Accurate Power Grid Analysis with Behavioral Transistor Network Modeling

Anand Ramalingam
anandram@cerc.utexas.edu

Giri V. Devarayanadurg
giri.venkata@intel.com

David Z. Pan
dpan@cerc.utexas.edu

Department of Electrical and Computer Engineering, The University of Texas, Austin, TX 78712
Intel Research, Hillsboro, OR 97124

ABSTRACT

In this paper, we propose fast and efficient techniques to analyze the power grid with accurate modeling of the transistor network. The solution techniques currently available for power grid analysis rely on a model of representing the transistor network as a current source. The disadvantage of the above model is that the drain capacitance of the PMOS transistors which are already *on* is not modeled. The drain capacitance of the PMOS transistors which are *on*, act much like a decoupling capacitance in the power grid. By ignoring the drain capacitance, the voltage drop predicted is pessimistic. This implies that a designer is likely to overestimate the amount of decoupling capacitance needed. In our proposed model, we model the transistor network as a simple switch in series with a RC circuit. The presence of switches leads to a non-constant conductance matrix. So, the switch is modeled behaviorally to make the conductance matrix a constant in presence of switches. The resulting conductance matrix is a $\mathcal{M}$-matrix thus making it amenable to linear algebraic methods presented in the literature. The proposed model is nearly as accurate as the SPICE model in predicting the voltage drop. We demonstrate that the current source model of the transistor network has an error of about 10% in predicting the voltage drop. The proposed model offers the middle ground between the accuracy of SPICE simulation and the speed of the current source model. The proposed model is $20 - 30\times$ faster than SPICE. It also reduces the size of the decoupling capacitance by $2 - 10\times$ in comparison with the methods presented in the literature.

Categories and Subject Descriptors

B.7.2 [**Design Aids**]: Simulation

General Terms

Algorithms, Performance

Keywords

Behavioral modeling of switch, Power grid, RC model of transistor

1. INTRODUCTION

Power grid related activities such as design, modeling, analysis and verification has been a scene of intense research activity in recent times. In this paper, we focus on power grid *analysis*. The solution techniques currently available for power grid analysis rely on a model of representing the transistor network as a current source [1]. This simplification enables decoupling the transistor network from the power grid. The most significant advantage of this simplification is that the power grid problem mathematically reduces to solving a linear system of equations. Thus we can apply techniques from numerical linear algebra to solve the linear system of equations arising from the power grid [2–9]. Hierarchical analysis has also been used to analyze power grid [10, 11]. In hierarchical analysis, the power grid is partitioned and a macromodel is created for each partition. The macromodel makes the problem of analyzing large power grids tractable. Since the deterministic techniques mentioned above solve the entire system, they are not suitable for incremental analysis. The need for incremental analysis gave rise to stochastic techniques such as random walk [12–14]. Stochastic techniques have also been applied to study the effect of process variations on the power grid [15–18]. When a transistor switches on and connects to the grid, the charge that is supplied to the transistor comes from the capacitors nearby. This *locality* effect has been exploited to design fast algorithms [19, 20]. Methods have also been proposed for power grid analysis in the context of floorplanning [21, 22].

In the literature reviewed so far, the nonlinear transistor network is modeled as a current source which results in a linear system of equations. But this modeling might lead to pessimism in the voltage drop prediction. In Figure 1, we illustrate a voltage drop at a node in a power grid. In the current source model, we replace the inverter chains with time-varying current sources which model the switching current drawn from the grid. The current source model does not take into account the decoupling capacitances provided by the PMOS transistors which are already on and currently not switching. This leads to pessimism in the voltage drop predicted. There is a difference of $0.025V$ in the voltage drop predicted. As we will see later in the results section, a pessimistic prediction will lead to a bigger decoupling capacitance.

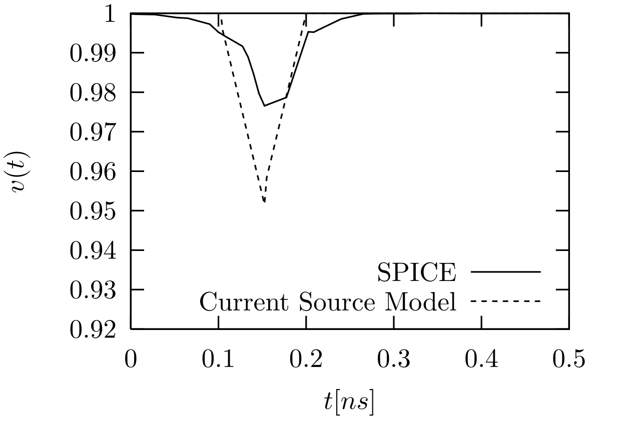

Figure 1: The voltage drop at a node from SPICE simulation and the current source based approach. There is a difference of $0.025V$ in the voltage drop predicted.

We summarize the disadvantages in modeling the transistor network as a current source:

1. When a PMOS transistor switches and connects to the power grid, some of the charge is supplied by the PMOS transistors that are already *on*. These PMOS transistors which are already *on*, act much like a 'decoupling capacitance' in the power grid. By ignoring this *local charge sharing* effect, the designer is likely to overestimate the amount of decoupling capacitance needed. This leads to wastage of power and silicon area. The *local charge sharing* effect is not captured in the current source model.

2. The number of transistors that get switched *on* differs from cycle to cycle. This implies the amount of capacitance seen by the power grid also varies from cycle of cycle. The *time-varying capacitance* is not captured in the current source model.

