Power Grid Optimization in 3D Circuits Using MIM and CMOS Decoupling Capacitors

Pingqiang Zhou
Department of ECE
University of Minnesota
Minneapolis, MN 55455
pingqiang@umn.edu

Karthikk Sridharan
Department of ECE
University of Minnesota
Minneapolis, MN 55455
sridh019@umn.edu

Sachin S. Sapatnekar
Department of ECE
University of Minnesota
Minneapolis, MN 55455
sachin@umn.edu

Abstract— In three-dimensional (3D) chips, the amount of supply current per package pin is significantly more than in two-dimensional (2D) designs. Therefore, the power supply noise problem, already a major issue in 2D, is even more severe in 3D. CMOS decoupling capacitors (decaps) have been used effectively for controlling power grid noise in the past, but with technology scaling, they have grown increasingly leaky. As an alternative, metal-insulator-metal (MIM) decaps, with high capacitance densities and low leakage current densities, have been proposed. In this paper, we explore the tradeoffs between using MIM decaps and traditional CMOS decaps, and propose a congestion-aware 3D power supply network optimization algorithm to optimize this tradeoff. The algorithm applies a sequence-of-linear-programs based method to find the optimum tradeoff between MIM and CMOS decaps. Experimental results show that power grid noise can be more effectively optimized after the introduction of MIM decaps, with lower leakage power and little increase in the routing congestion, as compared to a solution using CMOS decaps only, and motivate the stronger need for these decaps in 3D technology, as compared to 2D designs.

Index Terms—MIM decap, CMOS decap, Power grid, 3D

I. Introduction

Three dimensional (3D) circuit technologies, with multiple tiers of active devices stacked above each other, are a key approach to increased levels of integration and performance in the future. However, there are two significant limitations that 3D technologies must overcome before achieving their full potential, related to on-chip thermal issues and reliable power delivery. Both issues can be illustrated through a simple back-of-the-envelope calculation. A k-tier 3D chip that stacks k similar chips could use k times as much current as a single 2D chip of the same footprint. However, the packaging technology is not appreciably different: with a similar heat sink, the on-chip temperature on such a 3D chip can be up to k times higher than the 2D chip, and with a similar number of pins in the package, the current per pin is k times higher than the 2D case.

The above analysis operates under very coarse assumptions (for example, a smart 3D designer may not stack k layers with identical power levels), and a more nuanced approach is necessary for a more accurate analysis – but the eventual conclusions that thermal and power delivery issues are important in 3D – are inescapable. While much research has been conducted on thermal management strategies such as thermal via insertion, and the spatial distribution of power sources, the power delivery problem has attracted limited attention to date.

The power delivery problem can be summarized as follows. The parasitics in the power network, together with temporal variations in the current drawn by a circuit, result in a time-varying voltage drop/surge at nodes in the power grid. These variations can adversely impact the performance and the reliability of a circuit. Such shifts become more acute with technology scaling: on the one hand, noise margins become more stringent with reducing Vdd levels, and on the other hand, with increased switching speeds and larger currents, IR, LdI/dt, and electromigration effects become more prominent. In 3D circuits, robust power supply network design is more challenging, and significant resources have to be invested in building a bulletproof power grid for the 3D chip.

Several techniques are available to increase the reliability of power grids and control power grid noise, such as wire widening, grid topology optimization, and decap insertion. Of these techniques, decaps are arguably the most powerful method for reducing transient noise, and are therefore addressed in this paper. Decaps serve as local current reservoirs, and can be used to satisfy sudden surges in current demand by the functional blocks/cells, while keeping supply voltage levels relatively stable.

Conventional technologies for implementing decaps are based on SiO_2-based structures that are widely used in robust power delivery network design. In the recent past, the CMOS decap allocation and optimization problem has been investigated by numerous researchers for 2D [1]–[3] and 3D technologies [3]–[5].

Unlike the 2D case, new considerations come into play while optimizing a 3D power grid using CMOS decaps. Since CMOS decaps are usually fabricated using white space on the device layer, they must compete for area with through-silicon vias, or with the landing pads of 3D vias, for the limited white space. This leads to a new resource contention problem. One way to resolve this contention problem is to increase the chip size in order to make room for CMOS decaps. However, one of the advantages of 3D circuits over 2D implementations is their reduced chip footprint: increasing the chip size may counteract this benefit. Leakage power is an important issue in 3D circuit design. The CMOS decaps added to the 3D circuit will consume extra leakage power, and make things worse. While new high-k dielectrics have been proposed, they will provide temporary relief to the gate leakage problem.

