

Fast Decap Allocation Based on Algebraic Multigrid

Cheng Zhuo, Jiang Hu¹, Min Zhao² and Kangsheng Chen

Dept. of Information Science & Electronic Engineering, Zhejiang University, Hangzhou, 310027, China

¹Dept. of Electrical and Computer Engineering, Texas A&M University, College Station, TX, 77843

²Freescale Semiconductor, Inc., Austin, TX, 78729

{zhuocheng, chenks}@zju.edu.cn; ¹jianghu@ece.tamu.edu; ²minzhao@freescale.com

ABSTRACT

Decap (decoupling capacitor) is an effective technique for suppressing power supply noise. Nevertheless, over-usage of decap usually causes excessive power dissipation. Therefore, the total decap area needs to be minimized subject to power supply noise constraints. This is a complicated nonlinear optimization problem that may have as many as millions of variables. We propose an algebraic multigrid (AMG) based method to handle the high complexity. An error compensation scheme is developed to compensate the accuracy loss during the AMG reduction. A charge based back-mapping method and a few other techniques are suggested to further improve the computation efficiency. Our method is flexible to use and can be easily integrated with other existing decap allocation works. When compared to several previous works, the results from our method are usually the closest to the optimum. Our method also runs fast and can solve circuits with up to 1 million nodes in about 11 minutes. In addition, it has better scalability than the previous works.

1. INTRODUCTION

A stable power supply voltage is a fundamentally necessary condition for integrated circuits to maintain desired operating performance. This condition becomes increasingly difficult to reach when the VLSI technology scaling results in increasing power supply noise on one hand and decreasing noise margins on the other hand. The power noise can be alleviated by placing on-chip decoupling capacitors (decap), which behave like local charge reservoirs to cushion any ditch or overshoot of power supply level [1]. However, imprudent usage of decap may cause exorbitant power dissipation, especially leakage power [2]. Therefore, the area of decaps needs to be minimized while the constraint on power supply noise is satisfied. This is a difficult nonlinear optimization problem that may have as many as millions of variables.

Because of its importance and difficulty, the decap allocation problem has attracted many attentions recently. In [3, 4], a charge based model is proposed to approximately estimate the decap size for each circuit module. Other works [5, 6, 7, 8, 9] use adjoint sensitivity technique to guide the solution search in nonlinear optimization. Even though the number of transient simulations is greatly decreased by merged adjoint method [6, 7] or greedy search [7, 8], the sheer size of the problem still implies a huge computation cost. A rel-

atively direct approach is to reduce the problem size. In [8], the problem size reduction is achieved by divide-and-conquer assuming that the boundary voltages of each partition do not change during the decap optimization. The work of [9] uses geometric multigrid technique [10, 11] to reduce the problem size. However, the effectiveness of geometric multigrid is mainly restricted to regular power grids [12, 13]. Consequently, the method in [9] is not applicable for more general cases of irregular power grids [12, 13].

In this paper, we propose a fast decap allocation method based on algebraic multigrid (AMG) [14]. It first coarsens the original fine grid to a coarse grid that has manageable size. Then, a nonlinear programming is performed to obtain a decap allocation solution on the coarse grid. Last, the coarse solution is back-mapped to the original fine grid. The main contributions of this paper are summarized as follows.

- (1) We propose to utilize AMG based problem reduction that can be applied to either regular or irregular power grids. It is flexible to use and can be easily combined with other techniques such as the charge based technique [3, 4], conjugate gradient method [6, 7, 8] or the partitioning based approach [8].
- (2) We find the necessary condition that the solution on the coarse grid can lead to nonlinear programming solution on the fine grid. Based on this condition, an error compensation technique is developed and integrated with the nonlinear programming formulation on the coarse grid. This technique ensures that the solution on the coarse grid is still feasible after being mapped on the fine grid.
- (3) A new back-mapping scheme is proposed. Compared to previous work of linear programming based back-mapping [9], our charge based back-mapping method considers power noise and is much more scalable.
- (4) We introduce a few customized techniques to speed up sequential quadratic programming (SQP) which is employed to solve the nonlinear programming on the coarse grid. The technique such as violation aware decap pre-allocation could also possibly be applied to the other nonlinear optimizer.