The *modeling of transistor network* with respect to the power grid has received little attention. In this paper, *we focus on the modeling aspect* of the transistor network which results in an accurate power grid analysis.

We summarize the contributions of this paper:

- We analyze the power grid by modeling the transistor network accurately instead of replacing the transistor network with a time-varying current source.

- The transistor is modeled as a simple switch in series with a *RC* circuit. The switch is modeled *behaviorally* as a Norton current source model. The behavioral modeling of the switch is the key in making the proposed simulation efficient. Note that modeling the switch as a PWL resistor leads to convergence problems associated with abrupt non-linearities [23].

- The proposed model offers the middle ground between the accuracy of SPICE simulation and the speed of the current source model.

It should be noted that we have adapted techniques from the literature but the overall flow is original. The rest of the paper is organized as follows. The power grid modeling is reviewed in Section 2 and the proposed transistor network modeling is described in Section 3. The behavioral modeling of switch and its role in making the conductance matrix a constant in presence of switches is described in detail in Section 4. Simple speedup techniques are described in Section 5 and the overall flow is described in Section 6. The experimental results are discussed in Section 7 and we conclude in Section 8.

2. POWER GRID PRELIMINARIES

We adapt the power grid modeling described in [1] where the power grid is modeled as a passive Linear Time Invariant (LTI) network consisting of resistors, inductors, and capacitors. Since the ground grid is symmetrical we restrict our analysis to the power grid alone [24]. The power grid can be described using the Modified Nodal Analysis (MNA) formulation [25]:

$$\mathbf{G}V(t) + \mathbf{C}\frac{dV(t)}{dt} = I(t) \tag{1}$$

where

- $\mathbf{G} \in \mathbb{R}^{m\times m}$ is the conductance matrix which depends on the topology of the circuit. Also, m denotes the number of nodes in the power grid and the transistor network.

- $\mathbf{C} \in \mathbb{R}^{m\times m}$ is the admittance matrix resulting from capacitive and inductive elements.

- $V(t) \in \mathbb{R}^m$ is a time-varying vector of voltages at the nodes.

- $I(t) \in \mathbb{R}^m$ has two kinds of rows [4]:

  1. Rows with positive V_{DD} value corresponding to the nodes connected to voltage sources;

  2. Rows with 0, correspond to all other nodes.

Since we restrict our attention to the voltages, we ignore the KCL equations around the voltage sources. If all the voltage sources are grounded, this results in the conductance matrix $\mathbf{G}$ which is positive definite and it can be shown to be a $\mathcal{M}$-matrix [4]. This gives rise to efficient methods for solving the linear system [26, 27].

The ordinary differential equation in Eq. (1) can be solved in time domain using Backward Euler technique [25]. We use the Norton current source as the associated discrete circuit (ADC) model for both capacitor and inductor. This is because Thevenin's voltage source is not suited for MNA [25] since for every voltage source we need an extra row in the conductance matrix.

On applying ADC to the energy-storage elements, we get:

$$\left(\mathbf{G} + \frac{\mathbf{C}}{h}\right)V(t+h) = I(t+h) + \frac{\mathbf{C}}{h}V(t) \tag{2}$$

where h is the time-step taken in the transient simulation.

If we fix the time-step h to be a constant, then the matrix $\left(\mathbf{G} + \frac{\mathbf{C}}{h}\right)$ turns out to be a constant for the entire duration of simulation. This leads to greater efficiency in the transient solve since we need just one LU factorization of $\left(\mathbf{G} + \frac{\mathbf{C}}{h}\right)$ for the entire transient simulation and the cost can be amortized over many runs of the transient simulation. The nodal voltages at each time point in the transient is got by a Forward-Backward Solve (FBS) which is $O(m^2)$ compared to $O(m^3)$ for a direct solve [28].

3. TRANSISTOR NETWORK MODELING

In this section, we describe our transistor network modeling. We differ from the literature by modeling the transistor not as time-varying current source but as a simple *RC* circuit [29]. The transistor is connected to the power grid through a switch as shown in Figure 2.

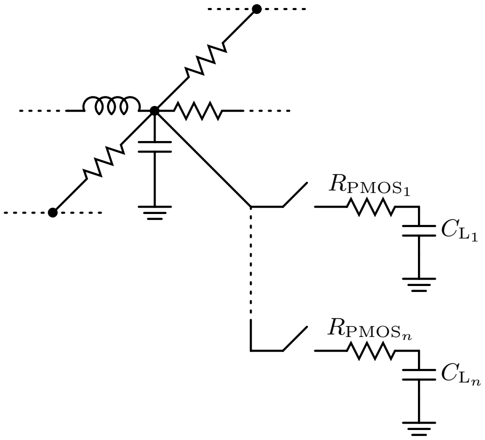

Figure 2: Transistor is modeled a simple switch in series with a *RC* circuit. Note that if a transistor gets switched on to the grid node, some of the charge will come from transistors which are already *on* which is not captured in previous models.

The advantage of this modeling is that it can accurately capture the self-loading effects of the transistor and the charge-sharing among the switching transistors. But the major disadvantage of this modeling is that based on whether the switch is on or off, the topology changes, leading to a different $\left(\mathbf{G} + \frac{\mathbf{C}}{h}\right)$ matrix during every switching instant. This makes the conductance matrix non-constant and it will make the transient simulation inefficient. To make the matters worse, if we have k transistors modeled as switches then potentially we have 2^k different topologies [30]. This fact is illustrated by building conductance matrices for the circuits in Figure 3 and Figure 4 which differ only in the state of the switch.

