國立交通大學

電子工程學系 電子研究所

碩士論文

封裝佈局上基於電源完整性且有效率成本導向之去耦合電容最佳化

Cost-Effective Decoupling Capacitor Selection for Beyond Die Power Integrity

研究生：陳以恩

指導教授：陳宏明 博士

中華民國一〇二年七月

封裝佈局上基於電源完整性且有效率成本導向之去耦合電容最佳化

研究生：陳以恩　　指導教授：陳宏明教授

國立交通大學 電子工程學系 電子研究所

摘 要

在基於電源完整性考量下的電源分配網路設計中，確保能提供穩定電壓至晶片上的元件是非常重要的。而普遍來說都是藉由放置去耦合電容(Decoupling Capacitor)來抑制元件的切換雜訊。目前已有多篇著作探討如何基於電源完整性考量去選擇最佳的去耦合電容組合給晶片、封裝、印刷電路板，但其所挑選出的去耦合電容基於成本以及可製造性的考量，很難應用於實際設計中。我們提出了一個有效率的演算法名爲"優先去耦合電容選擇的粒子群聚演算法"來自動化且最佳化地選擇去耦合電容組合。其利用粒子群聚演算法隨機搜尋的優點且優先採用較爲有效的去耦合電容。我們應用此演算法到三個實際業界的封裝設計中，而結果顯示與工程師根據經驗法則所選出的去耦合電容相比，我們的演算法能基於同樣甚至更低的成本中選出更好的組合，並縮短設計時程。我們的演算法亦能同時考慮晶片、封裝、印刷電路板的共同設計在不同的操作頻率下做最佳化。

關鍵字：電源完整性、去耦合電容

Cost-Effective Decoupling Capacitor Selection for Beyond Die Power Integrity

Student: Yi-En Chen Advisor: Dr. Hung-Ming Chen

Department of Electronics Engineering
Institute of Electronics
National Chiao Tung University

ABSTRACT

In designing reliable power distribution networks (PDN) for power integrity (PI), it is essential to stabilize voltage supply to devices on chip. We usually employ decoupling capacitor (decap) to suppress the noise generated by the switching of devices. There have been numerous prior works on how to select/insert decaps in chip, package, or board to maintain PI, however optimal decap selection is usually not applicable due to design budget and manufacturability. Moreover, design cost is seldom touched or mentioned. In this research, we propose an efficient methodology "PDC-PSO" to automatically optimizing the selection of available decaps. This algorithm not only takes advantage of particle swarm optimization (PSO) to stochastically search the design space, but takes the most effective range of decaps into consideration to outperform the basic PSO. We apply this to three real package designs and the results show that, compared to the original decap selection by rules of thumb, our approach could shorten the design period and we have better combination of decaps at the same or lower cost. In addition, our methodology can also consider package-board co-design in optimizing different operation frequencies.

Keywords: Power Integrity, Decoupling Capacitor

Acknowledgements

First, I would like to express my greatest appreciation to my advisor, Prof. Hung-Ming Chen, for his helpful guidance and the chance he gave to be an intern in Global Unichip Corporation (GUC). He strengthened my capabilities of researching and polished my English writing skill. I want to thank my intern supervisor, Mr. Shi-Hao Chen, for his practical advices and assistance, too. Besides, I am also grateful to GUC partners Willy Wang, Jacky Hong, and Richard Chiu, for all their support. I learned and got a lot of knowledge in GUC, and this experience was very wonderful and substantial. I would also like to thank all VLSI Design Automation Laboratory members for their help. I want to give my special thanks to Tu-Hsung Tsai and Meng-Ling Chen for collaboration and suggestion.

Finally, I want to give my greatest gratitude to my families and my girlfriend Wan-Yu Lu. Thanks for their support and encouragement, I could concentrate on my studies and finish this thesis successfully.

Yi-En Chen

July, 2013

Contents

List of Tables

List of Figures

Chapter 1

Introduction

As the semiconductor manufacturing technology advances, the noise margin of chip is much lower than before, and a small voltage ripple might cause the devices on chip malfunction. The authors in [27][28][29] show that the fluctuation of voltage would reduce the operation frequency, and the relationship of voltage and operation frequency is almost linear, as Figure 1.1 shows.

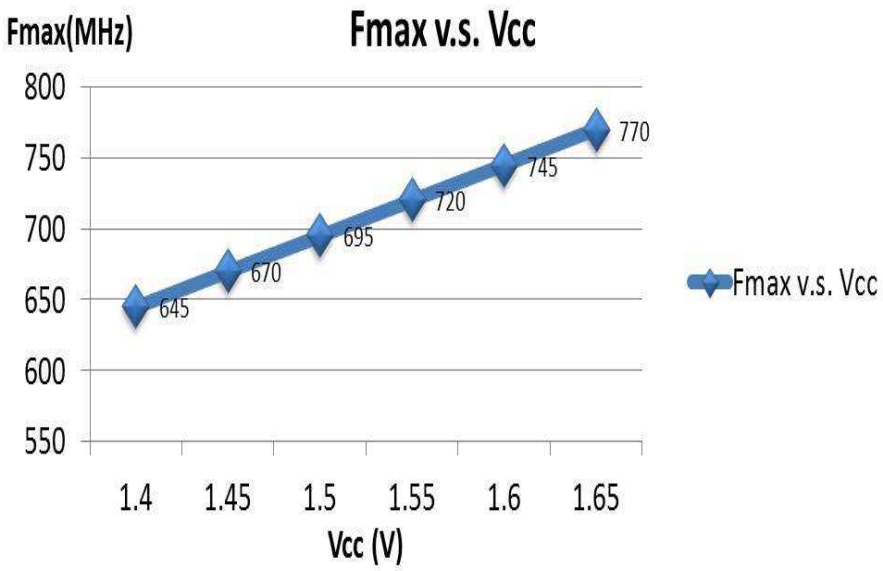


Figure 1.1: Relationship of supply voltage and maximum operation frequency.

Thus Power Integrity (PI) becomes more and more important, and it is about delivering clean power from voltage supplier to chip. Power distribution network (PDN) usually consists of Voltage Regulator Module (VRM), interconnections and

capacitors of PCB, package, and chip[1][22]. If the PDN is not well-designed, noise generated by devices on chip switching would exceed the tolerable range, and it might cause the Signal Integrity (SI), Electro-Magnetic Interference (EMI) problems and make the chip working incorrectly[20][21].

In PDN design, Decoupling Capacitor (decap) insertion is a common method to reduce voltage fluctuation. A decap acts as a temporary current pool and provides the low-noise return path for signals. However, it also acts as an inductor at the frequency higher than its self-resonance due to the intrinsic equivalent series inductance (ESL) decreasing its ability. Therefore, a good PDN usually includes several decaps to cover the targeted frequency range and to make the PDN robust. How to efficiently optimize the type, location and number of decaps to save cost and make PDN robust is critical in chip, package and PCB design[26]. Figure 1.2 shows that in a real package design, the engineers manually choose 16 decaps to meet the PDN specification. However, it could meet the same specification with only 5 decaps optimized by our program and the saving cost is very significant.

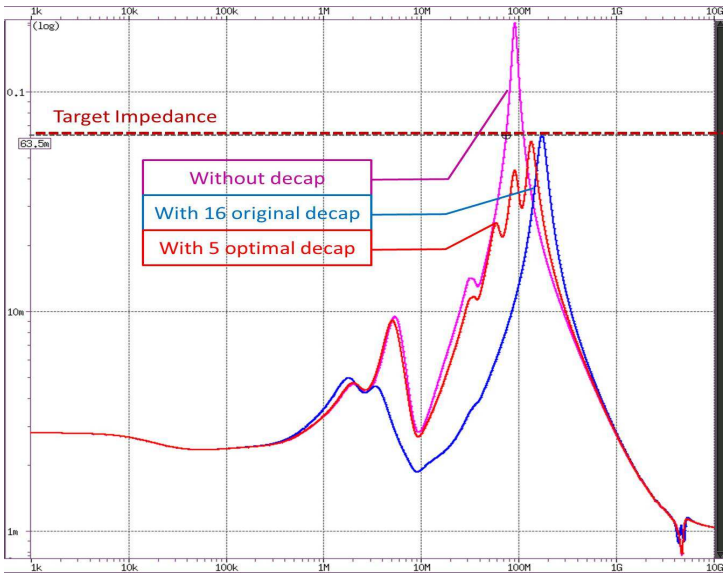


Figure 1.2: With the specification that target impedance is 0.0635Ω, our algorithm could reduce the total decaps from 16 to 5.

1.1 Previous Work

There are researches about decap selection optimization, such as [3][4][5][6], but they are manual rather than automatic optimizations. In [7][9] the authors use simulated annealing (SA) algorithm to choose the best location and type of decaps. However, compared to other stochastic algorithms like genetic algorithm (GA) or particle swarm optimization (PSO), SA is relatively ineffective and inefficient. Although PSO is applied to decap selection optimization problem in [10], it suffers from a problem that the result of decap selection is not commonly used in the industry, it would be expensive to manufacture or the design of package or PCB does not have enough area to place. There are researches using GA and sequential quadratic programming method to optimize decaps automatically (such as [11][12][13]). However, they have not taken the cost of decaps into consideration.

1.2 Our Contribution

In this thesis, we introduce an efficient algorithm named "Preferred Decap Choice Particle Swarm Optimization (PDC-PSO)" to optimize the decap combination for PDN design automatically. The constraints like type, amount, location, and cost of decaps could be taken into account to avoid over-design, thus this PDC-PSO algorithm is practical in real design. Since in [14][15][8] the authors demonstrate that in PSO each particle has its own acceleration coefficients and inertia weight changes with iterations, and p_1 should be larger and p_2 should be smaller, those would lead to a better solution than basic PSO. We blend those concepts in our decap optimization problem.

Since inductance would diminish the ability and shift the self-resonance of decap, we should take the inductance generated by traces and vias on board into account. Therefore, PDC-PSO modifies the acceleration coefficients of each particle according to the number of "preferred" decaps chosen by each particle. We apply PDC-PSO

to two real package designs to verify our algorithm and use the decaps of Murata[18] as our library, so the particle would search in a discrete solution space rather than continuous one, and that ensures the decaps we choose are manufacturable. The experimental results show that the decaps selected by our algorithm are effective in suppressing voltage fluctuation, and our algorithm could use less decaps to meet the same specification than manual decap selection.

1.3 Thesis Organization

The remainder of this thesis is organized as follows: Chapter 2 describes the power distribution network model and the definition of target impedance. Chapter 3 introduces the characteristic of decoupling capacitor and our objectives. Chapter 4 presents our methodology based on the particle swarm optimization. Chapter 5 reports the experimental results. Finally, Chapter 6 concludes this thesis.