2. REVIEW OF MULTIGRID METHOD

Multigrid is a technique to accelerate the convergence of solving differential equations numerically. The key idea of multigrid is to coarsen the original problem so that the low frequency errors appear to have high frequency and therefore can be quickly eliminated [10, 14]. The multigrid method can be categorized to geometric multigrid (GMG) and algebraic multigrid (AMG). GMG is based on geometric structure of the problem, preferably regular structures, while AMG is more flexible on handling general structures that may be irregular. Recently, AMG has been applied for fast power grid analysis [12, 13, 15].

3. PROBLEM FORMULATION

Power grid is usually a metal mesh where each edge can be modeled as a resistor. Each node of the mesh connects to a

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICCAD'06, November 5-9, 2006, San Jose, CA

Copyright 2006 ACM 1-59593-389-1/06/0011 ...\$5.00.

ground capacitance that represents the decap and parasitic capacitance. Active devices are modeled as timing varying current sources connected to the mesh. Through the power pad nodes, power grid is connected to the package, which could be modeled as voltage sources [16], or more complicated *RLCK* model.

The decap allocation problem is formulated as the following nonlinear programming problem, where \mathbf{C}^h is the decap vector, C_i^h is the i th element of \mathbf{C}^h , \mathbf{l}_b and \mathbf{u}_b represent the lower and upper bound for allowed decap size, respectively.

$$\text{Minimum} \quad \sum_{i \in PG^h} C_i^h \quad (1)$$

$$\text{Subject to:} \quad ceq(\mathbf{C}^h) = \sum_{i \in PG^h} s_i = 0 \quad (2)$$

$$\mathbf{l}_b < \mathbf{C}^h < \mathbf{u}_b \quad (3)$$

where $s_i = \int_0^T |\min(V_i(t) - V_{th}, 0)| dt = \int_{t1}^{t2} (V_{th} - V_i(t)) dt$ and $[t1, t2]$ is the time interval in which the violation occurs. This voltage drop noise metric is adopted from [5].

4. AMG BASED DECAP ALLOCATION

Since the size of a power grid is usually very large, sometimes with millions of nodes, solving the nonlinear programming of *DA* directly is extremely difficult. Therefore, we propose to reduce the problem size using the AMG technique. The reduced problem on the coarse grid is solved directly and the solution is mapped back to the original fine grid. Similar as other literatures, we use h to indicate fine grid and H for coarse grid.

4.1 Power Grid Reduction

For a power grid PG^h , by applying the modified nodal analysis (MNA), we may obtain the following equation[11]:

$$G\mathbf{V}(t) + C_0 \frac{d\mathbf{V}(t)}{dt} = \mathbf{I}(t) + \mathbf{U}(t) \quad (4)$$

where $\mathbf{V}(t)$ is the node voltage vector, G is the conductance matrix, C_0 is the capacitance matrix, $\mathbf{I}(t)$ is the current vector and $\mathbf{U}(t)$ indicates the items from the voltage sources.

In our AMG, we use G as the system matrix A^h to obtain the interpolation operator in the same method as in [12, 15]. Here, the interpolation operator is denoted as P_H^h . The reduction operator can be obtained easily as $R_h^H = (P_H^h)^T$. Then, the system A^h on the original fine grid can be reduced to a coarser grid as $A^H = R_h^H A^h P_H^h$. This procedure is repeated till the matrix is sufficiently small for direct solve.

Once the interpolation operators are obtained, the current source vector on the coarse grid can be obtained by:

$$\mathbf{I}^H(t) = R_h^H \mathbf{I}^h(t) \quad (5)$$

where $\mathbf{I}^h(t)$ is the current source vector on the fine grid. Correspondingly, the bounds for decap sizes are also updated to $\mathbf{l}_b^H < \mathbf{C}^H < \mathbf{u}_b^H$, where $\mathbf{l}_b^H = R_h^H \mathbf{l}_b$ and $\mathbf{u}_b^H = R_h^H \mathbf{u}_b$ [9].

