Fast Flip-chip Power Grid Analysis Via Locality and Grid Shells

Eli Chiprout
Strategic CAD, Intel labs, Chandler AZ, (eli.chiprout@intel.com)

Abstract

Full-chip power grid analysis is time consuming. Several techniques have been proposed to tackle the problem but typically they deal with the power grid as a whole or partition at unnatural boundaries. Using a locality effect under flip-chip packaging, we propose a natural partitioning approach based on overlapping power grid "shells". The technique makes more efficient any previous simulation techniques that are polynomial in grid size. It is also parallelizable and therefore extremely fast. Using complete partitions gives no loss of accuracy compared to a full matrix solution, while lesser partitions are conservative for droop and current. Results on a recent Pentium® microprocessor design show excellent speed and accuracy.

INTRODUCTION

In recent years, power and power delivery has become a major concern for chip design. New designs are drawing more active power, while technology scaling has yielded unwanted static leakage power. At the same time, increasing process resistivity has made the delivery of power on the chip a challenging design problem. On-chip power delivery networks which bring power from the main source to the transistors, typically consist of grids and have to be carefully designed to meet the new constraints [1].

Increasing numbers of devices, more metal layers and higher metal via densities have meant that the electrical models used to represent the power grid have grown in size exponentially. Typically, a full-chip power grid electrical model contains tens of millions of elements and nodes and is dense, making it prohibitive to simulate. The normal cost of simulation is at least one matrix inversion of a large dense system.

Recently, several new techniques have been proposed to tackle this problem. Because the models of the grid are typically linear, Padé type model-reduction strategies have been proposed [2], including using Krylov subspace techniques [3][4]. However, these have not been as effective because they work on the total system description which contains a large number of interface nodes and a high density of coupling. A newer approach has been to reduce the power grid model via multi-grid reduction [5]. In this technique the power grid is approximated by a smaller but less accurate grid, solved in the reduced space, and the solution projected back on the original grid. Still another approach has been to simulate the entire grid using random walk [6]. However, these techniques attempt to analyze the power grid as a whole without partitioning. They also may suffer from some accuracy loss due to the granularity of the approximation or the iterations required for convergence. Another technique has been proposed for partitioning the power grid [7], but the partitioning scheme is complex and not along the natural boundaries outlined in this paper.

With new advances in technology, C4 flip-chip packaging has come to prominence. Fig. 1 shows a flip-chip die with C4 bumps. Flip-chip allows power to be delivered at distinct and frequent C4 (Controlled Collapse Chip Connection) bumps above the power grid rather than at the sides of the chip as in the case of wire-bond technology. Flip-chip has become the technology of choice for high-performance power delivery designs [8]. Since flip-chip packages consist of almost ideal current sources/sinks distributed frequently and uniformly above the chip, they cause a "locality" effect. This means that the current for a particular area is generally drawn from the closest C4's. This was also noted in [6], whose technique appears to speed up in the presence of such effects.

In this paper, we present a "shell"-based partitioning technique that specifically takes advantage of such locality effects in flip-chip technologies. This technique allows us to partition the entire power grid into standard, uniform-size templates and simulate each sequentially or in parallel. Since matrix inversion grows polynomially with the size of the problem, simulation time is dramatically reduced compared to a full matrix solution. Further, any known simulation techniques may be used on the partitions. Parallelized simulation of the partitions is also enabled and has allowed us to simulate a full-chip microprocessor power grid, with the top 4 metals, in less than 30 minutes. If the standard partitioning template is used, there is no loss of accuracy compared to a full matrix solution. Smaller partitions yield reasonable approximations which are shown to be conservative for droop and current. This technique is completely general and may be applied to a DC or an RLC transient linear grid description. However, in the paper, we limit our description to a DC solution.

This paper is organized as follows: First we cover the concept of spatial locality in detail and show, qualitatively, how it behaves. Then, numerically, we show the impact of different power grid shells around a source. We then cover a partitioning scheme that is effective using these results. Finally, we show how parallel results can be combined in a total power grid simulation.

