Distributed Port Assignment for Extraction of Power Delivery Networks

Kan Xu, Eby G. Friedman

Department of Electrical and Computer Engineering
University of Rochester
Rochester, New York
(kxu8, friedman)@ece.rochester.edu

Mikhail Popovich, Gregory Sizikov

Google Inc.
Mountain View, California
(mpopovich, gsizikov)@google.com

***Abstract*—The stringent requirements of power noise on complex multi-domain power delivery networks (PDN), and the complicated relationship between signal integrity and power integrity (PI) have led to an ever challenging PI sign-off process. A lumped PDN model is widely used, where the power network is treated as a two-port network with the impedances extracted by an electromagnetic solver. A distributed model of the power network is however preferred during a PI sign-off flow, providing a more accurate circuit model for time domain simulations. Hundreds or even thousands of ports need to be properly evaluated during the PDN extraction process, which can be computationally expensive and error prone. A Python tool is described here to enable a fast and configurable process for distributed port assignment during the PDN extraction process. An enhanced automation flow, integrated with the Python tool, has also been developed to support early power network exploration. In one case study, a 360X speedup in the port assignment process is achieved while revealing a high risk power network within the package. The proposed automation flow is versatile and highly adaptive for different power network topologies.**

***Index Terms*—Power integrity sign-off, distributed power delivery network, PDN extraction, electronic design automation.**

I. Introduction

A power delivery network (PDN) is a complex and hierarchical system, spanning from the point-of-load voltage regulator to the on-chip load, including the printed circuit board (PCB), package, and integrated circuit (IC) [1]. In high performance computing, processors composed of a large number of cores providing high throughput are highly desirable, leading to greater on-chip current demand [2]. To balance the need for higher performance with low power in systems-on-chip (SoC), advanced technology nodes and multi-voltage power networks are utilized, leading to a reduction in power noise margin and less robust power planes for each voltage domain [3]. Power integrity (PI) sign-off in modern high performance processors and SoC has therefore become quite challenging [4], increasing the turnaround time for redesign, delaying the overall product development process.

One of the key steps during the PI sign-off process is PDN extraction, where electromagnetic (EM) solvers [5, 6] electrically characterize the power network within the package and PCB for subsequent analysis. A two-port network model is frequently utilized in package or PCB extraction, creating a one-dimensional model of the power delivery network [7]. In package level PDN extraction, for example, the bottom power and ground (P/G) balls and the top P/G bumps are individually grouped to form an electrical port, as illustrated in Fig. 1. In this way, the power network of the package is extracted as a two-port network. The extracted impedance is subsequently included within a lumped model of the power network for time domain analysis [8], as illustrated in Fig. 2a.

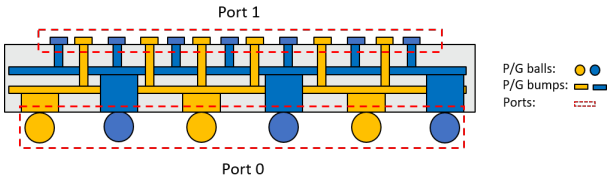


Fig. 1: Cross-sectional view of a simplified package power network with a lumped port model.

Although a lumped model of a power network is widely used, the complex and hierarchical structure of a power network exhibits a distributed nature, including dozens of power planes, thousands of traces, vias, and bumps, and millions of nodes. In a package power network, the P/G bumps of a specific voltage domain, for example, the power network for a memory, may not be evenly distributed across the entire package due to the sharing of metal resources with other voltage domains and signals. The impedance of the P/G bumps at different locations may therefore vary significantly regardless of manufacturing variation. In this case, a distributed n-port model, as illustrated in Fig. 2b, should be utilized for PDN extraction to achieve more realistic results. Port assignment is however often conducted manually within an EM solver [5, 6], which includes selecting bumps or areas, configuring the power and ground net, and assigning the characteristic impedance for each port. This process alone can be quite

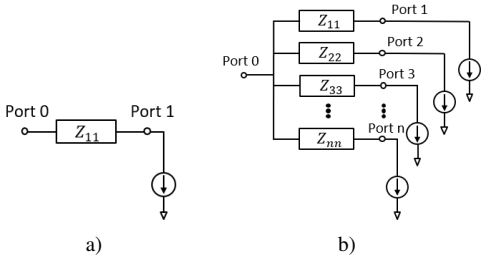


Fig. 2: Circuit model of a power network, where Z_{nn} is the impedance of the bump side to the ball side. a) Two-port model, and b) distributed port model.

This research is supported by Google Inc.

