Efficient Large-Scale Power Grid Analysis with Parallel Computing in Mathematica

Tamiyo Nakabayashi†, Keiji Nakabayashi‡, and Fujio Kako†
†Nara Women’s University
‡WDB Corporation
Kitauoyahigashi-machi, Nara, Japan 2-3-13, Azuchi-machi, Chuo-ku, Osaka, Japan
{nakaba, kako}@ics.nara-wu.ac.jp nakaba62@hotmail.com

Abstract: - We propose a technique for large scale power grid analysis with a parallel computing method in Mathematica. It models a power grid as a system of ordinary differential equations, and applies the parallel computing method to get a solution efficiently. We explain the technique using a linear RC elements network model of the power grid. Our experimental results demonstrate that the technique is capable of time domain analysis same as that by SPICE (a general-purpose circuit simulator), and is applicable to power grid design of VLSI chips. And they reveal the superiority of our technique under the condition of very large scale power grids, and also indicate the effectiveness of the parallel computing method in the power grid optimization design.

Key-Words: - power grid analysis, IR-drop, voltage fluctuations, Mathematica, parallel computing, Spice

I Introduction

Due to the increasing complexity and power consumption of VLSI chips, power grid analysis is an important issue. A robust power network design has to guarantee the correctness of circuit functionalities without slowing down operations. An improper design of power grid (power distribution system) can result in excessive IR-drops, and fluctuations in the voltages supplied to the active devices (transistors, functional blocks, etc.). If the voltage drop becomes too large, it increases the gate delays and causes logical errors (functional failures). And it degrades the circuit reliability. Many researchers have studied the impact, and proposed solutions to the problem [1], [13].

So far circuit simulators such as SPICE [11] and GridSim [6] have been used for power grid analysis in the VLSI design. And some high performance numerical solvers also were developed [3], [4], [7], [14], [15]. Mathematica is worldwide software of symbolic analysis (computer algebra) and a numerical analysis tool, and it is used in science, engineering, mathematics, and other areas [8]. In the area of electronic/electrical design, Mathematica has been applied to symbolic circuit analysis for analog circuits [5], [9], and numerical analysis for a power MOSFET circuit [10]. Furthermore, in the Wolfram Demonstrations Project, some examples of equation-based modeling for very simple electronic circuits were exhibited [2]. Recently Mathematica also supports a parallel computing method [12].

The goal of our study is to propose a technique for large scale power grid analysis with Mathematica. There are two supply grids in VLSI design: the power and ground grids. The two grids influence each other, and therefore a simultaneous simulation is preferred. However, if we take advantage of the fact that the power and ground grids are often symmetric, the combined power/ground grids can be reduced back to a single power grid [1], [13], [15]. Therefore, in this paper, we consider the analysis of the power grid. Furthermore we deal with the IR-drop (resistive voltage drop) which is mostly due to the voltage drop due to power line resistances on chip. First we consider a RC circuit model that is composed of a large scale linear interconnect network and the driving current sources due to the active devices in the power grid, and express it as a system of ordinary differential equations. Next we implement the system into Mathematica, and solve it. In particular, we perform verification for the IR-drop values under the condition of various driving currents with parallel computing method.
Finally we demonstrate our technique by several test results.

In the next section, we explain a RC elements network model of the power grid and derive the system of ordinary differential equations. Section III describes our technique for the power grid analysis and implements it into Mathematica. Section IV describes our experiments and compares the results. We conclude the paper in Section V.