Chapter 2

Power Distribution Network and Target Impedance

2.1 Power Distribution Network Model

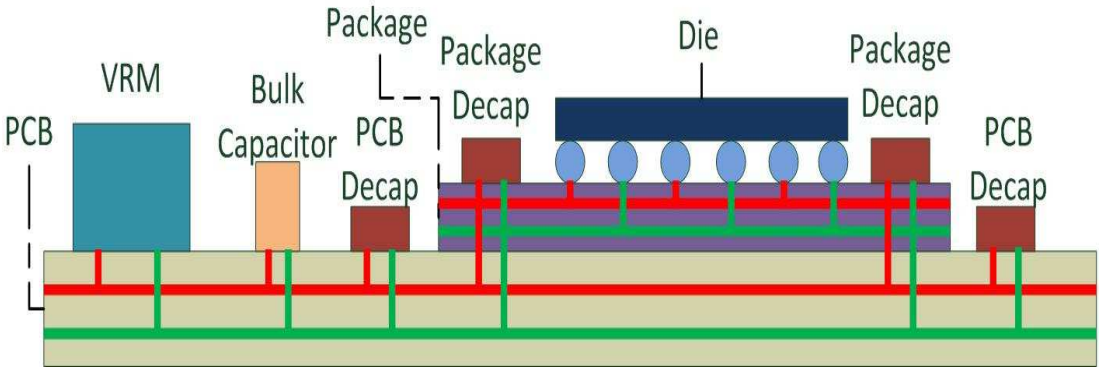
The PDN includes VRM, decaps, and the interconnections of power grid on PCB, package, and die as shown in Figure 2.1. The voltage sent by VRM to chip will be derated by the resistance and inductance of the PDN interconnection. When a DC current flows to the chip, the voltage would be decreased by the resistance of the interconnection of PDN according to Eq(2.1) and this leads to the IR-drop. On the other hand, if it is an AC current, according to Eq(2.2)(2.3), it would induce an electromotive force ε to resist the change of current and cause the Ldi/dt drop[23].

$$V = IR \tag{2.1}$$

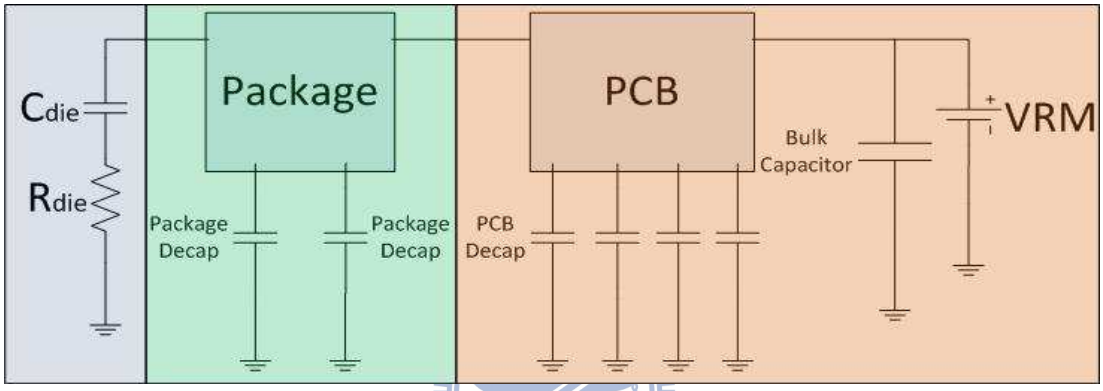
$$\varepsilon = -L\frac{di}{dt} \tag{2.2}$$

$$V = L\frac{di}{dt} \tag{2.3}$$

The fluctuation of voltage at the pads on chip may harm the circuit noise margin and cause those devices on chip malfunction[21]. Therefore, we have to shrink the



(a) Cross view of PDN system



(b) Lumped model of PDN system

Figure 2.1: PDN system includes chip, package, PCB, and VRM. (a) is the cross view of PDN system. (b) is the equivalent lumped model of PDN system.

fluctuation of voltage within an acceptable range to ensure the robustness of PDN.

Moreover, if we observe the impedance from the pads on chip, we would find that due to the PDN impedance, the fluctuant current causes different voltage drop at different frequency[2]. Figure 2.2 demonstrates an example that the PDN impedance is not a constant but varies in different frequencies. Since we want to control the fluctuation of voltage within a certain range, we should let the PDN impedance be below the target impedance.

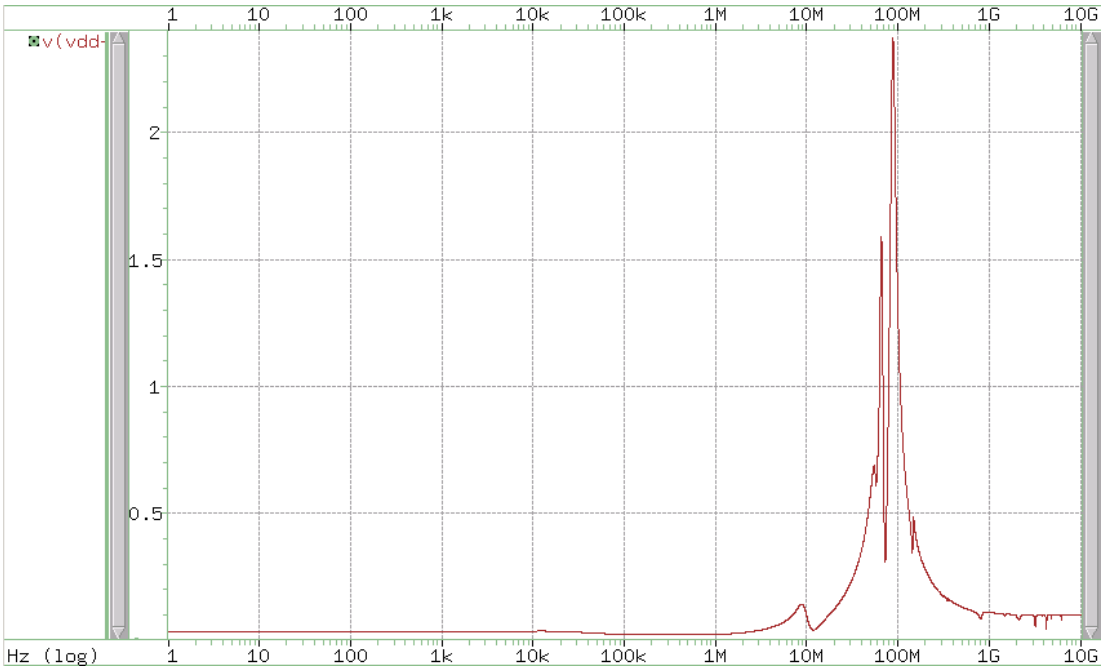


Figure 2.2: Impedance at different frequency of power distribution network.

2.2 Target Impedance

According to [16], the target impedance is defined as the following equation:

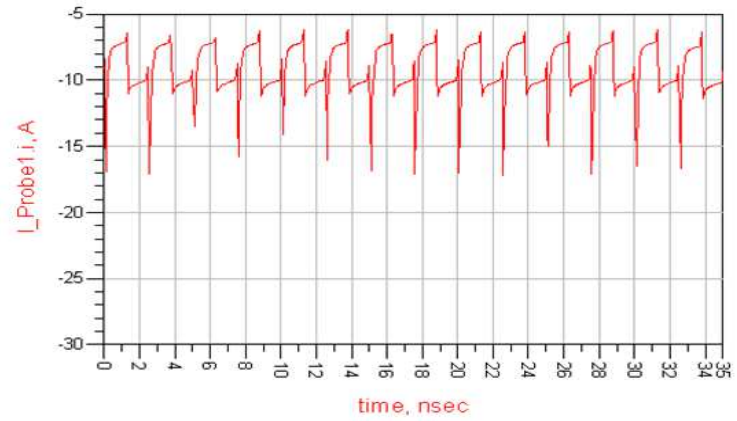
$$Z_{target} = \frac{V_{supply} \times \text{allowed ripple}}{\Delta I_{max}} \tag{2.4}$$

Eq(2.4) represents the maximum impedance of PDN in the situation that all devices on chip simultaneously operate and draw tremendous current and the fluctuation of voltage is still in an acceptable range. The PDN impedance should be below or meet the target impedance at the frequency in the transient state. To more accurately estimate the target impedance, in this research we apply the approach in [17] to get the real current profile and then use fast fourier transform (FFT) to translate the time-domain current spectrum to frequency-domain current spectrum. Figure 2.3 is an example of current profile, it is measured from the devices on chip and it records the change of current as the devices switch. After the current profile is translated by using FFT, it would be a frequency-domain spectrum and

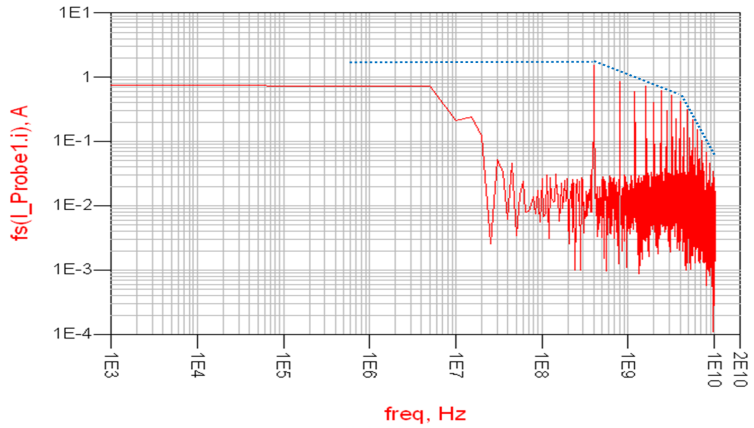
represents the compositions of current distributed in every frequency, as shown in Figure 2.3(b). The peak is usually the operation frequency (clock frequency) and the PDN impedance should be below target impedance on this frequency. Since the regular switching of devices would draw the current and lead to regular voltage drop, this could be regarded as a recurrent noise, and it is the main source of noise.

$$V(f) = I(f)Z \tag{2.5}$$

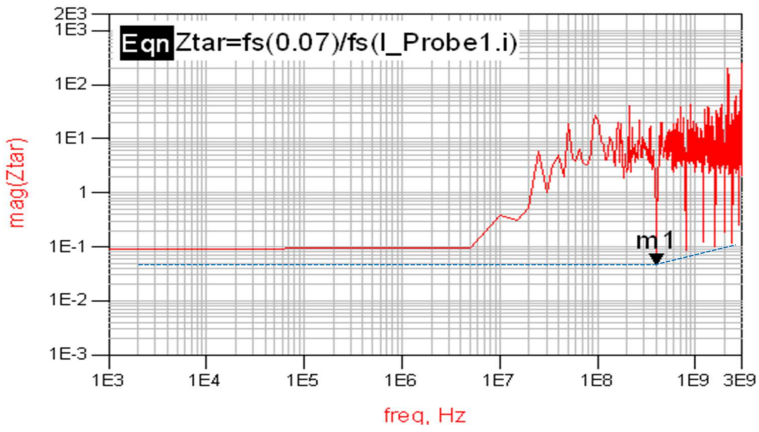
We use Eq(2.5) to translate the frequency-dependent current spectrum to the frequency-dependent impedance spectrum, where $V(f)$ is the allowed voltage ripple, and then we obtain the target impedance as the blue line in Figure 2.3(c).