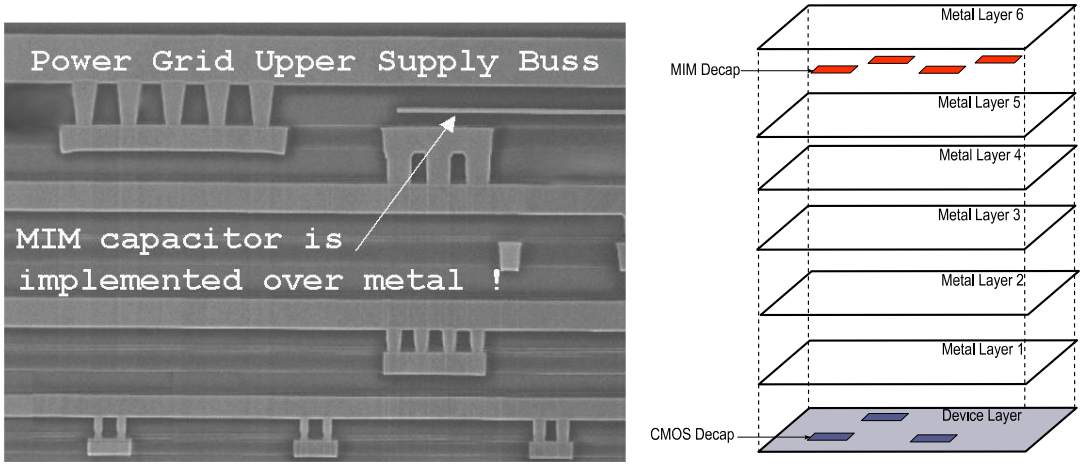


Fig. 1. (a) Schematic of a MIM decap [6]. (b) MIM and CMOS decaps in one 2D tier with 6 metal layers.

In this work, we address all of these issues. One of the novel features of our work is that it optimizes the power supply network using both conventional CMOS decaps and MIM decap technology, illustrated in Figure 1, which has high capacitance density and low leakage current density [6]–[8]. However, since MIM decaps are built between layers of metal interconnects, they present routing blockages to nets that attempt to cross them, and therein lies the tradeoff. The properties of MIM decaps makes them attractive for both 2D and 3D chips, but we pay particular attention to the 3D decap problem in this paper because (i) the power integrity problem is particularly critical in 3D, and requires novel approaches that leverage advances in materials, and (ii) the added complexity of handling routing blockages in a very constrained environment makes the 3D problem especially challenging.

We formulate the decap budgeting problem as a Linear Programming (LP) problem, and propose an efficient congestion-aware algorithm to optimize the power supply noise, while trying to find a balance between the routing congestion deterioration and leakage power increase.

II. Problem Formulation

We tile the layout using an uniform grid G' that is coarser than the original power grid, G, so that each tile of G' contains less than 20 power nodes in G. Our algorithm proceeds iteratively, adding a small amount of decap to the circuit in each iteration. An observation node is dynamically chosen from G for each tile in G', and all newly added decaps in this tile are connected to this observation node in each iteration. This helps in reducing the number of possible decap insertion spots, thus controlling the size of the problem that we solve.

A. Objective function

A key metric for the objective function is the noise violation area for the circuit. Given the transient voltage waveform, $v_i(t)$, at each node i of the power grid, the *violation area*, S_i, at the node is given by [1]:

$$S_i = \sum_j \int_{t_{s,j}}^{t_{e,j}} max\{V_{limit} - v_i(t),\ 0\}\, dt \tag{1}$$

where, V_{limit} is the voltage threshold, usually set to be 90% of V_{dd}, and $[t_{s,j},\ t_{e,j}]$ is the j^{th} interval during which the constraint is violated. The noise violation area, S, is the sum of S_i over all nodes. The goal of our optimization is to reduce the violation area to zero at all nodes, with optimal resource usage.

We denote the newly added CMOS and MIM decaps in tile k *by* Δx_k and Δy_k, respectively in each iteration. Let $S = \sum_{i=1}^{n} S_i$ be the total violation area over all the n nodes in the supply grid. The objective function in each iteration is to minimize the total increase in the leakage power, ΔP, while maximizing the reduction in the noise violation area, S, i.e.,

$$\text{minimize} \qquad \alpha \cdot \Delta S_{scaled} + (1-\alpha) \cdot \Delta P_{scaled} \tag{2}$$

Here, $\alpha\ is$ a weighting parameter that sets the objective to be a convex combination of the scaled noise violation metric, ΔS_{scaled}, and the scaled leakage power, ΔP_{scaled}, where the scaling ensures that the magnitudes of the two terms are similar. The parameter ΔS is the change in S when a small amount of CMOS decap and/or MIM decap is added to each tile k. Since the amount of decap inserted in each iteration is small, this change may be computed as

$$\Delta S = \sum_{k=1}^{m'} \{(\partial S/\partial x_k) \cdot \Delta x_k + (\partial S/\partial y_k) \cdot \Delta y_k\} \tag{3}$$

where m' is the number of tiles in G', $\partial S/\partial C$ is the sensitivity of S with respect to the decap $C \in \{x_k, y_k\}$, and Δx_k and Δy_k are as defined above. We note that $\partial S/\partial x_k$ and $\partial S/\partial y_k$ are nonpositive, since the violation area must decrease when decaps are added to the circuit. Therefore, minimizing ΔS, which is nonpositive, implies that we maximize the absolute reduction in S.