NOMENCLATURE

\[ t \] Simulation time domain [sec]
\((i,j)\) Coordinate system of power grid
(\(i\) and \(j\) ranges as \(i = 1, 2, \ldots\), \(j = 1, 2, \ldots, n\))
\(m\) A positive integer of 2 or more
\(n\) A positive integer of 2 or more
\(v_{i,j}\) Nodal voltage at node \((i,j)\) [V]
\(C_{int}\) Interconnect capacitance [F]
\(R_{int}\) Interconnect resistance [Ohm]
\(G_{int}\) Interconnect conductance [1/Ohm] (\(G_{int} = \frac{1}{R_{int}}\))
\(R_{off}\) Off resistance of active device [Ohm]
\(G_{off}\) Off conductance of active device [1/Ohm] (\(G_{off} = \frac{1}{R_{off}}\))
\(I_{i,j}\) Pulsed driving current of active device at node \((i,j)\) [A]
\(I_{high}\) High value of pulsed driving current [A] (\(I_{high} = \text{Max}(I_{i,j})\))
\(I_{low}\) Low value of pulsed driving current [A] (\(I_{low} = \text{Min}(I_{i,j})\))
\(\text{Max}()\) Maximum function
\(\text{Min}()\) Minimum function
\(C_{M}\) Capacitance matrix : \(C_{M} = [C_{int}]\)
\(G_{M}\) Conductance matrix : \(G_{M} = [G_{int}] + [G_{off}]\)
\(I_{M}\) Pulsed driving current matrix : \(I_{M} = [I_{i,j}]\)
\(v_{M}\) Nodal voltage vector : \(v_{M} = [v_{i,j}]\)
\(V_{dd}\) Power supply voltage [V]
\(T_{sw}\) Switching pulse period [sec]

II The Problem

Consider a simple power grid as depicted in Fig. 2.1. The power grid is a mesh structure, in which each edge is modelled as a uniform interconnect resistance. Each node \((v_{i,j})\) in the mesh has a uniform parasitic capacitance \((C_{int})\) to the ground. Active devices are modelled as pulsed current sources, and are connected to the mesh nodes. Each pulsed current source \((I_{i,j})\) is a driving current to be generated by the switching of the active device. Note that each driving current source has an off resistance \((R_{off})\) in parallel. Four corner nodes are connected to power pads (power supply voltage sources) that can be treated as ideal voltage sources \((V_{dd})\). Therefore our simple power grid is composed of a linear network of uniform distributed RC elements which excited by 4-corner ideal voltage sources and driving current sources.

Therefore our simple power grid is composed of a linear network of uniform distributed RC elements which excited by 4-corner ideal voltage sources and driving current sources. As depicted in Fig. 2.2, the KCL (Kirchhoff's Current Law) with respect to each node is expressed as Eq. (2-1).

We rewrite Eq. (2-1) and get the following equation.

\[
C_{int} \frac{dv_{i,j}}{dt} = \frac{v_{i+1,j} - v_{i,j}}{R_{int}} + \frac{v_{i,j+1} - v_{i,j}}{R_{int}} + \frac{v_{i,j-1} - v_{i,j}}{R_{int}} - \frac{v_{i+1,j} - v_{i,j}}{R_{off}} - I_{i,j}
\]
Hence the whole of the linear network can be represented as a system of ordinary differential equations, which is expressed as Eq. (2-3). Our problem is to find the IR-drops and fluctuations in the voltages supplied to the active devices on chip by solving the system of Eq. (2-3). Note the matrix form of Eq. (2-3).

\[
C_M \frac{dv_M}{dt} = G_M v_M - I_M
\]  

III Implementation into Mathematica

As described above, in our technique, we formulated the problem of the power grid analysis and derived the system of ordinary differential equations which expressed as Eq. (2-3). We develop a notebook source code to solve the system by using a numerical solver called NDsolve(), and implement it into Mathematica. In order to verify the results from the analysis with our technique, we compare with those from SPICE simulation.

In the remainder of this paper, we consider the stochastic behaviour of the switching of the active devices on chip and place the driving current sources at random with using random numbers. And we assume the synchronous switching of the active devices in the power grid.

As our experimental verification, we set the values of the main parameters as following: \(I_{\text{high}} = 2.0 \times 10^{-5} \, \text{A} \), \(I_{\text{low}} = 0.0 \, \text{A} \), \(R_{\text{int}} = 25.0 \, \text{Ohm} \), \(C_{\text{int}} = 3 \times 10^{-16} \, \text{F} \), \(R_{\text{off}} = 1.0 \times 10^{12} \, \text{Ohm} \), \(T_{\text{sw}} = 1.0 \times 10^{-3} \, \text{sec} \), \(V_{\text{dd}} = 1.0 \, \text{V} \), \(m = 30 \), \(n = 30 \). First we ran the power grid analysis with Mathematica-8 on HP dv9700 (OS Windows 7, CPU Intel(R) Single Core(TM) 2.5GHz, RAM 8.00 GB). And we obtained the nodal voltages of the power grid, and its run-time (CPU time) is 91.2 [sec]. Figure 3.1 (1) depicts the voltage fluctuations at the node \(v_{15,15} \) in the time domain.

