Verification of Large-Scale On-Chip Power Grids

Xuanxing Xiong

Illinois Institute of Technology
Chicago, Illinois, United States
April, 2013
Agenda

- Background
- Parallel Transient Simulation
- Vectorless Verification
  - Steady-State Verification
  - Transient Verification
- Conclusion
On-Chip Power Grids

- Also known as power rails, power distribution networks.
- Provide VDD and GND signals to gates and cells, e.g., inverters, NAND gates, flip-flops, SRAM cells, etc.

A multi-layer on-chip power grid [Mezhiba et al. TVLSI’04].
Power Supply Noises

- Types of Noises
  - IR drop
  - Capacitively coupled noise
  - Ldi/dt drop

- Impact of Noises
  - Timing violations
  - Logic failures
  - Chip failures
  - Poor reliability

- Grid verification is a must in modern design flow.

An example chiplet [Smith TAU'11].
Max IR drop = 17 mV
Structure Exploration

Power grid structure exploration [Sarkar EETimes’05].
Power Grid Verification

- **Grid Extraction**
  - Extract an RC/RLC circuit model of the grid and load currents from the layout.

- **Noise Evaluation**
  - Estimate power supply noises in the extracted circuit model.

- **Report power supply noises**

Power grid verification flow [Smith TAU’11].
Methodology

- Verify that the power supply noises are within certain acceptable range
  - Noises depend on the patterns of currents drawn
- General idea for power grid verification
  - First, specify currents
  - Second, compute noises
- Simulation-based verification
  - DC & Transient analysis
  - Use current excitations extracted from the circuit
  - Mainstream technique for sign-off verification of power grids
General circuit simulation tools (e.g. SPICE) are inefficient for power grid analysis because of extreme time and memory complexity.

Many algorithms have been developed, e.g.
- Preconditioned Conjugate Gradient (PCG) [Chen et al. DAC’01]
- Multigrid Method [Kozhaya et al. TCAD’02]
- Hierarchical Algorithm [Zhao et al. TCAD’02]
- Random Walk Approach [Qian et al. TCAD’05]
- Sparse Approximate Inverse Technique [Cauley et al. TCAD’10]

Challenges
- Handle large-scale power grids
- Parallel implementation
Limitations of Simulation

- Need to simulate a large number of current vectors to cover usual use scenarios
- No guarantee the worst noise (but not over pessimistic) can be found.
- Does not allow early power grid verification
  - Needs specific circuit implementations for current vector extraction
  - Can only be performed after the circuit design is done
  - But early power grid verification is preferred for ease of grid modification
- Remains mainsteam technique for sign-off verification of power grids
Objective
- Evaluate the worst-case voltage noise without enumerating all possible current vectors

Methodology
- Model all possible current vectors by a feasible set
- Solve an optimization problem that maximizes the voltage noise within the feasible set

Challenges
- Optimization usually consumes more resources, especially for large power grids
Steady-State Verification

Objective
- For worst-case DC scenarios and provide bounds for RC powergrid.
- **Early works are limited to small problem sizes.** But recent advances [Abdul Ghani et al. DAC’09] [Xiong et al. DAC’10, DAC’13] have improved solution efficiency drastically.

Current modeling
- Linear Current Constraints [Kouroussis et al. 03]
  - Bound the total currents drawn by circuit blocks
  - Linear Programming (LP) Formulation
- Work Modes [Qian et al. 04]
  - Use integer variables to model the working state of the circuit blocks
  - Integer Linear Programming (ILP) Formulation
  - But ILP is very hard to solve
Transient behaviors are more realistic
  - Steady-state verification could be over pessimistic.