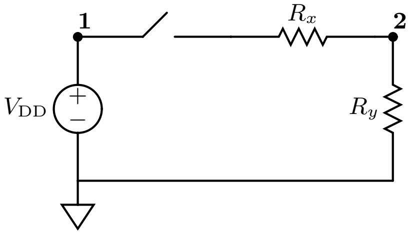

Figure 3: Network having an open switch. The topology due to an open switch is different when compared to a closed switch in Figure 4. This leads to the conductance matrix $\mathbf{G}_{\text{open}}$.

The MNA matrix equations corresponding to the circuit in Figure 3 containing an open switch is given by

$$\begin{array}{c} \\ \mathbf{1} \\ \mathbf{2} \end{array} \begin{array}{c} \begin{array}{cc} \mathbf{1} & \mathbf{2} \end{array} \\ \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{R_y} \end{pmatrix} \end{array} \begin{pmatrix} V_1 \\ V_2 \end{pmatrix} = \begin{pmatrix} V_{\text{DD}} \\ 0 \end{pmatrix}$$

$$\mathbf{G}_{\text{open}} V = I \tag{3}$$

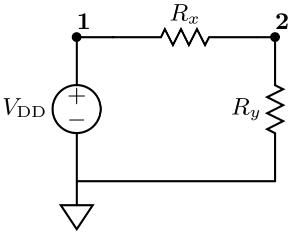

Figure 4: Networks having a closed switch. The topology due to a closed switch is different when compared to an open switch in Figure 3. This leads to the conductance matrix $\mathbf{G}_{\text{close}}$.

The MNA matrix equations corresponding to the circuit in Figure 4 containing a closed switch is given by

$$\begin{array}{c} \\ \mathbf{1} \\ \mathbf{2} \end{array} \begin{array}{c} \begin{array}{cc} \mathbf{1} & \mathbf{2} \end{array} \\ \begin{pmatrix} 1 & 0 \\ -\frac{1}{R_x} & \frac{1}{R_x} + \frac{1}{R_y} \end{pmatrix} \end{array} \begin{pmatrix} V_1 \\ V_2 \end{pmatrix} = \begin{pmatrix} V_{\text{DD}} \\ 0 \end{pmatrix}$$

$$\mathbf{G}_{\text{close}} V = I \tag{4}$$

Note that we drop the KCL equations around the voltage node **1** since we are interested only in the voltage at any given node. It is clear from Eq. (3) and Eq. (4), depending on whether the switch is closed or open we get a different conductance matrix **G**.

In the next section, we describe a technique to keep the conductance matrix $\left(\mathbf{G} + \frac{\mathbf{C}}{h}\right)$ in the MNA formulation in Eq. (2) a *constant*, irrespective of the state of the switches and thus get back the efficiency achieved by having a constant conductance matrix.

4. BEHAVIORAL SWITCH MODELING

In this section, we describe a discrete-time approximation to the ideal switch which models the switch behaviorally. This renders the conductance matrix $\left(\mathbf{G} + \frac{\mathbf{C}}{h}\right)$ in the MNA formulation in Eq. (2) a *constant*, irrespective of the state of the switches.

4.1 Ideal switch

The ideal switch shown in Figure 5 has zero resistance when *on* and infinite resistance when *off*. Also the ideal switch can move from one to state to another instantaneously. This change in resistance causes the change in topology and hence we get different conductance matrices.

But from the view of simulation, the behavior of the ideal switch (s) can be captured by the following equations:

$$s = \text{open} \Leftrightarrow i_s = 0 \tag{5}$$

$$s = \text{short} \Leftrightarrow v_s = 0 \tag{6}$$

This *behavioral modeling of switch* is a key in achieving efficiency in power grid simulation.

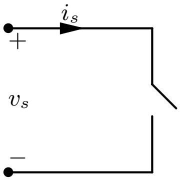

Figure 5: Ideal switch.

4.2 ADC for an approximate switch model

The industry standard circuit simulators like SPICE use a two-valued resistor for modeling the transistor switch. But it leads to convergence problems and long execution times for stiff networks [31]. For an efficient simulation with switches, we just need to model the behavior of the switch as captured in Eq. (5) and Eq. (6).

We now describe Approximate Discrete Circuit (ADC) for an approximate model of switch. This was independently developed by Hui and Morrall [32], and by Pejović and Maksimović [33] in 1994, motivated by the switching power system simulations.

Before describing the ADC formally, it would be instructive to get an intuition of the idea which makes the conductance matrix a constant irrespective of the state of the switches. To illustrate, consider the circuits in Figure 3 and Figure 4. We would like to have the same conductance matrix for both the circuits since they differ only in the state of the switch.

Let the switch be modeled by a voltage source (v_s) in series with a finite resistance (r_s) as shown in Figure 6.

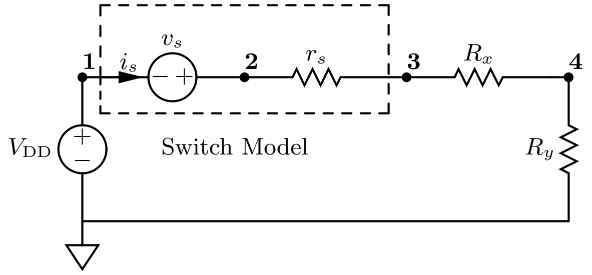

Figure 6: Modeling a switch with a voltage source. By varying the value of the voltage source, we can simulate the *on* or *off* behavior of the switch. Since we are just changing the value of the voltage source, the conductance matrix remains the same irrespective of the state of the switch.