In our AMG, the voltage source nodes are retained. The parasitic capacitance is used as the lower bound of decap optimization instead of being stamped in the capacitance matrix for the convenience of computation. Alternatively, the capacitance on the coarse grid can also be obtained by:

$$\mathbf{C}^H = R_h^H \mathbf{C}^h \quad (6)$$

4.2 Error Compensation

Obviously, the power grid reduction causes some information loss that is an inevitable price paid in change for the

improved computation speed. The question is how credible the solution obtained from the coarse grid is. For example, if we obtain an optimal solution on the coarse grid, does it imply an optimal solution on the fine grid? We will introduce a lemma and a theorem showing that the credibility of a coarse solution depends on the discrepancy between the transient response on the coarse grid and that on the fine grid. Based on that, we propose a compensation technique to reduce the error due to this discrepancy.

Lemma 1: *The sum of any row in the interpolation operator P_H^h (an $n \times m$ matrix and $n > m$) is 1.*

Proof: Omitted due to the page limit.

Theorem 1: *Suppose $\mathbf{V}(t)$ is the transient analysis result on the fine grid PG^h , and $\mathbf{V}^H(t)$ is the transient analysis result on the coarse grid PG^H . If the equation $\mathbf{V}(t) = P_H^h \mathbf{V}^H(t)$ holds, then the nonlinear programming *DA* on the fine grid is equivalent to the nonlinear programming *DA-Coarse* on the coarse grid which is:*

DA-Coarse:

$$\text{Minimum} \quad \sum_{i \in PG^H} C_i^H \quad (7)$$

$$\text{Subject to:} \quad ceq_H(\mathbf{C}^H) = \sum_{i \in PG^H} s_i^H = 0 \quad (8)$$

$$\mathbf{l}_b^H < \mathbf{C}^H < \mathbf{u}_b^H \quad (9)$$

where $s_i^H = \int_{t1}^{t2} (V_{th} - V_i^H(t)) dt$

Proof: Omitted due to the page limit.

When the nonlinear programming is carried out on the coarse grid as *DA-Coarse*, the corresponding transient analysis of the power grid is also performed on the coarse grid. Performing transient analysis on the coarse grid and directly mapping the result back to the original fine grid is equivalent to performing multigrid without the smoothing step. Omitting the smoothing often results in significant errors [12], *i.e.*, the condition $\mathbf{V}(t) = P_H^h \mathbf{V}^H(t)$ does not hold, especially when the current is not evenly distributed.

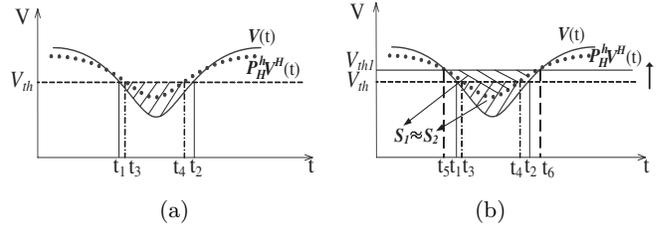


Figure 1: (a) Voltage drop noise metrics of $\mathbf{V}(t)$ and $P_H^h \mathbf{V}^H(t)$ (b) The raise of the threshold voltage

Figure 1(a) shows an example of the transient simulation results from $\mathbf{V}(t)$ and $P_H^h \mathbf{V}^H(t)$. It can be seen that the voltage $P_H^h \mathbf{V}^H(t)$ interpolated from the transient result on the coarse grid underestimates the voltage drop. We propose a compensation technique: raise the threshold voltage for the interpolated voltage by a constant δ such that its violation area is roughly equal to $ceq(\mathbf{C}^h)$. For example, in Figure 1(b), we raise the threshold voltage for the interpolated voltage from V_{th} to V_{th1} to make $S1 \approx S2$. Thus, we have:

$$\begin{aligned} ceq(\mathbf{C}^h) &= \sum_{i \in PG^h} \int_{t1}^{t2} (V_{th} - V_i(t)) dt \\ &\approx \sum_{i \in PG^h} \int_{t5}^{t6} (V_{th} + \delta - (P_H^h \mathbf{V}^H(t))_i) dt \end{aligned} \quad (10)$$

In order to avoid voltage drop violation or let Equation (10)

be zero, we just need to make $P_H^h \mathbf{V}^H(t) > V_{th} + \delta$, which is equivalent to requiring $\mathbf{V}^H(t) > V_{th} + \delta$.