Power grid modeling
  - Inductances [Abdul Ghani et al. ICCAD’06]
  - Capacitive couplings between VDD and GND networks [Avci et al. ICCAD’10]

Current modeling
  - Max delta constraints [Ferzli et al. TCAD’10]
  - Current slope constraints [Du et al. ISQED’10]
  - Current conservation constraints [Avci et al. ICCAD’10]
  - Hierarchical power constraints [Cheng et al. ISPD’11]

However, there is no constraint to restrict the transient behavior of individual current sources.
Our Contribution

- Parallel Transient Simulation
  - Parallel forward and back substitution [ICCAD’12]

- Vectorless Verification
  - Steady-State Verification
    - Dual bound algorithm [TCAD’11]
    - Hierarchical matrix inversion [ICCAD’10]
    - Constraint abstraction [DAC’13]

- Transient Verification
  - Transient constraints [ICCAD’11]
  - Variable reduction [TCAD’13]
Our Contribution

- Parallel Transient Simulation
  - Parallel forward and back substitution [ICCAD’12]

- Vectorless Verification
  - Steady-State Verification
    - Dual bound algorithm [TCAD’11]
    - Hierarchical matrix inversion [ICCAD’10]
    - Constraint abstraction [DAC’13]

- Transient Verification
  - Transient constraints [ICCAD’11]
  - Variable reduction [TCAD’13]
Agenda

- Background
- Parallel Transient Simulation [ICCAD’12]
- Vectorless Verification
  - Steady-State Verification
  - Transient Verification
- Conclusion
Preliminary

- Transient Analysis Equation
  \[ A v(t) = b(t) \]

- Direct Solver
  - Forward and back substitution at each time step

- Preconditioned CG Solver
  - Forward and back substitution to solve \( Mz_k = r_k \)
  - Matrix vector multiplication and vector operations

- Forward and back substitution is a dominating routine, but it is difficult to parallelize.
[Gupta et al SC’95]

- A post-factorization approach
- Supernode-based parallelization according to the elimination tree
- Achieve weak performance scaling using up to 256 processors with multiple right-hand-side (RHS) vectors (e.g., 30)
- However, it is often difficult to form supernodes for power grid transient analysis, and we only have a single RHS vector at each time step.
Levelized Parallelization

- **A post-factorization approach**
- It is implemented in a **direct solver**.
- For each level, compute the variables in parallel.
  - To reduce scheduling overhead, we only parallelize the levels with more than 10K variables in implementation.
ND-based Parallelization

- A pre-factorization approach
- It is implemented in a **preconditioned CG solver** with a stochastic preconditioner [Qian et al SIAM J.’08].

\[
A = L^T D L
\]

- Order the nodes by nested-dissection (ND)
  - Use RCM ordering for sufficiently small subsets of nodes
- The resultant matrix \( L \) has parallelizable structure.
To reduce scheduling overhead, we only solve the leaf subsets of nodes (P1, P2, ...) in parallel.
Direct Solver: Forward & Back Sub.

**Forward and Back Substitution Runtime (Direct Solver)**

- **1 thread**
- **2 threads**
- **4 threads**
- **8 threads**
- **12 threads**

**X-axis:** ibmpg1t, ibmpg2t, ibmpg3t, ibmpg4t, ibmpg5t, ibmpg6t

**Y-axis:** Runtime (s)
PCG Solver: Forward & Back Sub.

Forward and Back Substitution Runtime (PCG Solver)

- 1 thread
- 2 threads
- 4 threads
- 8 threads
- 12 threads

Runtime (s)
The direct solver is much faster than the PCG solver.

The pre-factorization approach (ND-based parallelization) is more efficient than the post-factorization approach (levelized parallelization) for parallel forward and back substitution.

Only 2X speedup is achieved because of high memory access to computation ratio (i.e., little data reuse).
Agenda

- Background
- Parallel Transient Simulation
- **Vectorless Verification**
  - Steady-State Verification
  - Preliminary
  - Constraint Abstraction
  - Transient Verification
- Conclusion
RC Power Grid
System Equations

- DC Model
  \[ G \mathbf{v} = \mathbf{i} \]

- Transient Model
  \[ G \mathbf{v}(t) + C \dot{\mathbf{v}}(t) = \mathbf{i}(t) \]

  \( G \): symmetric conductance matrix
  \( C \): diagonal capacitance matrix
Linear Current Constraints

[Kouroussis et al. DAC’03]