The leakage ΔP is calculated as $\sum_{k=1}^{m'}(a_k \cdot \Delta x_k + b_k \cdot \Delta y_k)$. In other words, it is the weighted sum of the increase in leakage due to the newly added decaps Δx_k and Δy_k. The weights a_k and b_k are given by

$$a_k = \frac{LD_{CMOS}}{CD_{CMOS}} \cdot \phi(T_k) \tag{4}$$

$$b_k = \frac{LD_{MIM}}{CD_{MIM}} \cdot \phi(T_k) \tag{5}$$

Here, LD_{CMOS}, LD_{MIM}, CD_{CMOS}, and CD_{MIM} are, respectively, the leakage densities of CMOS and MIM decaps, and the capacitance densities of CMOS and MIM decaps, and T_k is the average temperature in the tile k. The ratio $\frac{\Delta x_k}{CD_{CMOS}}$ provides the area of the added decap, which when multiplied by LD_{CMOS} determines the corresponding leakage. The penalty term $\phi(T_k) = T_k^2 \cdot exp(\mu/T_k^2)$ captures the effect of temperature on each leakage term, where μ is a constant negative number [9]. A higher temperature T_k corresponds to a larger $\phi(T_k)$, which means that the increase in leakage in tile k is controlled more strictly.

Considering that ΔS and ΔP may have different orders of magnitude, to better control the coefficients of the objective function, we scale them to ΔS_{scaled} and ΔP_{scaled} respectively. We normalize $\partial S/\partial x_k$ and $\partial S/\partial y_k$, scaling them by $\max\{|\partial S/\partial x_k|, |\partial S/\partial y_k|\}$ so that $|\partial S/\partial x_k|$, $|\partial S/\partial y_k|$ lie in $[0, 1]$. Similarly, a_k and b_k are also scaled by the factor $\max\{a_k, b_k\}$ so that they lie in $[0, 1]$.

The weighting parameter α is dynamically adjusted: at the beginning of the optimization, it is likely that CMOS decap resources are freely available and distributed over the whole chip area, but the leakage power cost is large if we use CMOS decaps to eliminate the noise. Therefore, we choose to use small α (in the range of 0.1 to 0.2) to control the increase of leakage power, and prefer to use MIM decaps at this stage. As the optimization proceeds, since more of the noise violations are eliminated and more regions become congested after inserting MIM decaps, we increase α to use more CMOS decap to help eliminate the remaining noise. At each iteration, we track n_{vio}, the number of grids that contain violating nodes, and n_{mim}, the number of grids where MIM decaps are inserted; if n_{mim}/n_{vio} is less than 5%, then we increase α by 0.1, unless it is already equal to 1.0.

B. Constraints

1) *Congestion constraints*. Since the MIM decaps inserted between metal layers may become potential routing blockages, it is necessary to impose a constraint that restricts the deterioration of congestion with MIM decap insertion. This constraint is written as:

$$\Delta Cong_k \leq \gamma \cdot Cong_k \tag{6}$$

where $Cong_k$ is the current congestion value in tile k, $\Delta Cong_k$ is the change of the congestion in tile k in the current iteration, and γ is a bounding parameter, which is empirically set to be 0.03 to 0.05 in our experiments.
Since each iteration imposes only a small change in the inserted decaps, it is reasonable to formulate $\Delta Cong_k$ as a linear function of the inserted MIM decaps $\Delta Cong_k = \sum_{i \in R_k}(c_i \cdot \Delta y_i)$, where the set R_k and the justification for this term are described in detail in Section III.

2) *Decap resource constraints*. For a tile k, the amount of CMOS decap that can be used is limited by its available white space, and the amount of MIM decap is restricted by its capacity. If C_{CMOS}^k and C_{MIM}^k are the current maximum allocatable amount of CMOS and MIM decaps in tile k, then the decap resource constraints for tile k can be formulated as:

$$0 \leq \Delta x_k \leq \min\{\Delta_{CMOS}, C_{CMOS}^k\} \tag{7}$$

$$0 \leq \Delta y_k \leq \min\{\Delta_{MIM}, C_{MIM}^k\} \tag{8}$$

where Δ_{CMOS} and Δ_{MIM} are upper bounds that are chosen to control the amount of CMOS and MIM decaps inserted in each iteration.

Equations (2)-(8) together formulate a linear programming problem.