Hence, the nonlinear programming on the coarse grid with error compensation is:

DA-Compensated:

$$\text{Minimum} \quad \sum_{i \in PG^H} C_i^H \quad (11)$$

$$\text{Subject to:} \quad ceq_H(\mathbf{C}^H) = \sum_{i \in PG^H} s_i^H = 0 \quad (12)$$

$$\mathbf{l}_b^H < \mathbf{C}^H < \mathbf{u}_b^H \quad (13)$$

where $s_i^H = \int_{t_5}^{t_6} (V_{th} + \delta - V_i^H(t)) dt$ and δ satisfies Equation (10).

The value of δ is different for each specific power grid with certain decap size. Moreover, the decap sizes are changed during the optimization. Therefore, it is not obvious how to choose the value of δ before the optimization.

With some numeric experiments, we notice that the maximum absolute value and average absolute value of $\delta(t) = P_H^h \mathbf{V}^H(t) - \mathbf{V}(t)$ decrease when the total decap area increases. When the decap area increases, more high frequency components of the voltage change is filtered out. Since the analysis on the coarse grid is relatively good at handling low frequency errors, the magnitude of the error or $\delta(t)$ consequently becomes smaller.

As a conservative approach, we obtain the value of δ based on using the minimum size for each decap. This scheme is to ensure that the error can always be fully compensated even in the worst case. Although it sometimes leads to some over-compensation, experimental results show that the degree of over-compensation is very limited in general. Through transient simulations, we can estimate the time-dependent error function $\delta(t)$. Then, a binary search is performed in the range between the average absolute value and the maximum absolute value of $\delta(t)$ so as to find a value of satisfying Equation (10).

4.3 Sequential Quadratic Programming Speedup

The compensated nonlinear programming problem *DA – Compensated* on the coarse grid is solved using a sequential quadratic programming (SQP) package [17]. The sensitivities of the voltage constraints with respect to decap sizes are calculated by the adjoint network method [5, 6]. SQP solves a nonlinear programming problem by converting it to a series of locally approximated quadratic programming problems [18]. Here, we suggest three simple yet effective speedup techniques to the SQP: (1) *violation aware decap pre-allocation*; (2) *variable removal for nodes at power pads*; (3) *search step scaling*.

SQP starts from an initial solution and then successively moves the solution toward the optimal point. Therefore, a good initial solution can greatly improve the convergence. We propose to find a good initial solution by pre-allocating more decap to nodes with relatively large current withdrawn during the violation time intervals:

- (1) Run transient simulation on the coarse grid with minimum-sized decap at each node.
- (2) Obtain the accumulated *violation charge* during the voltage violation period at each node: $Q_i^H = \int_{t_1}^{t_2} I_i^H(t) dt$ for each node i , where $[t_1, t_2]$ is the time interval of voltage drop violation.
- (3) Add decap equivalent to the amount of the charges during violation period: $\mathbf{C}_0^H = \mathbf{l}_b^H + \mathbf{Q}^H / (V_{DD} - V_{th})$.

During the power grid reduction of AMG, the nodes con-

nected to power pads are always kept [12, 15]. Thus, a significant portion of nodes in the coarse grid are connected to power pads. Since these nodes are directly connected to voltage sources in the circuit model, decaps at these nodes have no effect on the voltage drop there. Hence, we can remove many variables corresponding to these nodes in the nonlinear programming. Evidently, such removal may reduce the runtime of SQP.

Since the problem *DA – Compensated* is on the coarse grid, the allowed range ($\mathbf{u}_b^H - \mathbf{l}_b^H$) for each decap size is very large. This implies a large solution search space and slow runtime. In SQP [17], the searching step in each iteration is limited in order to maintain decent accuracy for the local quadratic approximation. We find that we can scale the searching step of [17] by a small factor $(1+\beta)$ with $\beta=0.1$ without significant influencing the solution quality. As a result, the computation speed is further improved.