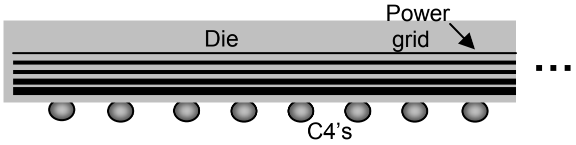


Figure 1. Flip-chip die showing C4 bump connections

SPATIAL LOCALITY

Flip-chip technology with C4 bumps provides a strong source of direct current at uniformly distributed points directly above the power distribution network. This is in comparison with wire bond technology which delivers power only at the edge of the die. Trends have led to more reliance on flip-chip technology for high performance designs and, for certain technologies, to increase the density of C4's.

When we observe the voltage droop around C4 bumps for a typical power grid, we note that these bumps act like poles holding up a tent where the physical droop of the tent represents the voltage droop and the top of the poles represent ideal sources of voltage directly above the bumps. In fact the sources are not perfectly ideal in that there is also the effect of package droop parasitics from the voltage regulation module (VRM). However, these parasitics are much larger than on die and provide a low frequency effect which is not dealt with in this paper. In the case of DC analysis, we only start with the voltages assumed to be delivered to the C4's.

A C4 bump acts as a low impedance path for local die currents to flow off-chip and therefore naturally attract currents to the nearest one. This effect is visible in the electrical models. To see this effect we simulate a Vcc resistive grid of a Pentium® microprocessor design within an area bound by 8x8 C4 bumps, of approximately 2500µ x 1600µ size, in a 7-metal layer process. Our grid was modeled electrically from M2 to M7, with ideal Vcc sources attached to M7 at the location of the C4 bumps. In the middle of this area we inserted a 10µ x10µ DC current source of 0.25A, attached to M2, and analyzed the grid both for voltage droop and current flows.

We observe in the contour plot of the voltage droop on M2 in Fig. 2 (each contour represents an equi-potential set of droop points, with the lowest droop in the middle) that the main droop is localized, with little or no droop far away. We also note that the contours conform themselves quit clearly to the C4 bump pattern, showing that these have a significant effect on the formation of the droop extension. The orientation of the contours in the horizontal direction is simply a reflection of the grid underneath with some metals providing a lower impedance horizontal path as compared to the vertical.

We can see things even more clearly in a contour plot of the currents magnitudes (highest currents in the middle) as they migrate their way up from the sources on M2 to the C4 bumps above M7, on the vias from M4

0-7803-8702-3/04/$20.00 ©2004 IEEE.

to M5 (Fig. 3). We note that the currents, as the voltages, show a tendency to stay localized and form a pattern conforming to the C4 bumps.

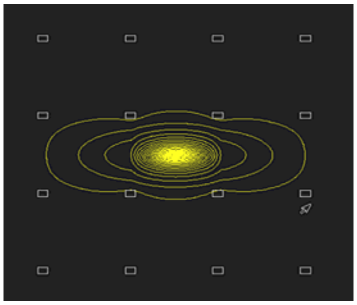


Figure 2. M2 voltage contour map of a Pentium® grid within a 4x4 array of C4 bumps.

These empirical results are similar for any kind of grid under a regular C4 distribution and are intuitive. They indicate that for any given source, only a small part of the grid is affected by a source perturbation. Since little or no perturbation reaches farther sections of the grid, and the C4 pattern is uniform, it is implied that one can remove grid metal segments and C4's which are beyond the perturbed grid, without effect on the simulation model.

Another way to think about it is that the grid extending outwards from the source represents an input-impedance to that source. Adding more metal segments and C4's to the grid is an effect of diminishing returns since, after a point, these would not change the input impedance as seen from the source.

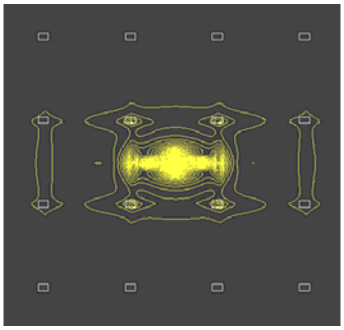


Figure 3. M4-M5 via Currents.

Note that this effect has nothing to do with the uniformity of the power grid but is only an artifact of the regularity and uniformity of the C4 bumps. A power grid may be uniform or non-uniform and this effect would still occur.