As other reference, we use a general-purpose circuit simulator known as Ngspice [11] (gEDA Project version of SPICE), and simulate the RC-elements network of the same power grid. In this case, the run-time with the same computer mentioned above is 4.54 [sec]. Figure 3.1 (2) depicts the voltage fluctuations at the node \(v_{15,15} \) which obtained by Ngspice.

The results from the analysis with our technique are in accord with those from SPICE simulation. Thus the results demonstrate that our technique is capable of the same analysis as SPICE, and is applicable to power grid design of VLSI chips.

IV Experimental Results

In the power grid design, the verification for the IR-drop values under the condition of various driving currents is very essential. We apply the technique described as above to the verification, and use the parallel computing method in Mathematica, in the sake of its efficiency. Figure 4.1 depicts an example of our notebook source code which used the parallel computing method.

In the first experimentation, on the power grid of the 20x20 grid nodes \((m=20, \, n=20)\), we consider the case in which the value of the pulsed driving current \(I_{\text{high}} \) takes the range from \(1.0 \times 10^{-5} \) to \(8.0 \times 10^{-5} \) which its step size is \(1.0 \times 10^{-5} \). Note that the values of other parameters are same as those in Section III. We ran the power grid analysis for...
each case at the same time, with Mathematica-8 on SUSE Linux Enterprise Server 11 which is composed of Intel Xeon Processor (8-core, 2.0GHz) and SGI Performance Suite software. Its run-time is 29.36 [sec], and is 3.79 times faster than that (111.27[sec]) of the sequential computing method in Mathematica. Table I shows the IR-drop value at the node $v_{10,10}$ for each pulsed driving current. Furthermore, the IR-drop values from Mathematica are in accord with that of those from Ngspice. The relative error is less than 2 percent. This supports the correct IR-drop values which produced by our technique. And it also indicates the effectiveness of the parallel computing method in the power grid optimization design.

Table I: IR-drop values at the node $v_{10,10}$ under the condition of various driving currents pulsed driving current.

<table>
<thead>
<tr>
<th>Driving Current $I_{sec}$ [A]</th>
<th>(1) Mathematica IRdrop [V]</th>
<th>(2) Ngspice IRdrop [V]</th>
<th>Relative Error $((1)-(2))/2$ [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00E-05</td>
<td>0.0194</td>
<td>0.0194</td>
<td>0.000</td>
</tr>
<tr>
<td>2.00E-05</td>
<td>0.0385</td>
<td>0.0387</td>
<td>-0.517</td>
</tr>
<tr>
<td>3.00E-05</td>
<td>0.0578</td>
<td>0.0588</td>
<td>-1.701</td>
</tr>
<tr>
<td>4.00E-05</td>
<td>0.0779</td>
<td>0.0774</td>
<td>0.646</td>
</tr>
<tr>
<td>5.00E-05</td>
<td>0.0963</td>
<td>0.0968</td>
<td>-0.517</td>
</tr>
<tr>
<td>6.00E-05</td>
<td>0.1169</td>
<td>0.1182</td>
<td>0.602</td>
</tr>
<tr>
<td>7.00E-05</td>
<td>0.1364</td>
<td>0.1355</td>
<td>0.664</td>
</tr>
<tr>
<td>8.00E-05</td>
<td>0.1541</td>
<td>0.1549</td>
<td>-0.516</td>
</tr>
</tbody>
</table>

In the second experimentation, under the same condition as that of the first experimentation described as above, we evaluate the performance of our technique. Table II shows the CPU time for each size ($m \times n$) of the power grid. The processing CPU time of Mathematica is proportional to the 2.5 power of grid size $m (=n)$. On the other hand, that of Ngspice is proportional to the 3 power of the grid size $m$. Note that the difference of CPU time between Mathematica and Ngspice becomes less in the case of large scale grid. Especially, Mathematica is able to analyze the power grid under the condition of grid size $m=n=300$. But Ngspice is unable to deal with it due to memory.

The results reveal the superiority of our technique under the condition of very large scale power grids. They also demonstrate that our technique is able to be applied to industrial design of power grid of VLSI chips in a practical time.

Table II: CPU time for different size of power grid.