4.4 Charge Based Back-mapping

After the solution for decap \mathbf{C}^H on the coarse grid is obtained, we need map it back to decap \mathbf{C}^h on the fine grid. The back-mapping is equivalent to spreading decaps on the coarse grid to nodes in the fine grid. Even though we know that $R_h^H \mathbf{C}^h = \mathbf{C}^H$, there are usually many solutions on the fine grid satisfying $R_h^H \mathbf{C}^h = \mathbf{C}^H$. The work of [9] finds a unique back-mapping by solving a linear programming of minimizing a weighted sum of total decaps area subject to $R_h^H \mathbf{C}^h = \mathbf{C}^H$. However, such back-mapping neglects the voltage drop constraint on the fine grid. Moreover, it tends to be very slow when the problem size is huge. For example, if there are one million nodes in the power grid, such back-mapping requires solving a linear programming with roughly one million variables.

We propose a charge based back-mapping. When we spread decaps on the coarse grid to nodes in the fine grid, we allocate greater portions to nodes with relatively large violation charge. Same as in Section 4.3, the *violation charge* for the nodes in the fine grid is the accumulated charge flowing into a node during the time interval of voltage drop violation: $Q_i^h = \int_{t_1}^{t_2} I_i^h(t) dt$ where $[t_1, t_2]$ is the time interval of voltage violation.

Procedure: <i>Charge Based Back-mapping</i>	
Input:	Interpolation operator P_H^h , the optimized decap vector \mathbf{C}^H on the coarse grid, the <i>violation charge vector</i> \mathbf{Q}^H on the coarse grid.
Output:	decap vector \mathbf{C}^h on the fine grid.
1:	For each node i in the coarse grid do
2:	Inspect the voltage waveform of node i to find the voltage violation interval $[t_1, t_2]$;
3:	For each node j that $P_H^h(j, i) \neq 0$ do
4:	$Q_j^h = \int_{t_1}^{t_2} I_j^h(t) dt$ is the <i>violation charge</i> ;
5:	end for
6:	$w = (C_i^H - l_{bi}^H) / Q_i^H$;
7:	$\mathbf{C}^h = \mathbf{C}^h + w \times P_H^h(:, i) \circ \mathbf{Q}^h$, where \circ represents entry-wise product, and $P_H^h(:, i)$ represents the i th column of P_H^h ;
8:	end for
9:	$\mathbf{C}^h = \mathbf{C}^h + \mathbf{l}_b$, check if $\mathbf{l}_b < \mathbf{C}^h < \mathbf{u}_b$;
10:	For those node that $\mathbf{C}^h > \mathbf{u}_b$ or $\mathbf{C}^h < \mathbf{l}_b$ do
11:	Average the redundant parts with its neighboring nodes until $\mathbf{l}_b < \mathbf{C}^h < \mathbf{u}_b$;
12:	end for

Figure 2: Algorithm for charge based back-mapping

Our proposed charge based back-mapping is outlined in Figure 2. Because the computations here are mostly vector operations and the violation time intervals are already avail-

able from the decap pre-allocation described in Section 4.3, our back-mapping is much faster and more scalable than the linear programming based back-mapping in [9].

Since $R_h^H \mathbf{I}^h(t) = \mathbf{I}^H(t)$, the violation charge on the coarse grid satisfies $Q_i^H = \int_{t_1}^{t_2} I_i^H(t) dt = \int_{t_1}^{t_2} R_h^H(i, :) \mathbf{I}^h(t) dt$, where $R_h^H(i, :)$ is the i th row of R_h^H . The violation time interval on the fine grid is approximated by that on the coarse grid. Then, we have:

$$\begin{aligned} Q_i^H &= \sum_j \int_{t_1}^{t_2} R_h^H(i, j) I_j^h(t) dt \\ &= \sum_j \int_{t_1}^{t_2} P_H^h(j, i) I_j^h(t) dt = \sum_j P_H^h(j, i) Q_j^h \end{aligned} \quad (14)$$

where $P_H^h(j, i)$ is the element at the j th row and i th column of P_H^h . This indicates that the total decap area remains the same after the back-mapping.

