Decoupling-Capacitor Planning and Sizing for Noise and Leakage Reduction

Eric Wong, *Student Member, IEEE*, Jacob Rajkumar Minz, *Student Member, IEEE*, and Sung Kyu Lim, *Senior Member, IEEE*

***Abstract*—Decoupling capacitors (decaps) are a popular means for reducing power-supply noise in integrated circuits. Since the decaps are usually inserted in the whitespace of the device layer, decap management during the floorplanning stage is desirable. However, a well-known existing work only allows the blocks to utilize the adjacent whitespace. In order to overcome this limit, we devise the effective-decap-distance model to analyze how functional blocks are affected by nonneighboring decaps. In addition, we propose a generalized network-flow-based algorithm to allocate the whitespace to the blocks and determine the oxide thicknesses for the decaps to be implemented in the whitespace. Experimental results show that our decap allocation and sizing methods can significantly reduce decap budget and leakage power with a small increase in area and wire length when integrated into 2-D and 3-D floorplanners.**

***Index Terms*—Decoupling capacitors (decaps), floorplanning, power-supply integrity.**

I. Introduction

SIGNAL integrity is a very important issue in very large-scale integration technology. Simultaneous switching of digital-circuit elements can cause considerable IR-drop and Ldi/dt noise in the power-supply network. This power-supply noise can cause logic faults. On-chip decoupling capacitors (decaps) are widely used to mitigate the power-supply-noise problem. By charging up during the steady state, decaps can assume the role of the power supply and provide the current needed during the simultaneous switching of multiple functional blocks.

Postfloorplanning or postroute power-supply synthesis can be applied to generate satisfactory power-supply distribution. In many cases, however, when the circuit-block locations are fixed, the constraints such as voltage drop and current density are so tight that there is no feasible power-network design capable of keeping power-supply noise within a specified margin. Hence, it is important to consider power-supply planning during the early design stage, where the circuit-block locations can be flexibly changed. Since the decaps are usually inserted in the whitespace of the device layer, decap management during the floorplanning stage is desirable. A pioneering work on decap-aware floorplanning was proposed by Zhao *et al.* [1]. However, a noticeable limitation of this work is that it allows the blocks to utilize the adjacent whitespace only. Although a majority of current is provided by neighboring decaps, it is still possible for a block to draw current from nonneighboring decaps.

The continued reduction of oxide thickness in advanced nanotechnology significantly increases the tunneling current and leakage power of thin oxide capacitors. This problem is addressed in the study in [2] by performing wire sizing of the power/ground network after decap insertion. Another possible solution for the decap-leakage reduction is to use thicker oxides [3], since the leakage power of gate-oxide capacitors is inversely proportional to the thickness of the oxide. However, thicker oxide reduces capacitance and increases the area required to implement the decaps. Therefore, a careful decision has to be made on the area, as well as the oxide thickness, of the decaps. Although dual-oxide-thickness decaps may increase manufacturing costs, the benefits include decap-leakage reduction and decap-area reduction.

The contributions of this paper are as follows.

1) We devise the effective-decap-distance modeling, where the effectiveness of a decap is dependent on the distance to the block that accesses it. Our experimental results show that the decap can be reduced significantly by allowing nonneighboring decap access when used in floorplanning.
2) We propose a generalized network-flow approach to accomplish two goals: To allocate the whitespace to the blocks and to determine the oxide thicknesses of the decaps to be implemented in the whitespace. Our experimental results show that the leakage power caused by decaps can be reduced significantly using our methods.
3) Having multiple device layers creates the possibility of allowing circuit modules to access decaps on other layers in 3-D IC. We show that the effective distance model and our decap-allocation/sizing schemes work very effectively for 3-D floorplanning.

The remainder of this paper is organized as follows. Section II presents the problem formulation and an overview of the algorithm. Section III presents the effective distance modeling. Section IV presents the generalized network-flow-based decap allocation and sizing method. Section V presents the application to 3-D floorplanning. Experimental results are provided in Section VI, and the conclusions are in Section VII.

Manuscript received July 20, 2006; revised December 1, 2006 and January 31, 2007. This work was supported in part by MARCO under Grant GSRC/C2S2 and in part by National Science Foundation CAREER Award CCF-0546382. This paper was recommended by Associate Editor R. Suaya.

E. Wong was with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA. He is now with Universal Avionics Systems Corporation, Norcross, GA 30071 USA.

J. R. Minz was with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA. He is now with Synopsys, Inc., Mountain View, CA 94043 USA.

S. K. Lim is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: limsk@ece.gatech.edu).

Digital Object Identifier 10.1109/TCAD.2007.906463

0278-0070/$25.00 © 2007 IEEE

II. Preliminary

A. Problem Formulation

The following are the inputs to the decaps planning and sizing (DCPS) problem: 1) a set of blocks that represent the circuit modules; 2) width, height, and maximum switching currents for each block; 3) a net list that specifies how the blocks are connected; 4) the oxide thicknesses available for decap fabrication; 5) the location of the power/ground pins; 6) the power-supply-noise constraint; and 7) decap leakage-power constraint. The goal of the DCPS problem is to find the following: 1) the location of the blocks and whitespace; 2) assignment of whitespace to blocks; and 3) thickness of decaps that are to be inserted in the whitespace so that the power-supply noise and leakage-power constraints are satisfied. The objective is to minimize $w_1 \cdot A + w_2 \cdot W + w_3 \cdot D$, where A and W, respectively, denote the total area and wire length of the floorplan and D denotes the total amount of decoupling capacitance required. w_1, w_2, and w_3 are the weights of the three objectives. If the existing whitespace cannot fill all of the decap demand, then the floorplan will be expanded to add additional whitespace. This area expansion is minimized under our area objective A.

B. Overview of the Algorithm

Our algorithm consists of two parts: floorplan optimization and decap insertion/sizing. Simulated annealing (SA) is a popular approach for floorplan optimization due to its high-quality solutions and flexibility in handling various constraints. We use sequence pair and its perturbation scheme [4] to represent and optimize our 2-D floorplans. In order to evaluate a candidate floorplanning solution, we use the following metrics: 1) area/wire length—the location of blocks is determined, and the area and wire length are computed—and 2) decap budget. First, we perform simultaneous switching noise (SSN) analysis to compute the noise level for each block. Then, the amount of decap needed for each block is computed based on its noise so that the overall SSN constraint is satisfied. Upon the completion of the floorplanning optimization, we perform the actual decap insertion and sizing on the final floorplan as follows.

1) The existing whitespace in the floorplan is detected.
2) A generalized network-flow graph is constructed. Solving the generalized flow network allocates whitespace for decap and assigns oxide thicknesses to the decaps.
3) If not all of the decap budgets of the blocks are filled, then area expansion is performed on the floorplan to add extra whitespace.
4) Go back to step 2) if the decap demands and leakage constraints of all of the blocks are not satisfied.

Note that the decap-insertion/sizing and floorplan-expansion step may be repeated several times until the noise/leakage constraints are met.

C. Existing Works

Recent works on decap-aware floorplanning for 2-D circuits include [1], [5]–[11]. Some existing floorplanning works on 3-D circuits include [12]–[14]. A pioneering work on decap-aware floorplanning for 2-D circuits is presented by Zhao *et al.* [1]. The authors proposed two algorithms, where the first one considers the decap placement as a postfloorplan step, while the second one considers the decap placement as an integral part of floorplanning, i.e., decap-aware floorplanning. In both cases, the objective is to minimize the floorplan area while suppressing the power-supply noise below the specified limit. A straightforward extension of this paper to 3-D by treating each layer separately would not take full advantage presented by the 3-D environment. For example, utilizing only the decaps adjacent to the blocks would limit interlayer access. Although a majority of current is provided by neighboring decaps, it is still possible for a block to draw current from nonneighboring decaps. We overcome this limitation by formulating effective decap distance, where the effectiveness of a decap is dependent on the distance to the block that accesses it. Our experimental results show that the area overhead induced by decap implementation can be reduced significantly by allowing nonneighboring-decap access. LP-based decap-to-block allocation is performed in the study in [1]. Instead, we propose a generalized network-flow-based approach, where a flow-approximation method is utilized to accelerate the decap-allocation step.

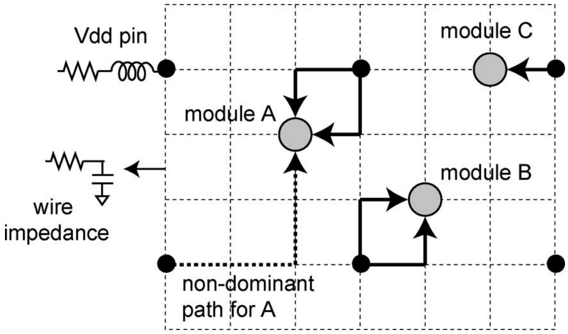

Fig. 1. Two-dimensional-mesh-based P/G network model. Dominant paths for each module are shown in solid lines.