GRID SIZE

Given this effect, one needs to know how much of the power grid may be removed while continuing to give the same voltage and current results for a source perturbation. In order to answer this question, a "shell" with an 8x8 C4 pattern and all the grid details immediately underneath in a "cookie-cut" pattern was used. In the middle, attached to M2, was placed a point source of 0.25A (chosen to give approximately 15% worst case droop), and the network was simulated with a Vcc of 1.0V. We performed the same analysis on a 6X6, 4x4 and a 2x2 C4 pattern shells and successively smaller grids underneath, with the same middle source. The numerical comparison of the worst case voltage value on M2 is given in Table 1:

Shells greater than 8x8 displayed no practical change in the results. Taking the 8x8 droop to be the accurate one, the errors for these various models are displayed in the last column of Table 1. Note the quick convergence of the results, with the 4x4 grid giving the correct results for all practical purposes.

Table 1. M2 min voltage for a point source (Vcc=1.0)

C4s	Max droop	% error
2x2	0.848359V	0.16
4x4	0.849702V	2.58e-05
6x6	0.849723V	1.18e-06
8x8	0.849724V	0.0

In order to cause the worst case for the locality effect, the same grids were taken, but attached to a uniform rectangular current source of 5.0 Amps which completely covered the middle 2x2 C4 area. The uniform source was divided evenly into every via descending from M2 into the source. Although giving a different range of error, the patterns showed quick convergence once again (Table 2)

Table 2. M2 min voltage with a uniform source.

C4s	Max droop	%error
2x2	0.797748V	2.58
4x4	0.818549V	4.10e-4
6x6	0.818879V	7.33e-6
8x8	0.818885V	0.0

We note that both in tables 1 and 2, the min voltage was just above the source in the middle 2x2 section of grid. However, the current is not completely confined to this 2x2 section (except in the 2x2 case) and there is always some droop further out. Using the worse uniform current source of 5.0 amps, we examine the supply voltage at the outer edge of each successive shell, outside of the middle 2x2 section, on M2, and we find that droop quickly dissipates with distance (2nd column, Table 3).

Table 3.Voltage supply and currents at the edge of the grid for a uniform source.

C4s	M2 Vmax	C4 Imin
4x4	0.999899V	7.848463e-8A
6x6	0.999999V	3.270416e-8A
8x8	1.000000V	2.952670e-9A

Finally, we also examine the worst case via currents in both the above cases. These are shown in Table 4 and also demonstrate a quick convergence rate as the shells increase in size, both in the point source and in the uniform source case. Similarly, in the 3rd column of Table 3, the C4 via currents (the largest of the via currents), at the edge of the shell and outside of 2x2 section, quickly diminish with shell size. This result is representative of grids under a C4 distribution. Depending on the density of the grid as compared to the C4 sources, and the strength of the current sources, it may be necessary to extend a grid size to 6x6 or perhaps even 8x8, but the principle of locality remains the same.

Table 4. M7 max current at C4 with a point uniform sources

C4s	Point source Imax	Uniform source Imax
2x2	7.302732e-2A	1.260141A
4x4	6.044550e-2A	1.051302A
6x6	6.010330e-2A	1.046146A
8x8	6.009613e-2A	1.046038A

One important point to note in the tables is that with smaller grid shells the results are always conservative. That is, the voltage droop is greater (the voltage supply smaller) and the currents are larger than a larger shell or the complete solution. This may be shown by a simple argument. The topology and the source are the same for the shell and the full grid in the area defined by the shell. However, the shell, as opposed to the full grid, confines some currents from spreading outside the boundary and therefore these are additive to the currents which are normally confined in the space of the shell, since all currents must exit in the shell and exit at the C4's above the shell.

PARTITIONING

Using the above principle, we note 3 points:

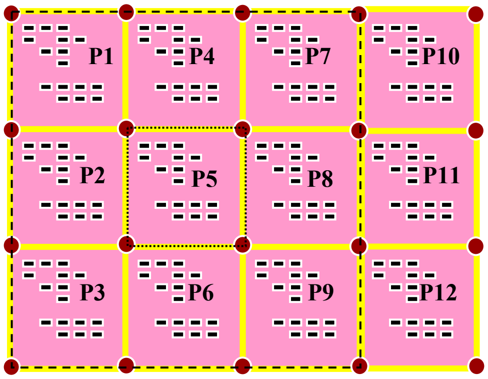
1. The grid as a whole is a linear network and the contribution of each current sources to the droop may be accumulated linearly (comments about nonlinear sources are added below).
2. For any individual current source, only a small subset of the grid (and C4's) is needed in order to determine the source's contribution to the droop.