5. EXPERIMENTAL RESULTS

The proposed method was implemented in C language with numeric libraries TAUCS [19] and RFSQP [17]. The experiments were carried out on PC with Pentium IV 2.6G CPU, 1GB memory and Windows operating system. The experiments are performed on 5 small testcases and 5 large testcases. All of them are in mesh structure with some local irregularities. The voltage supply is 1.8V. The information of the testcases, including the number of nodes, the threshold of voltage drop, the maximal violation time intervals and the percentage of violation nodes, are shown Table 1. The threshold of voltage drop is set artificially such that about 20-35% nodes have violations for each testcase.

Table 1: The testcases information

Case	#nodes	Threshold (mV)	Max(t2-t1) (ns)	Violation (%)
1	100	30	0.20	21.0%
2	225	58	0.20	27.6%
3	361	77	0.25	26.3%
4	625	75	0.35	31.2%
5	729	70	0.45	34.7%
6	10.0K	92	0.45	31.7%
7	62.5K	92	0.40	27.9%
8	108.9K	92	0.40	30.7%
9	324.9K	122	0.70	32.5%
10	1.0M	130	0.95	21.7%

5.1 Comparison with Previous Works

In order to demonstrate the efficiency of our AMG based method, we implemented and compared with the following previous works:

- **CG**: solving decap allocation problem using standard conjugate gradient method as in [6]. It generally provides very good results, but is relatively slow.
- **iCG**: the improved conjugate gradient method proposed in [8]. By replacing standard line search in CG with greedy search step sizes, it is significantly faster than CG. However, it tends to over-budget decap area.
- **Theta**: the charge based method introduced in [4]. It runs very fast, but may result in large over-estimation on decap area.

The comparisons are shown in Table 2. The data shows that our method runs much faster than CG which cannot even complete for the largest two cases. Our method is also faster than iCG on large cases. It is slower than Theta, but the gap becomes smaller when the size of power grid is increased. In fact, its runtime is less than that of Theta on the largest case. This is because the inaccurate decap

estimation of Theta method incurred multiple iterations of transient analysis and corrections. One can also observe that the speedup of our method vs. CG and iCG increases for large circuits. These evidences indicate that our method has better scalability than the other methods. This is an appealing feature when we face increasingly large circuit designs.

The solution quality are evaluated in terms of total decap area and *voltage slack*, which is the threshold of voltage drop minus worst voltage drop after optimization. It can be seen that our method almost always provides the minimum decap area except case 6. In contrast, Theta results in unnecessarily large decap area 30%~77% more than our method. With the similar or smaller decap, the voltage slack after being optimized by our proposed AMG is usually larger than that of CG. It indicates decaps are more reasonably allocated by our AMG method.

5.2 Effectiveness of AMG and Proposed Speedup for Sequential Quadratic Programming

The comparisons in this section are made to investigate effectiveness of using AMG and the SQP speedup techniques proposed in section 4.3. We compare the full version of our method with the following variants in Table 3:

- **SQP**: solving the decap allocation through SQP package [17] without using AMG and without using the speedup techniques introduced in Section 4.3.
- **Our SQP**: solving the decap allocation through SQP package [17] with the speedup techniques introduced in Section 4.3, but without using AMG.

Since the SQP package [17] is not able to handle large cases directly, the comparisons are made for only the small cases. The SQP [17] yields the minimum decap area, but has minor voltage violations. This is because the original SQP [17] terminates whenever the nonlinear constraint is smaller than a predefined ϵ .

The comparison of AMG results with that of our SQP shows excellent speed/quality compromise of the proposed AMG techniques. The comparison of the original SQP results with that of our SQP implies that the speedup techniques of Section 4.3 are very effective on improving the computation speed with very minor overestimation. Overall, the over-estimation on decap area from our method is no more than 6%. With the increase of the power grid size, we expect to see even larger speedup with our AMG method.