III. Effective Distance Modeling

A. Power-Supply-Noise Modeling

We use the method presented in [1] to calculate power-supply noise. A brief summary of their method is given here. A uniform RC-mesh is used to model the P/G network, as illustrated in Fig. 1. The edges in the mesh have resistive impedances.[1] The mesh contains power-supply and connection points. The connection points consume currents. The current is drawn from all the sources by the consumers, and the amount of current drawn along a path is inversely proportional to the impedance of the path in the power-supply mesh. The dominant supply for a block is defined as the voltage source supplying significantly more power to the block than any other neighboring sources. The dominant paths for a block are the paths from the dominant

[1] Note that we do not model inductive components in our mesh. This is due to the fact that floorplan optimization under time-varying current profile is a very complex problem, if not impossible. Instead, our optimization is targeting IR-drop minimization.

supply to the block carrying most of the current. It has been shown experimentally in [1] that the shortest paths between the dominant supply (nearest V_{dd} pins) and the block offers highly accurate SSN estimation within reasonable runtime. Let P_k be a dominant current path for block k. Then, $T^k = \{P_j : P_j \cap P_k \neq \emptyset\}$ denotes the set of all other dominating paths overlapping with P_k (T^k includes P_k itself). Let P_{jk} be the overlapping segments between path P_j and P_k. Let $R_{P_{jk}}$ denote the resistance of P_{jk}. After the current paths and their values have been determined for all blocks, the SSN for $B^{(k)}$ is given by

$$V_{\text{noise}}^{(k)} = \sum_{P_j \in T^k} i_j \cdot R_{P_{jk}} \tag{1}$$

where i_j is the current in the path P_j, which is the sum of all currents through this path to various consumers. The weight of i_j is the resistive components of the path.

In the worse case, a module would draw all of its switching current from its decap. Let $Q^{(k)} = \int_0^{t_s} I^{(k)}(t) \cdot dt$ denote the maximum charge drawn from the power supply by block $B^{(k)}$, where $I^{(k)}(t)$ is the current demand and t_s is the switching period. A greedy way to calculate decap budget is $C^{(k)} = Q^{(k)}/V_{\text{tol}}$, $k = \{1, 2, \ldots, M\}$, where V_{tol} is the noise tolerance of the block and M is the total number of blocks. It has been shown in [1] that this significantly overestimates the amount of decap needed. Instead, the decap budget is calculated as follows:

$$\Theta^{(k)} = \max\left(1, \frac{V_{\text{noise}}^{(k)}}{V_{\text{tol}}}\right), \qquad k = \{1, 2, \ldots, M\} \tag{2}$$

$$C^{(k)} = \frac{\left(1 - 1/\Theta^{(k)}\right) Q^{(k)}}{V_{\text{tol}}}, \qquad k = \{1, 2, \ldots, M\}. \tag{3}$$

This base decap budget is for the case where there is no resistance between a block and its decap. If m denotes the number of blocks, this $p \times q$-mesh-based decap analysis takes $O(mpq)$, where most of the time is spent on shortest path analysis. Note that it is possible to perform this decap analysis incrementally, where only the affected blocks and their dominant paths are updated from the SA-based floorplan perturbation. The worst case complexity still remains at $O(mpq)$, but the runtime can be significantly reduced if the perturbation causes minor change in the floorplan.

B. Decap Modeling With Effective Distance

The decap budget calculated using the method from [1] is only valid when decaps are adjacent blocks. This may result in unnecessary floorplan-area expansion due to being unable to utilize some of the existing whitespace due to the adjacency restriction. We introduce the concept of effective distance to overcome this limitation and to make use of nonadjacent whitespace for decap allocation. A decap placed far away from a block is less effective at reducing noise.

Definition 1: Effective distance $\gamma_{\text{eff}}(R_c)$ is the amount of decap needed when the resistance between the decap and the block is R_c, due to distance, to get the same noise reduction as a unit of decap adjacent to the block.

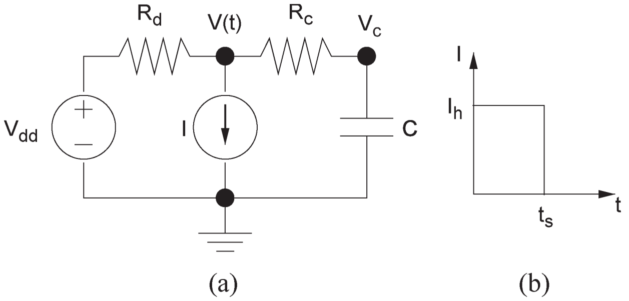

Fig. 2. (a) Circuit used for effective distance formulation. (b) Switching current of the block.

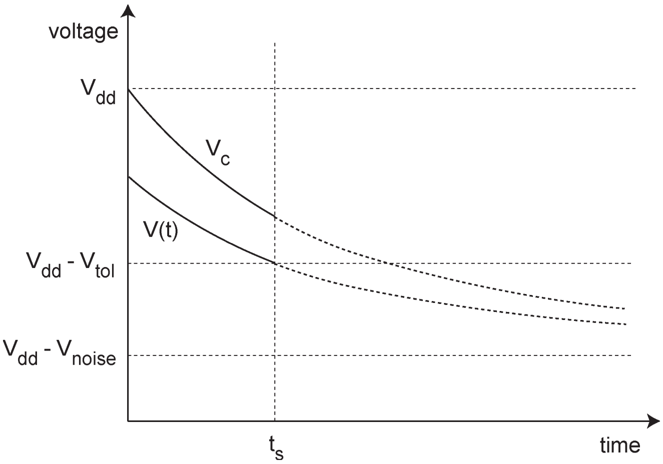

Fig. 3. Voltage of the circuit module $V(t)$ and the voltage of the capacitor V_c during switching. V_{dd} is the voltage of the power pin. V_{tol} is the maximum noise the block can handle. V_{noise} is the SSN.

The circuit shown in Fig. 2 was analyzed to find a relationship between the distance and the amount of decap needed by a block. In the circuit, V_{dd} represents the power pin, C represents the decap, and I represents the current demand of the block. R_d and R_c represent the resistances of the block to the power pin and to the decap, which depend on distance. We assume that the block draws I_h current during a switching interval of t_s time and negligible current when not switching. The voltage supplied to the block during switching is

$$V(t) = V_{dd} - V_{\text{noise}} + V_{\text{noise}} \frac{R_d}{R_c + R_d} \cdot e^{\frac{-t}{(R_c+R_d)C}} \tag{4}$$

where $V_{\text{noise}} = R_d \cdot I_h$ (see Fig. 3). This equation can be solved for C to find the amount of decap needed by the block

$$C = \frac{-t_s}{(R_c + R_d)\left[\ln \frac{(V_{\text{noise}} V_{\text{tol}})}{V_{\text{noise}}} + \ln \frac{R_c + R_d}{R_d}\right]}. \tag{5}$$

This equation only holds when $V_{\text{noise}} > V_{\text{tol}}$ and $R_c < R_{\max}$, where

$$R_{\max} = \frac{R_d \cdot V_{\text{tol}}}{V_{\text{noise}} - V_{\text{tol}}}. \tag{6}$$

The first condition is obvious since no decap would be needed if the noise were less than the tolerance. The second condition specifies the maximum resistance between a block and

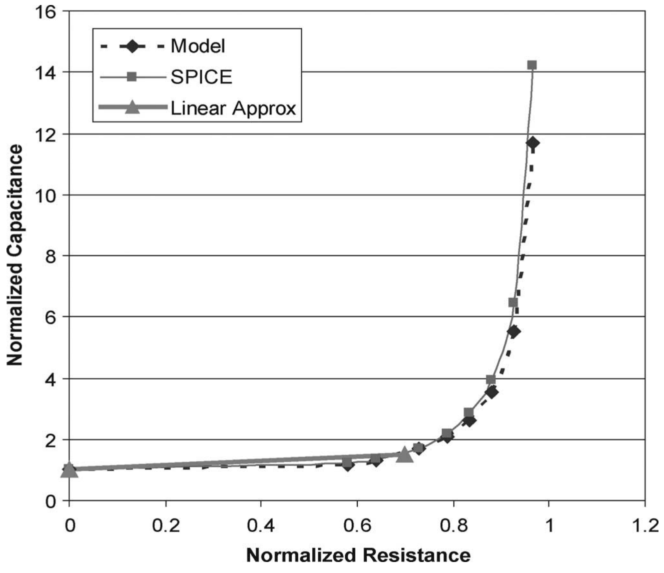

Fig. 4. SPICE modeling on decap requirement as a function of resistance R_c, which is normalized with respect to $R_{\max}$. Normalized capacitance is equivalent to γ_{eff}.