- Local Constraints

\[ 0 \leq i(t) \leq I_L, \forall t \]

- Global Constraints

\[ U \cdot i(t) \leq I_G, \forall t \]

U is a 0/1 matrix indicating the assignment of current sources to groups.
The MaxVN-LCC Problem

For every node $1 \leq l \leq n$,

\[
\text{Maximize } v_l \quad \text{s.t.} \quad Av = i, \quad 0 \leq i \leq I_L, \quad Ui \leq I_G
\]

Note: for DC model, $A = G$

for transient model, $A = G + \frac{C}{\Delta t}$
Problem Decomposition

- Original MaxVN-LCC Problem

\[
\begin{align*}
\text{Maximize} & \quad v_l \\
\text{subject to} & \quad Av = i, 0 \leq i \leq I_L, U i \leq I_G
\end{align*}
\]

- Decomposed Problem

I: Compute \(c_l\) by solving \(Ax = e_l\),

II: Maximize \(v_l = c_l^T i\) \(\text{s.t.}\)

\[0 \leq i \leq I_L, U i \leq I_G.\]

Two Steps: Power Grid Analysis & Noise Maximization
Minimize $D(\gamma)$ s.t. $\gamma \geq 0$,

$$D(\gamma) \triangleq I_G^T \gamma + \sum_{j=1}^{n} I_{L,j} \max(0, c_{l,j} - u_j^T \gamma)$$

- Employ the PCG method to compute $C_l$
- Apply the cutting-plane method to solve the simplified dual problem with error tolerance $\delta_{lp}$
- Very efficient when there are only a few number of global constraints
- However, the cutting-plane method become less efficiently when the number of global constraints increases.
Agenda

- Background
- Parallel Transient Simulation
- Vectorless Verification
  - Steady-State Verification
    - Preliminary
    - Constraint Abstraction [DAC’13]
  - Transient Verification
- Conclusion
Divide and Conquer?
Verifying a Subgrid

- ○ Internal Node of the Subgrid
- ● Neighboring Node of the Subgrid
- ○ External Node of the Subgrid

Subgrid
Boundary Constraints

- Define partial specification of boundary condition for each subgrid.

\[ 0 \leq v_{ex} \leq v_{\ell}, \quad \text{and} \quad \sum_{1 \leq j \leq \hat{m}} v_{ex,j} \leq v_g \]

- \( v_{\ell} \) is the vector of worst-case voltage noises at the neighboring global nodes, \( v_g \) is the maximum sum of voltage noises at the neighboring global nodes, which is computed by solving

\[
\text{Maximize} \quad \sum_{1 \leq j \leq \hat{m}} v_{ex,j} \quad \text{s.t.} \quad Av = i, \ 0 \leq i \leq I_L, \ Ui \leq I_G
\]
Constraint Abstraction

- Partition the power grid into disjoint subgrids, which are separated by global nodes.
- For each global node, compute its worst-case voltage noise (or an upper bound of voltage noise).
- For each subgrid, build boundary constraints, and then compute upper bounds of the worst-case voltage noises at internal nodes subject to boundary constraints.
Exploring Subgrid Size

![Graph showing the relationship between runtime and rough subgrid size](image)
Exploring Subgrid Size

![Graph showing the relationship between Overestimation of Voltage Noise (mV) and Rough Subgrid Size (rss). The graph compares two lines: Emax and Eavg. Emax shows a steep increase with increasing subgrid size, whereas Eavg increases more gradually.](image-url)
Speedup Relative to DualVN

![Graph showing speedup relative to DualVN vs number of nodes.](image-url)
Agenda

- Background
- Parallel Transient Simulation
- Vectorless Verification
  - Steady-State Verification
  - Transient Verification [TCAD’13]
- Conclusion
Integrated RLC Power Grid
The System Equation

- Time domain
  \[ Lv(t) + Gv'(t) + Cv''(t) = (\hat{I}(t))' \]
  - \( G \): conductance matrix
  - \( L/C \): represent self-inductance/capactiance links
  - \( v(t) \): nodal voltage noises
  - \( I(t) \): current excitations