978-1-7281-3320-1/20/$31.00 ©2020 IEEE

computationally expensive and error prone with hundreds or even thousands of ports within a system. A Python tool, Port Assigner, is therefore described in this paper to support fast port setup for n-port PDN extraction. In addition to this tool, an enhanced automation flow, PDN checker, has been developed to capture information describing a power network and to speed up the PDN extraction process.

The rest of the paper is organized as follows. The Python tool along with the enhanced automation flow are described in Section II. A case study of applying the PDN checker to a practical package is demonstrated in Section III, where port assignment, S parameter to Z parameter transformation, and impedance variance analysis are fully automated. Some conclusions are offered in Section IV.

II. Distributed port assignment and PDN checker

A typical system level PI sign-off flow consists of two stages: extraction of the power network impedances in the frequency domain, and time domain analysis, incorporating a voltage regulator and the on-chip load [9]. Multiple commercial electronic design automation tools are included in the PDN extraction stage. The package and/or PCB files are first developed in Cadence Allegro [10]. The electrical ports are assigned at the pinouts of the package and/or PCB to form an n-port network. S parameter characterization of this n-port network is subsequently extracted with an EM solver [5, 6]. ADS [11] transforms the S parameters into Z parameters to characterize the power network impedance. The target impedance Z_{target} is determined to evaluate the robustness of the power network across a wide frequency range. Z_{target} is

$$Z_{PDN} < Z_{target} = \frac{V_{DD} \cdot ripple}{\Delta I}, \tag{1}$$

where V_{DD} is the supply voltage of a specific voltage domain, $ripple$ is the maximum allowed ripple, ΔI is the maximum transient current within the system, and Z_{PDN} is the extracted impedance of the power network.

The procedure of Port Assigner in developing a mesh-based port assignment is described in Pseudocode 1. The Port Assigner is compatible with both Sigrity PowerSI and Ansys SIwave [5, 6], two industrial PDN extraction tools. In PowerSI, for example, the structural and electrical information characterizing the package and/or PCB is written within a package file PKG_{old}, which is the input file for Port Assigner. Other inputs include the target voltage domain $rail$, the P/G bumps used in $rail$, $padstack$, the size of the mesh N, and the vertices which describe the region of the mesh (V_{bot}, V_{top}). The output is the new package file with ports assigned, PKG_{new}. Port Assigner initially parses the original package file and determines the coordinates of each target bump. An NxN mesh is formed based on N and (V_{bot}, V_{top}), dividing the package into N^2 tiles. Based on the coordinates of the bumps, Port Assigner groups adjacent bumps into ports to form an n-port network. Based on the coordinates of the bumps and tiles, the bumps located within a certain tile are grouped into one port, producing an N^2+1 port model of a power network, including a port added at the bottom of the package.

A Python enhanced automation flow is developed with Port Assigner to characterize the power network and speed up the PDN extraction process. This enhanced automation flow is referred to here as the PDN checker. Three stages, port assignment, S-to-Z transformation, and illustration, are included in the PDN checker, as highlighted by the gray area in Fig. 3. Note that each stage can operate independently as a functional tool or can work together within this automation flow. During the first stage, Port Assigner performs port assignment and environmental setup. Along with Port Assigner, a configuration file is also generated by the PDN checker, where the port assignment specifications are determined. Parameters such as the name of the voltage domain, target frequency, and reference impedance can also be specified in the configuration file.

The PDN checker invokes the EM solver and performs PDN extraction. The second stage begins once PDN extraction is completed and the S parameter file is available. In this stage, the PDN checker automatically fetches the S parameter file and conducts an S-to-Z transformation, providing more intuitive results for PI evaluation. The S-to-Z transformation process in a two-port network has been proposed [12], and extended to an n-port network [13] as in (2),

$$\boldsymbol{Z} = \boldsymbol{G_0}^{-1} \cdot (\boldsymbol{I} - \boldsymbol{S})^{-1} \cdot (\boldsymbol{S}\boldsymbol{Z_0} + \boldsymbol{Z_0}^*) \cdot \boldsymbol{G_0}, \tag{2}$$

where

$$\boldsymbol{G_0} = \begin{bmatrix} \frac{1}{\sqrt{|\operatorname{Re}\{Z_{0,1}\}|}} & 0 & \dots & 0 \\ 0 & \frac{1}{\sqrt{|\operatorname{Re}\{Z_{0,2}\}|}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{\sqrt{|\operatorname{Re}\{Z_{0,n}\}|}} \end{bmatrix}, \tag{3}$$

and

$$\boldsymbol{Z_0} = \begin{bmatrix} Z_{0,1} & 0 & \dots & 0 \\ 0 & Z_{0,2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Z_{0,n} \end{bmatrix}. \tag{4}$$

$Z_{0,n}$ is the reference impedance of port n, $\boldsymbol{S}$ is the n-port S parameter matrix, $\boldsymbol{I}$ is the identity matrix, and $\boldsymbol{Z}$ is the n-port Z parameter matrix. The Z parameter matrix describing the power network is subsequently decoupled into the resistance and inductance of each port at the target frequency, preparing an impedance variance map for stage three. This stage bypasses ADS, automatically converting the S-to-Z parameters for each port.