its decap. Effective distance $\gamma_{\text{eff}}(R_c)$ can be defined as the capacitance needed as a function of resistance divided by the capacitance needed with no resistance

$$\gamma_{\text{eff}}(R_c) = \frac{C(R_c)}{C(0)} = \frac{R_d \cdot \ln \frac{V_{\text{noise}}-V_{\text{tol}}}{V_{\text{noise}}}}{(R_c+R_d)\left[\ln \frac{V_{\text{noise}}-V_{\text{tol}}}{V_{\text{noise}}} + \ln \frac{R_c+R_d}{R_d}\right]}. \quad (7)$$

To verify the effective distance model, resistive-power meshes were simulated in HSPICE. A block and a decap were inserted into the simulated power mesh. The location of the decap with respect to the block was varied, and the amount of capacitance needed to suppress the noise was found for each decap location. Fig. 4 compares the effective distance model with the HSPICE simulations. The model slightly underestimates the amount of decap needed when the resistance between the block and the decap approaches $R_{\max}$. To simplify effective distance calculations during decap allocation, a linear approximation of effective distance is used. In the linear approximation, the furthest that a block could access a decap is 0.7 $R_{\max}$, where 50% extra decap would be needed.

Let $D^{(k)}$ be the set of whitespace close enough to block $B^{(k)}$ to provide some decap. The decap that is allocated to block $B^{(k)}$ must satisfy

$$\sum_{j}^{D^{(k)}} \frac{C^{(j,k)}}{\gamma_{\text{eff}}(R_{j,k})} \geq C^{(k)} \quad (8)$$

where $C^{(j,k)}$ is the amount of decap allocated from whitespace j to block $B^{(k)}$, and $R_{j,k}$ is the resistance between whitespace j and block $B^{(k)}$. This constraint ensures that the actual decap allocation, which may include nonadjacent decap access, provides at least as much noise reduction that adjacent decap allocation with the decap budget from (3) would.

IV. Decap Allocation and Sizing Algorithms

A. Whitespace-Detection Algorithm

The whitespace present in a floorplan can be used to fabricate decap. If the existing whitespace is insufficient or unreachable by modules needing decap, then whitespace insertion through floorplan expansion may be necessary. Hence, detection of all existing whitespace in a floorplan is highly desirable. This can be done by drawing a grid on the floorplan and marking the grid cells that are covered by blocks. The grid cells that remain unmarked are detected as whitespace. If the grid is too coarse, the sizes and locations of the whitespaces will be inaccurate. In order to get the exact locations and sizes of the whitespaces, we use a nonuniform grid, as shown in Fig. 5(b). Vertical grid lines are drawn at the left and right edges of each block. Horizontal grid lines are drawn at the top and bottom edges of each block. If there are n blocks, there can be up to $2n$ vertical and $2n$ horizontal grid lines. The maximum number of grid cells is $(2n-1)^2 = 4n^2 - 4n + 1$, making this whitespace detection algorithm $O(n^2)$.

As shown in Fig. 5(c), many of the whitespaces in the floorplan are detected in small pieces rather than as a single large whitespace. Having too many whitespaces can slow down the decap-allocation algorithm. Therefore, whitespace merging is performed after whitespace detection to decrease the number of whitespaces. The first step of the whitespace-merging process is to traverse the grid horizontally and combine adjacent whitespace cells together. The next step of the whitespace-merging process is to traverse the grid vertically. For the vertical traversal, the width of adjacent whitespaces must match before being allowed to merge.

If sufficient decap cannot be allocated from the existing whitespace to suppress the SSN, then more whitespace is added by expanding the floorplan in the X and Y directions, as illustrated in Fig. 6.[2]

B. Decap Allocation and Sizing Algorithm

We model the decap allocation and sizing problem with generalized network flow. Generalized network flow generalizes traditional network flow by adding a gain factor $\gamma(e) > 0$ for each edge e. For each unit of flow that enters the edge, $\gamma(e)$ units must exit (see Fig. 7). For the traditional network flows, the gain factor is one. Capacity and node-conservation constraints are satisfied by the generalized networks, as in the traditional network flows. Generalized min-cost network flow can model the decap-allocation problem with dual-oxide-thickness capacitors and effective distance. Generalized network flow is a well-studied problem, but elegant exact and approximate algorithms have only been proposed recently [17], [18].

An example flow network for decap allocation is shown in Fig. 8. The nodes on the right represent the blocks. The capacities of the edges connecting to the sink are the decap demands

[2] We note that it is possible to slide the block x in Fig. 6(a) to the right to readjust the whitespace. This so-called whitespace redistribution in floorplanning has been used for wire-length minimization [15] and buffer insertion [16]. The investigation of this method for decap optimization is beyond the scope of this paper.

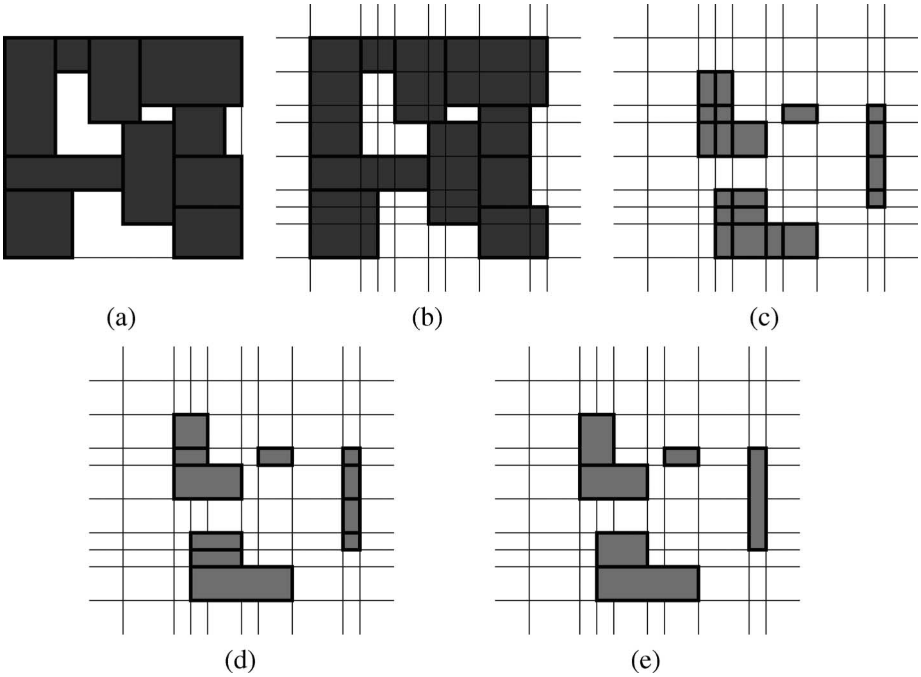

Fig. 5. Whitespace detection and whitespace merging. (a) Starting floorplan. (b) Nonuniform-grid generation, each grid line corresponds to a boundary of one of the blocks. (c) Twenty whitespace cells are detected. (d) After horizontal merging, there are 11 whitespace cells. (e) After vertical merging, there are six whitespace cells.

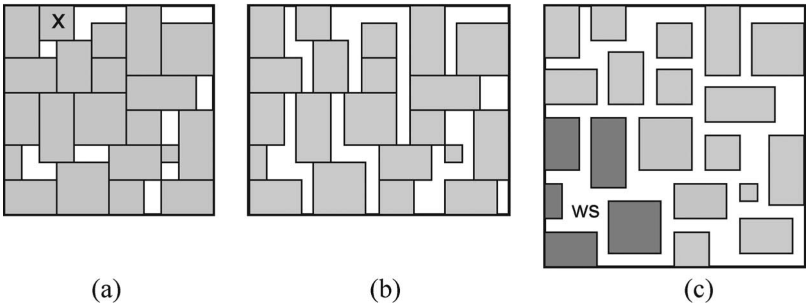

Fig. 6. Illustration of floorplan expansion. (a) Initial floorplanning. (b) X expansion. (c) $X-Y$ expansion, where the darker blocks denote the neighboring blocks of the decap (= whitespace) inserted.