Applying KVL to the circuit in Figure 6, we get,

$$-V_{\mathrm{DD}} - v_s + i_s(r_s + R_x + R_y) = 0 \tag{7}$$

To simulate the switch being *open* ($s = 0$) we need $i_s = 0$ as in Eq. (5). This can be easily achieved by setting $v_s = -V_{\mathrm{DD}}$. Similarly, to simulate the switch being *short* ($s = 1$), we set $v_s = 0$. Thus by changing the value of the voltage source we can simulate the behavior of a switch without changing the topology of the circuit. This is the intuition behind the switch modeling.

The above intuition is formalized by writing out the MNA equations for the circuit in Figure 6.

$$\begin{array}{c c} & \begin{array}{cccc} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} \end{array} \\ \begin{array}{c} \mathbf{1} \\ \mathbf{2} \\ \mathbf{3} \\ \mathbf{4} \end{array} & \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & -\frac{1}{r_s} & \frac{1}{r_s} + \frac{1}{R_x} & -\frac{1}{R_x} \\ 0 & 0 & -\frac{1}{R_x} & \frac{1}{R_x} + \frac{1}{R_y} \end{pmatrix} \end{array} \begin{pmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{pmatrix} = \begin{pmatrix} V_{\mathrm{DD}} \\ v_s \\ 0 \\ 0 \end{pmatrix} \tag{8}$$

By setting v_s to 0 or $-V_{\mathrm{DD}}$ in the RHS of Eq. (8) we can simulate the switch being on or off. Note that the conductance matrix on LHS remains a constant irrespective of the state of the switches.

4.3 Current source based ADC for switch

Since MNA lends itself more naturally to a current source compared to a voltage source [25], we use a current source instead of a voltage source to simulate the state of a switch. The ADC for an approximate switch model is shown in Figure 7. The ADC of an approximate switch can be thought of as a linearized, discrete equivalent circuit of a nonlinear resistor [33].

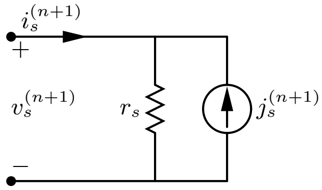

Figure 7: Associated Discrete Circuit (ADC) of an approximate switch model. The superscript $(n+1)$ refers to the simulation step.

The state of the switch is captured by changing the value of the current source $j_s^{(n+1)}$. This is similar to changing the value of voltage captured in Eq. (7). In our simulations, we modeled the switch as a current source as shown in Figure 7. The behavioral model of the switch is given by [33]:

$$j_s^{(n+1)} = \begin{cases} -i_s^{(n)} & \text{if } s^{(n+1)} = 0, \\ \frac{v_s^{(n)}}{r_s} & \text{if } s^{(n+1)} = 1. \end{cases} \tag{9}$$

Note that we use $\frac{v_s^{(n)}}{r_s}$ when $s^{(n+1)} = 1$ instead of 0 as we had discussed in the example above. But when the switch is closed there is almost no voltage drop across the switch. Hence $v_s^{(n)} \approx 0$ and thus consistent with our intuition. We adopt this notation just to be consistent with the power electronics literature. The *efficiency of our proposed algorithm* compared to SPICE is down to this behavioral modeling of the switch. We show that the algorithms previously presented in the literature can be applied to our new model without much change. This is done by proving that the resultant conductance matrix $\mathbf{G}$ is a $\mathcal{M}$-matrix.

4.4 Conductance matrix is a $\mathcal{M}$-matrix

Definition 4.1 ($\mathcal{M}$-matrix). A matrix $\mathbf{A} \in \mathbb{R}^{m \times m}$ is an $\mathcal{M}$-matrix if it satisfies the following conditions: (1) $a_{ii} > 0 \quad \forall i$; (2) $a_{ij} \leq 0 \quad \forall i \neq j$; (3) $a_{ii} \geq \sum_{j \neq i} |a_{ij}| \quad \forall i$; (4) $a_{ii} > \sum_{j \neq i} |a_{ij}|$ for at least one i;

Theorem 1. *The conductance matrix* $\mathbf{A} = (\mathbf{G} + \frac{\mathbf{C}}{h})$ *obtained by modeling switch by its ADC is an* $\mathcal{M}$*-matrix.*

Proof. We have to show that the conductance matrix $\mathbf{A}$ satisfies all the 4 conditions in Definition 4.1.

The first condition is $a_{ii} > 0 \quad \forall i$. This is satisfied due to the MNA stamping [25]. The diagonal entries of $\mathbf{A}$ is given by

$$a_{ii} = \sum_{j \in \text{nodes}} g_{ij} + \frac{c_i}{h} \tag{10}$$

By definition, a node is a junction of at least two elements. This implies the RHS of Eq. (10) has at least two distinct entries. Since the conductances and capacitances are positive values, we can conclude that the first condition $a_{ii} > 0 \quad \forall i$ holds for our conductance matrix $\mathbf{A}$.