(a) Current profile



(b) Current profile after FFT



(c) Target impedance(blue line) induced by current profile

Figure 2.3: To obtain the target impedance, we get the current profile in time domain (a) measured at the device on chip, and using FFT to translate it into frequency domain spectrum (b). Finally we derived the tolerable impedance at different frequency (c) from (b) and Eq(2.5), and the blue line is the target impedance.

Chapter 3

Preliminaries

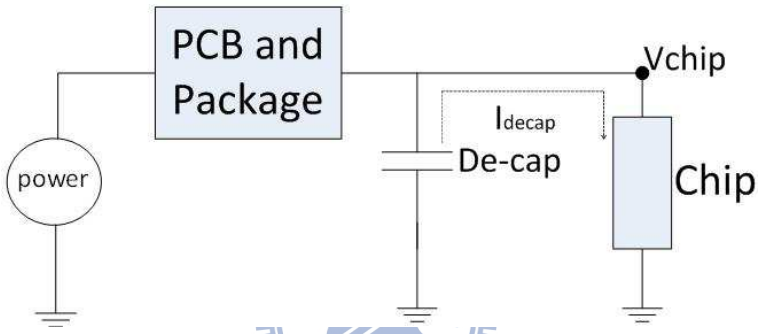
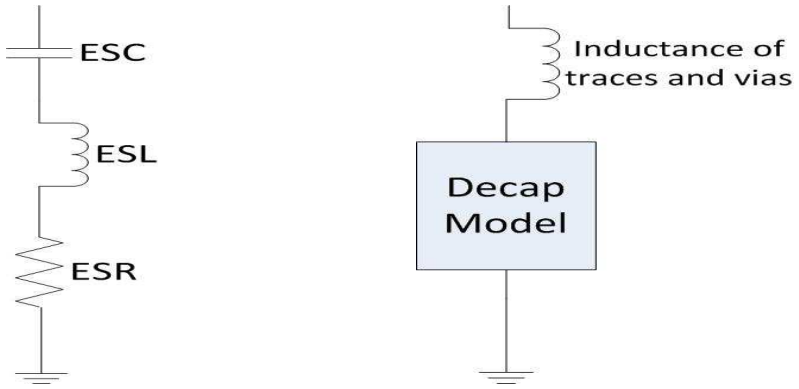


Figure 3.1: Power distribution network with decoupling capacitor

3.1 Decoupling Capacitor

Decoupling capacitor (decap) acts as a temporary current pool in PDN system, as shown in Figure 3.1, when the transient of the chip occurs, the VRM is not fast enough to provide sufficient current to the chip since the operation frequency and the switching of devices are very fast, making the voltage V_{chip} drop. In this situation, decap would provide current to chip when V_{chip} is lower than the voltage of decap. In general, the ideal capacitor does not exist in real world: besides its own equivalent series capacitance (ESC), there exists intrinsic equivalent series resistance (ESR) and inductance (ESL), as shown in Figure 3.2(a). The impedance equation of a decap is Eq(3.1). Eq(4.3) demonstrates that when the imaginary part of impedance of decap is zero, the impedance of decap is minimum and equals to its ESR and this



(a) (b)

Figure 3.2: Decap model. (a) is the decap lumped model consists of ESR, ESC, and ESL. (b) is the decap model including the inductance induced by traces and vias.

frequency is called "$self - resonance$".

$$Z_{decap} = ESR + j\left\{2\pi fESL - \frac{1}{2\pi fESC}\right\} \tag{3.1}$$

$$f_{self-resonance} = \frac{1}{2\pi\sqrt{ESL \times ESC}} \tag{3.2}$$

The effects of each component in different value are shown in Figure 3.3. We could see that both larger capacitance and inductance would cause lower self-resonance of decap. However, decap with large capacitance has wider low impedance range and better ability of decoupling. Large inductance may be a burden of decap since, according to Eq(2.2), inductance would resist the change of current and decrease the ability of current charge and discharge of decap. The resistance and the lowest impedance of decap at self-resonance are in positive correlation, and usually the lower ESR is easier to meet the target impedance. Therefore, an effective decap should include low ESR, low ESL and high ESC.

On the other hand, the impedance for the frequency beyond the self-resonance of decap is increasing, and it is a drawback for decap insertion to diminish the PDN impedance. A good solution to reduce the flaws of inductance is partitioning a large

decap into several small decaps because the parallel connection of several inductors would make a smaller inductor. However, the cost of this method is more layout area and it might make the routing of other cells more difficult. In this research we use the decap SPICE model of Murata Corp[18] to ensure our chosen decaps are manufacturable, but those models are more complex and accurate than lumped decap model. Figure 3.3(d) is the comparison of Murata model and lumped model including ESR, ESL and ESC.

3.2 Objectives

In real designs, the most important criterion is to work correctly, so in order to ensure the PDN is stable, the first objective function in our algorithm is defined as

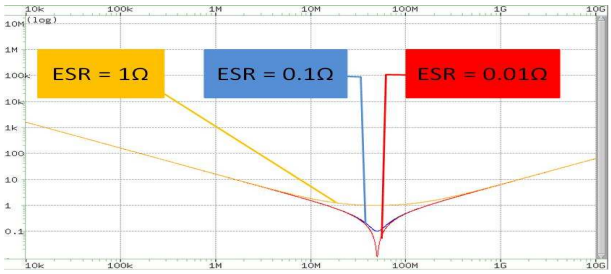
$$\boldsymbol{min} \int_{f_L}^{f_U} penalty(f) \times p(f) \tag{3.3}$$

where f_L and f_U are the lower and upper bound of interesting frequency respectively. $p(f)$ is the part of PDN impedance exceeding the target impedance and $penalty(f)$ is the penalty at each frequency. Although the PDN impedance should be below the target impedance in the entire frequency range in theory, there may be constraints such as cost and layout area in real design, and it is difficult to reach the goal by limited decaps. Therefore, we should take care the noisiest frequency as our top priority. And usually the greatest simultaneous switching noise (SSN) is in operation frequency, so we increase the penalty around it to treat lowering the PDN impedance as higher priority.

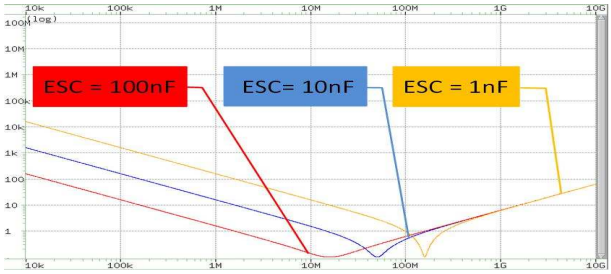
When we meet the target impedance, cost becomes the next important criterion in industry. When there are G decap combinations which could make the PDN impedance meet the target impedance, we use the following equation to choose the minimum cost one from those combinations:

$$\begin{aligned} \boldsymbol{min} \sum_{i=0}^{M} cost_i^g \times decap_i^g \\ subject\ to\ \ 0 \leq g \leq G \end{aligned} \tag{3.4}$$

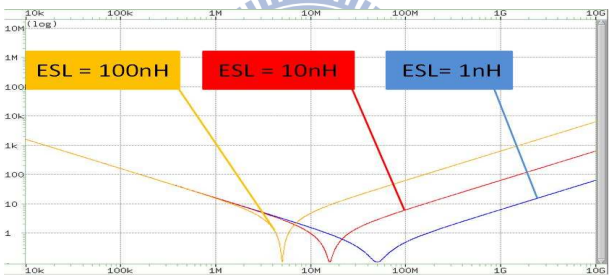
The $cost_i^g$ and $decap_i^g$ denote the retail price and the decap used in i_{th} port in g_{th} combination and M is the number of predefined ports for decap insertion.



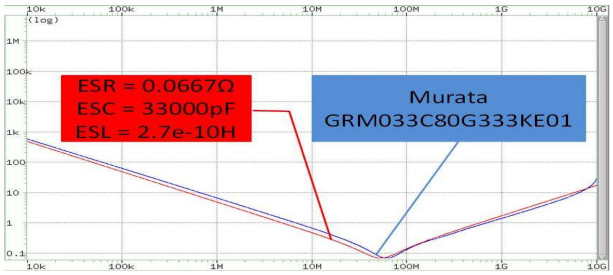
(a) Decap with different ESR



(b) Decap with different ESC



(c) Decap with different ESL



(d) Lumped and realistic decap

Figure 3.3: A good decap should have low ESR, ESL, and high ESC. (a) shows the lower ESR would lead to the lower minimal impedance at self-resonance of decap, and (b) shows the larger ESC would lower the self-resonance and increase the low impedance range, and (c) shows the smaller ESL would raise the self-resonance and increase the low impedance range. (d) is the comparison of a lumped decap model including only ESR, ESC, and ESL, and a realistic decap model.

Chapter 4

Methodology

4.1 Particle Swarm Optimization (PSO)

Particle Swarm Optimization is presented by Kennedy and Eberhart in 1995[19]. It is a stochastic algorithm and inspired by the fishes schooling and the birds flocking, by imitating the behavior of birds that consider not only their self-consciousness but also the corporate intelligence to find the best solution, and avoid being controlled by a specified individual. It has the advantages such as easy implementation, fast convergence and the ability to jump out from local optimal solution. Each bird (particle) in PSO changes its position by considering the best solution it ever found and the best solution the entire swarm found, and as time goes on, all particles would assemble to a best solution position.