III. Congestion Analysis And Linear Congestion Model

We estimate the routing congestion for decap optimization in 3D circuits using a probabilistic method, similar to [10], extended to 3D. However, any other congestion predictor could be used to replace this estimator with relatively few modifications, leaving the overall methodology unchanged. Given a placed 3D netlist, the core area is discretized using a 3D mesh, and the congestion in each tile of this mesh is estimated. For the purposes of our algorithm, the congestion in the Z direction is the

most important: since the uppermost two layers primarily consist of supply/clock wires rather than signal wires within a single tier, MIM decaps primarily affect signal routes in the Z direction. However, other terms in the objective function can act to provide disincentives to large area capacitors which would create significant bottlenecks to power/clock wires in the X, Y, and Z directions as well.

The initial congestion map for the circuit is thus calculated, and is predicated on the assumption that there are no blockages in the region. However, in case the design uses IP blocks that impose blockages for decaps, this information may easily be incorporated into the congestion estimator. When a MIM decap is inserted, it results in a blockage and causes a perturbation in the congestion values. We model this change in the congestion in a tile cell, assuming a small perturbation as a linear function. We now describe the procedure used to calculate this linear function using the initial congestion map.

Let R_k be a set of tile cells (including k) within a specified Manhattan radius, $maxDist$, of a tile cell k. We assume that the size of R_k is bounded by a small number, reflecting the fact that we operate under small perturbations that do not cause widespread congestion changes far away from k. For each tile cell $i \in R_k$, let W_i be the current number of routes in tile cell i, and let $CurCap_i$ and $NewCap_i$ be the current and new routing capacities in tile cell i after the insertion of a MIM decap.

Let ΔW_i be the number of routes in the tile cell i to be redistributed. The redistribution process proceeds as follows after a small additional MIM decap, Δy_i, is inserted in tile cell i. If W_i is smaller than the current capacity, $CurCap_i$, then none of the routes in tile i need to be redistributed but the congestion values are updated to reflect the reduction in the capacity. Otherwise, it is necessary for routes in tile i to be redistributed. The number of routes to be moved out of tile i, to neighboring tile cells, is calculated as:

$$\Delta W_i = W_i \times \frac{CurCap_i - NewCap_i}{CurCap_i} \tag{9}$$

The redistribution depends on the Manhattan distance of a cell from i. For a tile cell k that is at a distance d from cell i ($k \neq i$), the number of routes added is computed as:

$$\Delta W_{k,i} = \frac{1}{4d} \times \frac{\omega}{d} \times \Delta W_i \tag{10}$$

where

$$\omega = \frac{4}{\sum_{j=1}^{maxDist} (1/j^2)} \tag{11}$$

The term $\frac{\omega}{d}$ captures the fact that the number of routes added to a cell varies inversely with its distance d from cell i, and these are equally distributed among the $4d$ cells that lie at a Manhattan distance of d from i. The role of the factor, ω, is to ensure that the total number of routes redistributed equals ΔW_i. In our experiments, the value of $maxDist$ is set to be 1/3 of the smaller of the numbers of tile cells in X and Y directions.

We then calculate $\Delta Cong_{k,i}$, the increase in congestion in tile cell k caused by $\Delta W_{k,i}$, as $\Delta Cong_{k,i} = \frac{\Delta W_{k,i}}{CurCap_i} = c_i \cdot \Delta y_i$ ($k \neq i$). This leads to the following linear approximation

$$\begin{aligned} \Delta Cong_k &= \left(\sum_{i \in R_k, i \neq k} \Delta Cong_{k,i}\right) + (c_k \cdot \Delta y_k) \\ &= \sum_{i \in R_k} (c_i \cdot \Delta y_i) \end{aligned} \tag{12}$$

where $c_k \cdot \Delta y_k$ is the congestion increase caused by the MIM decap Δy_k added to tile k.

IV. Sequence-of-Linear-Program Based Solution

We use an iterative flow to solve the decap allocation problem. In each iteration we allocate a relatively small amount of decap to the current circuit, for two reasons. Firstly, the decap allocation problem is highly nonlinear, and this iterative approach permits us to control the optimization process by solving a sequence of linear programs, one in each iteration. Secondly, it avoids the excessive allocation of decaps that could invalidate the approximate linear model of congestion and violation area used in our algorithm: these models are predicated on the assumption of small perturbations.

The overall optimization flow can be formulated as follows:

1) Initial setup steps: solving the input 3D power grid, determining the set of nodes that violate the voltage specifications and computing the noise violation metric, building the coarser grid G' as described in Section II, generating the temperature map for the circuit using 3D thermal analysis, and evaluating $\phi(T_k)$ in each tile k of G'.
2) If violation node set is empty, then stop. Otherwise, for each tile k that contains at least one node that violates the voltage specification, select one observation node N_k. The node N_k is chosen to be the node i with the maximum violation area, S_i, in tile k.
3) For each tile that contains an observation node N_k, calculate $\partial S/\partial C_{N_k}$, the derivative of the total violation area S with respect to the decap C_{N_k} added at N_k using the adjoint analysis method.
4) For each tile k, calculate $\Delta Cong_k = \sum_{i \in R_k} (c_i \cdot \Delta y_i)$ using the method described in Section III.