- Discretization with time step \( \Delta t \) using trapezoidal rule
  \[
  v(t) = A^{-1}\left(\hat{I}(t) - \hat{I}(t - 2\Delta t) + Bv(t - \Delta t) + Dv(t - 2\Delta t)\right)
  \]
  \[
  A = G + \frac{\Delta t}{2}L + \frac{2}{\Delta t}C, \quad B = \frac{4}{\Delta t}C - \Delta tL, \quad D = G - \frac{\Delta t}{2}L - \frac{2}{\Delta t}C
  \]
Current Constraints

[Kouroussis et al. DAC’03] and [Avci et al. ICCAD’10]

- **Local Constraints**
  \[0 \leq I(t) \leq I_L, \forall t, \text{ or } 0 \leq I(k\Delta t) \leq I_L, \forall k\]

- **Global Constraints**
  \[UI(t) \leq I_G, \forall t, \text{ or } UI(k\Delta t) \leq I_G, \forall k\]

- **Current Conservation Constraints**
  \[EI(t) = 0, \forall t, \text{ or } EI(k\Delta t) = 0, \forall k\]
Our Transient Current Constraints

\[ \int_{0}^{N_{ts} \times \Delta t} I(t) \, dt \leq I_{T} \times \Delta t, \quad \text{or} \quad \sum_{k=1}^{N_{ts}} I(k \Delta t) \leq I_{T} \]

- \( N_{ts} \): number of time steps
- \( I_{T} \): nx1 upper bound vector
- Transient constraints may be extracted from the circuit by switching activity analysis, e.g.

[Morgado et al. ICSD’09] and [Morgado et al. TODAES’09]
Our Problem Formulation

- For each node $j$

Maximize/Minimize $v_j(N_{ts} \Delta t)$ subject to:

$$v(t) = A^{-1}\left(\hat{I}(t) - \hat{I}(t-2\Delta t) + Bv(t-\Delta t) + Dv(t-2\Delta t)\right)$$

$$0 \leq I(k\Delta t) \leq I_L, UI(k\Delta t) \leq I_G, EI(k\Delta t) = 0, \sum_{k=1}^{N_{ts}} I(k\Delta t) \leq I_T$$

- The formulation actually computes the worst noise at node $j$ for all time slots $k\Delta t$

- If the cumulative effects of voltage noises are of interests, e.g. similar to [Evmorfopoulos et al. ICCAD’10], the objective function can be

$$\text{Maximize/Minimize } \sum_{k'=1}^{N_{ts}} v_j(k' \Delta t)$$
Property of System Equation

\[ \mathbf{v}(\Delta t) = \mathbf{A}^{-1} \mathbf{I}(\Delta t), \]
\[ \mathbf{v}(2\Delta t) = \mathbf{A}^{-1} (\mathbf{I}(2\Delta t) + \mathbf{Bv}(\Delta t)) \]
\[ = \mathbf{A}^{-1} \mathbf{I}(2\Delta t) + \mathbf{A}^{-1} \mathbf{BA}^{-1} \mathbf{I}(\Delta t) \]

- There exists a unique series of nxn matrices \( H_1, H_2, \ldots, H_k, H_{k+1}, \ldots \), such that
  \[ \mathbf{v}(k\Delta t) = H_1 \mathbf{I}(k\Delta t) + H_2 \mathbf{I}((k - 1)\Delta t) + \cdots + H_k \mathbf{I}(\Delta t). \]

- \( j \)th column of \( S_k \) can be computed as
  \[ \mathbf{c}_{j,k} \triangleq \mathbf{v}(k\Delta t) \bigg| \mathbf{I}(\Delta t) = e_j, \mathbf{I}(p\Delta t) = 0, \forall 2 \leq p \leq k = H_k e_j \]

- \( H_k \) is symmetric. So
  \[ \mathbf{v}_j(k\Delta t) = \mathbf{c}_{j,1}^T \mathbf{I}(k\Delta t) + \mathbf{c}_{j,2}^T \mathbf{I}((k - 1)\Delta t) + \cdots + \mathbf{c}_{j,k}^T \mathbf{I}(\Delta t) \]
Symmetric Impulse Response
Problem Decomposition