To implement PSO algorithm in our problem, we regard the entire solution space as a multi-dimension grid, and each predefined decap insertion port corresponds to a dimension. The specification-matched decaps of each port form the coordinates, as Figure 4.5 shows. In the beginning, there would be P particles generated, and distributed randomly in the entire discrete solution space. Each particle is assigned a velocity randomly and that represents a solution to the optimization problem. Next, we calculate the fitness of all particles. Fitness is calculated according to

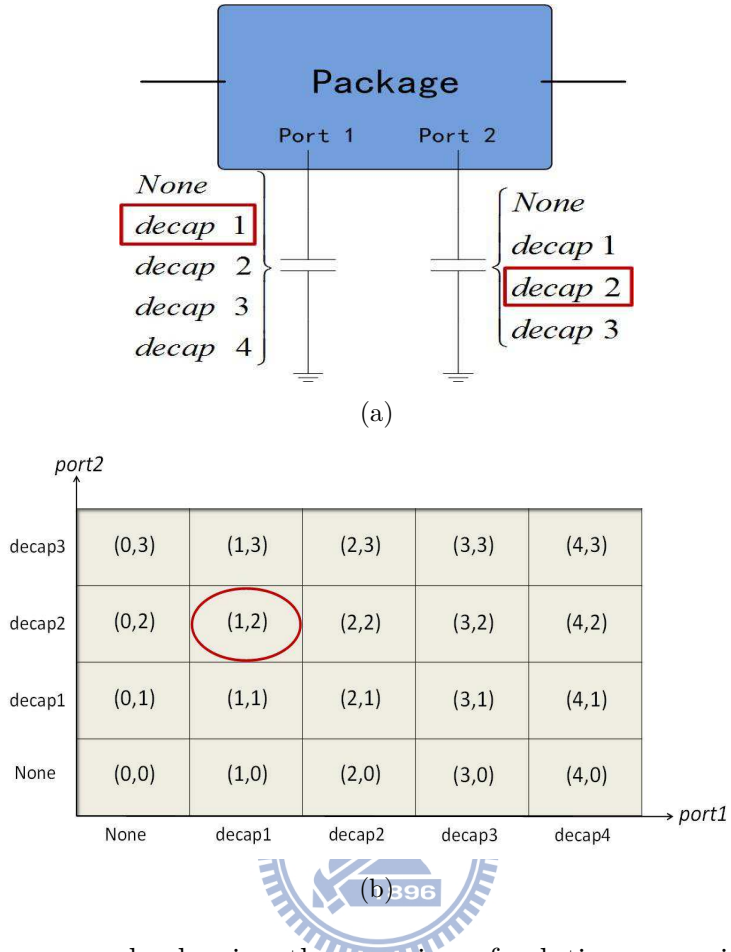


Figure 4.1: An example showing the meaning of solution space in PSO. (a) is a package design with two predefined decap insertion ports, and there are several specification-matched decaps which could be chosen in each port. (b) is the discrete PSO solution space we map. If a particle is on (1,2), it means we choose decap1 for port1 and decap2 for port2. Besides, "None" means there is no decap placed in this port.

the objective function(Eq(3.3)) in the optimization problem, and usually the lower fitness represents better solution. After the fitness calculation of all particles is performed, each particle memorizes its own fitness as its p_{best}, and best fitness of those particles is defined as global best g_{best}. If there are the particles whose fitness is 0, the p_{best} and g_{best} would be decided by objective function(Eq(3.4)). After that, particles adjust their position by the following equations:

$$v_i^{t+1} = \omega(t)v_i^t + p_1 r_1(pbest_i^t - x_i^t) + p_2 r_2(gbest^t - x_i^t) \tag{4.1}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{4.2}$$

where x_i^t is the position of i_{th} particle in t_{th} iteration, and v_i^t is the velocity of i_{th} particle in t_{th} iteration. $pbest_i^t$ is the best solution ever found by i_{th} particle till t_{th} iteration, and $gbest^t$ is the best solution ever found by all particles till t_{th} iteration. r_1 and r_2 are randomly number distributed between [0, 1]. ω is the "inertia", and p_1 and p_2 are the coefficients of acceleration.

4.2 Preferred Decap Choice (PDC)

Since in industry flow engineers should consider many constraints such as the routability, size or cost of chip, package and PCB, thus there may be a situation that we provide a series of decap size and number that could make PDN impedance meet target impedance, but the white space in the layout is not large enough or there is no such decap. To avoid this, we choose decaps from the library of capacitor manufacturer like Murata[18] then take the specification into consideration. This could make sure the result could be manufactured and the shape and size match the predefined port.

As [6] shows, to reduce the impedance in the specified frequency range, using the combination of different decaps to make PDN impedance meet the target impedance is more effective than using the combination of the same decaps. Figure 4.2 demonstrates how we define a "$Preferred$" decap. When using the same total amount of decaps, if we use more "$Preferred$" decap, we could make the PDN impedance meet the target impedance easier. Therefore, the optimal solution of decap selection usually includes several "$Preferred$" decaps, we want the particles in PSO to search the area around the location with more "$Preferred$" decaps to find the optimal solution.

However, in real PDN design, there are traces and vias between the predefined

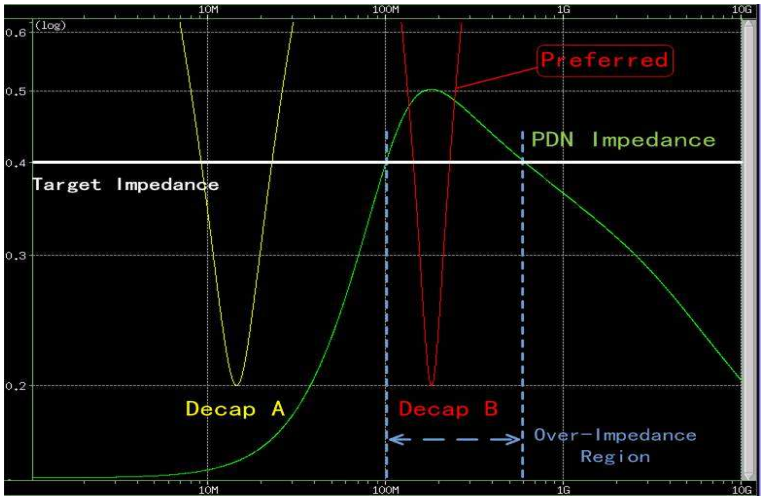


Figure 4.2: To choose an effective decap from DecapA and DecapB, since the resonance frequency of DecapB is within the over-impedance region, it is more effective and we mark it as "$Preferred$".

decap port and pads on chip, and we know that the traces and vias have their intrinsic inductance that would increase the equivalent inductance and it would decrease the low impedance frequency range of decap. If we only consider the self-resonance of decap as the standard to choose which decap we preferred, it would be inaccurate. To minimize the inaccuracy, we roughly extract the inductance between predefined decap insertion port and the pads on chip by measuring the impedance from predefined decap insertion port while grounding the pads on chip, and use an inductor to fit the impedance curve and get the inductance as shown in Figure 4.3. Therefore, the ESL of decaps should add the inductance between the predefined decap insertion port and pads on chip. After the modification of ESL of each decap, we could recalculate the self-resonance of each decap according to Eq(4.3), and mark the decap as "$Preferred$" if its self-resonance is at the frequency that PDN impedance is exceeding the target impedance.

$$f_{self-resonance} = \frac{1}{2\pi\sqrt{ESL \times ESC}} \tag{4.3}$$

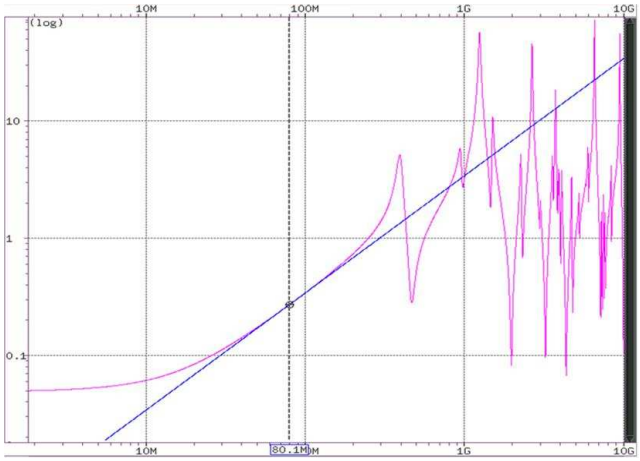


Figure 4.3: To obtain intrinsic inductance of traces and vias between predefined decap port and pads on chip and make our estimation of the decap self-resonance more accurate, we measure the impedance from a predefined decap port(purple line) and use an inductance(blue line) to fit it.

4.3 PDC-PSO

The basic PSO usually chooses the decap whose self-resonance is not at the non-meeting target impedance frequency, and it wastes time to search the solution consisting of those decaps. In [8], the authors show that in the PSO algorithm, there would be a better result if p_2 is less than "1" and p_1 is between $[4, 10]$, and p_1 should decrease and p_2 should increase as the number of iteration increases. Therefore, we would set parameter p_{1max}, p_{1min}, p_{2max} and p_{2min} to define the boundaries of p_1 and p_2. In addition to the methods described in [8], we give more information about which decap should be chosen to make the PSO algorithm have higher chances to find the optimal solution.

In our algorithm, each particle has its own coefficients of acceleration p_1 and p_2, and when a particle moves to a better location and updates *pbest* or *gbest*, the particle would check how many decaps it chooses are marked as "$Preferred$", and

renew its p_1 and p_2 according to Eq(4.4)(4.5)(4.6)(4.7).

$$p_{1new} = p_1 + \phi_1 \tag{4.4}$$

$$p_{2new} = p_2 + \phi_2 \tag{4.5}$$

$$\phi_1 = (p_{1max} - p_1) \times \frac{N_{Local}}{N_{Local} + N_{Global}} \tag{4.6}$$

$$\phi_2 = (p_{2max} - p_2) \times \frac{N_{Global}}{N_{Local} + N_{Global}} \tag{4.7}$$

where p_{1max} and p_{2max} are the user defined upper bound of p_1 and p_2. N_{Local} is the amount of "$Preferred$" decaps used in *pbest* location, and N_{Global} is the amount of "$Preferred$" decaps used in *gbest* location. In the beginning of PDC-PSO, we set p_1 for p_{1max}, and p_2 for p_{2min}.

Since we know that the coefficients of acceleration influence the PSO significantly, we let every particle have its own p_1 and p_2, and we know they are relative to *pbest* and *gbest* respectively, so we renew those coefficients when its *pbest* or *gbest* is updated. When the *pbest* or *gbest* location of a particle uses more "$Preferred$" decaps, we know that it has higher possibility that the global optimal solution is nearby. Eq(4.4) and Eq(4.6) demonstrate that if the *pbest* of a particle uses more "$Preferred$" decaps, p_1 would be increased so that this particle would tend to search the area around *pbest*. Similarly, if *gbest* uses more "$Preferred$" decaps than *pbest*, according to Eq(4.5)and Eq(4.7), the coefficient p_2 would be larger than p_1, and that would make the particle tend to search the area around *gbest*. To avoid our algorithm using the decaps whose resonance frequencies are at the same over-impedance region and being trapped to local optimal solution, we set maximum capacity for each over-impedance region to prevent there are too many "$Preferred$" decaps in the same over-impedance region. The maximum capacity is decided by user. Once the amount of "$Preferred$" decaps used in *pbest* or *gbest* location is more than the maximum capacity, it would increase the maximum capacity in N_{Local}

or N_{Global}, as Eq(4.8) shows.

$$\begin{aligned} &For\ an\ over-impedance\ region, \\ &if\ \#Decap_{Preferred} > Capacity, \\ &N_{Local(Global)} = N_{Local(Global)} + Capacity \end{aligned} \tag{4.8}$$

Since we map the solution space into a multi-dimension grid, the particle must be on the first index of the dimension to let no decap be placed in the port. If the decap library of port is very large, the probability of a particle being on the first index would be small. Thus we add more none-decap-insertion locations, which means there is no decap placed in the port, in each dimension to increase the probability that particle would find the lower cost solution, as shown in Figure 4.4. In the process of PDC-PSO, if the best solution of particles let the PDN impedance meet the target impedance, when particles move to new locations, it would compare the cost of the decap combinations and set the lower cost one as its best solution. By those modifications in our algorithm, the particles would not be trapped in local best and we have more probability to find the global optimal solution.