<table>
<thead>
<tr>
<th>$m \times n$</th>
<th>(1) Mathematica (\text{(Parallel Computing)}) CPU Time [sec]</th>
<th>Ratio [-]</th>
<th>(2) Ngspice CPU Time [sec]</th>
<th>Ratio [-]</th>
<th>(1) / (2) CPU Time Ratio [-]</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 \times 10</td>
<td>4.92</td>
<td>1.00</td>
<td>0.29</td>
<td>1.00</td>
<td>17.1</td>
</tr>
<tr>
<td>20 \times 20</td>
<td>29.36</td>
<td>5.97</td>
<td>1.68</td>
<td>5.83</td>
<td>17.5</td>
</tr>
<tr>
<td>30 \times 30</td>
<td>72.90</td>
<td>14.82</td>
<td>6.98</td>
<td>24.22</td>
<td>10.4</td>
</tr>
<tr>
<td>40 \times 40</td>
<td>144.68</td>
<td>29.41</td>
<td>19.14</td>
<td>66.44</td>
<td>7.6</td>
</tr>
<tr>
<td>50 \times 50</td>
<td>259.48</td>
<td>52.75</td>
<td>55.34</td>
<td>192.14</td>
<td>4.7</td>
</tr>
<tr>
<td>75 \times 75</td>
<td>650.80</td>
<td>132.30</td>
<td>274.46</td>
<td>953.00</td>
<td>2.4</td>
</tr>
<tr>
<td>100 \times 100</td>
<td>1480.68</td>
<td>301.01</td>
<td>912.60</td>
<td>3168.75</td>
<td>1.6</td>
</tr>
<tr>
<td>200 \times 200</td>
<td>8799.37</td>
<td>1788.85</td>
<td>7124.40</td>
<td>24800.00</td>
<td>1.2</td>
</tr>
<tr>
<td>300 \times 300</td>
<td>24260.27</td>
<td>4930.95</td>
<td>NG</td>
<td>NG</td>
<td>-</td>
</tr>
</tbody>
</table>

(* * * Pulse function * * *)

\[
fpulse[x_\text{Real}, \text{delay}, \text{rise}, \text{width}, \text{fall}, \text{periode}, \text{lowval}, \text{highval}] = \begin{cases} 
\text{lowval}, & x \leq \text{delay} \\
\frac{\text{lowval} - \text{highval}}{\text{fall}}(x - \text{delay} - \text{rise} - \text{width}) + \text{highval}, & \text{Mod}[x - \text{delay}, \text{periode}] > \text{rise}, \\
\frac{\text{lowval} - \text{highval}}{\text{rise}}(x - \text{delay} - \text{periode}) + \text{lowval}, & \text{Mod}[x - \text{delay}, \text{periode}] \leq 0.0, \\
\text{lowval}, & \text{Mod}[x - \text{delay}, \text{periode}] \leq 0.0, \\
\end{cases}
\]

(* * * Size: 20 x 20 * * *)

Clear \[v2xy1, v3xy1, v4xy1, v5xy1, v6xy1, v15xy20, v16xy20, v17xy20, v18xy20, v19xy20\];

(* * * A system of ordinary differential equations * * *)