The second condition is $a_{ij} \leq 0 \quad \forall i \neq j$. This is again satisfied due to the MNA stamping or in other words it is correct by construction. The only exception comes when there is a voltage source connected to the node. KCL needs to account for the current through the voltage source (i_{v_s}). In MNA stamping, it turns out that the coefficient of i_{v_s} is $+1$ causing the second condition to fail. Since we are interested only in the voltages at a given node, this situation is avoided by ignoring the KCL around the node connected to the voltage source.

The third condition is $a_{ii} \geq \sum_{j \neq i} |a_{ij}| \quad \forall i$. This follows from Eq. (10).

$$\begin{aligned} a_{ii} &= \sum_{j \in \text{nodes}} g_{ij} + \frac{c_i}{h} \\ &= \sum_{j \in \text{nodes}} |g_{ij}| + \frac{c_i}{h} \quad (\text{Since } g_{ij} \text{ is positive, } g_{ij} = |g_{ij}|) \\ &\geq \sum_{j \in \text{nodes}} |g_{ij}| \qquad \geq \sum_{j \neq i} |g_{ij}| \\ &= \sum_{j \neq i} |a_{ij}| \qquad (\text{Since } a_{ij} = g_{ij} \quad \forall i \neq j) \end{aligned} \tag{11}$$

The fourth condition is $a_{ii} > \sum_{j \neq i} |a_{ij}|$ for at least one i. We need at least one voltage source for our power grid. In our MNA construction, the KCL equations around the voltage source node are dropped. We can state the MNA construction corresponding to voltage source mathematically as follows. If there is a voltage source at node i then $a_{ii} = 1, \quad a_{ij} = 0 \quad \forall i \neq j$. The RHS corresponding to this row is set to V_{DD}. Thus the fourth condition directly follows from this construction. □

It should be noted that it is straightforward to make $\mathbf{A}$ a symmetric matrix. The matrix is not symmetric due to the fact that we drop the KCL around the voltage source node. The technique to make $\mathbf{A}$ symmetric is illustrated by an example [4]. Consider the circuit shown in Figure 4. The MNA matrix equations corresponding to the circuit in Figure 4 is reproduced below for convenience.

$$\begin{array}{cc} & \begin{array}{cc} \mathbf{1} & \qquad \mathbf{2} \end{array} \\ \begin{array}{c} \mathbf{1} \\ \mathbf{2} \end{array} & \begin{pmatrix} 1 & 0 \\ -\frac{1}{R_x} & \frac{1}{R_x} + \frac{1}{R_y} \end{pmatrix} \end{array} \begin{pmatrix} V_1 \\ V_2 \end{pmatrix} = \begin{pmatrix} V_{\mathrm{DD}} \\ 0 \end{pmatrix}$$

Since $V_1 = V_{\mathrm{DD}}$, we can rewrite our system of equations as:

$$\begin{array}{cc} & \begin{array}{cc} \mathbf{1} & \qquad \mathbf{2} \end{array} \\ \begin{array}{c} \mathbf{1} \\ \mathbf{2} \end{array} & \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{R_x} + \frac{1}{R_y} \end{pmatrix} \end{array} \begin{pmatrix} V_1 \\ V_2 \end{pmatrix} = \begin{pmatrix} V_{\mathrm{DD}} \\ \frac{V_{\mathrm{DD}}}{R_x} \end{pmatrix}$$

Now the conductance matrix turns out to be a symmetric matrix. Note that the conductance matrix is still an $\mathcal{M}$-matrix even after its transformation to a symmetric matrix.

5. SPEEDUP TECHNIQUES

In this section, we discuss speedup techniques to improve the runtime of the proposed modeling. To get an intuition behind these techniques, it is instructive to look into the phases that occur in power grid simulation. There are two phases which occur in power grid simulation and it shown in Figure 8 which is a stylistic depiction of the phases. Please note that the graph is not drawn to scale.

1. *Local charge redistribution.* This happens when the PMOS transistor gets switched on to the power grid. Most of the charge is supplied by the local capacitors.
2. *Global recovery phase.* This happens when the power supply starts supplying charge to the capacitors. It brings back all the capacitors to their original state of being fully charged.

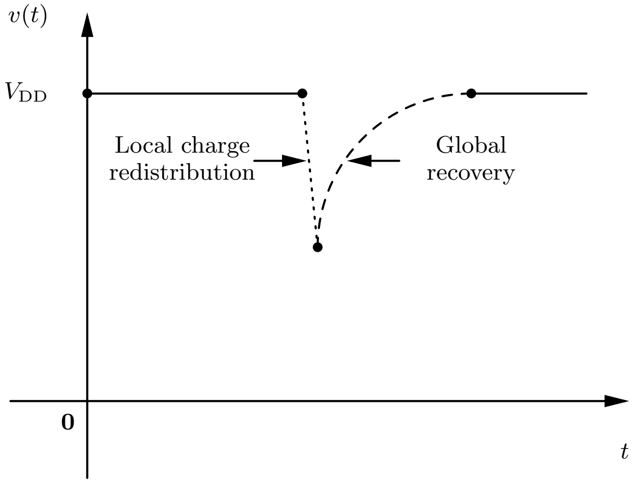

Figure 8: The figure is *not* drawn to scale. This is a stylistic depiction of the 2 phases that occur in power grid simulation. There are two phases, the first phase (dotted) is local charge redistribution and the second phase (dashed) is global recovery.