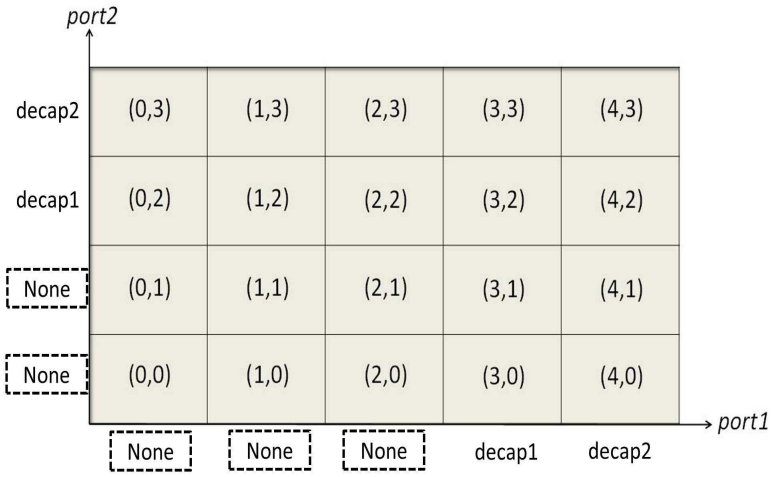


Figure 4.4: An example that demonstrates how we add more none-decap-insertion location to each dimension.

4.3.1 Boundary Condition

Since the particles might exceed the solution space boundary, we have to deal with this situation. According to [30], we use the following three types of boundary conditions: absorbing, reflecting and damping.

1. Absorbing: When a particle overflies the boundary of solution space in one dimension, it would be stuck on the boundary and the velocity would be reset a random number in that dimension. Differ from [30], we do not make the velocity be zero since our solution space is discrete, the velocity which is zero would make the particle move too slowly and be limited in a certain range. Thus we reset the velocity a random value to enhance the searching range of particle, as Figure 4.5(a) shows.

2. Reflecting: When a particle overflies the boundary of solution space in one dimension, it would be stuck on the boundary and the sign of its velocity in the dimension would be changed. This represents the energy which making the particle overfly the boundary is reflected by a wall and draws the particle back to the solution space, as Figure 4.5(b) shows.

3. Damping: When a particle overflies the boundary of solution space in one dimension, it would be stuck on the boundary and the sign of its velocity in the dimension would be changed and multiply a random value between 0 and 1. This represents the energy which making the particle overfly the boundary is imperfectly reflected by a wall and the lost part of energy is determined by a random value, as Figure 4.5(c) shows. All the three types of boundary conditions mentioned above could make the particle be within the solution space, so in this thesis when particles overfly the boundary, we randomly choose one of the boundary conditions to deal with the problem. In this way, we could enhance the searching diversity and have a better solution.

4.3.2 Particle Swarm Optimization with Random Sampling in Variable Neighborhoods

To avoid that the particles are trapped into local optimum and hard to escape, we use the PSO-RSVN algorithm[31] to make the particle escape from the local optimum. In PSO-RSVN, it would detect the premature convergence state of PSO, and when the premature convergence or stagnation state is detected, the particle swarm would be dispersed. In other words, the algorithm redistributes the particles into several neighborhoods which are around the global best point G. The following equations are used to generate the set of particles in each neighborhood:

$$\lambda_{jd}^{-} = \begin{cases} \tau_{jd}, & \tau_{jd} \geq \sigma_{min} \\ \sigma_{min}, & \tau_{jd} < \sigma_{min} \end{cases} and\ \tau_{jd} = G_d - \xi_j \left| \sigma_{max} - \sigma_{min} \right| \tag{4.9}$$

$$\lambda_{jd}^{+} = \begin{cases} \varsigma_{jd}, & \varsigma_{jd} \leq \sigma_{max} \\ \sigma_{max}, & \varsigma_{jd} > \sigma_{max} \end{cases} and\ \varsigma_{jd} = G_d + \xi_j \left| \sigma_{max} - \sigma_{min} \right| \tag{4.10}$$

$$X_t \in \Psi_j | X_{td} \sim \bigcup \left(\lambda_{jd}^{-}, \lambda_{jd}^{+} \right),\ t = 1, \cdots, |\Psi_j|\ ;\ j = 1, \cdots, M \tag{4.11}$$

Where d is the dimension of particle; σ_{max} and σ_{min} are the boundaries of each dimension; ξ_j is a fractional number between 0 and 1, and it is calculated according to the following equation: ξ_j=j/M. M is the number of neighborhood and it is decided by users. Ψ_j is the j-th set of particles in the domain within the interval $[\lambda_{jd}^{-},\lambda_{jd}^{+}]$. To select the proper particle for each subset ϕ_j, we adopt the suggestion of the author and let number of each set of particles be $|\phi_j|=|\Psi_j|=|\Omega|$/M. By this modification, the swarm diversity would increase the exploration of the solution space and the possibility of escape from local optimum.

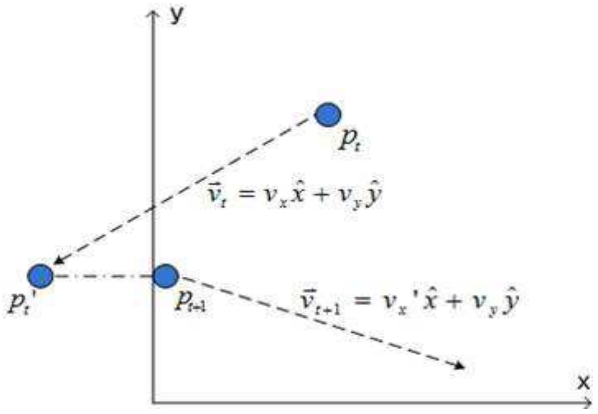
4.3.3 Algorithm and Flow

First, we select the decaps matching specification for each predefined port from library and sort them according to their self-resonance, and add the some none-

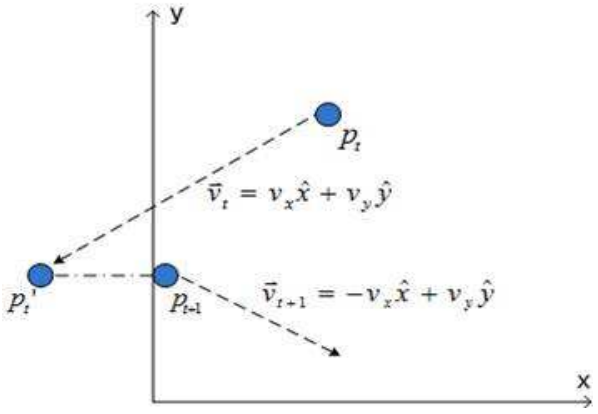
Algorithm 1 PDC-PSO

```
Choose matched spec decaps from library.
Initialize P particles with random position and velocity.
Calculate fitness of each particle, define pbest and gbest.
for t = 1 to MAX ITERATION do
    Renew the position, velocity, and inertia.
    Calculate fitness of each particle.
    Renew pbest and gbest, take cost into consideration.
    if pbest or gbest is updated then
        Update p1 and p2.
    end if
end for
Solution ← gbest.
```

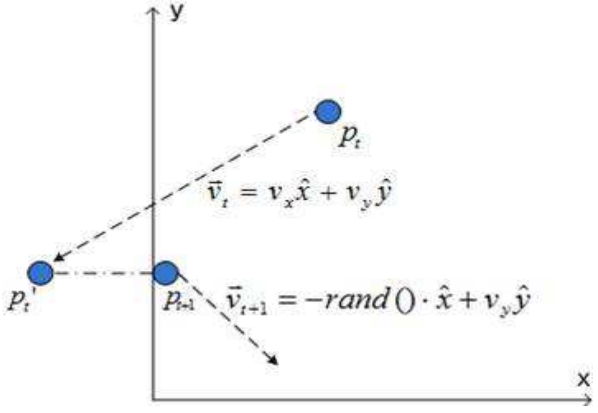
decap-insertion location into the solution space. Second, we define P particles and assign random position in the searching space and velocity for each particle, then calculate the fitness of each particle and update *pbest*, and choose the particle with best fitness as the *gbest*. Next, if the iteration number does not reach the max iteration, each particle renews its individual position and velocity according to Eq(4.1)(4.2), then calculate the fitness of each particle, the fitness will take cost into consideration. If the new fitness or cost of a particle is better than *pbest* or *gbest*, the particle updates its *pbest* or *gbest*, then using Eq(4.4)(4.5)(4.6)(4.7) to get the new acceleration coefficients. After whole iterations are terminated, *gbest* is the best selection of decaps for each predefined port.



(a)



(b)



(c)

Figure 4.5: The three boundary conditions for our problem. Where p_{t+1} and v_{t+1} are the position and velocity after modifying.