During the third stage, the PDN checker determines the resistance and inductance of each port, producing an impedance map. Note that the resistance and inductance match the physical location of the ports on the package and/or PCB, providing physical intuition characterizing the power network. PI evaluation is subsequently conducted based on the magnitude and

Pseudocode 1: Port Assigner

Input: PKG_{old}, $rail$, $padstack$, (V_{bot}, V_{top}), N
Output: PKG_{new}

$lines \leftarrow parse(PKG_{old})$
for each $line$ in PKG_{old} **do**
  if $padstack$ **AND** $rail \in line_i$ **then**
    $pinouts \in (pin_i, vertex_i)$
  end if
end for
Configure $mesh$ based on N and (V_{bot}, V_{top})
Configure vertices of each $tile$ within $N \times N$ mesh
for each $vertex$ in $pinouts$ **do**
  for each $tile$ in $mesh$ **do**
    if $vertex_i$ is in $tile_j$ **then**
      $port_j \in pin_i$
    end if
  end for
end for
for each entry in $port$ **do**
  Write $port_i$ back to PKG_{old}
end for
$PKG_{new} \leftarrow PKG_{old}$

location of the impedance. If, for example, the impedance is significantly greater than the target specification, the power network of the package and/or PCB can be redesigned. If the impedance is not too high but rather in a sensitive location, the target region can be changed.

III. Case Study

A case study is described in this section to demonstrate the flow and advantages of the PDN checker in a practical PI sign-off process. A power network of a package with multiple voltage domains is the input of the PDN checker, where a mesh-based port assignment scheme is utilized in this demonstration system. The process of applying the PDN checker to a practical power network, and capturing the hotspots within the power network is described in Subsection III-A. The capability of enabling faster turnaround time is described in Subsection III-B.

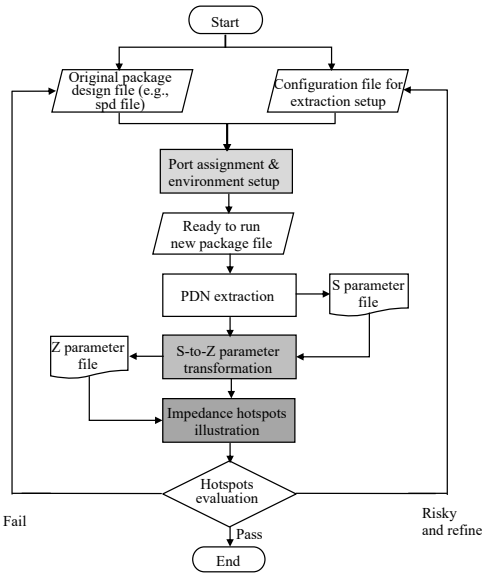


Fig. 3: Proposed automation flow for PDN extraction.

A. Hotspots within power network

The PDN checker is compatible with both PowerSI and SIwave [5, 6], where Sigrity PowerSI [6] is utilized in this demonstration system. The target voltage domain and frequency in this demonstration system are, respectively, VDD_SRAM and 100 MHz. The P/G pinouts on the ball side are grouped into one port. Based on the size and distribution of the P/G bumps of the package, a 15x15 mesh distributed port assignment is utilized on the bump side. The reference impedance of the port is set to 0.1 ohms for high accuracy PDN extraction. The system requires less than 30 seconds to complete the port assignment process with the PDN checker. As compared with a manual port assignment, which requires approximately three hours to assign 225 ports, a 360X speedup is achieved.

