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

Tamiyo Nakabayashi†, Keiji Nakabayashi‡, and Fujio Kako†
†Nara Women's University
Kitauoyahigashi-machi, Nara, Japan
{nakaba, kako}@ ics.nara-wu.ac.jp
‡WDB Corporation
2-3-13, Azuchi-machi, Chuo-ku, Osaka, Japan
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 Mathemtica 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 (The index i and j ranges as $i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n$)
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}=1/ R_{int})
R_{off}	Off resistance of active device [Ohm]
G_{off}	Off conductance of active device [1/Ohm] (G_{off}=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}=Max ($I_{i,j}$))
I_{low}	Low value of pulsed driving current [A] (I_{low}=Min ($I_{i,j}$))
Max ()	Maximum function
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).

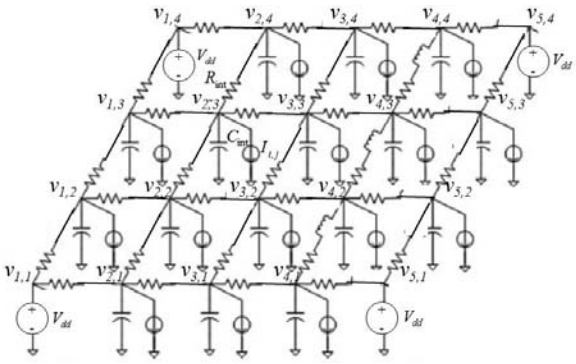


Fig. 2.1: Our power grid. Note that this example is a case of grid size 5x4 (m=5, n=4).

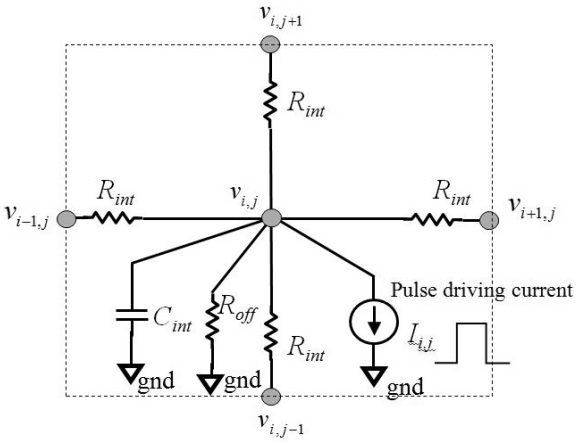


Fig. 2.2: KCL with respect to each node in the power grid.

$$C_{\text{int}}\frac{dv_{i,j}}{dt} = \frac{v_{i-1,j}-v_{i,j}}{R_{\text{int}}} + \frac{v_{i+1,j}-v_{i,j}}{R_{\text{int}}} + \frac{v_{i,j-1}-v_{i,j}}{R_{\text{int}}} + \frac{v_{i,j+1}-v_{i,j}}{R_{\text{int}}} - \frac{v_{i,j}}{R_{off}} - I_{ij} \tag{2-1}$$

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

$$C_{\text{int}}\frac{dv_{i,j}}{dt} = G_{\text{int}}\left(v_{i-1,j}-v_{i,j}\right) + G_{\text{int}}\left(v_{i+1,j}-v_{i,j}\right) + G_{\text{int}}\left(v_{i,j-1}-v_{i,j}\right) + G_{\text{int}}\left(v_{i,j+1}-v_{i,j}\right) - G_{off}v_{i,j} - I_{ij}$$

(2-2)

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).

$$C_M \frac{dv_M}{dt} = G_M v_M - I_M \qquad (2\text{-}3)$$

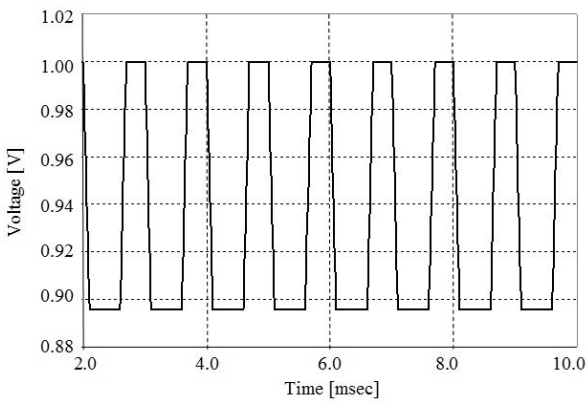
III Implementation into Mathemtica

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.