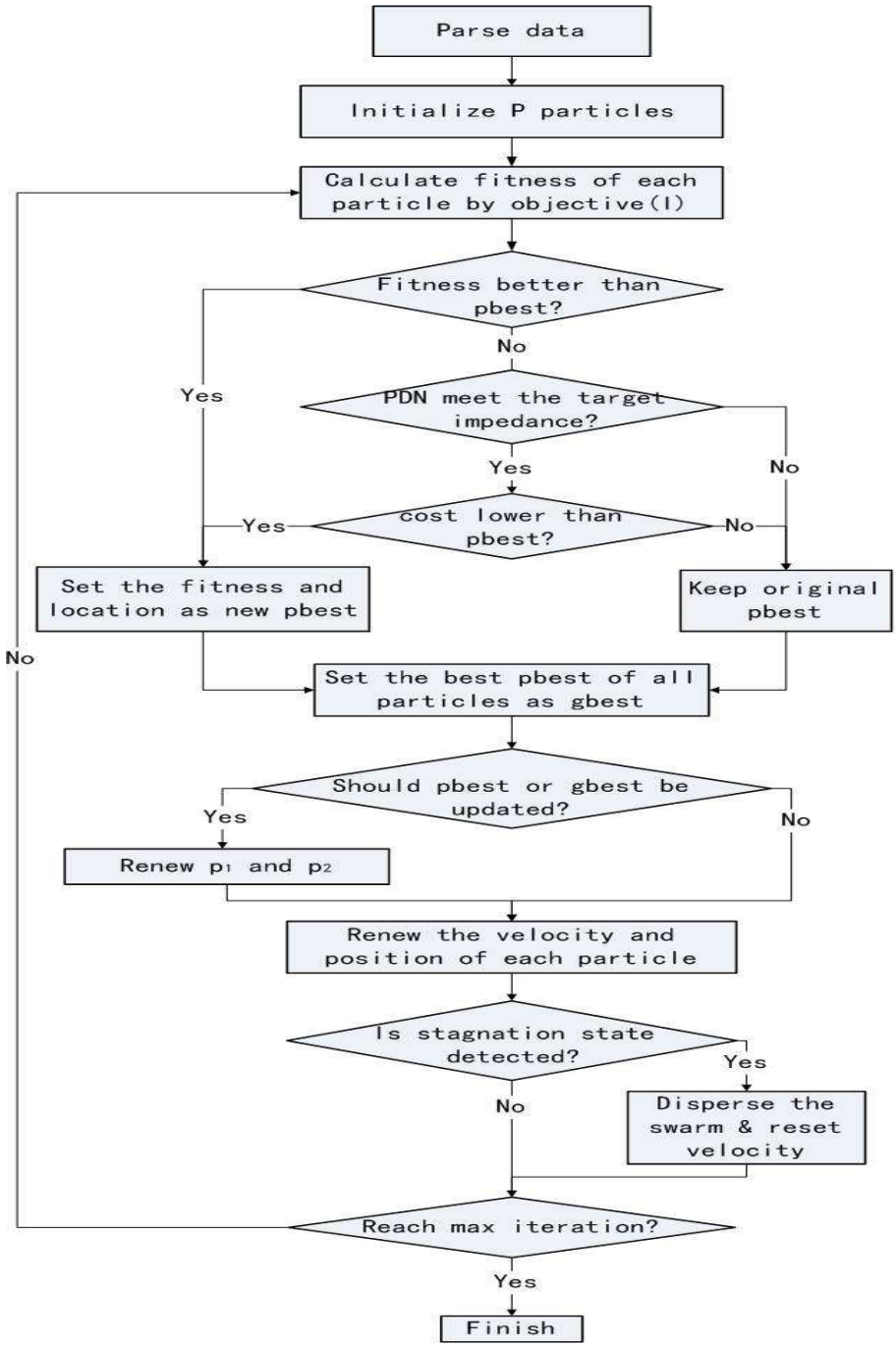


Figure 4.6: The entire flow of the algorithm.

Chapter 5

Experimental Results

We implement our algorithm with C++ language and apply it to three package designs. We use HSPICE to get the PDN impedance. The package, PCB SPICE models are extracted by SIwave[24] for all cases, and the chip is modelled by a resistor (R_{chip}) and a capacitor (C_{chip}).

The information of the three cases and the parameter setting for PSO, PDC-PSO, and SA are shown in Table 5.1 and Table 5.2 respectively. We run our algorithm, basic PSO and simulated annealing(SA) 50 times respectively and record the results to show the performance of each algorithm. We set the runtime limitation for 1 day, and we also parallelize each particle in PSO and PDC-PSO to accelerate the program.

Table 5.1: Information of all cases. T-I means the target impedance.

Case Information			
	Case-1	**Case-2**	**Case-3**
Process	40nm	28nm	28nm
# ports	2	5	16
Max port size(mm^2)	1.6*0.8	2*1.25	2*1.25
Op frequency	800Mhz	100Mhz	200Mhz
Supply voltage	1.5V	0.7V	0.9V
Voltage tolerance	10%	10%	10%
T-I(Performance)	0.75	0.01	0.036
T-I(Cost)	0.9	0.1	0.07

Table 5.2: Parameters For PSO & PDC-PSO & SA

Parameters For PSO			
	Case-1	**Case-2**	**Case-3**
# Particles	**3**	**5**	**10**
Max Iteration	**30**	**40**	**150**
ω	**1**	**1**	**1**
p1	**2**	**2**	**2**
p2	**2**	**2**	**2**
Parameters For PDC-PSO			
	Case-1	**Case-2**	**Case-3**
# Particles	**3**	**5**	**10**
Max Iteration	**30**	**40**	**150**
ω	**0.4**	**0.4**	**0.4**
Capacity	**1**	**3**	**8**
Stagnation	**15**	**20**	**20**
$\mathbf{p1}_{min}$	**2.5**	**2.5**	**2.5**
$\mathbf{p1}_{max}$	**9**	**9**	**9**
$\mathbf{p2}_{min}$	**1**	**1**	**1**
$\mathbf{p2}_{max}$	**2**	**2**	**2**
Parameters For SA			
	Case-1	**Case-2**	**Case-3**
Initial temperature	**100**	**100**	**100**
Final temperature	**0.98**	**0.0035**	**2.80E-05**
Decreasing step	**0.95**	**0.95**	**0.99**

5.1 Cost Driven Decap Selection

We slightly relax the target impedance of each case to let some of the predefined ports be empty and still could make PDN meet the target impedance, and apply algorithms to find the minimum cost decap combination. We set each decap cost to 1, that is, using less decaps stands for the lower cost.

Figure 5.1 is the histogram showing the decap cost of each algorithm run for 50

times, and compared to PSO and simulated annealing(SA), our algorithm use less decaps to make the PDN meet the target impedance and saves the area and cost.

This result and Figure 1.2 show that the manual and careless decap selection usually causes over-design, and compared to PSO and SA, our algorithm is more effective in decap selection while maintaining the PDN stable and taking the cost into consideration. In Case-3, since we set the runtime limitation is 1 day and the runtime of SA exceeds the limited time, we do not show the SA result in Case-3 histogram.

5.2 PDC-PSO v.s. PSO v.s. SA in Lowering PDN Impedance

We apply these three algorithms PDC-PSO, PSO and SA to all cases and compare their effect in lowering PDN impedance. Figure 5.2 demonstrates that compared to PSO and SA, PDC-PSO has more possibility to find better solution. Table 5.3 shows that the arithmetic mean and best solution found by PDC-PSO running 50 times are better than PSO and SA. The value in Table 5.3 represents the over target impedance area, the lower value means better solution. In Case-3, since we set the runtime limitation is 1 day and the runtime of SA exceeds the limited time, we do not show the SA result in Case-3 histogram.

5.3 Optimized Decap for Voltage Fluctuation Reduction

In this experiment, we compare the effect of decap combination chosen by the experience of engineers, and our algorithm in lowering PDN impedance and reducing voltage fluctuation. Figure 5.3 is comparison of decap combination chosen by our algorithm and rules of thumb in all cases.

Figure 5.3(a) demonstrates that the design without decaps does not meet target

Table 5.3: Comparison of PDC-PSO, PSO and SA. The values are calculated according to objective function(Eq(3.3))

		SA	PSO	PDC-PSO
Case-1	**Best (x10^3)**	4.29	4.29	4.29
	Avg (x10^4)	7730.17	2.24	1.29
	runtime (hr)	6.5	2.5	2.5
Case-2	**Best (x10^7)**	1.94	1.69	1.67
	Avg (x10^7)	2.38	1.84	1.79
	runtime (hr)	14.00	3.00	3.00
Case-3	**Best (x10^6)**	N/A	5.62	3.84
	Avg (x10^6)	N/A	8.58	7.91
	runtime (hr)	> 24	11	11

impedance on 90MHz and 65MHz, and after inserting two decaps (GRM033R60J104KE84) selected by hand, the design does not meet the target impedance on 100MHz either. But with our program, we could observe that the result uses the same amount of decaps and makes the PDN impedance meet the target impedance.

Figure 5.3(b) and Figure 5.3(c) show that both the manual and our program result could not let PDN meet the target impedance, and it means the amount of predefined ports is not enough to make the PDN system stable in entire frequency range. The engineers should redesign the package or release the tolerance of voltage in this situation. However, our algorithm could lower the impedance around the operation frequency, and make the whole PDN impedance approach the target impedance closely. That means the relaxation of specification can be less.

5.4 Discussions

We run the SSN simulation to verify the decap combination for voltage fluctuation reduction, and the results are shown in Table 5.4, Figure 5.4, and Figure 5.5. We found that the improvement of Case-1 is minimal, it is because the PDN impedance at operation frequency is below the target impedance already. That means if there

is no other noise, the PDN system without decap is stable. However, besides the operation frequency, there are still many noise occurred anywhere unexpectedly. To prevent the unexpected noise from causing the PDN system unstable, we should be conservative and make the entire frequency range meet the target impedance.

From viewpoint of chip-package-PCB co-design, we should not just care the impedance at operation frequency, but mind the entire frequency range the unexpected noise would occur. Although using both the decap combinations selected by rules of thumb and our program could maintain the PDN system within the specification 300mV at operation frequency, as Figure 5.4(a) shows. The whole PDN system includes chip, package, PCB, and VRM, and the unexpected noise exists in low, middle, and high frequency. There might be unexpected noise in low, middle, and high frequency. Therefore, we measure the voltage fluctuation when there is a noise coming from PCB or chip at 90MHz, as Figure 5.4(b) shows. The manual selection is not effective to suppress the noise, and our result could still keep the PDN system voltage fluctuation under 300mV. Table 5.4 shows the improvement is 46.69%.

Another problem we should take care is that sometimes the performance of PDN with decaps is worse than PDN without decaps since the anti-resonance[25] might occur at the noisy frequency. As Figure 5.5(a) and Figure 5.5(b) show, the decaps selected by rules of thumb cause the voltage fluctuation larger than the original design without decaps. Table 5.4 shows the improvements of decaps selected by manual are -54.7% in Case-2 and -37.4% in Case-3. Therefore, choosing decap should consider its own characteristic rather than rules by thumb, or we may obtain the PDN system worse than the original design.

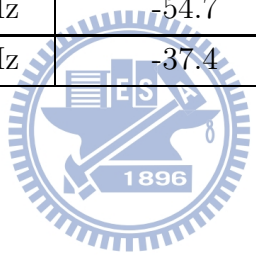
Table 5.4 shows the improvement of PDN system with and without decap optimization, and we could observe that with decap optimization by our program, the voltage fluctuation could be improved obviously and this could make the design

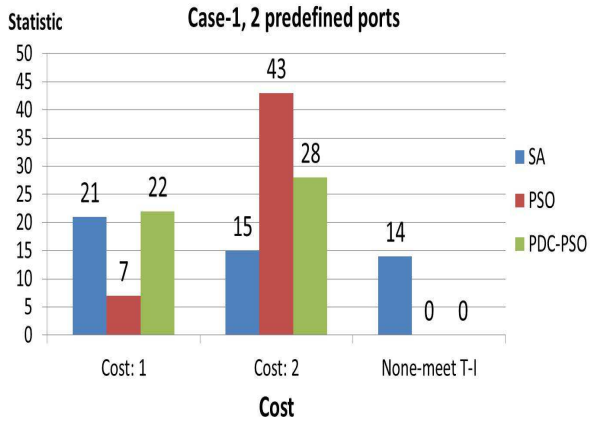
more robust at the same cost and avoid over-design.

