kj}) to the neighbors. Then, we compute the corresponding row of the approximate inverse, multiply the row vector by g_{kj}, and add the result to the k^{th} row of $\mathbf{T}$. Calculation of moments is also efficiently done by factorizing $\mathbf{G}_{33}$, followed by standard system solves.

5. EXPERIMENTAL RESULTS

A `C++` implementation has been written to test the proposed approach. We use SPAI [4] to compute the approximate inverses, and solve the linear programs using MOSEK [12]. The test grids were generated from user specifications, including grid dimensions, metal layers, pitch and width per layer, and supply voltage sites and current sources distribution. The supply voltages and current sources were randomly placed on the grid. The technology specifications were consistent with 1.1 V 65nm CMOS technology. A global constraint is specified for the subgrid and other global constraints were specified to cover the entire chip. The subgrid nodes are also identified by the user. Computations were done using a 2.6 GHz Linux machine with 24 GB of RAM. A SPAI error tolerance value of $\delta_1 = 0.1$mV is used to compute the approximate inverse for the original and modified power grids. A lower value of tolerance $\delta_2 = 0.01$mV is used to construct $\mathbf{T}$.

Table 1 shows the speed and accuracy of the proposed bounds computation approach (section 3.1). Since we are interested in analyzing only the external nodes, we report maximum error and average percentage error values for the upper bounds on worst-

Table 1: Speed and accuracy after using efficient bounds computation

Power Grid		Subgrid		Max Error	Avg. %	CPU time		Speed
Name	Nodes	n_{int}	n_{prt}	(mV)	Error	Original	Fast v_{ub}	Up
G1	8,413	3,891	118	0.08	0.075	22.92 min.	10.58 min.	2.16x
G2	18,678	10,788	176	0.08	0.102	68.19 min.	24.97 min.	2.73x
G3	32,554	15,714	208	0.07	0.057	2.61 h.	1.12 h.	2.33x
G4	50,444	29,458	290	0.07	0.055	4.5 h.	1.77 h.	2.54x
G5	72,692	42,764	348	0.07	0.047	7.71 h.	3.11 h.	2.47x
G6	98,162	68,972	402	0.08	0.079	11.87 h.	3.10 h.	3.82x
G7	128,241	95,294	413	0.08	0.064	18.71 h.	4.19 h.	4.46x
G8	162,087	124,824	518	0.08	0.078	25.30 h.	5.27 h.	4.8x

Table 2: Speed and accuracy after applying macromodeling

Power	Max Error	Avg. %	CPU time		Speed Up	Total Speed Up
Grid	(mV)	Error	Fast v_{ub}	Reduced	Reduced *vs* Fast v_{ub}	Reduced *vs* Original
G1	2.4	1.96	10.58 min.	5.03 min.	2.1x	4.55x
G2	2.06	2.05	24.97 min.	12.26 min.	2.03x	5.56x
G3	1.4	1.20	1.12 h.	0.93 h.	1.2x	2.81x
G4	2.04	1.29	1.77 h.	1.26 h.	1.4x	3.57x
G5	2.01	0.88	3.11 h.	2.36 h.	1.31x	3.26x
G6	2.79	1.14	3.10 h.	2.22 h.	1.39x	5.34x
G7	3.13	1.11	4.19 h.	2.64 h.	1.58x	7.08x
G8	2.6	1.14	5.27 h.	3.14 h.	1.67x	8.05x

case voltage drops at external nodes only. The runtime and accuracy are compared with the original approach based on finding the worst-case voltage drop at every node, and then computing the upper bound using (7). The results show that we are able to achieve significant runtime savings with negligible error values.

The macromodeling approach was tested with user-specified values for nodal time constant ($\tau_N = 5ps$), and conductance threshold ($\kappa = 5 \times 10^{-3}\mho$). Table 2 gives the speed and accuracy obtained after applying macromodeling. The runtime for the reduced grid includes the time taken to perform macromodeling of the subgrid, and then using the efficient bounds computation approach to find the upper bounds. The accuracy is compared with the original approach while speed-up is measured with respect to the efficient bounds computation approach. We also report the total speed-up with respect to the original approach. The results show that we incurred an average error of about 1% for large grids while extracting a total speed-up in the range of 3-8x.

Of course, it is to be expected that, if fewer nodes are to be verified, then corresponding time savings would be the result. However, in our case, the speed-ups are much higher than would be obtained based solely on this argument. For example, if one wants to verify only 15% of the grid nodes, one would expect a speed-up of 100/15=6.67x. However, we can verify 15% of the nodes with a typical total speed-up of 27x. Thus, the benefits of our upper-bounding and macromodeling techniques are quite significant.

6. CONCLUSIONS

We describe an early incremental verification approach for *RC* grids under a constraints-based power grid verification framework, in which only the nodes that are *external* to a subgrid region are to be verified. Our approach gives a fast and accurate way to compute the upper bounds on worst-case voltage drops at external nodes, based on two contributions: 1) an upper-bound method that eliminates the need to perform multiple iterations, and 2) a macromodeling method that drastically reduces the internals of the subgrid. As a result, 3-8x speed-ups are obtained, with negligible 1-2% error. With this proposed approach, it becomes practical to perform early incremental design verification of the on-die power grid under dynamic conditions.

7. REFERENCES

[1] D. Kouroussis and F. N. Najm. A static pattern-independent technique for power grid voltage integrity verification. In *ACM/IEEE DAC*, pages 99–104, Anaheim, CA, Jun. 2-6 2003.

[2] M. Nizam, F. N. Najm, and A. Devgan. Power grid voltage integrity verification. In *ACM/IEEE ISLPED*, pages 239–244, San Diego, CA, Aug. 8-10 2005.

[3] I. A. Ferzli, F. N. Najm, and L. Kruze. A geometric approach for early power grid verification using current constraints. In *ACM/IEEE ICCAD*, pages 40–47, San Jose, CA, Nov. 5-8 2007.

[4] N. H. Abdul Ghani and F. N. Najm. Fast vectorless power grid verification using an approximate inverse technique. In *ACM/IEEE DAC*, San Fransisco, CA, Jul. 26-31 2009.

[5] D. Kouroussis, I. A. Ferzli, and F. N. Najm. Incremental partitioning-based vectorless power grid verification. In *ACM/IEEE International Conference on Computer Aided Design*, pages 358–364, San Jose, CA, November 6-10 2005.

[6] Y.-M. Lee, Y. Cao, T.-H. Chen, J.M. Wang, and C.C.-P. Chen. HiPRIME: hierarchical and passivity preserved interconnect macromodeling engine for RLKC power delivery. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 24(6):797–806, June 2005.

[7] Haifang Liao and W. Wei-Ming Dai. Partitioning and reduction of RC interconnect networks based on scattering parameter macromodels. In *Digest of Technical Papers, ACM/IEEE International Conference on Computer-Aided Design*, pages 704–709, Nov 1995.

[8] P. Miettinen, M. Honkala, J. Roos, C. Neff, and A. Basermann. Study and development of an efficient RC-in-RC-out MOR method. In *IEEE International Conference on Electronics, Circuits and Systems*, pages 1277–1280, Aug 2008.

[9] B. Sheehan. Realizable reduction of RC networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26(8):1393–1407, August 2007.

[10] Samuel L. Oppenheimer, Jean Paul Borchers, and F. Roger Hess. *Direct and Alternating Currents.* McGraw-Hill, New York, 1973.

[11] K.J. Kerns and A.T. Yang. Stable and efficient reduction of large, multiport rc networks by pole analysis via congruence transformations. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 16(7):734–744, 1997.

[12] The MOSEK optimization software (www.mosek.com).