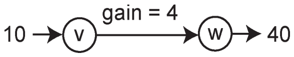

Fig. 7. Example of a generalized network-flow arc.

of the blocks. The gains of these edges are unity, and the costs are zero. The nodes on the left represent the whitespace. The capacities of the edges connecting to the source are the areas of the whitespace. The costs of these edges are zero, and the gains are unity. The nodes in the middle represent the oxide thicknesses. Each whitespace is connected to a thin oxide node and a thick oxide node. Additional oxide thicknesses can be considered by adding more oxide nodes. The edges connecting the whitespace to the oxide nodes have gain factors equal to the capacitance per unit area of the oxide thicknesses. The costs of these edges are the leakage per unit area of the oxide thicknesses, and the capacities of the edges are infinite.

If a circuit module is close enough to draw decap from a whitespace module, the circuit module is connected to the two oxide nodes corresponding to that whitespace. They are connected with an edge of infinite capacity, zero cost, and gain factor $1/\gamma_{\text{eff}}$ to represent the effectiveness of the whitespace. Maximizing the flow in this generalized flow network allocates the maximum possible decap to blocks. Fig. 9 shows an example of a floorplan and its corresponding generalized flow network. Minimizing the cost in the generalized flow network minimizes the leakage of the decaps. After the network-flow graph has been solved, several details about the decap allocation can be determined. The proportion of flow that goes through the thin and thick oxide nodes corresponding to a whitespace determines the proportion of thin and thick oxide decaps that are to be fabricated on that whitespace. The total decap leakage can be calculated by taking the edges between the whitespace and the oxide nodes and, then, multiplying the flows of the edges by the costs of the edges. If the flow in the sink edges are saturated, then the decap demands of all the circuit modules can be met. If the flow in some of the sink edges are less than capacity, then there is not enough whitespace to fulfill the decap

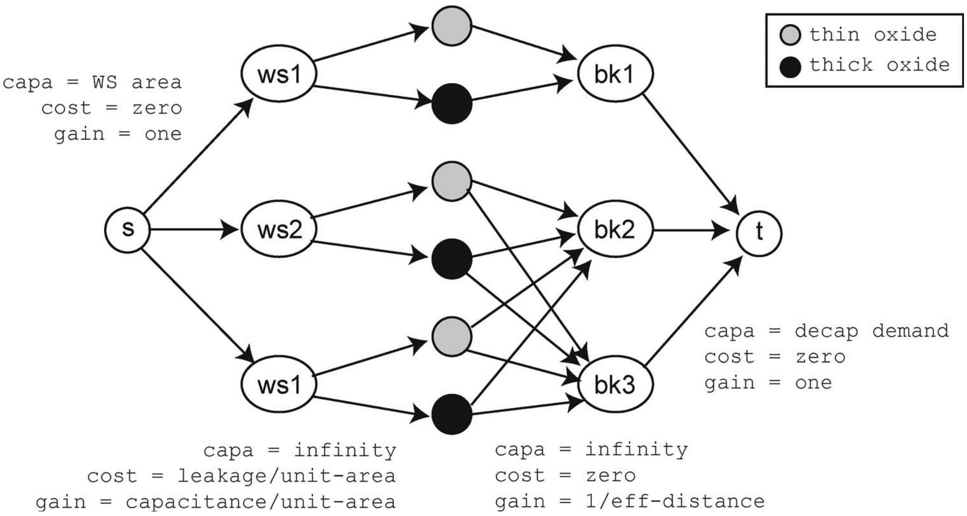

Fig. 8. Generalized flow network for decap allocation. bk1, bk2, and bk3 are blocks needing decap. ws1, ws2, and ws3 are whitespace. (capa = capacity).

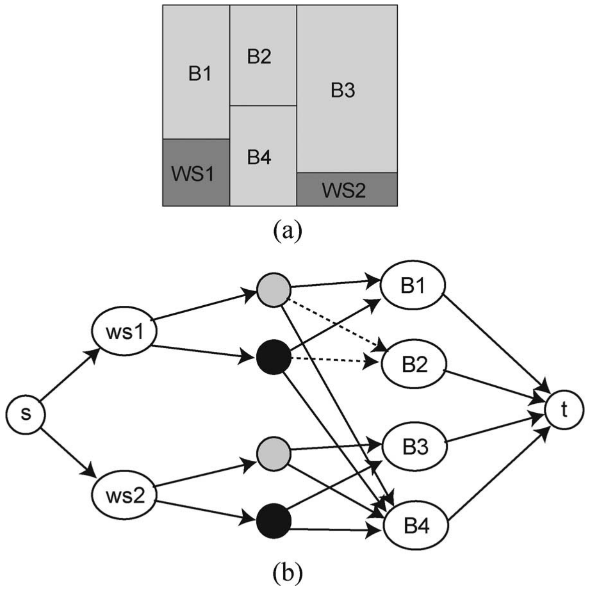

Fig. 9. Construction of a generalized flow network for a floorplan. (a) Floorplan needing decap. (b) Generalized flow network for decap allocation. Whitespace and blocks that are adjacent are connected. Block 2 is connected to whitespace 1 because effective distance allows access (shown in dotted line).

demands of the circuit modules. In this case, the floorplan must be expanded for additional whitespace.

Exact generalized min-cost max-flow algorithms are $O(n^3)$. This is too slow for iteration between decap allocation and whitespace insertion, so we used an approximation algorithm [18]. This algorithm runs in $O(\epsilon^{-2} \cdot n^2)$, where ϵ is the error-bound percentage from the maximum flow, and n is the number of nodes. Since the amount of flow returned by the approximation algorithm can be anywhere from $(1 - \epsilon) \cdot \text{flow}_{\max}$ to $\text{flow}_{\max}$, it could underallocate decap. To prevent underallocation, all of the decap demands are divided by $(1 - \epsilon)$. For example, if a module had a decap demand of 100 and ϵ were set to 0.2, then anywhere from 80 to 100 decap would be allocated by the approximation algorithm for generalized network flow if there was plenty of whitespace. If the decap demand was divided by $(1 - \epsilon)$ to get 125 before sending it to the generalized network-flow algorithm, then the allocation would be between 100 and 125. This would satisfy the decap demand or exceed it by up to $\epsilon/(1 - \epsilon)$.

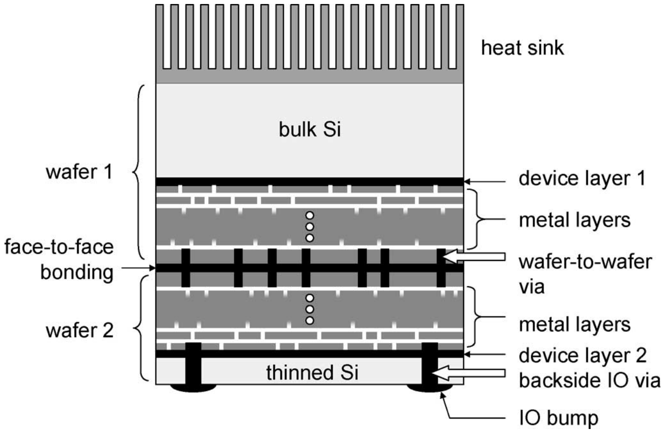

Fig. 10. Two-die 3-D IC with face-to-face bonding.

V. Application to 3-D Floorplanning

A. Motivation

Three-dimensional integrated circuits are an emerging technology with great potential to improve performance and power. Several different approaches in fabricating 3-D integrated circuits or 3-D-compatible transistors have been taken [19]–[22]. These vary in terms of the maximum number of device layers and the maximum density of interconnects between these layers. The wafer-bonding approach shown in [22], where discrete wafers are "glued" together using a copper interconnect interface, permits multiple wafers and multiple 3-D interconnects, overcoming the above limitations (see Fig. 10). The ability to route signals in the vertical dimension enables distant blocks to be placed on top of each other. This results in a decrease in the overall wire length, which translates into less wire delay, less power, and greater performance.

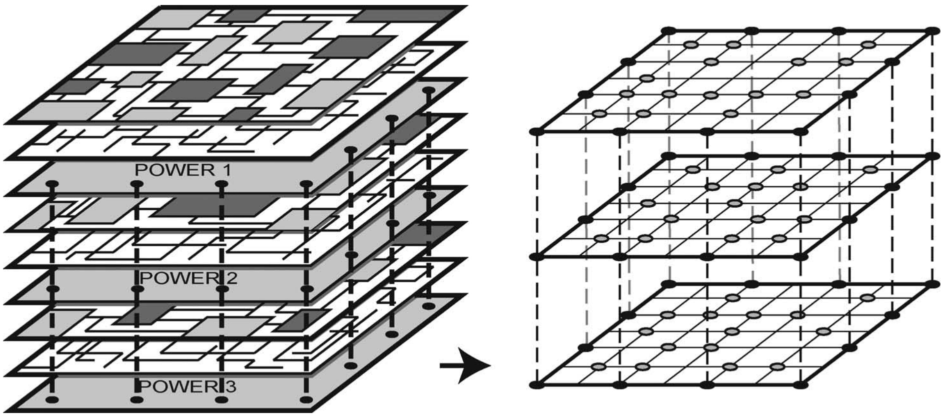

Fig. 11. Three-dimensional power-supply network modeling.