5) Formulate the linear programming problem described in Section II and solve it.
6) Update the decap budget using the solution from LP solver. For each tile k, if the solution Δx_k or Δy_k of current iteration is not zero, then we insert corresponding Δx_k CMOS decap or Δy_k MIM decap to tile k. Next, we update the current maximum allocatable amount of decap resource C^k_{CMOS} or C^k_{MIM} in tile k correspondingly.
7) Solve the circuit using the updated decap allocation, and update the set of violating nodes.
8) Update the current total violation area S.

V. Experimental Results

The overall 3D power grid optimization flow has been written using Tcl, and the 3D power grid analyzer and the congestion and leakage aware decap allocation algorithm are implemented in C++. All experiments are performed on an Intel Pentinum 4 CPU 2.8GHz Linux machine with 1G memory running Redhat Linux 2.6.9.

The 3D placement tool in [11] is first applied to generate the 3D layouts from the IBM-PLACE benchmarks using four tiers. Next, we scale all the layouts to the 90*nm* technology node. Since the time-varying current sources, which model the behavior of each functional unit, are not originally available in these benchmarks, we use a method similar to [1] to generate the waveforms in each circuit. Six layers of regularly distributed power grid are generated for each 2D tier of a 3D circuit when building the 3D power grid. The supply voltage is set to be 1.2V and the voltage drop threshold is chosen to be 0.12V in each of the experiments. The capacitance densities for CMOS and MIM decaps are, respectively, set to be 17.3 $fF/\mu m^2$ (the oxide thickness is assumed to be 20 Å) and 8.0 $fF/\mu m^2$. The leakage density of a CMOS decap is set to be $6.5 \times 10^{-5}\, mA/\mu m^2$, which is obtained from the simulation of a CMOS decap using PTM model [12]. For all of our experiments, the leakage density of the MIM decap is sufficiently small that it can be neglected.

A. Comparison Of Optimization Efficiency

TABLE I
Parameters Of Benchmarks

Circuit	# Nodes	Worst voltage droop (V)	# Violation nodes	Violation Area S (V ·ns)	Power Density Range ($\times 10^7$ W/m^2)	Temperature Range (°C)
ibm123	18634	0.135	3330	13.739	[0, 1.27]	[22.5, 88.9]
ibm05	12026	0.122	1359	72.260	[0, 1.33]	[26.2, 84.7]
ibm08	17030	0.125	3191	41.305	[0, 1.29]	[25.9, 74.3]
ibm10	29262	0.159	5935	91.286	[0, 1.25]	[26.1, 84.9]
ibm18	75042	0.163	6392	108.649	[0, 2.28]	[29.7, 92.3]

Table I lists the parameters of the benchmarks used in our experiments. The circuit ibm123 is the combination of three ibm benchmarks: ibm01, ibm02 and ibm03.

TABLE II
Comparison Of Optimization Efficiency

Ckt	CMOS only						MIM only					CMOS + MIM						
	VNs	S (V ns)	Lkg (mA)	CMOS (pF)	#Iter	Time (s)	maxC (%)	avgC (%)	MIM (pF)	#Iter	Time (s)	Lkg (mA)	maxC (%)	avgC (%)	CMOS (pF)	MIM (pF)	#Iter	Time (s)
ibm123	375	0.024	2.0	543	28	141	15.8	3.9	607	7	59	1.0	8.4	1.7	271	345	5	49
ibm05	33	0.050	1.7	462	7	28	19.7	1.7	550	23	111	1.4	0.0	1.2	360	178	23	114
ibm08	38	0.011	1.1	302	19	88	30.6	1.5	768	24	134	0.5	0.0	1.0	145	622	22	121
ibm10	371	0.184	1.5	408	15	123	10.6	5.9	511	11	186	0.8	4.5	2.6	220	296	4	135
ibm18	157	0.082	2.5	673	16	450	39.5	5.3	812	9	339	1.3	7.0	3.7	344	472	9	331

Table II lists the results of decap optimization in three different cases. First, only CMOS decaps are used: in this case, it is not possible to add enough CMOS decaps to eliminate the the violation area **S** (see column 3) for any of the five circuits. However, we list the results for the best available solution that minimizes this metric, showing the final number of violating nodes (**VNs**) that fail to meet the constraints, the corresponding violation area (**S**), the total leakage current of the CMOS decaps (**Lkg**), the total amount of CMOS decap allocated (**CMOS**), the total number of iterations required by the optimizer **(#Iter**), and the total CPU time (**Time**).

Next, only MIM decaps are used: in this case, the violation area is completely eliminated by our procedure. Considering that the allocated MIM decaps will affect the routing congestion, we list the following results: the percentage increase in maximum and average Z-direction routing congestion after optimization (**maxC**, **avgC**), the total amount of MIM decap allocated (**MIM**), as well as the total number of iterations (**#Iter**) and the total CPU time (**Time**) for this case. Since MIM decaps have much smaller leakage density than CMOS decaps, for all practical purposes, their leakage is zero and is not shown in the table.