As illustrated in Fig. 4, at the end of the PDN checker flow, an impedance map is generated, capturing the hotspots within the power network. Each bar shown in Fig. 4 represents the resistance or inductance of each port on the bump side of the package. The height of the bar implies the magnitude of the resistance or inductance, from the bump side to the ball side. The dark blue bars represent invalid ports. The location of the bars on the impedance map reflects the physical location of the P/G bumps on the package, as illustrated in Fig. 5. Note that the hotspots represent the location of the high impedance P/G bumps within the power network. The highest resistance and

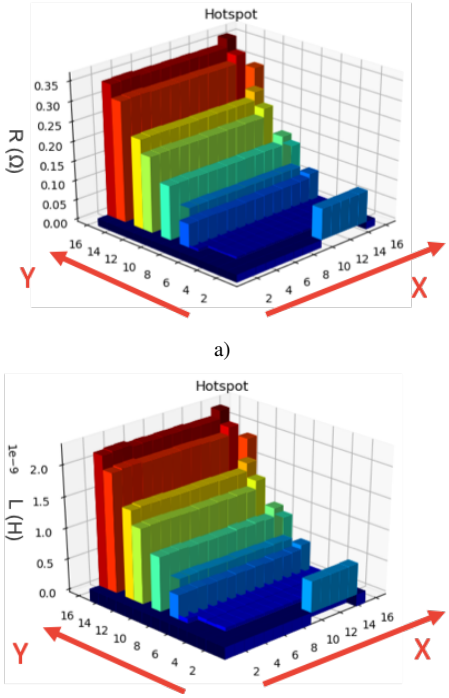


Fig. 4: Impedance variance of VDD_SRAM voltage domain across the entire package surface. (a) Map of resistance, and (b) map of inductance.

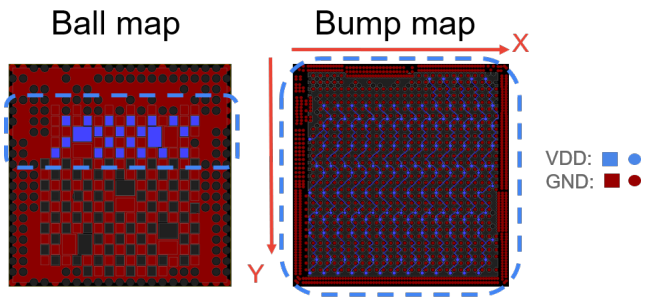


Fig. 5: Distribution map of P/G pinouts on the BGA and bump sides of a prototype package.

inductance of the package power network is, respectively, 0.35 ohms and 2.2 nH, as illustrated in Fig. 4. The gradient of the impedance distribution across the P/G bumps of the package is depicted by the impedance map, illustrating the location of the largest impedances.

B. PDN redesign with fast turnaround time

A greater impedance is observed deeper into the width of the package, as illustrated in Fig. 4. The variance between the maximum and minimum resistance and inductance is, respectively, 90X and 40X. Consider, as an example, the inductance. The maximum and minimum inductance is, respectively, 2.2 nH and 51 pH, exhibiting more than a 40X variance. Assuming the transient current of the loads connected to these ports is identical, a 40X difference in Ldi/dt noise is produced. In comparison, a lumped two-port PDN extraction only exhibits a single inductance of 1.59 pH, not revealing the impedance variance among the pinouts.

The reason for the significant increase in resistance and inductance is that the P/G bumps are connected through metal traces in the direction of the width of the package, where the current is distributed horizontally. The P/G balls for the VDD_SRAM voltage domain are only placed within a specific area along the bottom side of the package due to congestion with other voltage domains, as illustrated in Fig. 5. The bumps, alternatively, are distributed across the top side of the package. A horizontal current distribution path is therefore required in the power network within the package, which is achieved by connecting the bumps with metal traces, as illustrated in Fig. 6. These thin metal traces exhibit a significant resistance and inductance, which is not revealed by a lumped two-port PDN extraction process.

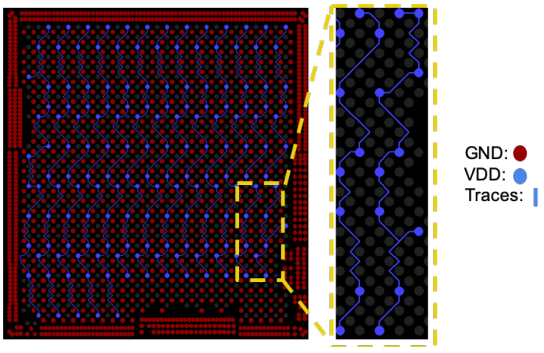
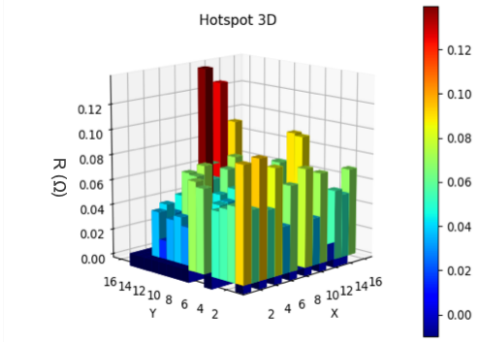
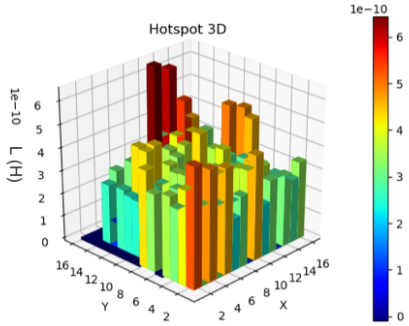