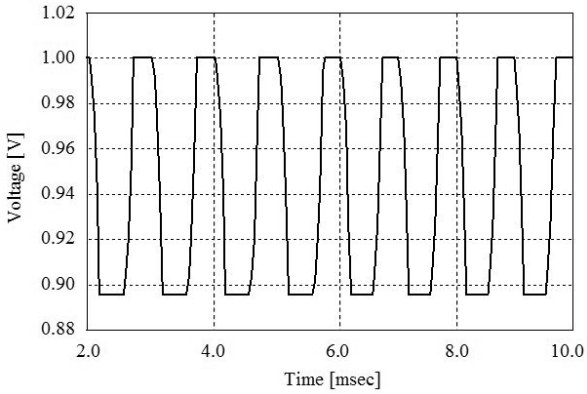
In the remainder of this paper, we consider the stochastic behaviour of the switching of the active devices on chip and place the driving current sources at random with using random numbers. And we assume the synchronous switching of the active devices in the power grid.

As our experimental verification, we set the values of the main parameters as following: I_{high}=2.0e-5 [A], I_{low}=0.0 [A], R_{int}=25.0 [Ohm], C_{int}=3e-16 [F], R_{off}=1.0e12 [Ohm], T_{sw}=1.0e-3 [sec], V_{dd}=1.0 [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.



(1) Our technique with Mathematica



(2) Ngspice

Figure 3.1: Voltage fluctuations at the node $v_{15,15}$.

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_{high}) takes the range from 1.0e-5 to 8.0e-5 which its step size is 1.0e-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 Mathemtica-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 Mathemtica. Table I shows the IR-drop value at the node $v_{10,10}$ for each pulsed driving current. Furthermore, the IR-drop values from Mathemtica 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.

Driving Current I_{high} [A]	(1) Mathemtica IRdrop [V]	(2) Ngspice IRdrop [V]	Reative Error ((1)-(2))/(2) [%]
1.00E-05	0.0194	0.0194	0.000
2.00E-05	0.0385	0.0387	-0.517
3.00E-05	0.0578	0.0588	-1.701
4.00E-05	0.0779	0.0774	0.646
5.00E-05	0.0963	0.0968	-0.517
6.00E-05	0.1169	0.1162	0.602
7.00E-05	0.1364	0.1355	0.664
8.00E-05	0.1541	0.1549	-0.516

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 x 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 Mathemtica and Ngspice becomes less in the case of large scale grid. Especially, Mathemaica 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.