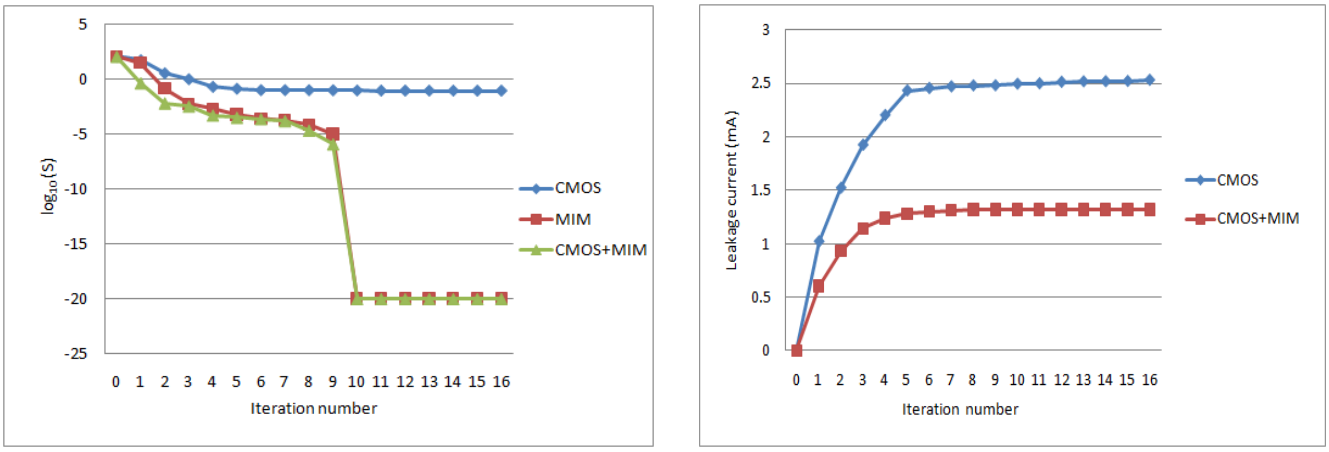
Finally, when both CMOS and MIM decaps are used, again, the violation area is completely eliminated. We list the total leakage current of the CMOS decaps (**Lkg**), the percentage increase in maximum and average Z-direction routing congestion

after optimization (**maxC**, **avgC**), the total amount of CMOS decap allocated (**CMOS**), the total amount of MIM decap allocated (**MIM**), the total number of iterations (**#Iter**) and the total CPU time (**Time**).

From Table II we can see that for each of the five circuits, the violation area cannot be eliminated through the use of CMOS decaps only. This is due to the fact that the amount of CMOS decap that can be added in a circuit is limited by the available area of white space; moreover, for decaps to be effective, it is important for sufficient white space to be available near the area where the voltage constraints are violated. Placing decaps far away from the voltage violation area is of little help in alleviating noise violations. Therefore, unless we disturb the current placement or enlarge the chip size to make more white space available near the violation area, it is not possible to completely eliminate these violations.

The introduction of MIM decaps can effectively eliminate the voltage violations and greatly reduce the decap leakage, at the cost of worsened routing congestion. Table II shows that the use of MIM decaps alone leads to severe congestion problems. Comparing the results of using MIM and CMOS decaps individually with using them together, it can be seen that replacing part of the MIM decaps with CMOS decaps can obtain a better tradeoff between congestion and leakage, while effectively eliminating voltage violations.

Comparing the total decap values for the MIM only and the CMOS+MIM cases, we can see that the decap values are similar (the values for CMOS-only are significantly different, since the constraints are not met in this case). The slight difference is attributable to approximations in linearizing the cost function in each iteration: specifically, in each iteration of our decap budgeting algorithm, an approximate formula, $\Delta S = (\partial S/\partial C) \cdot \Delta C$, is used to estimate the effect of added decap on the violation area, and this holds only when ΔC is small enough. In other words, in order to make the linear model more accurate, a smaller ΔC should be used, implying that the upper bounds for CMOS and MIM decaps in each iteration should be set to be very low (see Section II-B). This may lead to an increase in the number of iterations, impairing the computational efficiency of our approach. In our experiments, we found that a good balance between efficiency and accuracy can be obtained when Δ_{CMOS} and Δ_{MIM} are chosen to be in the region $[0.5pF, 1.0pF]$.



(a) (b)

Fig. 2. Change in the total (a) noise violation area, and (b) leakage current, over each iteration.