5.1 Local charge redistribution phase

The major drawback of this proposed modeling is that the time-step (h) in Eq. (2) is decided by the fast transients during the switching of transistors which cause a voltage drop in the grid near instantaneously (*local charge redistribution*). Since we need to track these transients accurately, the time-step (h) is set to tens of picoseconds. When the power grid recovers back after this fast transient voltage drop, we can track the voltages well with time-step in hundreds of picoseconds. Thus if we can calculate this voltage drop using a fast approximation without doing a detailed simulation then we can use a bigger time-step in rest of the simulation. This implies a faster runtime for the proposed power grid modeling.

The researchers in power electronics community have addressed this issue of charge redistribution [23]. But they assume instantaneous charge redistribution by ignoring the resistances which is not a good approximation in the power grid problem. So we need an approximation which also takes into account the resistances of the power grid.

The idea behind the approximate method to calculate the voltage drop is that when a transistor switches on at a grid, most of the charge comes from nearby capacitors (*local charge redistribution*) and the recovery of the nearby capacitors to their original fully charged state is due to the charge being supplied from the nearby voltage sources (*global recovery phase*).

The voltage drop is due to *local charge redistribution.* The idea of *locality* is exploited to devise a fast method to calculate the drop. It has been shown in that the using first two shortest paths (in terms of impedance) from the node to the voltage source to calculate the drop gives an approximation within 10% of the SPICE results [34]. Thus having more shortest paths will lead to a better approximation of the voltage drop.

But the drawback of this method is that it works only if the switching events are isolated. Though we do not define *isolated switching event* formally in this paper, it can be stated informally as follows. A switching event is said to be isolated if there are no switching events in its locality. Since our proposed modeling results in a non-linear system, we cannot use superposition when switching events occur simultaneously in nearby power grid nodes. We defer the voltage drop calculation during simultaneous switching to future work.

5.2 Global recovery phase

A simplifying assumption in scheduling gate switching helps speedup the global recovery phase. We assume that all gates that can switch in a given cycle, switch during the positive edge of the cycle. For example, consider an inverter chain (inv1 – inv4 – inv16) hooked to the same power grid node. If inv1 is switched on during a cycle, inv16 also gets switched on but after some finite delay. But in our model, we switch on both inv1 and inv16 at the same time. This makes our voltage drop to be near the positive edge of the clock. Since our goal is to observe the dynamics at power grid nodes over several cycles, we are less concerned at the exact time at which the voltage drop occurs.

Thus in *global recovery phase*, observe that once grid recovers back to the supply voltage in a given cycle it is going to stay at the supply voltage. This is because all the gates that were scheduled to be switched during a given cycle gets switched during the start of the cycle. Thus the power grid simulation can be fast-forwarded to the start of the next cycle once all the nodes recover back to the supply voltage.

6. OVERALL ALGORITHM

The proposed power grid simulation with the behavioral modeling of transistor is given in Algorithm 1.

Algorithm 1 Power-Grid-Simulation

```
Input: Transistor representation of the blocks
Input: Input switching patterns at the primary inputs of
    the block
Input: RC representation of the power grid
Input: $h$ = time-step, set by the fastest transient in local
    charge redistribution
Output: Power grid voltage at various user specified points
    on the power grid
// (Section 3)
Model transistor as a switch connected to a RC circuit.
// (Section 4)
Model switch by its ADC in Eq. (9).
Generate the conductance matrix for the power grid
    model as in Eq. (2).
while time < time-stop do
    if positive edge of clock then
        // Conductance matrix in LHS of Eq. (2) is
        // constant irrespective of the state of switches
        Update RHS in Eq. (2) based on the state of
            switches. Please note that the state of switch de-
            notes whether the PMOS transistor is on or off.
        // Speedup technique for local charge redistribution
        // (Section 5.1)
        if switching events are isolated then
            Use approximate methods to calculate the volt-
                age drop due to the local charge redistribution.
            Update time.
        end if
    end if
    // use any efficient $\mathcal{M}$-matrix solver
    Solve for unknown voltages in Eq. (2).
    Update ADC of energy-storage elements.
    time = time + $h$.
    // Speedup technique for global recovery phase
    // (Section 5.2)
    if all the nodes have recovered to supply voltage then
        Fast forward time to the start of next cycle.
    end if
end while
```

7. EXPERIMENTAL RESULTS

We implemented our model and algorithm using a combination of awk/perl/matlab scripts since our main purpose is to demonstrate the concept. Hence if implemented in C/C++ with well tuned data structure/code, the speedup compared to SPICE is expected to at least 2 orders of magnitude. We report the results of experiments run on random benchmarks [35]. The experiments were done using a 32-bit Linux machine with 4 GB RAM and running at 3.4 GHz. The delay models were generated using the 90*nm* Berkeley Predictive Technology Model [36].

The results are presented in Table 1. The error in our proposed model is very small compared to the current source based model. Note that the error in voltage drop predicted by current source model for ckt9 compared to ckt1 is bigger. This is because in ckt9, there are more transistors hooked to the power grid node compared to ckt1. Hence the decoupling capacitance provided by the PMOS transistors which are on, is much bigger in ckt9 compared to ckt1. Since the current source model has no knowledge of the drain capacitance provided by the PMOS transistors which are on, there is a higher error in voltage drop predicted for ckt9 compared to ckt1.

While solving $\mathbf{Ax} = \mathbf{b}$ we employed simple *LU* factorizations rather than the specialized algorithms for solving $\mathcal{M}$-

matrix presented in the literature. Thus an implementation using compiled language and specialized algorithms for solving $\mathbf{Ax} = \mathbf{b}$, would lead to a greater speedup.