Table 5.4: Peak-to-peak voltage fluctuation comparison

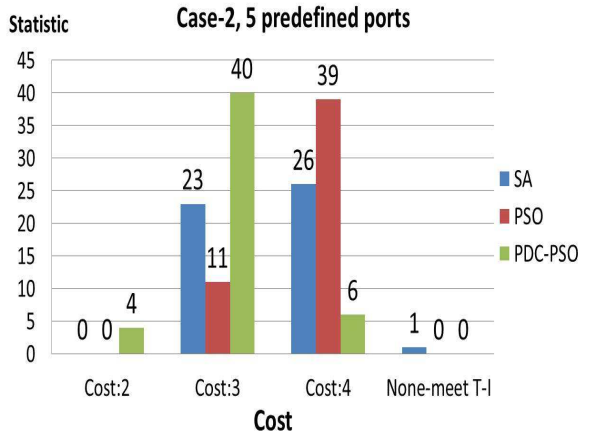
Case	**Operation Frequency**	**Without Decap**	**With Original Decaps**	**With PDC-PSO**
Case-1 (2-port)	800MHz	338mV	253mV	247mV
Case-1 (2-port)	90MHz	374mV	330mV	278mV
Case-2 (5-port)	100MHz	724mV	1120mV	386mV
Case-3 (16-port)	200MHz	270mV	371mV	179mV

Case	**Operation Frequency**	**Improvement of Original Decaps(%)**	**Improvement of PDC-PSO(%)**
Case-1 (2-port)	800MHz	25.15	26.92
Case-1 (2-port)	90MHz	11.76	25.67
Case-2 (5-port)	100MHz	-54.7	46.69
Case-3 (16-port)	200MHz	-37.4	33.7

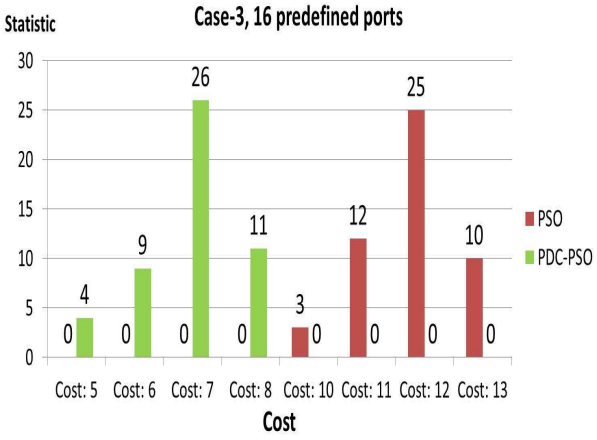




(a)

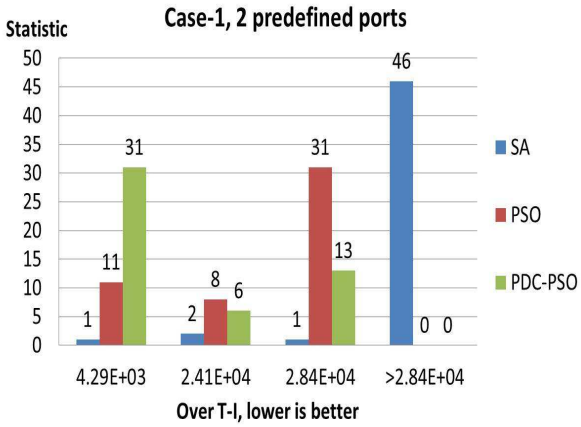


(b)

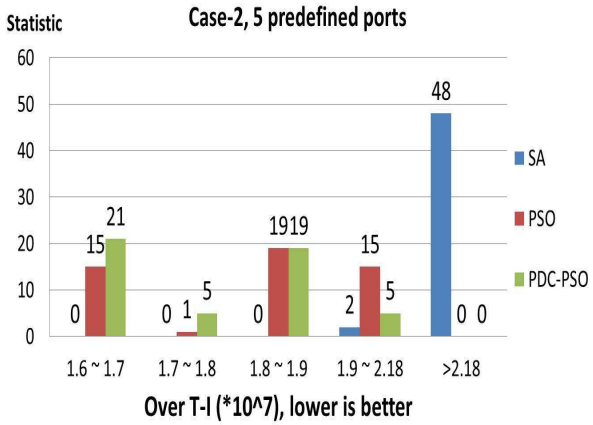


(c)

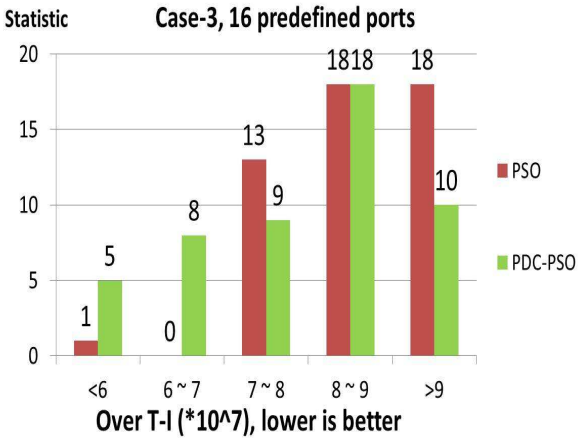
Figure 5.1: Comparison of decap combination cost chosen by PDC-PSO, PSO and SA. We run each algorithm 50 times and record its result. T-I means the target impedance. (a) is the algorithm comparison of Case-1. (b) is the algorithm comparison of Case-2. (c) is the algorithm comparison of Case-3.



(a)

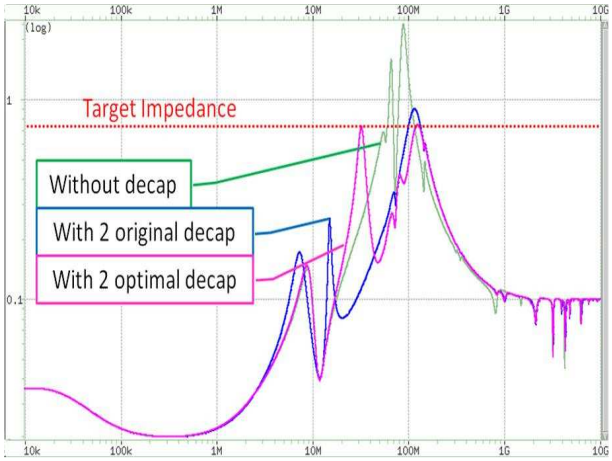


(b)

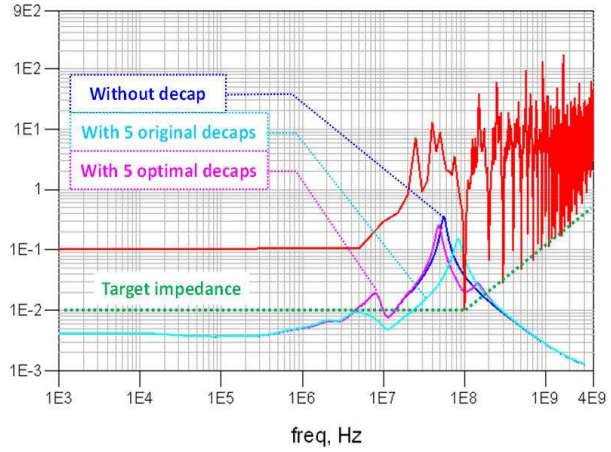


(c)

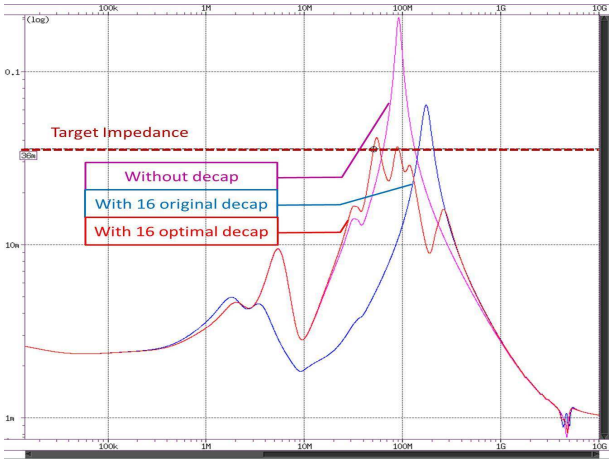
Figure 5.2: Comparison of PDC-PSO, PSO and SA in lowering PDN impedance. We run each algorithm 50 times and record its result. T-I means the target impedance. (a) is the algorithm comparison of Case-1. (b) is the algorithm comparison of Case-2. (c) is the algorithm comparison of Case-3.



(a)

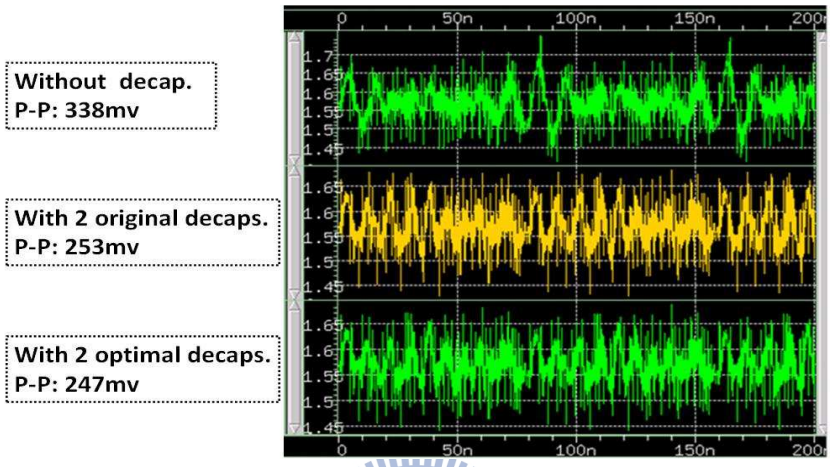


(b)

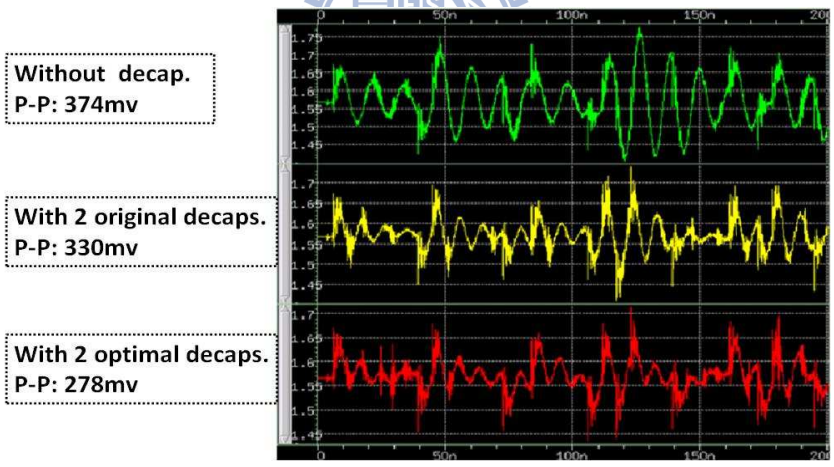


(c)

Figure 5.3: All cases frequency spectrum. (a) is the Case-1 comparison of original and optimal decap combination in frequency domain. (b) for Case-2. (c) for Case-3.