Figures 2 shows how the total violation area and total leakage current of circuit ibm18 change as the iterative process progresses. It can be seen that the CMOS-only case cannot bring the violation area down beyond some threshold, while the MIM-only and the CMOS+MIM methods are both successful (note that the extremely low violation value of approximately 10^{-20} is essentially zero). Figure 2 shows that the total violation area decreases rapidly in the first 5 iterations, and most of the violations are eliminated after this stage (Note that the y-axis in this figure is on a log scale). The reason is as follows: most of the violations of the power nodes are relatively easily resolved by inserting a small amount of decap. Although the violation area of these nodes is individually small, their sum, taken over a large number of nodes, is large. Eliminating these "easy" violations at the beginning of the iterative process cause the violation area to decrease rapidly at first. Beyond this point, a relatively small number of "hard" violation nodes remain, and the change in the violation area is harder to view on the scale of this graph, but is definitely visible at a magnified scale. For the same reason, most of the white space resources that are effective in reducing noise violations are consumed in early iterations, resulting in a fast initial increase in the leakage power. The leakage component of the objective function implies that MIM decaps are preferred over CMOS decaps when both are available, and when the insertion of MIM decaps does not significantly affect congestion.

B. Effect Of Power Grid Density

In this section, we further investigate how power grid density affects the results of decap budgeting provided by our algorithm.

The circuit ibm123 with a size of $2480\mu m \times 2000\mu m$ was selected, and three power grids with different densities were built. In **Case1**, the power pitches in both the x and y directions, for the lowest two metal layers in each 2D tier, are set to be the cell row height. In **Case2**, the power pitch in the y direction in these layers is set to be half of the cell row height, while that in the x direction is set to be the cell row height, and in **Case3** the power pitches in both the x and y directions in these layers are set to be half of the cell row height. In all three cases, the power pitches for the higher metal layers, as well as the number of interlayer vias connecting adjacent 3D tiers, are set proportionately.

TABLE III
OPTIMIZATION RESULTS OF DIFFERENT POWER GRID DENSITIES

Cases	Power Grid Density	# Nodes	# Violation Nodes	Worst-case voltage droop (V)	Violation Area (V ns)	Decap (fF)	Lkg (mA)	maxC (%)	avgC (%)	#Iter	Time (s)
Case1	Nominal	18634	3330 (17.87 %)	0.135	13.739	616234	1.0182	8.35	1.69	5	48.7
Case2	Denser	36433	4210 (11.56 %)	0.126	2.615	436972	0.6755	27.14	4.01	2	48.8
Case3	Densest	72114	4671 (6.48 %)	0.124	1.482	237234	0.3208	58.41	7.62	1	51.3

Our decap optimization algorithm, using both CMOS and MIM decaps, was then used to individually optimize the power grids in all three cases. The results are shown in Table III. From the table, we can see that:

- First, a denser grid helps to reduce the voltage droop in a circuit. When we increase the power grid density, both the worst-case voltage droop and violation area will be reduced (see column 4 and column 5).
- Second, a denser grid implies a larger number of grid nodes, resulting in larger cost for transient analysis and adjoint sensitivity analysis. Therefore, it takes more time to solve the problem in each iteration. On the other hand, the total number of iterations decreases because the violation area in the circuit is reduced. Therefore, we can see from Table III that the total CPU time for our algorithm increases much more slowly than the power grid size.
- Third, a denser grid implies more power connections in Z direction, and therefore a higher routing congestion, which is more sensitive to the inserted MIM decaps. This can be seen in Table III: when the power grid density increases, so does the percentage increases in the maximum and average congestion values.

C. Comparison of Power Grid Performance between 2D and 3D circuits

In this section, we compare the performance of power grids in 2D and 3D circuits. In order to do so, we generate a pair of 2D circuits and a 3D circuit, all with the same footprint. The placement of these circuits is not necessarily optimized, but they are adequate for our power delivery experiments.

Several test circuits were generated by applying the 3D placement tool in [11] to find a four-tier 3D placement of the circuit ibm08. The 4 tiers of the circuit are then flattened out on the 2D plane to obtain a 2D circuit, A_1, by placing tier0 at the lower left, tier1 at the lower right, tier2 at the upper right and tier3 at the upper left. A different 2D circuit, A_2, was obtained from the same 3D placement result by changing the order in which the tiers were placed: this time, tier2 is placed at the lower left, tier3 at the lower right, tier0 at the upper right and tier1 at the upper left. Finally, we stacked circuit A_2 on A_1 to build a separate 3D circuit, B.

By construction, the circuits A_1, A_2 and B all have the same footprint, and circuits A_1 and A_2 have the same average current densities. The first three rows of Table IV show the characteristics of these three circuits. As expected, it can be seen that before optimization, the 3D circuit has a significantly worse voltage drop, and a larger fraction of the total number of nodes experience noise violations.