5.3 Other Details of Our AMG Based Method

Table 4 displays other details of our AMG based method including the number of iterations for the grid reduction, the number of nodes after reduction, the compensation constant described in Section 4.2, the number of iterations for the SQP and the amount of total decap allocated. It can be seen that the coarse grids usually contain only hundreds of nodes.

Table 4: Detail results of our AMG based method

Case	#iter. in reduction	#nodes after reduction	δ (mV)	#iter. SQP	Decap (nF)
1	2	42	5.9	8	0.4
2	2	93	8.1	20	1.2
3	3	118	11.3	7	1.6
4	5	139	17.8	21	2.7
5	6	127	18.8	19	3.0
6	13	348	25.1	32	42.1
7	23	503	28.3	31	271.0
8	25	771	28.7	17	478.0
9	30	684	20.2	71	1270.0
10	37	763	10.4	103	3910.0

Table 2: Comparison among CG, iCG, Theta and the proposed AMG based method

Case	CPU Time (sec) / speedup factor							Normalized decap area				Voltage slack(mV)			
	AMG	CG		iCG		theta		AMG	CG	iCG	theta	AMG	CG	iCG	theta
1	1.3	7.4	5.7x	1.4	1.1x	0.06	0.05x	1	1.02	1.16	1.3	0.3	0	1.1	3.3
2	2.5	14.6	5.8x	2.2	0.88x	0.06	0.02x	1	1.02	1.12	1.63	0.5	0.2	2.7	16.6
3	2.9	16	5.5x	2.3	0.79x	0.08	0.03x	1	1.04	1.03	1.64	1.5	0.8	0.5	25.3
4	4.4	18.6	4.2x	2.5	0.57x	0.1	0.02x	1	1.05	1.15	1.61	4.6	0.8	6.6	21.4
5	4.6	27.5	6.0x	2.6	0.57x	0.2	0.04x	1	1.05	1.11	1.68	3.3	0.8	3.9	22
6	42.7	91.4	2.1x	9.5	0.22x	1.8	0.04x	1	0.98	1.03	1.59	0.7	1.3	1.6	2
7	47.4	783	16.5x	56.6	1.2x	17	0.36x	1	1.53	1.51	1.47	4.6	1.7	0	2.4
8	62.3	1415	22.7x	101	1.6x	34.8	0.56x	1	1.05	1.41	1.51	3.9	0.2	0.7	2.8
9	211		×	538	2.6x	162	0.77x	1	×	1.20	1.77	0.1	×	0.6	2.6
10	674		×	4786	7.1x	1601	2.4x	1	×	1.12	1.48	0.5	×	0	1.2

Table 3: Comparison among SQP, our SQP and the proposed AMG based method

Case	CPU Time (sec) / speedup factor						Normalized decap area			Voltage slack(mV)		
	AMG	SQP		our SQP		AMG	SQP	our SQP	AMG	SQP	our SQP	
1	1.3	2.1	1.6x	1.6	1.2x	1	0.94	0.96	0.3	-0.1	0.5	
2	2.5	16.8	6.7x	3.2	1.3x	1	0.97	0.99	0.5	-0.2	0.2	
3	2.9	113	39.0x	10.9	3.8x	1	0.98	1.0	1.5	-0.2	1.6	
4	4.4	792	180.0x	63.6	14.4x	1	0.94	0.98	4.6	-0.2	2.4	
5	4.6	1360	295.7x	177	38.5x	1	0.94	0.96	3.3	-0.2	0.2	

5.4 Charge Based vs. Linear Programming Based Back-mapping

In Table 5, we compare our charge based back-mapping and the linear programming (LP) based back-mapping [9] on the four large cases. Since both mappings give the same total decap area, we compare only the CPU time and the voltage slack here. It is evident that our charge based back-mapping provides better voltage slack at faster speed.