m	n	(1) Mathemtica (Parallel Computing)		(2) Ngspice		(1) / (2) CPU Time Ratio [-]
		CPU Time [sec]	Ratio [-]	CPU Time	Ratio [-]	
10	10	4.92	1.00	0.29	1.00	17.1
20	20	29.36	5.97	1.68	5.83	17.5
30	30	72.90	14.82	6.98	24.22	10.4
40	40	144.68	29.41	19.14	66.44	7.6
50	50	259.48	52.75	55.34	192.14	4.7
75	75	650.80	132.30	274.46	953.00	2.4
100	100	1480.68	301.01	912.60	3168.75	1.6
200	200	8799.37	1788.85	7142.40	24800.00	1.2
300	300	24260.27	4930.95	NG	NG	—

```
(*** Pulse function ***)
fpulse[x_Real,delay_,rise_,width_,fall_,periode_,lowval_,highval_]=Which[x<=delay,lowval,Mod[x-delay,periode]>rise+width
+fall,lowval,Mod[x-delay,periode]>rise+width,(lowval-highval)/fall*(Mod[x-delay,periode]-rise-width)+highval,Mod[x-delay,p
eriode]>rise,highval,Mod[x-delay,periode]>0.0,(highval-lowval)/rise*(Mod[x-delay,periode])+lowval,Mod[x-delay,periode]<=0
.0,lowval];

(*** Size: 20 x 20 ***)
Clear [v2xy1, v3xy1, v4xy1, v5xy1, v6xy1,
                                        .
                                        .
                                        .
v15xy20, v16xy20, v17xy20, v18xy20, v19xy20] ;

(*** A system of ordinary differential equations ***)
eqns={cint v2xy1'[t]==-v2xy1[t]/roff+(v1xy1-v2xy1[t])/rint+(v3xy1[t]-v2xy1[t])/rint+(v2xy2[t]-v2xy1[t])/rint,
cint v3xy1'[t]==(-1) isrc
fpulse[t,0+delay,rise,width,fall,periode,lowval,highval]-v3xy1[t]/roff+(v2xy1[t]-v3xy1[t])/rint+(v4xy1[t]-v3xy1[t])/rint+(v3xy2[
t]-v3xy1[t])/rint,
cint v4xy1'[t]==(-1) isrc
fpulse[t,0+delay,rise,width,fall,periode,lowval,highval]-v4xy1[t]/roff+(v3xy1[t]-v4xy1[t])/rint+(v5xy1[t]-v4xy1[t])/rint+(v4xy2[
```

```
t]-v4xy1[t])/rint,
cint v5xy1'[t]==(-1) isrc fpulse[t,0+delay,rise,width,fall,periode,lowval,highval]-v5xy1[t]/roff+(v4xy1[t]-v5xy1[t])/rint+(v6xy1[t]-v5xy1[t])/rint+(v5xy2[t]-v5xy1[t])/rint,
cint v6xy1'[t]==(-1) isrc fpulse[t,0+delay,rise,width,fall,periode,lowval,highval]-v6xy1[t]/roff+(v5xy1[t]-v6xy1[t])/rint+(v7xy1[t]-v6xy1[t])/rint+(v6xy2[t]-v6xy1[t])/rint,
.
.
.
cint v15xy20'[t]==-v15xy20[t]/roff+(v14xy20[t]-v15xy20[t])/rint+(v16xy20[t]-v15xy20[t])/rint+(v15xy19[t]-v15xy20[t])/rint,
cint v16xy20'[t]==-v16xy20[t]/roff+(v15xy20[t]-v16xy20[t])/rint+(v17xy20[t]-v16xy20[t])/rint+(v16xy19[t]-v16xy20[t])/rint,
cint v17xy20'[t]==-v17xy20[t]/roff+(v16xy20[t]-v17xy20[t])/rint+(v18xy20[t]-v17xy20[t])/rint+(v17xy19[t]-v17xy20[t])/rint,
cint v18xy20'[t]==-v18xy20[t]/roff+(v17xy20[t]-v18xy20[t])/rint+(v19xy20[t]-v18xy20[t])/rint+(v18xy19[t]-v18xy20[t])/rint,
cint v19xy20'[t]==-v19xy20[t]/roff+(v18xy20[t]-v19xy20[t])/rint+(v20xy20-v19xy20[t])/rint+(v19xy19[t]-v19xy20[t])/rint,

v2xy1[0]==0, v3xy1[0]==0, v4xy1[0]==0, v5xy1[0]==0, v6xy1[0]==0,
.
.
.
v15xy20[0]==0, v16xy20[0]==0, v17xy20[0]==0, v18xy20[0]==0, v19xy20[0]==0};

eqns=eqns/.{cint->3/10000000000000000,roff->1000000000000,rint->25,v1xy1->1,v1xy20->1,v20xy1->1,v20xy20->1,delay->0,rise->1/10000,width->5/10000,fall->1/10000,periode->10/10000,lowval->0,highval->1};

(*** Parallel computing ***)
sol:=ParallelTable[NDSolve[eqns,{v2xy1, v3xy1, v4xy1, v5xy1, v6xy1
.
.
.
v15xy20, v16xy20, v17xy20, v18xy20, v19xy20},
{t,0,10 10/10000},MaxStepSize->1/10000,MaxSteps->1000000],{isrc,1/10000,8/10000,1/10000}];

DistributeDefinitions[sol];

(*** Evaluation and Plot graph ***)
Table[Plot[Evaluate[v10xy10[t]/.sol[[i]]],{t,0,10 10/10000},PlotLabel->isrc],{i,1,Length[sol],1}]
```

Fig 4.1: Our notebook source codes with parallel computing method in Mathematica

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 Mathemtica 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 Mathemtica.