Table 1: Runtime over 10 cycles for random circuits. The drop predicted by both the proposed and the current source model are pessimistic. (Cycle-time = $750ps$)

ckt	Nodes	Runtime [s]		Speedup	Error[e][%]	
		Prop[p]	SPICE		CS[c]	Prop[p]
1	3658	4.86	153.41	31.57×	8.5	2.2
2	3958	5.70	165.19	28.98×	8.6	0.5
3	4258	6.00	179.26	29.88×	9.7	1.1
4	4558	6.56	188.70	28.77×	10.9	0.55
5	4858	7.13	199.07	27.92×	13.1	1.2
6	5158	7.80	205.73	26.38×	12.4	0.2
7	5458	8.37	215.99	25.81×	13.6	0.15
8	5758	8.95	234.18	26.17×	14.8	0.25
9	6058	9.57	252.39	26.37×	15.4	1.1

[e] Error [%] is given by $\frac{V_{\text{predicted}} - V_{\text{SPICE}}}{V_{\text{SPICE}}} \times 100$

[p] Proposed Model

[c] Current Source Model

The speedup technique discussed in Section 5.2 was also implemented. This technique improved runtime by nearly an order of magnitude. If the switching is isolated, then we can use the speedup technique discussed in Section 5.1. This can improve the runtime at least by 2×.

The major gain in using the proposed modeling comes when sizing the decoupling capacitances (decap). We considered a power grid node under the worst case switching conditions of an inverter chain ($\text{inv1} - \text{inv4} - \text{inv16}$) hooked to it.

The decap C_{decap} at a given node is given by [24]:

$$C_{\text{decap}} = \frac{\left(\frac{I_{\text{peak}} t_{\text{rise}}}{2}\right)}{\Delta V} \tag{12}$$

where the maximum voltage ripple allowed is ΔV.

When we take into account the capacitances of the gates connected to the node while simulating, the decap size needed goes down drastically as shown in Table 2. The column labeled Literature reports decap size based on Eq. (12). The proposed column has decap values obtained manually using our proposed modeling algorithm and verified using SPICE.

Table 2: Decoupling capacitances for a given tolerance level. The decap is at node for a random circuit.

$\Delta V(V)$	Proposed(pF)	Literature [24](pF)	Gain
0.01	2	6.75	3.4×
0.02	0.630	3.38	5.4×
0.03	0.236	2.25	9.5×
0.04	0.158	1.69	10.7×
0.05	0.105	1.35	12.9×

It is clear that we can get an order of magnitude reduction in decap when the ripple tolerance is higher ($\Delta V \geq 0.04V$). But even when the ripple tolerance is very low we still get a significant reduction in decap size.

8. CONCLUSION

We have proposed a new model for power grid simulation while retaining the features that make power grid simulation amenable to linear algebraic methods presented in the literature. The proposed model can be seamlessly integrated into the existing power grid models. It is more accurate compared to the current source model and it also retains the efficiency of current source model by having a constant conductance matrix. The proposed model offers the middle ground between the accuracy of SPICE simulation and the speed of the current source model. The proposed model also reduces the size of decoupling capacitance since it takes into account the drain capacitance of the transistors into account. We are working on fast approximation methods to find the voltage drop in the local charge distribution faster. This will help to improve time-step (h) and hence the runtime.

9. REFERENCES

[1] Howard H. Chen and J. Scott Neely. Interconnect and circuit modeling techniques for full-chip power supply noise analysis. *IEEE Transactions on Components, Packaging, and Manufacturing Technology, Part B: Advanced Packaging*, 21(3):209–215, August 1998.

[2] Tsung-Hao Chen and Charlie Chung-Ping Chen. Efficient large-scale power grid analysis based on preconditioned krylov-subspace iterative methods. In *DAC '01: Proceedings of the 38th conference on Design automation*, pages 559–562, New York, NY, USA, 2001. ACM Press.

[3] Yu Min Lee and Charlie Chung-Ping Chen. Power grid transient simulation in linear time based on transmission-line-modeling alternating-direction-implicit method. In *ICCAD '01: Proceedings of the 2001 IEEE/ACM international conference on Computer-aided design*, pages 75–80, Piscataway, NJ, USA, 2001. IEEE Press.

[4] Joseph N. Kozhaya, Sani R. Nassif, and Farid 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.

[5] Haihua Su, Emrah Acar, and Sani R. Nassif. Power grid reduction based on algebraic multigrid principles. In *DAC '03: Proceedings of the 40th conference on Design automation*, pages 109–112, New York, NY, USA, 2003. ACM Press.

[6] Yu Zhong and Martin D. F. Wong. Fast algorithms for IR drop analysis in large power grid. In *Proceedings of the International Conference on Computer-Aided Design*, pages 351–357, 2005.

[7] Quming Zhou, Kai Sun, Kartik Mohanram, and Danny C. Sorensen. Large power grid analysis using domain decomposition. In *DATE '06: Proceedings of the conference on Design, automation and test in Europe*, pages 27–32, 2006.

[8] Jin Shi, Yici Cai, Sheldon X.-D. Tan, and Xianlong Hong. High accurate pattern based precondition method for extremely large power/ground grid analysis. In *ISPD '06: Proceedings of the 2006*

international symposium on Physical design, pages 108–113, 2006.

[9] Hao Yu, Yiyu Shi, and Lei He. Fast analysis of structured power grid by triangularization based structure preserving model order reduction. In *DAC '06: Proceedings of the 43rd annual conference on Design automation*, pages 205–210, 2006.