In general, the distance between the functional blocks and power pins in a 3-D design is reduced as compared to its 2-D counterpart, so we expect that the overall decap cost (= the area overhead) will be less in 3-D designs. However, the number of blocks accessing each power pin also increases due to the vertical interconnects available in 3-D ICs. In addition, thermal-hotspot problem is generally considered a formidable challenge in 3-D IC designs. Since the leakage power increases with the higher temperatures, we may have to give up the area saving by using decaps with greater area and oxide thickness. Thus, we believe that an in-depth tradeoff study that involves these 3-D-specific issues during 3-D floorplanning is crucial.

The decap-allocation problem in a 3-D IC has a couple of additional factors not present in the 2-D case. First, having multiple device layers creates the possibility of allowing circuit modules to access decaps on other layers. In this case, our effective distance model is the perfect means to allow interlayer nonneighboring decap access. Second, in case the existing whitespace in a floorplan is insufficient to supply the needed decap, the floorplan needs to be expanded to add additional whitespace. In 3-D ICs, expanding different layers can have different effects on the footprint area of the chip. For example, expanding a small layer might not increase the footprint area because there is a larger layer. To take advantage of this, we perform footprint-aware area expansion, which includes expanding smaller layers more than larger layers.

B. Footprint-Aware Decap Insertion

We extend the existing 2-D sequence-pair scheme [4] to represent 3-D floorplans. Specifically, k sequence pairs are used to represent the block placements of k device layers. This representation only encodes relative block positions among the blocks in the same layer. However, it is straightforward to determine the interlayer-position relationships of the blocks by computing the block coordinates. We use a 3-D mesh to model the P/G network in 3-D ICs, as shown in Fig. 11. An illustration of our 3-D SSN analysis is shown in Fig. 12. The dominant current source for block A is s_1, which is not located in the same layer. The dominant (shortest) path p_0 carries $I_A/6$ amount of current, where I_A denotes the current demand of A. The block C draws current from s_2 and s_3 using p_1, p_2, and p_3 (each of these carries $I_C/3$ amount of current). The resistance of p_{34}, the overlap between p_3 and p_4, contributes to the SSN at B and C.

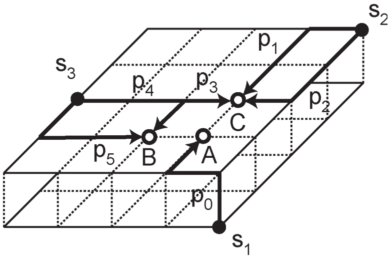

Fig. 12. Illustration of 3-D SSN calculation.

Our footprint-aware area-expansion algorithm finds the X and Y slack of each layer relative to the footprint and expands in the direction with more slack. If a particular layer is the bottleneck layer, i.e., it has maximum width and height, then some of the expansion is shifted to adjacent layers. Allowing blocks to use decaps in other layers is made possible by effective distance. The $X-Y$ expansion of each layer is controlled by α and β parameters, where α and β are the percent expansions in the X and Y directions. A simple expansion would set α and β to be equal to each other. In footprint-aware expansion, the X and Y slack of each layer are defined as $S_x = \text{Footprint}_{\mathtt{width}} - \text{Layer}_{\mathtt{width}}$. Then, the equation $\beta/\alpha = S_y/S_x$ is used to make the whitespace insertion favor the direction with more slack. After each iteration, the α and β are increased until the decap demands are met.

VI. Experimental Results

Our power-supply noise-aware floorplanner and generalized network-flow-based decap allocator were implemented in C++. The experiments were run on Pentium IV 2.4-GHz systems running Linux. The power/ground networks were modeled as uniform meshes. The power pins were placed at the four corners of each layer of the power-supply network. The error bound for the algorithm used to solve generalized network flow ϵ was set to 0.3 for the experiments.

TABLE I
COMPARISON TO AN EXISTING WORK [1]. THE RATIO VALUES ARE BASED ON THE POSTDECAP-INSERTION METHOD IN [1]

	results from [1]						ours					
	post decap insertion			noise-aware			area/wirelength-driven			decap-driven		
ckt	area	decap	runtime	area	decap	runtime	area	decap	runtime	area	decap	runtime
apte	50705710	20.72	12	50235794	16.36	119	48815100	13.82	24	49662800	13.75	24
xerox	20850453	6.74	18	20581079	5.85	193	21929600	5.28	24	21678300	5.20	29
hp	10876803	4.45	16	10559300	4.12	215	10156900	2.11	28	9988280	1.76	34
ami33	1254350	0.09	45	1253960	0.08	956	1237540	0.00	203	1237540	0.00	182
ami49	37766000	9.34	57	37548000	8.00	1582	40316000	11.15	431	40624800	10.83	448
RATIO	1.000	1.000	1.000	0.989	0.886	16.615	1.000	0.623	3.429	1.000	0.598	3.540

TABLE II
IMPACT OF EFFECTIVE DECAP DISTANCE. WE REPORT EXTRA DECAP AREA NECESSARY TO SATISFY THE NOISE CONSTRAINT

	2D floorplanning		3D floorplanning	
ckt	w/o ED	w/ ED	w/o ED	w/ ED
n50	0	0	3	0
n50b	3914	3896	5	0
n50c	2334	1957	4	0
n100	12431	11302	3098	93
n100b	17812	17866	5978	5
n100c	18860	18675	1365	0
n200	8230	8227	2563	0
n200b	21489	21372	5117	0
n200c	22452	22445	2586	0
RATIO	1.000	0.970	1.000	0.003

TABLE III
PERCENTAGE OF DECAPS ALLOCATED TO ADJACENT VERSUS NONADJACENT BLOCKS WHEN EFFECTIVE DECAP DISTANCE IS USED. WE ALSO REPORT THE PERCENTAGE OF DECAPS ALLOCATED TO BLOCKS IN OTHER DEVICE LAYERS IN THE 3-D FLOORPLANS

	2D floorplanning		3D floorplanning		
ckt	adj	non	adj	non	inter
n50	95.2%	4.8%	31.4%	68.6%	44.8%
n50b	96.1%	3.9%	34.9%	65.1%	53.1%
n50c	94.6%	5.4%	25.0%	75.0%	55.3%
n100	95.4%	4.6%	30.7%	69.3%	56.0%
n100b	99.5%	0.5%	35.0%	65.0%	56.6%
n100c	99.2%	0.8%	34.9%	65.1%	53.5%
n200	99.1%	0.9%	42.1%	57.9%	46.5%
n200b	99.6%	0.4%	44.5%	55.5%	48.5%
n200c	99.2%	0.8%	39.0%	61.0%	53.1%
AVE	97.5%	2.5%	35.3%	64.7%	51.9%

A. Comparison to Existing Work

To verify our floorplanner and noise analyzer, we performed 2-D floorplanning on the Microelectronics Center of North Carolina (MCNC) benchmarks using the 0.25-μm technology parameters, as in [1]. The MCNC blocks were assigned random current densities between 10^6 A/m^2 and $2 \cdot 10^6$ A/m^2, as in [1]. Table I shows the comparison of our 2-D floorplanning results to those reported in [1]. As with the case in [1], our floorplanner was able to reduce decap budget when noise or decap aware. The decap values are slightly different than [1], because the current densities of the blocks are randomly assigned. Nevertheless, our decap-aware floorplanner reduced the decap relative to our area/wire-length-driven floorplanner, just as the noise-aware floorplanner reduced the decap relative to the postfloorplanner in [1].

B. Effective Distance Results

Due to the small number of blocks in the MCNC benchmarks, we used Gigascale Systems Research Center (GSRC) benchmarks for 3-D floorplanning. The GSRC benchmarks do not specify the current demands of the blocks, so we randomly assigned maximum current densities to the blocks between 10^6 and 10^7 A/m^2. The values for wire resistance, inductance, decap capacitance, and decap leakage used for the 3-D floorplans were taken from the ITRS for the 90-nm technology node. The 3-D floorplanning results are based on four-die stacks.

Table II shows the impact of effective distance on 2-D and 3-D floorplans. We obtain floorplans with wire+area objective and insert decaps as a postprocess. For both 2-D and 3-D floorplans, effective distance reduces the amount of area expansion required to insert sufficient decap to suppress power-supply noise, which is set to 10% of V_{dd}. The improvement in area expansion from effective distance is quite small at 3% for 2-D floorplans. For 3-D floorplans, the need for area expansion was almost completely eliminated by effective distance.