Fig. 6: V_{DD} pinouts connected by metal traces on the bump side of the package.



a)



b)

Fig. 7: Revised impedance variance of VDD_SRAM domain. (a) Resistance map, and (b) inductance map. Note the significantly lower resistance and inductance at the hotspot as compared to Fig. 4.

This impedance variance map efficiently illustrates a problematic power network for the VDD_SRAM voltage domain. This impedance variance across the P/G pinouts is significant, regardless of the magnitude of the impedance. A high resistance and inductance lead, respectively, to a larger IR voltage drop and Ldi/dt noise within the power network of the package. A package redesign is therefore required in this early power network exploratory stage, avoiding a costly redesign process during a later development stage. The package is subsequently redesigned, followed by another run of the PDN checker flow. The revised impedance map is illustrated in Fig. 7. The peak inductance, as an example, is reduced by more than 3X as compared with the original package. Moreover, the impedance variation is much smaller, leading to a more balanced and robust power network.

IV. Conclusions

Distributed port extraction of power delivery networks is required to provide accurate circuit models for time domain analysis. A Python based extraction tool is described to enable fast distributed port assignment during impedance extraction of the power delivery network. A Python enhanced automation flow for the power delivery network during early stages of the power network exploration and PI sign-off stages is presented in this paper. The three stage flow, including distributed port assignment, S-to-Z parameter transformation, and impedance variance analysis, provides more accurate PDN extraction information. These three stages can be decoupled into individual tools or can operate together within an overall automation flow.

REFERENCES

[1] I. P. Vaisband, R. Jakushokas, M. Popovich, A. V. Mezhiba, S. Kose, and E. G. Friedman, *On-Chip Power Delivery and Management, Fourth Edition*, Springer International Publishing, 2016.

[2] ITRS Technology Working Groups, "International Technology Roadmap for Semiconductors," 2013.

[3] E. Salman and E. G. Friedman, *High Performance Integrated Circuit Design*, McGraw-Hill Professional, 2012.

[4] Z. Yang, "Fundamentals of Power Integrity," *Proceedings of the IEEE Symposium on Electromagnetic Compatibility, Signal Integrity and Power Integrity*, pp. 1–50, July 2018.

[5] Ansys SIwave, "https://www.ansys.com/products/electronics/power-integrity," 2018.

[6] Sigrity PowerSI, "https://www.cadence.com/content/dam/cadence-www/global/en_US/documents/tools/pcb-design-analysis/sigrity-powersi-ds.pdf," 2019.

[7] K. Xu, B. Vaisband, G. Sizikov, X. Li, and E. G. Friedman, "Power Noise and Near-Field EMI of High-Current System-in-Package With VR Top and Bottom Placements," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, Vol. 9, No. 4, pp. 712–718, April 2019.

[8] R. Sjiariel, "Power Integrity Simulation of Power Delivery Network System," *Proceedings of the IEEE Workshop on Signal and Power Integrity*, pp. 1–5, May 2015.

[9] L. Smith and E. Bogatin, "Principles of Power Integrity for PDN Design," *Proceedings of the IEEE Symposium on Electromagnetic Compatibility & Signal/Power Integrity*, pp. 1–74, July 2018.

[10] Cadence Allegro, "https://www.cadence.com/content/cadence-www/global/en-US/home/tools/pcb-design-and-analysis/pcb-layout/allegro-pcb-designer.html," 2018.

[11] Keysight Technologies ADS, "https://www.keysight.com/en/pc-1297113/advanced-design-system-ads," 2019.

[12] D. A. Frickey, "Conversions between S, Z, Y, H, ABCD, and T Parameters which are Valid for Complex Source and Load Impedances," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 42, No. 2, pp. 205–211, February 1994.

[13] T. Reveyrand, "Multiport Conversions between S, Z, Y, h, ABCD, and T Parameters," *Proceedings of the Workshop on Integrated Nonlinear Microwave and Millimetre-Wave Circuits*, pp. 1–3, July 2018.