[10] Min Zhao, Rajendran Panda, Sachin S. Sapatnekar, and David Blaauw. Hierarchical analysis of power distribution networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 21(2):159–168, February 2002.

[11] Sachin S. Sapatnekar and Haihua Su. Analysis and optimization of power grids. *IEEE Design & Test of Computers*, 20(3):7–15, 2003.

[12] Weikun Guo, Sheldon X.-D. Tan, Zuying Luo, and Xianglong Hang. Partial random walks for transient analysis of large power distribution networks. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer*, E87-A(12):3265–3272, December 2004.

[13] Haifeng Qian, Sani R. Nassif, and Sachin S. Sapatnekar. Early-stage power grid analysis for uncertain working modes. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 24(5):676–682, May 2005.

[14] Haifeng Qian, Sani R. Nassif, and Sachin S. Sapatnekar. Power grid analysis using random walks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 24(8):1204–1224, August 2005.

[15] Imad A. Ferzli and Farid N. Najm. Statistical verification of power grids considering process-induced leakage current variations. In *ICCAD '03: Proceedings of the 2003 IEEE/ACM international conference on Computer-aided design*, pages 770–777, Washington, DC, USA, 2003. IEEE Computer Society.

[16] Sanjay Pant, David Blaauw, Vladimir Zolotov, Savithri Sundareswaran, and Rajendran Panda. A stochastic approach to power grid analysis. In *DAC '04: Proceedings of the 41st annual conference on Design automation*, pages 171–176, New York, NY, USA, 2004. ACM Press.

[17] Peng Li. Variational analysis of large power grids by exploring statistical sampling sharing and spatial locality. In *Proceedings of the International Conference on Computer-Aided Design*, pages 645–651, 2005.

[18] Praveen Ghanta, Sarma Vrudhula, Sarvesh Bhardwaj, and Rajendran Panda. Stochastic variational analysis of large power grids considering intra-die correlations. In *DAC '06: Proceedings of the 43rd annual conference on Design automation*, pages 211–216, 2006.

[19] Eli Chiprout. Fast flip-chip power grid analysis via locality and grid shells. In *ICCAD '04: Proceedings of the 2004 IEEE/ACM International conference on Computer-aided design*, pages 485–488, Washington, DC, USA, 2004. IEEE Computer Society.

[20] Sanjay Pant and Eli Chiprout. Power grid physics and implications for CAD. In *DAC '06: Proceedings of the 43rd annual conference on Design automation*, pages 199–204, 2006.

[21] Su-Wei Wu and Yao-Wen Chang. Efficient power/ground network analysis for power integrity-driven design methodology. In *DAC '04: Proceedings of the 41st annual conference on Design automation*, pages 177–180, 2004.

[22] Chen-Wei Liu and Yao-Wen Chang. Floorplan and power/ground network co-synthesis for fast design convergence. In *ISPD '06: Proceedings of the 2006 international symposium on Physical design*, pages 86–93, New York, NY, USA, 2006. ACM Press.

[23] Jiří Vlach and Kishore Singhal. *Computer Methods for Circuit Analysis and Design*. Van Nostrand Reinhold, New York, NY, USA, 1993.

[24] Atsushi Muramatsu, Masanori Hashimoto, and Hidetoshi Onodera. Effects of on-chip inductance on power distribution grid. In *ISPD '05: Proceedings of the 2005 international symposium on Physical design*, pages 63–69, 2005.

[25] Leon O. Chua and Pen-Min Lin. *Computer-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques*. Prentice Hall Professional Technical Reference, Eaglewood Cliffs, NJ, USA, 1975.

[26] William L. Briggs, Van Emden Henson, and Steve F. McCormick. *A multigrid tutorial: second edition*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

[27] Yousef Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2003.

[28] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2001.

[29] Mark Alan Horowitz. *Timing Models for MOS Circuits*. PhD thesis, Stanford University, January 1984.

[30] Yoshishige Murakami. A Method for the Formulation and Solution of Circuits Composed of Switches and Linear *RLC* Elements. *IEEE Transactions on Circuits and Systems*, 34(5):496–509, May 1987.

[31] Huang-Jin Wu and Wu-Shiung Feng. Efficient simulation of switched networks using reduced unification matrix. *IEEE Transactions on Power Electronics*, 14(3):481–494, May 1999.

[32] Shu-Yuen Ron Hui and S. Morrall. Generalised associated discrete circuit model for switching devices. *IEE Proceedings on Science, Measurement and Technology*, 141(1):57–64, January 1994.

[33] Predrag Pejović and Dragan Maksimović. A method for fast time-domain simulation of networks with switches. *IEEE Transactions on Power Electronics*, 9(4):449–456, July 1994.

[34] Shiyou Zhao, Kaushik Roy, and Cheng-Kok Koh. Decoupling capacitance allocation and its application to power-supply noise-aware floorplanning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 21(1):81–92, 2002.

[35] Maha Nizam, Farid N. Najm, and Anirudh Devgan. Power grid voltage integrity verification. In *ISLPED '05: Proceedings of the 2005 international symposium on Low power electronics and design*, pages 239–244, 2005.

[36] Yu Cao, Takashi Sato, Michael Orshansky, Dennis Sylvester, and Chenming Hu. New paradigm of predictive MOSFET and interconnect modeling for early circuit simulation. In *Proceedings of Custom Integrated Circuits Conference*, pages 201–204, 2000.