Table III shows the percentage of decaps allocated to adjacent and nonadjacent blocks when effective decap distance is used. Only 2.5% of the decaps are allocated to nonadjacent blocks in the 2-D floorplans. This is why effective distance did not reduce the area expansion by very much for the 2-D floorplans. For the 3-D floorplans, the effect is much larger, with a majority of the decaps allocated to nonadjacent blocks. Most of the nonadjacent decap allocations were between decaps and blocks in different layers. This interlayer decap allocation is why effective distance is so much more effective at reducing area expansion for 3-D floorplans than for 2-D floorplans.

C. 2-D and 3-D Floorplanning Results

Table IV compares area- and wire-length-driven floorplanning to decap-driven floorplanning for 2-D and 3-D chips. Only thin oxide decaps are used in this experiment. We observe that, in both the 2-D and 3-D cases, the decap-driven floorplanner was able to reduce the decap at the expense of area and wire length. The reduction in decap for the 3-D floorplans is greater than the reduction for 2-D floorplans. This is due to the larger solution space for 3-D floorplans.

TABLE IV
AREA/WIRE-LENGTH-DRIVEN AND DECAP-DRIVEN FLOORPLANNING RESULTS. EFFECTIVE DECAP DISTANCE IS USED

2D floorplanning

	area/wirelength-driven							decap-driven						
ckt	area before	wire length	decap cost	area after	decap leakage	decap time	total run time	area before	wire length	decap cost	area after	decap leakage	decap time	total run time
n50	231610	87278	81.8	231610	3.6	8	140	234898	88520	79.1	235968	3.2	17	157
n50b	231072	73482	80.2	234968	3.5	42	175	228541	82306	76.3	228687	3.3	16	161
n50c	224725	85971	71.7	227059	3.1	37	170	242039	89591	67.5	245182	3.0	38	182
n100	231099	162619	179.7	242401	7.5	292	812	233835	171853	184.1	247862	7.8	485	1095
n100b	206847	124037	209.9	224713	8.7	383	904	218240	126941	208.2	234709	8.6	455	1003
n100c	232460	156793	189.1	251135	8.0	436	950	227532	152519	189.0	239596	7.8	236	780
n200	259140	332469	239.9	267367	8.8	345	2405	262032	321292	238.2	278169	9.9	921	3063
n200b	264616	329730	270.9	285988	11.1	2266	4411	263467	336893	270.5	286166	11.3	1263	3695
n200c	263500	310327	254.3	285945	10.5	678	2869	268475	397419	249.1	325735	10.6	284	2721
RATIO	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.017	1.055	0.983	1.030	0.991	1.174	1.050

3D floorplanning

	area/wirelength-driven							decap-driven						
ckt	area before	wire length	decap cost	area after	decap leakage	decap time	total run time	area before	wire length	decap cost	area after	decap leakage	decap time	total run time
n50	65484	37909	39.3	65484	2.0	22	136	67053	44418	23.8	67053	1.1	12	135
n50b	63662	33980	41.1	63662	2.0	22	138	66123	38387	23.4	66123	1.1	19	144
n50c	63888	40657	33.0	63888	1.7	24	138	70356	44159	19.7	70356	1.0	11	136
n100	63252	74959	140.0	63345	6.6	245	642	67375	84409	119.0	67492	5.6	137	593
n100b	56304	56607	160.7	56309	7.4	128	521	63300	63927	143.0	63314	6.7	148	600
n100c	59290	69079	141.1	59290	6.6	121	517	63612	78271	111.7	63612	5.3	69	480
n200	69948	146638	197.2	69948	9.2	297	1769	67750	154319	186.9	67750	8.8	354	1889
n200b	66483	153188	224.3	66483	10.2	297	1796	67276	167804	214.7	67282	9.9	656	2202
n200c	63308	148654	205.7	63308	9.4	332	1827	52675	169227	195.5	63691	8.8	316	1863
RATIO	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.033	1.070	0.837	1.055	0.849	1.087	1.119

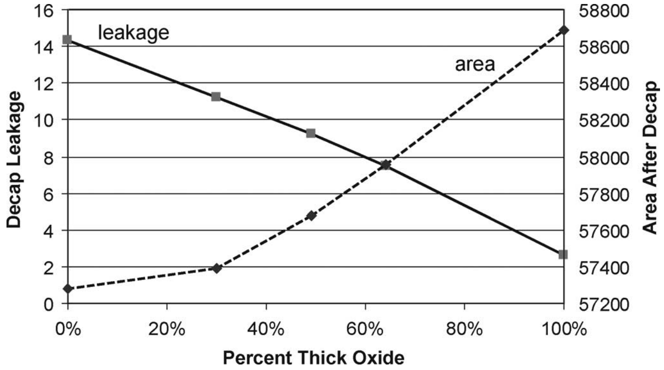

Fig. 13. Decap insertion with dual oxide thicknesses for n100. Effective decap distance and footprint-aware whitespace insertion are used. The area before decap insertion is 56 925.

D. Decap Oxide Thickness Results

The generalized min-cost network-flow-based decap allocator is able to trade increased area for decreased decap leakage. The proportion of thin and thick oxide decaps were controlled by adjusting the cost of leakage. When the cost of leakage is zero, the decap allocator ignores leakage and assigns all decaps as thin oxide, minimizing area expansion. As the cost of leakage is raised, the decap allocator will increase the proportion of thick oxide decaps. Fig. 13 shows the effect that different oxide thickness proportions had on area and decap leakage, where the proportion of thin and thick oxide decaps were varied for the n100 benchmark by adjusting the leakage cost. Only using thin oxide decaps minimized the area expansion but had high decap leakage. As more thick oxide decaps were used, the leakage decreased, but the area expansion increased. Using all thick oxide decaps resulted in the greatest area expansion but decreased the decap leakage to approximately one fifth of the thin oxide leakage.

Table V shows the impact of dual-oxide-thickness decaps for 2-D and 3-D floorplans. With dual-oxide-thickness decaps, the generalized min-cost network-flow-based decap allocator was able to reduce the decap leakage of all circuits to 5 A or less. The flow-based decap allocator minimizes the area expansion by using as many thin oxide decaps as possible without violating the leakage constraint. The decap allocator assigned some thick oxide decaps to the smaller circuits even though the leakage was already below the constraint. This is due to the approximation algorithm used to solve the generalized min-cost network flow.

E. Impact of Flow Approximation

Table VI shows the effect that, varying the error bound, ϵ has on the decap allocation. The error bound allows for a tradeoff between runtime and solution quality in terms of area. Reducing ϵ from 0.50 to 0.40 resulted in an area savings for five of the nine circuits, while increasing the runtime by 3.5%. Reducing ϵ to 0.3 resulted in additional area savings for the n100 and n100b circuits, at a cost of a 9% increase in runtime. Decreasing ϵ for area savings has diminishing returns. For example, decreasing ϵ from 0.30 to 0.20 resulted in negligible area reductions for the n100 and n100b circuits, with no area reductions in the other seven circuits.

TABLE V
FLOORPLANNING WITH DUAL-OXIDE-THICKNESS DECAPS. EFFECTIVE DECAP DISTANCE IS USED

2D floorplanning

				thin oxide only			dual oxide				
ckt	area before	wire length	decap cost	area after	decap leakage	run time	area after	decap leakage	run time	thin ox %	thick ox %
n50	231610	87278	81.8	231610	3.6	140	231610	3.4	153	85.6%	14.4%
n50b	231072	73482	80.2	234968	3.5	175	234972	3.2	175	81.2%	18.8%
n50c	224725	85971	71.7	227059	3.1	170	227059	3.0	171	94.8%	5.2%
n100	231099	162619	179.7	242401	7.5	812	247124	5.0	787	47.5%	52.5%
n100b	206847	124037	209.9	224713	8.7	904	240145	5.0	1261	36.9%	63.1%
n100c	232460	156793	189.1	251135	8.0	950	252498	4.9	1000	40.4%	59.6%
n200	259140	332469	239.9	267367	8.8	2405	291704	5.0	6347	26.6%	73.4%
n200b	264616	329730	270.9	285988	11.1	4411	307088	5.0	6104	22.1%	77.9%
n200c	263500	310327	254.3	285945	10.5	2869	305299	5.0	6394	25.1%	74.9%
RATIO	-	-	-	1.000	1.000	1.000	1.036	0.686	1.419	-	-