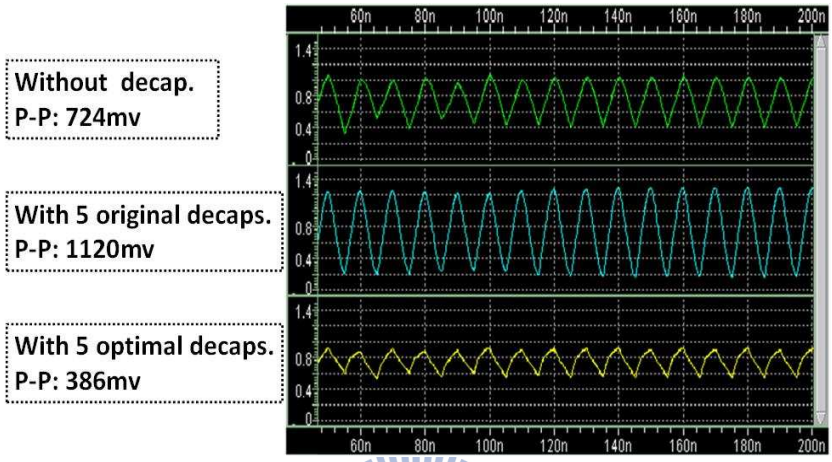


(a)

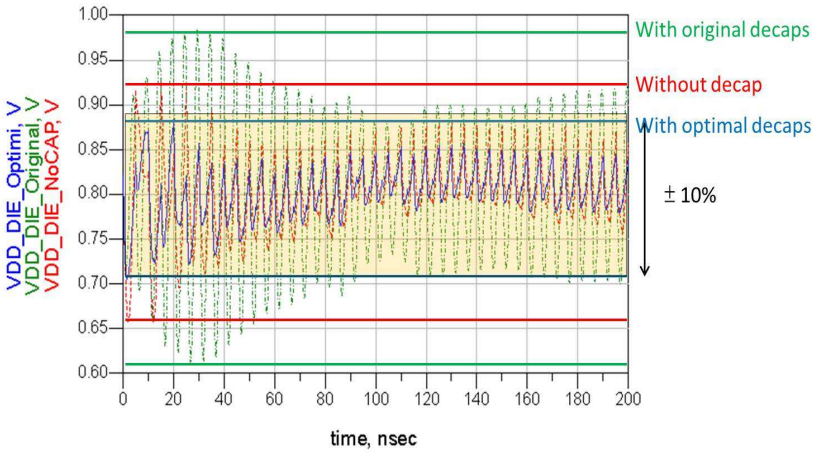


(b)

Figure 5.4: Case-1 time domain spectrum. P-P means peak-to-peak. (a) is the Case-1 time-domain comparison of original and optimal decap combination in 800MHz. (b) is the Case-1 time-domain comparison of original and optimal decap combination in 90MHz.



(a)



(b)

Figure 5.5: Case-2 and Case-3 time domain spectrum. P-P means peak-to-peak. (a) is the Case-2 time-domain comparison of original and optimal decap combination in 100MHz. (b) is the Case-3 time-domain comparison of original and optimal decap combination in 200MHz.

Chapter 6

Conclusions

A well-designed PDN is essential for high speed system. To maintain the power integrity, adding decaps is an effective way. Since the more decaps would cost more money and area, how to choose decaps becomes a critical issue. In this thesis, we introduce an efficient algorithm named "PDC-PSO" to optimize the type and location of decaps automatically. The results show that, compared to the decaps chosen by rules of thumb, our algorithm could effectively shrink the voltage fluctuation at pads on chip within the tolerable range at the same or lower price in a relatively short execution time.

Bibliography

[1] Istvan Novak, "Frequency-Domain Characterization of Power Distribution Networks," *Artech House Publishers*, 2007.

[2] Eric Bogatin, "Signal and Power Integrity - Simplified (2nd Edition)," *Prentice Hall*, 2009.

[3] Xiaoping Yang, Q. Chen, and C. Chen, "The optimal value selection of decoupling capacitors based on FDFD combined with optimization," in *Electrical Performance of Electronic Packaging*, pp. 191–194, 2002.

[4] R. Fizesan, and D. Pitica, "Simulation for power integrity to design a PCB for an optimum cost," in *International Symposium for Design and Technology in Electronic Packaging*, pp. 141–146, 2010.

[5] S. M. Nabil, A. B. El-Rouby, and A. Hussin, "A complete solution for the power delivery system (PDS) design for high-speed digital systems," in *International Conference on Design Technology of Integrated Systems in Nanoscal Era*, pp. 179–183, 2009.

[6] L. D. Smith, "Frequency Domain Target Impedance Method for Bypass Capacitor Selection for Power Distribution Systems," in *DesignCon*, 2006.

[7] Jun Chen, and Lei He, "Efficient In-Package Decoupling Capacitor Optimization for I/O Power Integrity," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 945–960, 2006.

[8] Jun Chen, and Lei He, "Experimental Analysis of Acceleration Coefficient in Particle Swarm Optimization Algorithm," in *Computer Engineering*, vol. 36, no. 4, 2010.

[9] Hui Zheng, B. Krauter, and L. Pileggi, "On-package decoupling optimization with package macromodels," in *Custom Integrated Circuits Conference*, pp. 723–726, 2003.

[10] J. N. Tripathi, R. K. Nagpal, N. K. Chhabra ,R. Malik , and J. Mukherjee, "Maintaining Power Integrity by damping the cavity-mode anti-resonances' peaks on a power plane by Particle Swarm Optimization," in *International Symposium on Quality Electronic Design (ISQED)*, pp. 525–528, 2012.

[11] Kai-Bin Wu, Gus-Hwa Shiue, and Ruey-Beei Wu, "Optimization for the Locations of Decoupling Capacitors in Suppressing the Ground Bounce by Genetic Algorithm," in *Progress In Electromagnetics Research Symposium*, 2007.

[12] K. Bharath, A. Ege Engin, and M. Swaminathan, "Automatic package and board decoupling capacitor placement using genetic algorithms and M-FDM," in *Design Automation Conference*, pp. 560–565, 2008.

[13] A. Ege Engin, "Efficient Sensitivity Calculations for Optimization of Power Delivery Network Impedance," in *IEEE Transactions on Electromagnetic Compatibility*, vol. 52, no. 2, pp. 332–339, 2010.

[14] Praveen Kumar Tripathi, Sanghamitra Bandyopadhyay, and Sankar Kumar Pal, "Multi-Objective Particle Swarm Optimization with time variant inertia and acceleration coefficients," in *Information Sciences*, vol. 177, no. 22, pp. 5033–5049, 2007.

[15] Zhengjia Wu, and Jianzhong Zhou, "A Self-Adaptive Particle Swarm Optimization Algorithm with Individual Coefficients Adjustment," in *International Conference on Computational Intelligence and Security*, pp. 133–136, 2007.

[16] L. D. Smith, R. E. Anderson, D. W. Forehand, T. J. Pelc, and T. Roy , "Power distribution system design methodology and capacitor selection for modern CMOS technology," in *IEEE Transactions on Advanced Packaging*, vol. 22, no. 3, pp. 284–291, 1999.

[17] Dirack Lai, "Achieve optimized power delivery using Adaptive target impedance," in *http://www.ansoft.com/firstpass/pdf/AchieveOptimizedPowerDelivery.pdf*, 2007.

[18] "Murata Manufacturing Co.," in *http://www.murata.com/*.

[19] J. Kennedy, and R. Eberhart, "Particle swarm optimization," in *IEEE International Conference on Neural Networks, Proceedings.*, vol. 4, pp. 1942–1948, 1995.

[20] Yiyu Shi , Jinjun Xiong, Chunchen Liu, and Lei He, "Efficient Decoupling Capacitance Budgeting Considering Operation and Process Variations," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 7, pp. 1253–1263, 2008.

[21] Hang Li, Zhenyu Qi, S. X. Tan, Lifeng Wu, Yici Cai, Xianlong Hong, "Partitioning-based approach to fast on-chip decap budgeting and minimization," in *Design Automation Conference*, pp. 170–175, 2005.

[22] Peng Li, "Design analysis of IC power delivery," in *International Conference on Computer-Aided Design*, pp. 664–666, 2012.

[23] Yiyu Shi, and Lei He, "Modeling and design for beyond-the-die power integrity," in *International Conference on Computer-Aided Design*, pp. 411–416, 2010.

[24] `ANSYS.http://www.ansys.com/`.

[25] Iijima You, Matsumura Masataka, and Sudo Toshio, "Anti-resonance peak damping of PDN impedance by on-board snubber circuits," in *Electrical Design of Advanced Packaging and Systems Symposium (EDAPS)*, pp. 127–130, 2012.

[26] Hao Yu, Chunta Chu, and Lei He, "Off-chip Decoupling Capacitor Allocation for Chip Package Co-Design," in *Design Automation Conference*, pp. 618–621, 2007.

[27] R. Heald, K. Aingaran, C. Amir, M. Ang, M. Boland, P. Dixit, G. Goudsberry, D. Greenley, J. Grinberg, J. Hart, T. Horel, W. J. Hsu, J. Kaku, Chin Kim, Song Kim, F. Klass, H. Kwan, G. Lauterbach, R. Lo, H. McIntyre, A. Mehta, D. Murata, S. Nguyen, Yet-Ping Pai, S. Patel, K. Shin, K. Tam, S. Vishwanthaiah, J. Wu, G. Yee, and E. You, "A third-generation SPARC V9 64-b microprocessor," in *IEEE Journal of Solid-State Circuits*, vol. 35, no. 11, pp. 1526–1538, 2000.

[28] A. Waizman, O. Vikinski, and G. Sizikov, "CPU Power Delivery Impedance Profile Resonances Impact on Core FMAX," in *IEEE Electrical Performance of Electronic Packaging*, pp. 119–122, 2006.

[29] Madhavan Swaminathan, and A. Ege Engin, "Power Integrity Modeling and Design for Semiconductors and Systems," *Academic Internet Publishers*, 2007.

[30] Shenheng Xu, and Y. Rahmat-Samii, "Boundary Conditions in Particle Swarm Optimization Revisited," in *IEEE Transactions on Antennas and Propagation*,vol. 55, no. 3, pp. 760–765, 2007.

[31] Gonzalo Napoles, Isel Grau, and Rafael Bello, "Constricted Particle Swarm Optimization based Algorithm for Global Optimization," in *Polibits, Research journal on Computer science and computer engineering with applications*,vol. 46, no. 1, pp. 5–11, 2012.