\[
eqns = \{\text{cint} v2xy1'[t] == -v2xy1[t]/\text{roff} + (v1xy1[v2xy1[t]]/\text{rint} + (v3xy1[t] - v2xy1[t])/\text{rint} + (v2xy2[t] - v2xy1[t])/\text{rint}, \\
\text{cint} v3xy1'[t] == (-1) \text{isrc} fpulse[t, 0 + \text{delay}, \text{rise}, \text{width}, \text{fall}, \text{periode}, \text{lowval}, \text{highval}] - v3xy1[t]/\text{rint} = \text{isrc} fpulse[t, 0 + \text{delay}, \text{rise}, \text{width}, \text{fall}, \text{periode}, \text{lowval}, \text{highval}] - v4xy1[t]/\text{rint} = \text{isrc} fpulse[t, 0 + \text{delay}, \text{rise}, \text{width}, \text{fall}, \text{periode}, \text{lowval}, \text{highval}] - v4xy1[t]/\text{rint} = \text{isrc} fpulse[t, 0 + \text{delay}, \text{rise}, \text{width}, \text{fall}, \text{periode}, \text{lowval}, \text{highval}] - v4xy1[t]/\text{rint} = \text{isrc}
\]
\( t \cdot v4xy1[t]/\text{rint}, \)
cint \( v5xy1'[t] == (-1) \) isrc
fpulse\( t, 0 + \text{delay}, \text{rise}, \text{width}, \text{fall}, \text{periode}, \text{lowval}, \text{highval} \)-v5xy1[t]/roff+(v6xy1[t]-v5xy1[t])/rint+(v5xy2[t]-v5xy1[t])/rint,
cint \( v6xy1'[t] == (-1) \) isrc
fpulse\( t, 0 + \text{delay}, \text{rise}, \text{width}, \text{fall}, \text{periode}, \text{lowval}, \text{highval} \)-v6xy1[t]/roff+(v7xy1[t]-v6xy1[t])/rint+(v6xy2[t]-v6xy1[t])/rint,
\( \cdots \)
\( \cdots \)
cint \( v15xy20'[t] == -v15xy20[t]/\text{roff}+(v14xy20[t]-v15xy20[t])/\text{rint}+(v16xy20[t]-v15xy20[t])/\text{rint}+(v15xy19[t]-v15xy20[t])/\text{rint}, \)
cint \( v16xy20'[t] == -v16xy20[t]/\text{roff}+(v15xy20[t]-v16xy20[t])/\text{rint}+(v17xy20[t]-v16xy20[t])/\text{rint}+(v16xy19[t]-v16xy20[t])/\text{rint}, \)
cint \( v17xy20'[t] == -v17xy20[t]/\text{roff}+(v16xy20[t]-v17xy20[t])/\text{rint}+(v18xy20[t]-v17xy20[t])/\text{rint}+(v17xy19[t]-v17xy20[t])/\text{rint}, \)
cint \( v18xy20'[t] == -v18xy20[t]/\text{roff}+(v17xy20[t]-v18xy20[t])/\text{rint}+(v19xy20[t]-v18xy20[t])/\text{rint}+(v18xy19[t]-v18xy20[t])/\text{rint}, \)
cint \( v19xy20'[t] == -v19xy20[t]/\text{roff}+(v18xy20[t]-v19xy20[t])/\text{rint}+(v19xy19[t]-v19xy20[t])/\text{rint}+(v19xy19[t]-v19xy20[t])/\text{rint}, \)
v2xy1[0]==0, v3xy1[0]==0, v4xy1[0]==0, v5xy1[0]==0, v6xy1[0]==0,
\( \cdots \)
v15xy20[0]==0, v16xy20[0]==0, v17xy20[0]==0, v18xy20[0]==0, v19xy20[0]==0;
eqns=eqns/.{\text{cint} \rightarrow 3/10000000000000000, \text{roff} \rightarrow 1000000000000, \text{rint} \rightarrow 25, \text{v1xy1} \rightarrow 1, \text{v1xy20} \rightarrow 1, \text{v20xy1} \rightarrow 1, \text{v20xy20} \rightarrow 1, \text{delay} \rightarrow 0, \text{rise} \rightarrow 1/10000, \text{width} \rightarrow 5/10000, \text{fall} \rightarrow 1/10000, \text{periode} \rightarrow 10/10000, \text{lowval} \rightarrow 0, \text{highval} \rightarrow 1};

(* Parallel computing *)
sol:=ParallelTable[NDSolve[eqns,\{v2xy1, v3xy1, v4xy1, v5xy1, v6xy1\}, \{t,0,10\10/10000\},MaxStepSize->1/10000,MaxSteps->100000,\{\text{isrc},1/10000,8/10000,1/10000\}];
DistributeDefinitions[sol];

(* Evaluation and Plot graph *)
Table[Plot[\text{Evaluate[v10xy10[t]/sol[i] \[Rule] isrc]}\text{,\{t,0,10/10000\},\text{PlotLabel} \[Rule] \text{isrc}}\text{,\{i,1,Length[sol],1\}}]

---

**V Conclusions**

We proposed a technique for large scale power grid analysis. It starts with the modelling of a power grid as a system of ordinary differential equation systems, and uses the parallel computing method in Mathematica to obtain a solution of the system. Our technique was demonstrated by experimental results. They revealed that our technique is capable of circuit analysis same as that by SPICE, and also revealed the superiority of our technique under the condition of very large scale power grids. And they indicated the effectiveness of the parallel computing method in the power grid optimization design. In the future work, we have a plan to extend the technique to be able to deal with RLC interconnect network with parasitic inductive effects, and develop a new parallel numerical solver to achieve more high performance in Mathematica.

**References**