3D floorplanning

				thin oxide only			dual oxide				
ckt	area before	wire length	decap cost	area after	decap leakage	run time	area after	decap leakage	run time	thin ox %	thick ox %
n50	65484	37909	39.3	65484	2.0	136	65484	1.5	136	56.5%	43.5%
n50b	63662	33980	41.1	63662	2.0	138	63662	1.7	139	68.0%	32.0%
n50c	63888	40657	33.0	63888	1.7	138	63888	1.4	141	67.3%	32.7%
n100	63252	74959	139.8	63345	6.6	642	63347	4.4	659	45.3%	54.7%
n100b	56304	56607	160.7	56309	7.4	521	56310	4.6	535	39.9%	60.1%
n100c	59290	69079	141.1	59290	6.6	517	59290	4.1	527	38.3%	61.7%
n200	69948	146638	197.2	69948	9.2	1769	69948	4.9	2137	32.8%	67.2%
n200b	66483	153188	224.3	66483	10.2	1796	66604	5.0	2049	29.3%	70.7%
n200c	63308	148654	205.7	63308	9.4	1827	63321	5.0	2355	32.9%	67.1%
RATIO	-	-	-	1.000	1.000	1.000	1.001	0.652	1.082	-	-

TABLE VI
IMPACT OF ERROR-BOUND ϵ ON THE DECAP-ASSIGNMENT RESULTS. WE REPORT THE FINAL AREA AND THE RUNTIME

	$\epsilon = 0.50$		$\epsilon = 0.40$		$\epsilon = 0.30$		$\epsilon = 0.20$	
ckt	area	run time	area	run time	area	run time	area	run time
n50	65484	133	65484	128	65484	136	65484	178
n50b	63662	134	63662	128	63662	138	63662	167
n50c	63888	134	63888	139	63888	138	63888	168
n100	64338	505	63521	528	63345	642	63295	920
n100b	56898	475	56355	501	56309	521	56304	540
n100c	59290	475	59290	464	59290	517	59290	669
n200	69971	1641	69948	1638	69948	1769	69948	2155
n200b	66508	1640	66483	1795	66483	1796	66483	2175
n200c	63357	1673	63308	1987	63308	1827	63308	2244
ratio	1.000	1.000	0.997	1.035	0.997	1.089	0.997	1.354

TABLE VII
3-D DECAP RESULTS FOR DIFFERENT POWER-PIN CONFIGURATIONS

	Sixteen Power Pins							Thirty Six Power Pins						
ckt	area before	wire length	decap cost	area after	decap leakage	decap time	total run time	area before	wire length	decap cost	area after	decap leakage	decap time	total run time
n50	65484	37909	39.3	65484	2.0	22	136	67405	37782	7.2	67405	0.5	2	117
n50b	63662	33980	41.1	63662	2.0	22	138	63888	34661	4.7	63888	0.3	1	117
n50c	63888	40657	33.0	63888	1.7	24	138	60928	38780	3.2	60928	0.2	1	115
n100	63252	74959	140.0	63345	6.6	245	642	67830	73844	38.4	67830	2.4	10	407
n100b	56304	56607	160.7	56309	7.4	128	521	57024	58869	37.3	57024	2.4	10	408
n100c	59290	69079	141.1	59290	6.6	121	517	61944	70353	35.5	61944	2.3	10	413
n200	69948	146638	197.2	69948	9.2	297	1769	65455	149514	46.6	65455	2.9	138	1762
n200b	66483	153188	224.3	66483	10.2	297	1796	68845	152839	74.5	68845	4.8	275	1903
n200c	63308	148654	205.7	63308	9.4	332	1827	63646	146880	57.0	63646	3.5	143	1652
RATIO	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.010	1.002	0.223	1.010	0.304	0.261	0.980

F. Different Power-Pin Configurations

Table VII shows decap results for two different power-pin configurations: 16 pins along the boundary and 64 pins in an 8 × 8 grid. The configuration with 36 power pins has a much lower decap than the configuration with 16 power pins. The lower decap also reduces the decap leakage. The 36-pin configuration has also a much faster decap-allocation step because fewer blocks need decap, which reduces the size of the generalized network-flow graph.

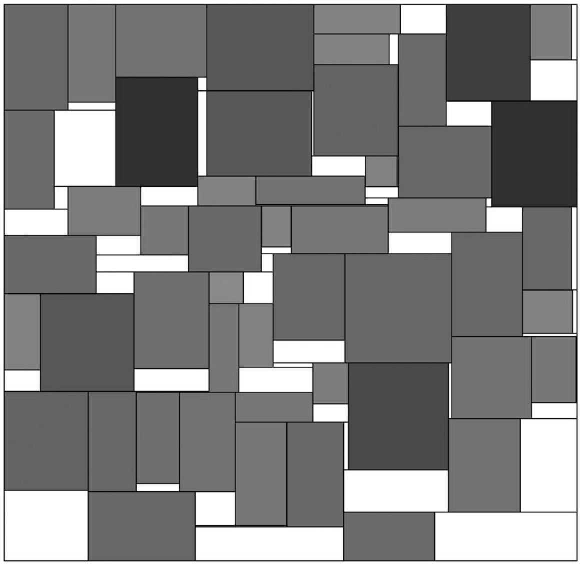

Fig. 14. Two-dimensional floorplan of n50. Blocks are shown in red. Lighter blocks need less decap and darker blocks need more decap.

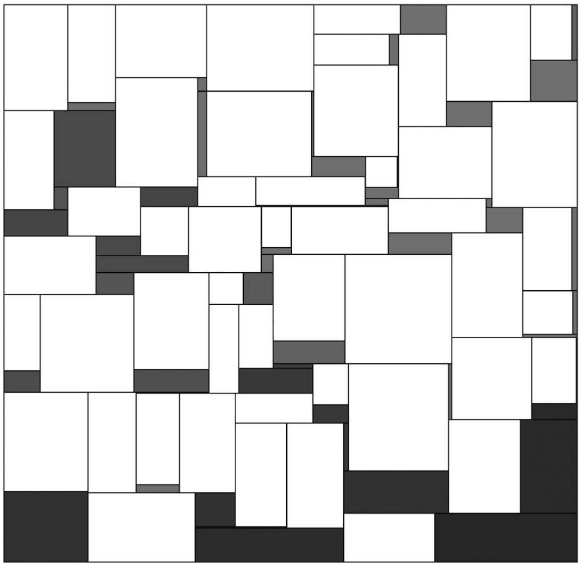

Fig. 15. Decap allocation for 2-D floorplan of n50. Whitespaces are shown in blue. Whitespaces with more thin oxide decaps are lighter. Whitespaces with more thick oxide decaps are darker.

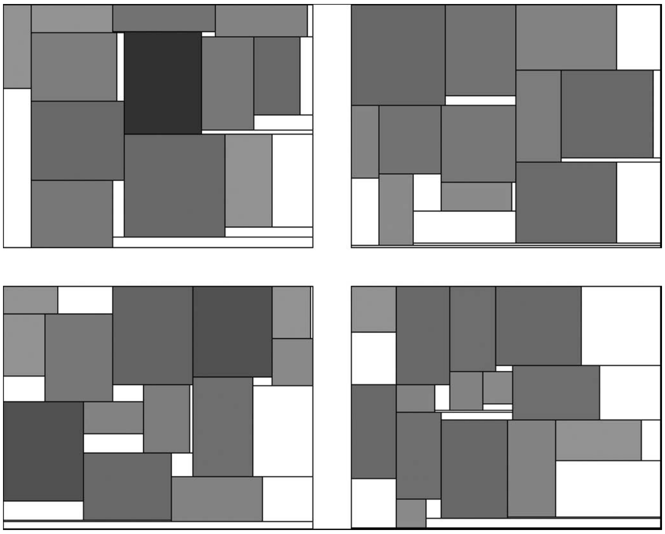

Fig. 16. Three-dimensional floorplan of n50. Blocks are shown in red. Lighter blocks need less decap and darker blocks need more decap.

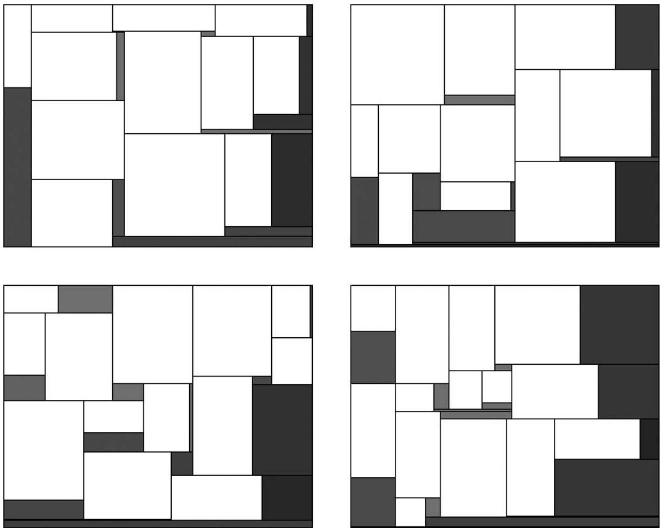

Fig. 17. Decap allocation for 3-D floorplan of n50. Whitespaces are shown in blue. Whitespaces with more thin oxide decaps are lighter. Whitespaces with more thick oxide decaps are darker.