- For each node $j$:
  
  I: Compute $c_{j,k}$, $\forall 1 \leq k \leq N_{ts}$,
  
  II: Maximize/Minimize $v_j(N_{ts}\Delta t) = \sum_{k=1}^{N_{ts}} c_{j,k}^T \hat{I}((N_{ts} + 1 - k)\Delta t)$

  Subject to: $0 \leq I(k\Delta t) \leq I_L$, $UI(k\Delta t) \leq I_G$,

  $EI(k\Delta t) = 0$, $\sum_{k=1}^{N_{ts}} I(k\Delta t) \leq I_T$.

- Sub-problem I: transient analysis with current excitation $e_j$ to compute $c_{j,k}$

- Sub-problem II: linear programming (LP) to compute worst-case voltage noises
A Simple Case Study

\[ I_1(t) \leq 10\text{mA} \]
\[ I_2(t) \leq 10\text{mA} \]
\[ I_3(t) \leq 10\text{mA} \]
Frequency Response

Voltage Drop vs Current Frequency

Magnitude (dBV) vs f(Hz)

Phase (deg) vs f(Hz)

1
2
3
Worst-Case Waveforms

- Worst-case waveforms can be found.

- Ignoring transient contraints leads to a 35.6% overestimation on the voltage drop (61.8 to 83.8mV ).

Condition: $N_{ts}=10,000$, $\Delta t = 10\,\text{ps}$, $IT = 10,000\,\text{mA}$
Distribution of Coefficients

The Distribution of the Absolute Values of Coefficients

- $[1e-1, +\infty)$
- $[1e-2, 1e-1)$
- $[1e-3, 1e-2)$
- $[1e-4, 1e-3)$
- $[1e-5, 1e-4)$
- $[1e-6, 1e-5)$
- $[1e-7, 1e-6)$
- $[1e-8, 1e-7)$
- $[1e-9, 1e-8)$
- $(-\infty, 1e-9)$
Variable Reduction

- Assume hierarchical global constraints, use a sorting-deletion algorithm to estimate the maximum voltage noise caused by each group of currents.
- Remove insignificant current variables according to a user-specified error bound.
Performance Speedup over Standard LP Solver

Number of Nodes

- 5mV Error
- 10mV Error
- 20mV Error
Agenda

- Background
- Parallel Transient Simulation
- Vectorless Verification
  - Steady-State Verification
  - Transient Verification
- Conclusion
Conclusion

- Investigated parallel forward and back substitution algorithms for power grid transient simulation.

- Proposed novel algorithms to speedup vectorless steady-state verification for up to 17X.

- Proposed efficient approach for vectorless transient verification with novel transient current constraints.
Future Work

- Parallel Transient Simulation
  - Explore temporal decomposition for parallel implementation

- Vectorless Verification
  - Build a systematic solution of steady-state verification based on the proposed algorithms
  - Transient verification for all possible time t.
  - A comparison study of power grid simulation and vectorless verification
Thank You !!
Constraint Abstraction Results

<table>
<thead>
<tr>
<th>Power Grids</th>
<th>Constraint Abstraction</th>
<th>Speedup Relative to</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Name</td>
<td>( n )</td>
</tr>
<tr>
<td>pg1000</td>
<td>5875</td>
<td>46.31</td>
</tr>
<tr>
<td>pg2000</td>
<td>22939</td>
<td>39.91</td>
</tr>
<tr>
<td>pg2500</td>
<td>35668</td>
<td>28.80</td>
</tr>
<tr>
<td>pg3000</td>
<td>51195</td>
<td>43.63</td>
</tr>
<tr>
<td>pg4000</td>
<td>90643</td>
<td>54.38</td>
</tr>
<tr>
<td>pg5000</td>
<td>141283</td>
<td>( \approx 45.91 )</td>
</tr>
<tr>
<td>pg10000</td>
<td>562363</td>
<td>( \approx 23.11 )</td>
</tr>
<tr>
<td>ibmpg1</td>
<td>10242</td>
<td>677.67</td>
</tr>
<tr>
<td>ibmpg2</td>
<td>65228</td>
<td>357.92</td>
</tr>
<tr>
<td>ibmpg3</td>
<td>150687</td>
<td>( \approx 179.91 )</td>
</tr>
<tr>
<td>ibmpg4</td>
<td>478094</td>
<td>( \approx 3.42 )</td>
</tr>
<tr>
<td>ibmpg5</td>
<td>291382</td>
<td>( \approx 42.44 )</td>
</tr>
<tr>
<td>ibmpg6</td>
<td>430337</td>
<td>( \approx 109.96 )</td>
</tr>
</tbody>
</table>
## Average Runtime per Node