Table 5: Comparison between the charge based and linear programming based back-mapping

Case	CPU Time (sec)		Voltage slack (mV)	
	Charge based	Linear programming	Charge based	Linear programming
6	<0.1	2.86	0.7	0.4
7	0.24	16.97	4.6	3.8
8	0.42	33.84	3.9	2.8
9	0.62	261.27	0.1	0.0

6. CONCLUSIONS

In this paper, we propose an algebraic multigrid based decap allocation method. The condition for the equivalency between the nonlinear programming on the fine grid and that on the coarse grid is derived. An error compensation technique is developed to make the method robust to uneven current distributions. Furthermore, we also propose a new back-mapping scheme, which is faster and provides better solution than the previous work. Our method can be applied to irregular power grids and can be easily combined with other existing techniques. Experimental results confirm that our method can lead to high quality solution at fast speed and it has better scalability than several previous works

7. REFERENCES

- [1] S. Bobba, T. Thorp, K. Aingaran, and D. Liu. IC power distribution challenges. In *Proc. ICCAD*, pages 643–650, 2001.
- [2] Q. K. Zhu. *Power Distribution Network Design for VLSI*. John Wiley, 2004.
- [3] H. H. Chen and D. D. Ling. Power supply noise analysis methodology for deep-submicron vlsi chip design. In *Proc. DAC*, pages 638–642, 1997.
- [4] C. K. S. Zhao, C.-K. Koh, and K. Roy. Decoupling capacitance allocation and its application to power-supply

noise-aware floorplanning. *IEEE Trans. on Computer-Aided Design*, 21(1):81–92, Jan. 2002.

- [5] H. Su, S. S. Sapatnekar, and S. R. Nassif. Optimal decoupling capacitor sizing and placement for standard cell layout designs. *IEEE Trans. on Computer-Aided Design*, 22(4):428–436, Apr. 2003.
- [6] J. Fu, Z. Luo, X. Hong, Y. Cai, S. X.-D. Tan, and Z. Pan. A fast decoupling capacitor budgeting algorithm for robust on-chip power delivery. In *Proc. ASP-DAC*, pages 505–510, 2004.
- [7] Z. Qi, H. Li, S. X.-D. Tan, L. Wu, Y. Cai, and X. Hong. Fast decap allocation algorithm for robust on-chip power delivery. In *Proc. ISQED*, pages 542–547, 2005.
- [8] H. Li, Z. Qi, S. X.-D. Tan, L. Wu, Y. Cai, and X. Hong. Partitioning-based approach to fast on-chip decap budgeting and minimization. In *Proc. DAC*, pages 170–175, 2005.
- [9] K. Wang and M. Marek-Sadowask. On-chip power supply network optimization using multigrid-based technique. *IEEE Trans. on Computer-Aided Design*, 24(3):407–417, Mar. 2005.
- [10] W. Hackbusch. *Multigrid Methods*. Science Press, Beijing, China, 1988.
- [11] J. N. Kozhaya, S. R. Nassif, and F. N. Najm. A multigrid-like technique for power grid analysis. *IEEE Trans. on Computer-Aided Design*, 21(10):1148–1160, Oct. 2002.
- [12] H. Su, E. Acar, and S. R. Nassif. Power grid reduction based on algebraic multigrid principles. In *Proc. DAC*, pages 109–112, 2003.
- [13] Z. Zhu, B. Yao, and C. K. Cheng. Power network analysis using an adaptive algebraic multigrid. In *Proc. DAC*, pages 105–108, 2003.
- [14] K. Stuben. *Algebraic Multigrid (AMG): An Introduction with Applications*, volume Guest appendix, in *Multigrid Methods*. Academic Press, New York, 2000, also available as GMD Report 70, Nov. 1999.
- [15] C. Zhuo, J. Hu, and K. Chen. An improved AMG-based method for fast power grid analysis. In *Proc. ISQED*, pages 290–295, 2006.
- [16] H. H. Chen and J. S. Neely. Interconnect and circuit modeling techniques for full-chip power noise analysis. *IEEE Trans. on Components Packaging II*, 21, Issue 3:209–215, Aug. 1998.
- [17] <http://aemdesign.com/downloadrfsqp.htm>.
- [18] Y. Yuan and W. Sun. *Optimization Theory and Method*. Science Press, Beijing, China, 1997.
- [19] <http://www.tau.ac.il/stoledo/taucs/>.