G. Floorplan Examples

Fig. 14 shows a 2-D floorplan of n50. The darker blocks have higher decap demands. Fig. 15 shows the decap allocation for the 2-D floorplan. Whitespaces with higher proportions of thick oxide decaps are darker. The top half of the floorplan has more blocks with high decap demand so more thin oxide decaps are allocated there, since they provide more decap per unit area. The bottom half of the floorplan uses more thick oxide decaps because the blocks near the bottom require less decap. Fig. 16 shows a 3-D floorplan of n50, and Fig. 17 shows its decap allocation. The whitespaces on the left sides of the four layers have more thin oxide decaps allocated to them because there is less whitespace area available for decap allocation than there is on the right sides.

VII. Conclusion

We presented the effective distance model to analyze how functional blocks are affected by nonneighboring decaps. A generalized network-flow-based decap allocation and sizing algorithm incorporated dual-oxide-thickness decaps to reduce leakage. Our algorithm significantly reduced decap budget and leakage power with a small increase in area and wire length when integrated into the 2-D and 3-D floorplanner. Future work includes adapting whitespace-redistribution techniques to further reduce the area expansion required for decap insertion.

REFERENCES

[1] S. Zhao, C. Koh, and K. Roy, "Decoupling capacitance allocation and its application to power supply noise aware floorplanning," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 21, no. 1, pp. 81–92, Jan. 2002.

[2] J. Fu, Z. Lou, X. Hong, Y. Cai, S. X.-D. Tan, and Z. Pan, "VLSI on-chip power/ground network optimization considering decap leakage currents," in *Proc. Asia South Pacific Des. Autom. Conf.*, 2005, pp. 735–738.

[3] H. H. Chen, J. S. Neely, M. F. Wang, and G. Co, "On-chip decoupling capacitor optimization for noise and leakage reduction," in *Proc. IEEE Symp. Integr. Circuits Syst. Des.*, 2003, pp. 251–255.

[4] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, "Rectangle packing based module placement," in *Proc. IEEE Int. Conf. Comput.-Aided Des.*, 1995, pp. 472–479.

[5] H. Chen, L. Huang, I. Liu, and M. Wong, "Simultaneous power supply planning and noise avoidance in floorplan design," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 4, pp. 578–587, Apr. 2005.

[6] H. Su, S. Sapatnekar, and S. R. Nassif, "An algorithm for optimal decoupling capacitor sizing and placement for standard cell layouts," in *Proc. Int. Symp. Phys. Des.*, 2002, pp. 68–73.

[7] J. Choi, S. Chun, N. Na, M. Swaminathan, and L. Smith, "A methodology for the placement and optimization of decoupling capacitors for gigahertz systems," in *Proc. VLSI Des. Symp.*, 2000, pp. 156–161.

[8] H. Chen, L. Huang, I. Liu, M. Lai, and D. Wong, "Floorplanning with power supply noise avoidance," in *Proc. Asia South Pacific Des. Autom. Conf.*, 2003, pp. 427–430.

[9] I. Hattori, A. Kamo, T. Watanabe, and H. Asai, "Optimal placement of decoupling capacitors on PCB using Poynting vectors obtained by FDTD method," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2002, pp. V-29–V-32.

[10] S. Zhou, S. Dong, X. Wu, and X. Hong, "Integrated floorplanning and power supply planning," in *Proc. Int. Conf. ASIC*, 2001, pp. 194–197.

[11] I. Liu, H.-M. Chen, T.-L. Chou, A. Aziz, and D. Wong, "Integrated power supply planning and floorplanning," in *Proc. Asia South Pacific Des. Autom. Conf.*, 2001, pp. 589–594.

[12] P. Shiu, R. Ravichandran, S. Easwar, and S. K. Lim, "Multi-layer floorplanning for reliable system-on-package," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2004, pp. V-69–V-72.

[13] R. Ravichandran, J. Minz, M. Pathak, S. Easwar, and S. K. Lim, "Physical layout automation for system-on-packages," in *Proc. IEEE Electron. Compon. Technol. Conf.*, 2004, pp. 41–48.

[14] J. Cong, J. Wei, and Y. Zhang, "A thermal-driven floorplanning algorithm for 3-D ICs," in *Proc. IEEE Int. Conf. Comput.-Aided Des.*, 2004, pp. 306–313.

[15] X. Tang, R. Tian, and M. Wong, "Optimal redistribution of white space for wire length minimization," in *Proc. Asia South Pacific Des. Autom. Conf.*, 2005, pp. 412–417.

[16] S. Chen, X. Hong, S. Dong, Y. Ma, and C. Cheng, "Floorplanning with consideration of white space resource distribution for repeater planning," in *Proc. Int. Symp. Quality Electron. Des.*, 2005, pp. 628–633.

[17] N. Garg and J. Konemann, "Faster and simpler algorithms for multicommodity flow and other fractional packing problems," in *Proc. IEEE Symp. Foundations Comput. Sci.*, 1998, pp. 300–309.

[18] K. D. Wayne and L. Fleischer, "Faster approximation algorithms for generalized flow," in *Proc. ACM/SIAM Symp. Discrete Algorithms*, 1999, pp. 981–982.

[19] C. W. Eichelberger, "Three-dimensional multichip module system," U.S. Patent 5 111 278, May 5, 1992.

[20] G. Roos, B. Hoefflinger, M. Schubert, and R. Zingg, "Manufacturability of 3-D-epitaxial-lateral-overgrowth CMOS circuits with three stacked channels," *Microelectron. Eng.*, vol. 15, no. 1–4, pp. 191–194, Oct. 1991.

[21] V. Subramanian, P. Dankoski, L. Degertekin, B. Khuri-Yakub, and K. Saraswat, "Controlled two-step solid-phase crystallization for high-performance polysilicon TFTs," *IEEE Electron Device Lett.*, vol. 18, no. 8, pp. 378–381, Aug. 1997.

[22] A. Fan, A. Rahman, and R. Reif, "Copper wafer bonding," in *Electrochem. Solid-State Lett.*, vol. 2, 1999, pp. 534–536.

Eric Wong (S'05) received the B.S. degree in electrical engineering from the State University of New York, Binghamton, in 2004 and the M.S. degree in electrical and computer engineering from Georgia Institute of Technology, Atlanta, in 2006.

He is currently a Software Engineer with Universal Avionics Systems Corporation, Tucson, AZ.

Jacob Rajkumar Minz (S'05) received the B.Tech. degree in computer science and engineering from the Indian Institute of Technology (IIT), Kharagpur, India, in 2001 and the Ph.D. degree in electrical and computer engineering from Georgia Institute of Technology, Atlanta, in 2006.

He was with the Advanced VLSI Design Laboratory, IIT, for a year, where he was involved in the design of digital chips. He is currently with Synopsys Corporation, Sunnyvale, CA. His areas of interest are in physical-design automation, logic synthesis, and algorithms for electronic computer-aided design.

Sung Kyu Lim (S'94–M'00–SM'05) received the B.S., M.S., and Ph.D. degrees from the Computer Science Department, University of California at Los Angeles (UCLA), Los Angeles, in 1994, 1997, and 2000, respectively.

From 2000 to 2001, he was a Postdoctoral Scholar with UCLA and a Senior Engineer with Aplus Design Technologies, Inc. In 2001, he was with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, where he is currently an Associate Professor. His research focus is on the physical-design automation for 3-D circuits, 3-D system-on-packages, microarchitectural physical planning, and field-programmable analog arrays.

Dr. Lim was the recipient of the Design Automation Conference Graduate Scholarship in 2003 and the National Science Foundation Faculty Early Career Development (CAREER) Award in 2006. He was the recipient of the Outstanding Junior Faculty Member Award from the School of Electrical and Computer Engineering, Georgia Institute of Technology, in 2007. He has been on the Advisory Board of the Association for Computing Machinery (ACM) Special Interest Group on Design Automation since 2003. He is an Associate Editor of the IEEE TRANSACTIONS ON VERY LARGE-SCALE INTEGRATION (VLSI) SYSTEMS and served as a Guest Editor for the ACM TRANSACTIONS ON DESIGN AUTOMATION OF ELECTRONIC SYSTEMS. He was with the Technical Program Committee of several ACM and IEEE conferences on electronic-design automation.