These principles yield a grid shell partitioning scheme.

- The sources are partitioned in non-overlapping tiles, covering the analysis area.
- The grid is partitioned in overlapping shells, one for each source partition.

We also observe the following. The source partitions may include multiple sources that are simulated together. The purpose of a grid shell is to "cover" a source partition; that is, provide enough power grid, so that the source currents will flow as they would in a complete power grid. Several methods are possible in order to deduce how extensive a grid is necessary for the source partitions. A basic one will be outlined in this paper. Note that the grid partitions are distinct from the source partitions and do not correspond to the source partitions in terms of area.

Figure 4. Current source partitions with P5 source partition highlighted, with its power grid shell, P1-P9.



For example, Fig. 4 illustrates a small area contained in 3x4 C4's (large dots). Uniform area source partitioning is illustrated by the 2x2 source partitions P1-P12, each under a 2x2 C4 area. (A uniform area partitioning is simplest to implement). The grid shell partitions surround the current partitions. In this example, the P5 source partition is highlighted together with its shell grid partition which includes P1-P9.

Note that this is not a necessary partition size. The general idea is to partition as many sources as are practical to simulate in one run together with their necessary power grid shell. This will depend on the compute server configuration, the CPU times desired, the error accrued and complexity of the assembly of information. We have typically used all sources under a 3x4 C4 segment since we found this to be a good compromise between simulation time, assembly time and granularity.

SELECTION OF THE GRID SHELL SIZE

The purpose of a grid shell partition is to provide enough surrounding power grid wires and C4 outlets for the currents so that they will flow no (or little) differently than they would if all the power grid metals and C4's had been included. The extent of the droop effect will depend on the strength of the currents, the resistivity of the grid or the vias and the error tolerance required. Strong currents will tend to have more important residual effects at the edge of small shells. More resistive grids will tend to have more droop, and therefore have a tendency to have less locality.

Thus, the most resistive grid, one in which the minimum number and width of possible metals exist, will be most resistive and therefore tend to spread the droop most. If entire sections of grid are missing, then the current will spread widely to flow around the open areas, but it is assumed in this paper (and actualized in practical designs because of electro-migration and power deliver constraints) that such grids don't exist in practice. This implies that the path impedance to a physically nearer C4 will be significantly less than to a farther C4. In either case, we describe a check to determine if the grid shell partition has produced the correct simulation results and if not, to substitute a larger grid shell size. We have not encountered these issues on our designs.

While there are several potential algorithms that are open to research, we outline a simple and practical technique to determine a uniform shell partition for a uniform current partition in practical cases. The main point to remember is that one can easily overcompensate by providing a slightly bigger grid shell than required in the estimated worst case. Since a large problem is being partitioned, there is already significant savings in terms of full matrix inversion and memory required. Further, if partitions are simulated in parallel, a small extra cost for each partition does not present a large impact either.

Given uniform current partitions, an offline simulation may be performed with the worst case potential grid/source combination, causing the least locality and most droop, and the same partition size re-used for the less severe combinations. The worst case DC currents are obtained by taking either the most power hungry area of the die or a max of the currents for any Cartesian position, across all partitions.

For determining a worst case grid, we either have a uniform grid or a non-uniform one. In the case of uniformity across grid shells, there is no need for a worst case shell, since all partitions will behave the same. In the case of non-uniformity of grid shells, the worst case grid is the most resistive grid for any partition. This may be obtained by taking a uniform grid with a wider pitch than any partition, and at minimum width for each wire. Another, more accurate approach uses a synthesized simpler electrical model of the partition that gives a measure of impedance of the grid in various directions and therefore of locality. Such a simple equivalent circuit may by generated by combining metals/vias in parallel. Space prevents discussion of the details in this paper.

With the worse case source and worse case grid, enough power grid and C4's are allocated to the current source partition to yield a selected error bound. Typically for our designs, a ring of single C4's extending outwards from the current partition (e.g. Fig. 4) was more than enough.