<table>
<thead>
<tr>
<th>Power Grids</th>
<th>Nodes</th>
<th>${N_{cs}}^1$</th>
<th>$^2$Time (s)</th>
<th>$^3$Time (s)</th>
<th>$^4$Variables</th>
<th>$^5$Time (s)</th>
<th>$^6\times$</th>
</tr>
</thead>
<tbody>
<tr>
<td>pg1</td>
<td>3088</td>
<td>1352</td>
<td>0.12</td>
<td>12.41</td>
<td>43646</td>
<td>3.42</td>
<td>3.63</td>
</tr>
<tr>
<td>pg2</td>
<td>11768</td>
<td>5202</td>
<td>0.46</td>
<td>105.21</td>
<td>84769</td>
<td>16.62</td>
<td>6.33</td>
</tr>
<tr>
<td>pg3</td>
<td>45928</td>
<td>20402</td>
<td>1.65</td>
<td>415.16</td>
<td>92093</td>
<td>7.66</td>
<td>54.19</td>
</tr>
<tr>
<td>pg4</td>
<td>71408</td>
<td>31752</td>
<td>2.66</td>
<td>612.32</td>
<td>143613</td>
<td>6.32</td>
<td>96.95</td>
</tr>
<tr>
<td>pg5</td>
<td>102488</td>
<td>45602</td>
<td>3.80</td>
<td>1064.45</td>
<td>200959</td>
<td>20.84</td>
<td>51.08</td>
</tr>
<tr>
<td>pg6</td>
<td>181448</td>
<td>80802</td>
<td>6.84</td>
<td>1846.32</td>
<td>346159</td>
<td>29.18</td>
<td>63.28</td>
</tr>
<tr>
<td>pg7</td>
<td>282808</td>
<td>126002</td>
<td>10.98</td>
<td>3455.03</td>
<td>576164</td>
<td>30.15</td>
<td>114.60</td>
</tr>
<tr>
<td>pg8</td>
<td>1 125 608</td>
<td>502 002</td>
<td>45.61</td>
<td>7NA</td>
<td>1857217</td>
<td>131.57</td>
<td>NA</td>
</tr>
</tbody>
</table>
Accuracy Results

<table>
<thead>
<tr>
<th>Power Grids</th>
<th>Backward Euler</th>
<th>Trapezoidal Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$E_{max}$</td>
<td>$E_{avg}$</td>
</tr>
<tr>
<td>ibmpg1t</td>
<td>1.08e-03</td>
<td>1.61e-04</td>
</tr>
<tr>
<td>ibmpg2t</td>
<td>8.92e-04</td>
<td>1.53e-04</td>
</tr>
<tr>
<td>ibmpg3t</td>
<td>8.73e-04</td>
<td>1.31e-04</td>
</tr>
<tr>
<td>ibmpg4t</td>
<td>2.11e-03</td>
<td>1.63e-04</td>
</tr>
<tr>
<td>ibmpg5t</td>
<td>5.63e-04</td>
<td>9.08e-05</td>
</tr>
<tr>
<td>ibmpg6t</td>
<td>6.47e-04</td>
<td>1.11e-04</td>
</tr>
</tbody>
</table>

- Trapezoidal rule is much more accurate than backward Euler for power grid transient simulation.