TABLE IV
COMPARISON OF POWER GRID PERFORMANCE BETWEEN 2D AND 3D CIRCUITS

circuit	# Nodes	Supply Voltage (V)	Worst voltage droop (V)	#Violation nodes	Violation Area S (V ·ns)	Leakage (mA)	Total decap (pF)	maxC (%)	avgC (%)
A_1	16529	1.2	0.1348	1368 (8.28%)	3.728	1.909	795.3	0.0	0.12
A_2	16529	1.2	0.1354	1537 (9.30%)	4.292	2.083	876.1	0.0	0.13
B	33033	1.2	0.1868	11421 (34.57%)	229.327	23.411	8210.9	0.0	0.23
$A_2^{'}$	16529	1.2	0.1693	3110 (18.82%)	54.632	6.635	2641.9	0.0	0.32
$B^{'}$	33033	1.2	0.2098	15827 (47.91%)	445.633	32.238	11130.7	0.0	0.38
$A_2^{''}$	16529	1.2	0.1016	0 (0 %)	0	-	-	-	-
$B^{''}$	33033	1.2	0.1638	6287 (19.03%)	92.913	10.680	3532.5	0.0	0.10

Next, we applied our MIM+CMOS algorithm to optimize these three circuits, and the results are summarized in the last four columns of the table. It can be seen that circuit B requires about 5 × the amount of decaps as the combined sum for A_1 and A_2 to meet the voltage specifications, which results in a larger amount of leakage. This motivates the need for using MIM decaps, to control the leakage.

Next, we consider two scenarios where the current distribution on the upper layer is increased by 25% (circuit $A_2^{'}$) or decreased by 25% (circuit $A_2^{''}$). This mimics the fact that circuit designers may choose to use more memory on one layer than the other. Correspondingly, these circuits are stacked on A_1 to obtain 3D circuits $B^{'}$ and $B^{''}$, respectively. These results are shown in

the lower half of Table IV. As expected, both A_2' and B' require more decaps than A_2 and B, respectively. Moreover, There is no voltage violation for circuit A_2'', although B'' does see violations that must be fixed, at a cost lower than that for B.

VI. Conclusion

We have proposed an efficient decap allocation algorithm to optimize 3D power supply network using both MIM and CMOS decaps. MIM decaps have the desirable properties of high capacitance density and low leakage density, and can be a good complement to the on chip SiO_2-based CMOS decap. Our algorithm uses 3D congestion analysis and a linear congestion model, as well as linearized noise models based on adjoint sensitivity analysis, to guide the decap allocation among CMOS and MIM decaps. Experimental results show that power grid noise can be more effectively optimized using both MIM and CMOS decaps, with lower leakage power and low routing congestion costs. Although this work has used MIM decaps, the ideas can be extended to other types of potential decap technologies as well, such as deep-trench decaps, and similar methods may be used for such technologies.

References

[1] H. Su, S. S. Sapatnekar, and S. R. Nassif, "An algorithm for optimal decoupling capacitor sizing and placement for standard cell layouts," in Proceedings of the International Symposium on Physical Design, 2002, pp. 68–73.

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

[3] E. Wong, J. Minz, and S. K. Lim, "Decoupling capacitor planning and sizing for noise and leakage reduction," in Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, 2006, pp. 395–400.

[4] J. R. Minz, S. K. Lim, and C.-K. Koh, "3D module placement for congestion and power noise reduction," in Proceedings of the Great Lakes Symposium on VLSI, 2005, pp. 458–461.

[5] G. Huang, et al., "Power delivery for 3D chip stacks: Physical modeling and design implication," in Proceedings of the IEEE Electrical Performance of Electronic Packaging Meeting, 2007, pp. 205–208.

[6] D. Roberts, et al., "Application of on-chip MIM decoupling capacitor for 90nm SOI microprocessor," in Proceedings of the IEEE International Electron Devices Meeting, 2005, pp. 72–75.

[7] Y. L. Tu, et al., "Characterization and comparison of high-k metal-insulator-metal (MiM) capacitors in 0.13μm Cu BEOL for mixed-mode and RF applications," in Proceedings of the IEEE International Symposium on VLSI Circuits, 2003, pp. 79–80.

[8] H. Sanchez, et al., "Increasing microprocessor speed by massive application of on-die high-k MIM decoupling capacitors," in Proceedings of the IEEE International Solid-State Circuits Conference, 2006, pp. 2190–2199.

[9] P. Zhou, et al., "3D-STAF: Scalable temperature and leakage aware floorplanning for three-dimensional integrated circuits," in Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, 2007, pp. 590–597.

[10] J. Lou, et al., "Estimating routing congestion using probabilistic analysis," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 21, no. 1, pp. 32–41, Jan. 2002.

[11] B. Goplen and S. S. Sapatnekar, "Placement of 3D ICs with thermal and interlayer via considerations," in Proceedings of the ACM/IEEE Design Automation Conference, 2007, pp. 626–631.

[12] "Predictive technology model," Device Group at Arizona State University, Available at http://www.eas.asu.edu/~ptm.