The shell partition is large enough, if the droops and currents are zero (or an error tolerance) at the outer boundaries. This shows that the grid shell is larger than the extent of the locality effects. This is also a check for error, for every simulation should produce zero (or small) effects at the boundary. As we have seen in our grid example, a 4x4 grid with a 2x2 current partition is generally sufficient but it could vary in the case of much more resistive grids. It is effective in early design, when the accuracy of design data is low, to select a grid partition area that is equivalent to a source partition area for faster simulation time but less accuracy.

ASSEMBLY OF THE RESULTS

In the linear source model case, each partition may be simulated separately. Once the results are obtained, assembly of the results is required. The results of a grid partition with its sources are accumulated to the result of any other grid partition simulation that overlaps the current one. For example, when simulating current partition P5 (Fig. 4), the voltage and current results in the metals over P5 are obtained, and to these are added the simulation results of current partitions P1-P4, P6-P9 (with their grid partitions – not shown), but not source partitions P10-P12 since that is outside of their locality, determined to be a ring of C4's.

The assembly above assumes a linear power grid model with linear sources. Typically, if the effect of nonlinear sources is required, one has to run an iterative simulation between the large linear power grid model and a nonlinear simulation of the impact of the droop on the devices. However we note that this technique offers a good compromise between the two. Most of the impact of sources is confined to a source partition where they cause the greatest currents and droop. Droop and current outside of these areas will be smaller. Given the smaller impact, these residuals may be added linearly to the impacted nonlinear source partitions, as a first order approximation. A small scale iterative algorithm may also be set up to obtain the nonlinear results, however we avoid the full discussion in this paper.

ANALYSIS

This technique partitions a large design but adds the additional constraint of requiring the simulation of extra nodes per partition and assembly of results. The assembly of results is simply a set of additions and does not consume much. We therefore analyze the speedup of partition simulations over a full solution.

Assume that there are N_A nodes per a given area A of the grid. The total number of nodes is then $A * N_A$. If there are k source partitions, then each source partition has an area of A/k. If the grid partition extends the source partition by an area of a all around the partition, then the total number of nodes of a grid partition will be $(A/k+a)*N_A$. Given a simulation algorithm that grows with the power of p in the number of nodes in a partition, the original simulation problem is proportional to $(A*N_A)^p$ while the partitioned simulation, in serial, is proportional to $k((A/k+a)*N_A)^p$. If a has a fixed relationship to A/k, then $a = n * A/k$ the ratio of the total smaller simulations to the larger one will be approximately: $k(n+1/k)^p$ with $n << k$. Parallel simulation is not considered here.

EXAMPLES

The main dependencies for CPU time will be the size of the partition, the number of metals in a partition, and number of partitions that can be processed in parallel. This will vary from design to design and from site to site. What we can show here is an example of how much CPU time one partition consumes and then present the practical results for a full-chip run.

For the dense power grid given above (with metals 2-7) we show the growth in simulation time with the growth of shell size. The simulation was run on an Intel Xeon CPU 2.8GHz with 512 L2 Cache running Linux 2.4.9.

Notice that this simulation is done on an internal simulator, but any technique may be used to simulate an individual partition including other published techniques mentioned in [3]-[7]. In addition to the simulation time, there is the time associated with the assembly of the results. Given a large partition, such as a 3x4 current partition and 5x6 "shell", the areas of overlap are few and addition, used for the accumulating the contribution of individual partitions is not an expensive operation, so the cost is negligible.

Table 5. Growth in CPU time with growth of "shell" size for each partition using metals 2-7.

C4s	metals	vias	resistors	CPU (s)
2x2	3739	12568	33977	3.9
4x4	28547	98650	267415	68.9
6x6	76,595	266,316	722365	451.1
8x8	147,883	515,566	1,398,827	1628

The top 4 metal layers of a power grid of a complete microprocessor design in a 90nm process were analyzed. The grid was modeled resistively, and a detailed floor-plan of approximately 2,000,000 floorplanned DC current sources was also generated. The grid was partitioned by 5x4 current partitions and 7x6 grid "shells", giving a shell buffer of 2x2 width all around the current partition. This yielded 18 x 13 = 234 partitions. The edge of the chip did not need any shell buffer.