References

[1] Y. Cai, L. Kang, J. Shi, X. Hong, and S. X.-D. Tan “Random Walk Guided Decap Embedding for Power/Ground Network Optimization,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems SYSTEMS—II: EXPRESS BRIEFS*, Vol. 55, No. 1, pp. 36-40, Jan. 2008.

[2] Circuit Design – Wolfram Demonstrations Project [Online]. Available:

http://demonstrations.wolfram.com/topic.html?limit=20&topic=Circuit+Design

[3] Z. Feng and P. Li,"Multigrid on GPU: Tackling Power Grid Analysis on parallel SIMT platforms," *Proc. IEEE Int'l Conf. on Computer-Aided Design*, San Jose, CA, Nov. 2008, pp. 647-654.

[4] Z. Feng and Z. Zeng, "Parallel multigrid preconditioning on graphics processing units (GPUs) for robust power grid analysis," *Proc. 47th Design Automation Conference, New York, NY,*, USA, June 2010, pp. 661-66.

[5] E.H.-A. Gerbracht, On the Engineers' New toolbox or How to Design Linear) Analog Circuits, Using Symbolic Analysis, Elementary Network Transformations, Computer Algebra System, *Proc. Int'l Workshop on Symbolic and Numerical Methods, Modeling and Applications to Circuit Design (SM2ACD'08)*, Erfurt, Germany, Oct. 2008, pp. 127-134.

[6] GRID simulation technology. [Online] Available: http://www.gridsimtech.org/products/nanorail.html

[7] A. Korobkov, "Power-grid (PG) analysis challenges for large microprocessor designs," *Proc. of the 2012 ACM international symposium on International Symposium on Physical Design (ISPD'12)*, New York, NY, USA, Mar. 2012, pp. 95-96.

[8] Mathematica Website [Online]. Available: http://www.wolfram.com/mathematica/new-in-8

[9] T. Nakabayashi, K. Nakabayashi, and F. Kako, "Application of Computer Algebra Approach to Solve Engineering Problems － A Case Study: Power Supply Stabilization Loop Circuit," *Proc. the 2012 American Conference on Applied Mathematics (AMERICAN-MATH'12)*, Harvard, Cambridge, Jan. 2012, pp. 228-233.

[10] T. Nakabayashi, K. Nakabayashi, and F. Kako, "A New Technique of Electro-thermal Modeling and Reliability Circuit Analysis of Power MOSFETs with Mathematica," *Proc. 3rd Int'l Conference on Mathematical Models for Engineering Science (MMES'12)*, France, Paris, Dec. 2012.

[11] Ngspice gEDA Website [Online]. Available: http://Ngspice.sourceforge.net/presentation.html

[12] Parallel Computing – Wolfram Mathematica [Online]. http://reference.wolfram.com/mathematica/guide/ParallelComputing.html

[13] A. Ramalingam, G. V. Devarayanadurg, and D. Z. Pan, "Accurate Power Grid Analysis with Behavioral Transistor Network Modeling," *Proc. of the 2007 international symposium on Physical design (ISPD'07)*, New York, NY, USA, Mar. 2007, pp. 43-50.

[14] J. Shi, Y. Cai, W. Hou, L. Ma, S. X.-D. Tan, P.-H. Ho, and X. Wang, "GPU friendly Fast Poisson Solver for Structured Power Grid Network Analysis," *Proc. 46th Design Automation Conference*, San Francisco, CA, USA, July 2009, pp. 178-183.

[15] C. Zhuo, J. Hu, M. Zhao, and K. Chen, "Power Grid Analysis and Optimization Using Algebraic Multigrid," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 27, No. 4, pp. 738-751, Apr. 2008.