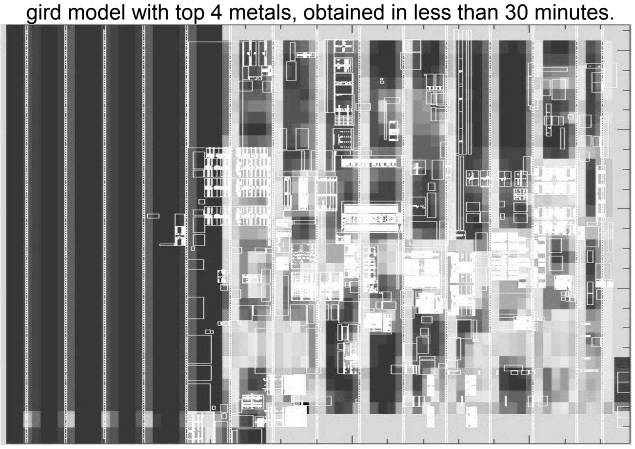
Each partition was slightly different but contained approximately 1000 metals, 50,000 vias, 150,000 resistors, 100-500 currents, and 100,000 nodes.

In our case, with a large design center, there were enough machines to simulate all the partitions in parallel. Each partition took 25 minutes to complete and the entire run was finished in approximately 30 minutes, with the resultant full-chip top-level currents being displayed in Fig. 5. The results were accurate within the range described Section 2 for a shell buffer of 2x2. The final results assembled the voltages of 18,484,366 nodes and 45,110,490 via and metal currents.

CONCLUSION

A new partitioning scheme has been proposed for a power grid in a flip-chip package. Using locality, it is based on power grid "shells" and current partitions and allows one to efficiently simulate the entire problem with any known solver and in parallel. This technique has enabled a full chip DC power grid solution of a large microprocessor design in less than 30. For simplicity, we illustrated using a DC example, but this technique is applicable to dynamic and even power grids with nonlinear sources, that attach to a flip-chip package, with the only variation being the size of the shell needed. Future research will concentrate on more efficient early estimates of locality and application to dynamic models.

Figure 5. Currents of a complete Pentium® microprocessor power gird model with top 4 metals, obtained in less than 30 minutes.



ACKNOWLEDGMENTS

The author wishes to thank Thomas Toms, Steven J. Morris and Sandip Kundu for helpful in collaboration and discussions.

REFERENCES

[1] H. H. Chen and D. D. Ling, "Power supply noise analysis methodology for deep-submicron VLSI chip design," Proceedings of the ACM/IEEE Design Automation Conference, pp. 638-643, 1997.

[2] E. Chiprout and T. V. Nguyen, "Power analysis of large interconnect grids with multiple sources using model reduction", Proceedings European Conference on Circuit Theory and Design, Stressa, Italy, Sept. 1999.

[3] J. M. Wang and T.V. Nguyen, "Extended Krylov subspace method for reduced order analysis of linear circuits with multiple sources", Proceedings ACM/IEEE Design Automation Conference, pp. 247 – 252, 2000.

[4] T. Chen and C. C. Chen, "Efficient large-scale power grid analysis based on preconditioned Krylov-subspace iterative methods," Proceedings of the ACM/IEEE Design Automation Conference, pp. 559-562, 2001.

[5] J. Kozhaya, S. R. Nassif, and F. N. Najm, "A multigrid-like technique for power grid analysis," IEEE Trans. On Computer-Aided Design, vol. 21, no. 10, pp. 1148-1160, 2002.

[6] H. Qian, S. R. Nassif, and S. S. Sapatnekar, "Random walks in a supply network," Proceedings of the ACM/IEEE Design Automation Conference, pp. 93-98, 2003.

[7] A. Dharchoudhury, R. Panda, D. Blaauw, R. Vaidyanathan, B. Tutuianu and D. Bearden, "Design and analysis of power distribution networks in PowerPCTM microprocessors," Proceedings of the ACM/IEEE Design Automation Conference, pp. 738-743, 1998.

[8] Dietrich Tönnies, "A Review and Trends in Flip-Chip Technology", Chip scale review, April 2004.